Chinese Journal of Aeronautics, (2025), 38(10): 103495

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

FULL LENGTH ARTICLE

A hyperelliptic curve-based authenticated key agreement scheme for unmanned aerial vehicles in cross-domain environments☆

Muhammad Asghar KHAN [a], Insaf ULLAH [b,*], Haralambos MOURATIDIS [b], Abdulmajeed ALQHATANI [c], Pascal LORENZ [d]

[a] *Department of Electrical Engineering, Prince Mohammad Bin Fahd University, Al Khobar 31952, Saudi Arabia*
[b] *Institute for Analytics and Data Science, University of Essex, Colchester CO4 3SQ, UK*
[c] *Department of Information Systems, Najran University, Najran 55461, Saudi Arabia*
[d] *Department of Network and Telecommunication, University of Haute Alsace, Colmar 68008, France*

Received 19 August 2024; revised 7 October 2024; accepted 18 December 2024
Available online 19 March 2025

KEYWORDS

Computation cost;
Cross-domain authentication;
Hyperelliptic curve cryptography;
Key agreement;
ROM;
ROR;
Security;
Unmanned aerial vehicles

Abstract Unmanned Aerial Vehicles (UAVs) are increasingly recognized for their pivotal role in military and civilian applications, serving as essential technology for transmitting, evaluating, and gathering information. Unfortunately, this crucial process often occurs through unsecured wireless connections, exposing it to numerous cyber-physical attacks. Furthermore, UAVs' limited onboard computing resources make it challenging to perform complex cryptographic operations. The main aim of constructing a cryptographic scheme is to provide substantial security while reducing the computation and communication costs. This article introduces an efficient and secure cross-domain Authenticated Key Agreement (AKA) scheme that uses Hyperelliptic Curve Cryptography (HECC). The HECC, a modified version of ECC with a smaller key size of 80 bits, is well-suited for use in UAVs. In addition, the proposed scheme is employed in a cross-domain environment that integrates a Public Key Infrastructure (PKI) at the receiving end and a Certificateless Cryptosystem (CLC) at the sending end. Integrating CLC with PKI improves network security by restricting the exposure of encryption keys only to the message's sender and subsequent receiver. A security study

* Corresponding author.
E-mail address: insaf.ullah@essex.ac.uk (I. ULLAH).
☆ Special Issue: Secure and Covert UAV Communication.
Peer review under responsibility of Editorial Committee of CJA

https://doi.org/10.1016/j.cja.2025.103495

employing ROM and ROR models, together with a comparative performance analysis, shows that the proposed scheme outperforms comparable existing schemes in terms of both efficiency and security.

1. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs), more commonly known as drones, have experienced significant technological advancements and expanded applications in both military and civilian domains.[1] UAVs have the potential to serve as autonomous communication nodes, airborne relays, or aerial Base Stations (BSs), supporting traditional networks in challenging propagation environments characterized by obstacles and highly mobile, remotely located nodes.[2] UAVs can help set up a flexible and reliable multi-hop communication backbone by flying at low altitudes, high elevation angles, and over urban, suburban, and rural terrains to support demanding applications like disaster and crisis management, agriculture, transportation, environmental monitoring, remote sensing, and healthcare.[3] Nevertheless, several scientific and technological hurdles must be overcome before UAV-aided communication networks can be reliably deployed in highly dynamic and diverse settings.

Typically, UAVs receive command and control signals from ground control stations to carry out their missions remotely. These signals are sent across various communication channels at varying transmission rates. Considering that the majority of information sent to or by UAVs takes place in the air and that most of the data transferred is sensitive, security and privacy are key concerns in these communications.[4–6] From a security and privacy standpoint, the sensing aspect of UAVs is likewise worthy of attention. For instance, in the worst-case scenario, a sensor could transmit incorrect information, causing UAVs to make faulty decisions. Similarly, the situation with the malfunctioning sensor is a lot more concerning, which may drastically impair the capacity of the UAV to acquire information. In the worst-case scenario, it can even cause an accident if the sensor is wholly inoperable. In addition, a reliable communication link is essential to enable the exchange of information between a BS and a UAV.[7] On the other hand, a link that is not secure is susceptible to various cyber-attacks.[8]

In recent years, several research efforts have been made to ensure the security of communication for UAVs. Nonetheless, their solutions either need a substantial amount of computation and a high degree of communication costs on the part of the UAV, or their schemes have security flaws in some cases.[9] As a result, a lightweight Authenticated Key Agreement (AKA) scheme is an ideal solution for UAVs to cater to the constants encountered by a typical UAV. The proposed AKA scheme can be formulated using either the Public Key Infrastructure (PKI), a Certificateless Cryptosystem (CLC), or an Identity-based Cryptosystem (IBC). However, a drawback of PKI lies in its impractical application for UAVs due to significant certificate management overhead involving tasks like storage, distribution, and revocation.[10] In response, an Identity-based Cryptosystem (IBC) is introduced to alleviate the burden on traditional PKI.[11] IBC utilizes a publicly recognized string as a public key, aiming to reduce the costs associated with PKI certificate renewal.

Being Identity-Based, the IBC appears more susceptible to hacking by external parties (key escrow problem). The key escrow problem was addressed by developing CLC, a kind of ID-based encryption.[12] The Key Generation Center (KGC) is responsible for generating partial private keys and distributing them to users over a secure network. The next step for the user is to generate their private and public keys by combining the previously obtained private key fragment with some random digits produced by the computer. In each of the previous exchanges, homogeneous cryptography was employed. This type of cryptography meant that both the sender and the receiver shared the same security domain, which made the network more vulnerable. The vulnerability requires deploying a cross-domain mechanism in which the sender and the receiver both have independent security domain attributes to protect a diverse range of cyber-attacks. Integrating CLC and PKI confers the advantage of fortifying the network against unauthorized access. This is achieved by limiting the revelation of the original keys exclusively to the sender and any subsequent recipients of the message. Within the framework of the proposed AKA scheme, implemented in a cross-domain environment, the transmitting end employs CLC, while the receiving end utilizes PKI.

In addition, existing studies of AKA schemes for UAV environments have designed an array of schemes with a common deficiency in their mathematical algorithmic roots and high computational and communication costs. For the most part, the Rivest-Shamir-Adleman (RSA), Bilinear Pairing (BP), and Elliptic Curve Cryptography (ECC) algorithms are the ones that are used to enhance the security and efficiency of cryptographic methods.[13] The key size of RSA is enormous, but BP's pairing and map-to-point function processing make it a less secure algorithm overall. ECC is a public-key cryptography system based on the algebraic structure of elliptic curves over finite fields. It offers equivalent security to traditional schemes like RSA but with much shorter key lengths, making it more efficient in terms of computation and memory usage. ECC is beneficial in environments with limited resources, such as UAVs, because it provides robust security with lower power consumption and faster encryption and decryption processes. Its strength lies in the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), which underpins its security. ECC was used to bring together the disparities between RSA and BP. Security and efficiency are both provided by 160-bit ECC keys. Hyperelliptic Curve Cryptography (HECC) was created to give an alternative to the security provided by ECC.[14] HECC offers the same security level that ECC, BP, and RSA provide but with a shorter key length of 80 bits. This attribute makes HECC more efficient. HECC is the most suited solution, most cost-effective, and most efficient

for use with devices with limited resources, such as UAVs. As a result, we have added the following new features to the AKA scheme, which characterize our most significant contributions:

(1) This research article introduces an efficient and secure AKA scheme designed for UAVs operating in cross-domain environments. The proposed scheme ensures efficiency and security by facilitating a seamless transition between the sending side's utilization of CLC and the receiving side's adoption of PKI.
(2) In the proposed scheme, on the sender side, which utilizes CLC, the central authority will send the partial public and private keys to UAVs in an encrypted form through an open network. Also, during the generation of the request for partial public and private keys, UAVs transmit their identity in an encrypted manner, guaranteeing sender anonymity and preventing adversaries from accessing the sender's identity.
(3) The proposed scheme incorporates HECC, an advanced iteration of ECC with a reduced 80-bit key size, ensuring an equivalent level of security as conventional methods. The primary objective of the proposed scheme is to substantially reduce computation and communication costs, making it highly suitable for easy integration into UAVs.
(4) After a thorough security analysis using ROM and ROR models, the proposed scheme proves to be resistant to various widely known attacks. In addition, a performance comparison with other relevant schemes determines that it is efficient in computation and communication costs.

The subsequent sections of this article are organized in the following manner. Section 2 of the manuscript encompasses a comprehensive literature review, which critically evaluates the most relevant schemes. Section 3 contains preliminaries, which provide details about syntax, the Random Oracle Model (ROM), the threat model, and the network model. Section 4 presents the construction and correctness of the proposed AKA scheme. Section 5 of the paper discusses the security analysis under ROM, which aims to demonstrate the scheme's resilience against various attacks. In Section 6, we discussed security analysis using the ROR model. Section 7 details the informal security analysis. Section 8 of the manuscript delves into the analysis of performance, encompassing computation and communication costs. Section 9 presents a discussion of the conclusion of the proposed work.

2. Literature review

In recent years, the extensive deployment of UAVs in several civilian and commercial applications and the security challenges of UAV communication have attracted the attention of many researchers. UAVs need a secure connection since they usually communicate through an unsecured wireless channel. Existing UAV security protocols primarily focus on authenticity, data integrity, and confidentiality. A robust Authenticated Key Agreement (AKA) scheme can offer a strong defense against attacks. Table 1 presents a comparative analysis of the most pertinent existing schemes, categorized by year, applied techniques, and limitations. Wazid et al.[15] developed a lightweight AKA scheme designed for remote user authentication. The proposed method relies only on highly effective and one-way cryptographic hash algorithms and bitwise XOR operations. The scheme developed by Wazid et al. is resistant to a wide range of attacks, although it is susceptible to user impersonation attacks and privileged insider attacks. An AKA protocol was developed by Ali et al. [16] to ensure that continuous communication takes place between the user and the UAV. The SHA-160 hashing algorithm and the XOR function are used in this approach. Nevertheless, the method proposed by Ali et al.[16] is susceptible to user impersonation attacks, privileged insider attacks, forgery attacks, and Denial-of-Service (DoS) attacks.

Wang et al.[17] introduced a novel efficient mechanism for authenticated key agreement. The proposed protocol exhibits innovation by incorporating a dynamic switching mechanism between the PKI and IBC methods. This feature is in contrast to prior protocols that employ a uniform cryptosystem throughout; however, once this method encounters the issue of a single point of system failure, it will lead to problems such as user privacy leakage. Tao et al.[18] proposed a security analysis of an AKA protocol designed for cross-domain applications, which utilizes a heterogeneous signcryption method. The initial authentication stage in this methodology fails to satisfy the criteria for the unforgeability of signcryption ciphertext. The authors proposed a new cross-domain authentication system utilizing two-way heterogeneous signcryption to enhance security. The research proves that the system is unforgeable under the random oracle model and against adaptive selection message attacks. In their presentation of a heterogeneous cross-domain AKA protocol, Liu et al.[19] incorporated symptom-matching in TMIS, which stands for CDAKA. The authors found their proposed AKA approach's computation and communication costs negligible. Also, when using ROM and assuming the Elliptic Curve Computable Diffie-Hellman Problem (ECDHP), it is secure against widely known security attacks.

Recently, Pan et al.[20] proposed a Heterogeneous Authenticated Key Agreement (HAKA) scheme to enable a UAV to connect with a GS. In this configuration, the UAV would be part of IBC, whereas the GS would be part of PKI. A rigorous security analysis has shown that the proposed scheme is secure. On the other hand, the scheme proposed by Pan et al.[20] is not suitable for use with UAVs that have a limited amount of computing resources. The key escrow is a crucial issue in the proposed schemes.[18–20] The identity-based cryptography-related device will encounter a problem with key escrow. In addition, these schemes require a secure channel to transmit the private key to the device for identity-based cryptography. They lack anonymity because the device transmits its true identity over an open network during registration. Aithekar et al.[21] proposed an identity-based heterogeneous scheme for UAV networks. Their proposed scheme is based on ECC without pairing operations, aiming to reduce the computation resources required for authentication and enhance authentication efficiency. Xie et al.[22] proposed a blockchain-based authentication scheme for a UAV-assisted IoV system for registration and authentication services and permits dynamic addition and removal. The authors designed a novel combined scheme based on multiple PKGs for encryption and signature to prevent information tampering and identity deception.

All the schemes mentioned above[15–22] rely on bilinear pairing, exponential operations and ECC, which entails significant computation and communication costs or security flaws. Due to UAVs' resource-constrained nature, onboard processing capabilities are very restricted. Such operations would be challenging for a UAV to complete. HECC, on the other hand, provides the same degree of security as a bilinear pairing, exponential operations, and ECC but with shorter key lengths of 80 bits. This feature increases the efficiency of HECC. HECC is the most appropriate, cost-effective, and efficient use option for low-resource devices (see Table 1).

3. Preliminaries

This section details syntax, the ROM, the threat model, the network model, and the construction of the proposed AKA scheme.

3.1. Syntax of proposed scheme

The syntax of the proposed cross-domain AKA scheme for UAVs is discussed in the subsequent sub-steps.

(1) Setup. The Service Provider (SP_{TA}) that serves as the Trusted Authority (TA) and sets $A_{SP_{TA}}$ as his private key and $B_{SP_{TA}}$ as his public key. Then, SP_{TA} creates and distributes $PRM_{SP_{TA}}$ to a network.
(2) UAV Secret Value Generation (UAVSVG). In this sub-step, UAV acts as the initiator/sender, represented as (INT_{UAV}), and first generates and then sends ($END_{INT_{UAV}}, Q_{INT_{UAV}}$) to SP_{TA} via an open network.
(3) Partial Private Key Generation (PPKGN). When SP_{TA} receives ($END_{INT_{UAV}}, Q_{INT_{UAV}}$), then it generates and sends $EPRT_{SP_{TA}}$ to INT_{UAV} through an open network.
(4) UAV Private and Public Key Generation (UPBG). When INT_{UAV} receives ($EPRT_{SP_{TA}}$), it sets ($J_{SP_{TA}}$, $Q_{INT_{UAV}}$) as his public key and ($PRT_{SP_{TA}}, P_{INT_{UAV}}$) as his private key.
(5) GS Private and Public Key Generation (GSPBG): GS assumes the role of responder/receiver (R_{GS}) and can generate his private and public keys ($P_{R_{GS}}, Q_{R_{GS}}$).
(6) Authenticated Key Agreement: INT_{UAV} first chooses the symmetric key β from AES, generates and sends ($L_{INT_{UAV}}, C, S_{INT_{UAV}}$) to R_{GS} that contains the ciphertext, which is generated on a symmetric key. When R_{GS} receives ($L_{INT_{UAV}}, C, S_{INT_{UAV}}$), it first recovers the symmetric key β, then verifies the signature. If the verification process is successful, it sets β as the symmetric secret key between R_{GS} and INT_{UAV}.

3.2. ROM

Bellare and Rogaway[23] developed the ROM in 1993. By seeing hash functions as random oracles, we can quickly gauge the robustness of cryptographic methods that use them. According to this paradigm, every given input will always produce an output at a fixed time. If the input has been requested before, the oracle will provide the same value it offered before. If the input is not what the oracle has seen, it will give an unexpected result. Rather than relying on a hash function, it uses a freely accessible random function (sometimes known as the "random

Table 1 A comparative study of the most relevant existing schemes by year, applied techniques, and limitations.

Scheme	Year	Technique	Limitation
Xie et al.[22]	2024	* Bilinear pairing * Blockchain	* Suffers from high computation and communication costs
Aithekar et al.[21]	2024	* ECC	* Vulnerable to impersonation attack * Suffers from high computation and communication costs
Pan et al.[20]	2023	* Bilinear pairing * Exponential operation * Hash function	* Suffers from a key escrow problem * Requires a secure channel to send the private key * Lacks of anonymity * Bilinear pairing and exponential operations need heavy computational efforts
Ali et al.[16]	2020	* Exclusive-OR * AES * Hash function	* Vulnerable to IND-CPA security model * There is no support for user anonymity and untraceability features
Tao et al.[18]	2020	* Bilinear pairing * Exponential operation * Hash function	* Suffers from a key escrow problem * Requires a secure channel to send the private key * Bilinear pairing and exponential operations require heavy computational efforts
Wazid et al.[15]	2019	* Exclusive-OR * Hash function	* Vulnerable to user impersonation and privileged insider attacks
Liu et al.[19]	2018	* ECC * Hash function	* Suffers from a key escrow problem * Requires a secure channel to send the private key * Lacks of anonymity * ECC needs to put more effort into the scheme's processing than HECC
Wang et al.[17]	2017	* Bilinear pairing * Exponential operation * Hash function	* This method encounters the issue of a single point of system failure, which will lead to problems such as user privacy leakage

oracle"). To guess the hash function's output, a potential attacker must employ a random number generator.

3.3. Threat model

The threat model for the proposed cryptographic scheme involves two main classes of adversaries, Type 1 and Type 2, each with distinct capabilities and goals.

Type 1. Adversaries ($A_{\mathrm{INT}_{GS1}}$)

Type 1 adversaries are considered to be the external threat actors who aim to falsify digital signatures and compromise the confidentiality of the proposed scheme. They have limited access to the system and are considered "outsider" actors. The key characteristic of Type 1($A_{\mathrm{INT}_{GS1}}$) adversaries include their inability to access the user's private but can modify the public key used in communication at the same time. However, they lack the means to derive or directly obtain the corresponding private key, which limits their ability to fully compromise the integrity of secure communications. The primary objective of Type 1 ($A_{\mathrm{INT}_{GS1}}$) adversaries is to manipulate or generate parameters in such a way that they can create valid ciphertexts or forged signatures. To do this, they attempt to exploit weaknesses in the cryptographic process, such as tampering with the public key or intercepting data during transmission, with the ultimate goal of creating a false signature.

Type 2. Adversaries ($A_{\mathrm{INT}_{GS2}}$)

Type 2 adversaries ($A_{\mathrm{INT}_{GS2}}$) represent insider threats, potentially entities such as service providers (denoted by $\mathrm{SP}_{\mathrm{TA}}$) in the proposed scheme. Type 2 adversaries ($A_{\mathrm{INT}_{GS2}}$) have more extensive access than Type 1 adversaries ($A_{\mathrm{INT}_{GS1}}$). Specifically, they can modify the private key of the ($\mathrm{SP}_{\mathrm{TA}}$), enhancing their ability to compromise security. However, even these actors don't have that much ability to access the user's public key, meaning they are still restricted from some of the most critical information needed to break the scheme entirely. The primary danger posed by Type 2 adversaries ($A_{\mathrm{INT}_{GS2}}$) is their ability to tamper with the private key infrastructure, which might allow them to forge signatures and compromise the integrity of sensitive data. By doing so, they could masquerade as legitimate entities or users, potentially breaching the confidentiality of communications between users and the ($\mathrm{SP}_{\mathrm{TA}}$).

Both types of adversaries have a shared goal: to determine the conditions necessary for generating the secret key and the ciphertext. This involves creating or retrieving parameters used in cryptographic calculations, with the ultimate aim of forging a legitimate-looking signature or intercepting sensitive communications. These adversaries exploit different system vulnerabilities by altering public or private keys or manipulating the cryptographic process.

3.4. Security goals

The following are the security goals to be achieved by the proposed scheme.

(1) Unforgeability. The proposed scheme must be unforgeable, which means the attacker will not be able to generate a forged signature.
(2) Secret Key Security. The scheme must have the property of secret key security, which means the attacker will not be able to generate a secret shared key.
(3) Resist against Reply Attack. The proposed scheme will be safe from reply attacks, which means the attacker will not be able to resend the old messages.
(4) Resist against Impersonation Attack. This scheme must be safeguarded from an impersonation attack, which means the attacker cannot impersonate any user.
(5) Man-in-the-Middle Attack. The scheme will resist man-in-the-middle attacks, which means the attacker cannot impersonate any user or generate a forged signature or secret key.

3.5. Network model

The network model proposed in this paper contains three main entities: UAVs, GS, and Service Providers ($\mathrm{SP}_{\mathrm{TA}}$). Each UAV has several helpful components, such as cameras, a Global Positioning System (GPS), an Inertial Measurement Unit (IMU), and sensors, all of which may be used in various application situations. The network is established when the UAVs begin their flight into the air, bringing height sensors, IMU, GPS units, and any other embedded equipment, such as the flight controller. As building a network consisting of many clusters of UAVs starts, we assume that the UAVs already know their neighbors' zone ID, location, altitude, and velocity. In our case, the UAVs belong to certificateless cryptography; when they want to communicate with GS, they first register themselves with $\mathrm{SP}_{\mathrm{TA}}$, and for this purpose, UAVs will send their identity in an encrypted form to $\mathrm{SP}_{\mathrm{TA}}$. When $\mathrm{SP}_{\mathrm{TA}}$ receives encrypted identity, then it first recovers the original identity of UAVs by using a standard shared secret key and generates the partial public and private key, encrypts both partial public and private key through a standard shared secret key and sends it by using the insecure network to UAVs. Then, UAVs can decrypt the ciphertext containing the partial public and private key, generate the wholly private and public key pairs, generate signcryption on collected data, and send the encryption text to GS using an open/insecure network.

The second entity, GS, which belongs to public key infrastructure cryptography, first generates its public and private keys. When it receives a signcryption text from UAVs, its task is to verify the signature and get plain text after decrypting the ciphertext. The third entity is the Service Provider ($\mathrm{SP}_{\mathrm{TA}}$), which is responsible for creating public parameters and partial public and private keys for UAVs. Fig. 1 illustrates the network model the proposed scheme would use to function.

4. Construction of proposed scheme

The construction of the proposed cross-domain AKA scheme for the UAV environment is discussed in further detail in the sub-steps below. The symbols used in the proposed scheme are listed in Table 2.

4.1. Setup

The Service Provider ($\mathrm{SP}_{\mathrm{TA}}$) that serves as the Trusted Authority (TA) first suggests the Hyper Elliptic Curve ($\mathrm{HEC}_{\mathrm{SP}_{\mathrm{TA}}}$) with

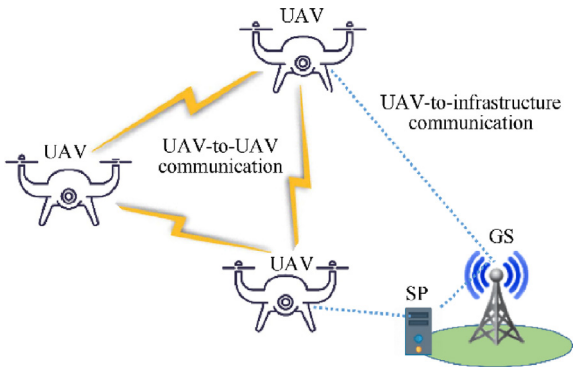

Fig. 1 A network model of proposed scheme.

genus 2, defined over a finite field $F^{(n)}$, where (n) is the order and its value is 80 bits. Second, SP_{TA} chooses $A_{SP_{TA}} \in F^{(n)}$, computes $B_{SP_{TA}} = A_{SP_{TA}} D$, and selects three hash functions $(H_{SP_{TA1}}, H_{SP_{TA2}}, H_{SP_{TA3}})$, sets $A_{SP_{TA}}$ as his private key and $B_{SP_{TA}}$ as his public key. Further, SP_{TA} makes a param $PRM_{SP_{TA}} = \{F^{(n)}, (n), B_{SP_{TA}}, HEC_{SP_{TA}}, H_{SP_{TA1}}, H_{SP_{TA2}}, H_{SP_{TA3}}, D\}$ and distributes $PRM_{SP_{TA}}$ to a network. The illustration of setup and key generation is given in Fig. 2.

Table 2 Notation table.

Notation	Description
SP_{TA}	It represents the service provider that serves as the Trusted Authority (TA).
$HEC_{SP_{TA}}$	It represents the Hyperelliptic Curve, suggested by SP_{TA}
$F^{(n)}$	Indicates a finite field over $HEC_{SP_{TA}}$
(n)	Indicates the order of finite field $F^{(n)}$
$A_{SP_{TA}}$	Indicates the private key of SP_{TA}
$B_{SP_{TA}}$	Indicates the public key of SP_{TA}
$H_{SP_{TA1}}, H_{SP_{TA2}}, H_{SP_{TA3}}$	Indicates the hash functions suggested by SP_{TA}
$PRM_{SP_{TA}}$	Indicates the public parameter form suggested and published by SP_{TA}
D	Indicates the devisor, which belongs to the Jacobian of $HEC_{SP_{TA}}$
INT_{UAV}	Indicates the UAV performing the role of initiator/sender
R_{GS}	Indicates the GS performing the role of responder/receiver
$(J_{SP_{TA}}, Q_{INT_{UAV}})$	Indicates the public key pair of INT_{UAV}
$(PRT_{SP_{TA}}, P_{INT_{UAV}})$	Indicates the private key pair of INT_{UAV}
$P_{R_{GS}}$	Indicates the private key of R_{GS}
$Q_{R_{GS}}$	Indicates the public key of R_{GS}
$\oplus$	Used for encryption and decryption
$K_{INT_{UAV}}$	The public key cryptography-based secret key, which is shared between INT_{UAV} and SP_{TA}
$E_{K_{INT_{UAV}}}$	Indicates encryption through the public key cryptography-based secret key $K_{INT_{UAV}}$
$D_{K_{INT_{UAV}}}$	Indicates decryptions through the public key cryptography-based secret key $K_{INT_{UAV}}$
$ID_{INT_{UAV}}$	Indicates the identity of INT_{UAV}
β	The symmetric key, which is shared between INT_{UAV} and R_{GS}

4.2. CL-key generation

This phase consists of the four steps listed below.

(1) UAV Secret Value Generation (UAVSVG): Here, UAV is acting as the initiator/sender, which could be represented as (INT_{UAV}) and can conduct the actions below.
   (A) First, it picks $P_{INT_{UAV}} \in F^{(n)}$ and computes $Q_{INT_{UAV}} = P_{INT_{UAV}} D$
   (B) Computes $K_{INT_{UAV}} = P_{INT_{UAV}} B_{SP_{TA}}$ and $END_{INT_{UAV}} = E_{K_{INT_{UAV}}}(ID_{INT_{UAV}})$
   (C) Sends $(END_{INT_{UAV}}, Q_{INT_{UAV}})$ to SP_{TA} via an open network and stores the tuple for $(END_{INT_{UAV}}, Q_{INT_{UAV}})$ in his database for future use.

Note that if the attacker wants to launch a Man in the Middle attack by accessing the identity of the initiator/sender, which could be represented as (INT_{UAV}), then he first decrypts $END_{INT_{UAV}}$ as $ID_{INT_{UAV}} = D_{K_{INT_{UAV}}}(END_{INT_{UAV}})$ and this process the attacker needs to compute $K_{INT_{UAV}} = P_{INT_{UAV}} B_{SP_{TA}}$, which is not feasible for him to extract the private number $P_{INT_{UAV}}$ from $Q_{INT_{UAV}} = P_{INT_{UAV}} D$ that leads to solving a hard problem called HECDLP.

(2) Partial Private Key Generation (PPKGN): When SP_{TA} receives $(END_{INT_{UAV}}, Q_{INT_{UAV}})$, it performs the following steps.
   (A) Computes $K_{INT_{UAV}} = Q_{INT_{UAV}} A_{SP_{TA}}$ and recovers $ID_{INT_{UAV}}$ as $ID_{INT_{UAV}} = D_{K_{INT_{UAV}}}(END_{INT_{UAV}})$.
   (B) Chooses $G_{SP_{TA}} \in F^{(n)}$, and then computes $J_{SP_{TA}} = G_{SP_{TA}} D$, and $PRT_{SP_{TA}} = G_{SP_{TA}} + A_{SP_{TA}} H_{SP_{TA1}} \cdot (ID_{INT_{UAV}}, J_{SP_{TA}}, Q_{INT_{UAV}})$.
   (C) Computes $EPRT_{SP_{TA}} = E_{K_{INT_{UAV}}}(PRT_{SP_{TA}}, J_{SP_{TA}})$ and sends $EPRT_{SP_{TA}}$ to INT_{UAV} through an open network.

(3) UAV Private and Public Key Generation (UPBG): When INT_{UAV} receives $(EPRT_{SP_{TA}})$, it performs the following steps.
   (A) It recovers $(PRT_{SP_{TA}}, J_{SP_{TA}})$ as $(PRT_{SP_{TA}}, J_{SP_{TA}}) = D_{K_{INT_{UAV}}}(EPRT_{SP_{TA}})$.
   (B) INT_{UAV} sets $(J_{SP_{TA}}, Q_{INT_{UAV}})$ as his public key and $(PRT_{SP_{TA}}, P_{INT_{UAV}})$ as his private key.

(4) GS Private and Public Key Generation (GSPBG): Here, GS implies the role of responder/receiver (R_{GS}) and can generate his private and public keys as it selects $P_{R_{GS}} \in F^{(n)}$, and computes $Q_{R_{GS}} = P_{R_{GS}} D$, sets $P_{R_{GS}}$ as his private key and $Q_{R_{GS}}$ as his public key.

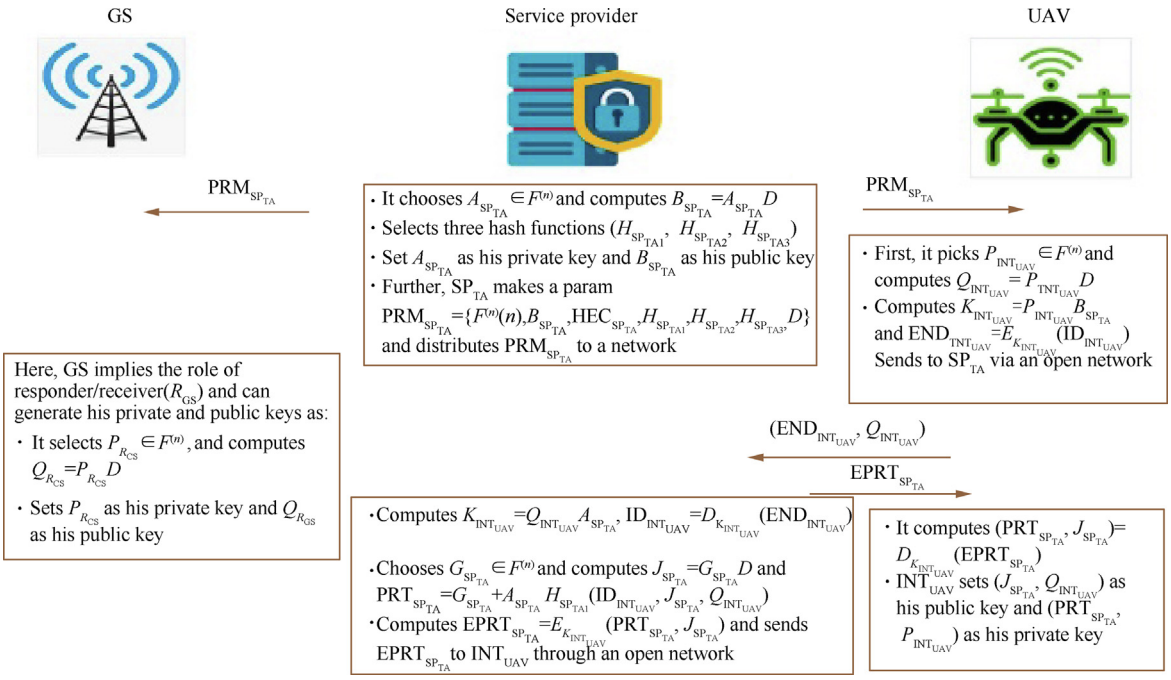

Fig. 2 Illustration of setup and key generation.

4.3. Authenticated key agreement

INT_{UAV} first chooses the symmetric key β from AES and picks $\delta_{INT_{UAV}} \in F^{(n)}$, computes $L_{INT_{UAV}} = \delta_{INT_{UAV}} D$, calculates $SK_{INT_{UAV}} = \delta_{INT_{UAV}} Q_{R_{GS}}$, generates a hash value as $H = H_{SP_{TA2}}(\beta, Q_{INT_{UAV}}, L_{INT_{UAV}}, ID_{INT_{UAV}})$, encrypts the symmetric key β as $C = H_{SP_{TA3}}(SK_{INT_{UAV}}, ID_{INT_{UAV}}) \oplus \beta$, computes $S_{INT_{UAV}} = \frac{P_{INT_{UAV}}}{PRT_{SP_{TA}} + H} \bmod(n)$, and then sends ($L_{INT_{UAV}}$, C, $S_{INT_{UAV}}$, T_{im}) to R_{GS}. When R_{GS} receives ($L_{INT_{UAV}}$, C, $S_{INT_{UAV}}$, T_{im}), it first checks the validity of T_{im}, then computes $SK_{R_{GS}} = L_{INT_{UAV}} P_{R_{GS}}$, recovers symmetric key β as $\beta = H_{SP_{TA3}}(SK_{INT_{UAV}}, ID_{INT_{UAV}}) \oplus C$, then checks if the following equation holds: $Q_{INT_{UAV}} = S_{INT_{UAV}}(J_{SP_{TA}} + VB_{SP_{TA}} + HD)$, where $V = H_{SP_{TA1}}(ID_{INT_{UAV}}, J_{SP_{TA}}, Q_{INT_{UAV}})$, then it sets β as the symmetric secret key between R_{GS} and INT_{UAV}. The illustration of authentication and key agreement is given in Fig. 3.

4.4. New device adding phase

This phase includes the four steps outlined below.

(1) New UAV Secret Value Generation (UAVSVG): Here, we represent a new UAV denoted by (INT_{NUAV}) capable of conducting the tasks below.
   (A) First, it picks $P_{INT_{NUAV}} \in F^{(n)}$ and computes $Q_{INT_{NUAV}} = P_{INT_{NUAV}} D$.
   (B) Computes $K_{INT_{NUAV}} = P_{INT_{NUAV}} B_{SP_{TA}}$ and $END_{INT_{NUAV}} = E_{K_{INT_{NUAV}}}(ID_{INT_{NUAV}})$.
   (C) Sends ($END_{INT_{NUAV}}$, $Q_{INT_{NUAV}}$) to SP_{TA} via an open network.

(2) New UAV Partial Private Key Generation (PPKGN): When SP_{TA} receives ($END_{INT_{UAV}}$, $Q_{INT_{UAV}}$), it performs the following steps.
   (A) Computes $K_{INT_{NUAV}} = Q_{INT_{NUAV}} A_{SP_{TA}}$ and recovers $ID_{INT_{NUAV}}$ as $ID_{INT_{NUAV}} = D_{K_{INT_{NUAV}}}(END_{INT_{NUAV}})$
   (B) Chooses $G_{SP_{TA}} \in F^{(n)}$, and then computes $J_{SP_{TA}} = G_{SP_{TA}} D$, and $PRT_{SP_{TA}} = G_{SP_{TA}} + A_{SP_{TA}} \cdot H_{SP_{TA1}}(ID_{INT_{NUAV}}, J_{SP_{TA}}, Q_{INT_{NUAV}})$
   (C) Computes $EPRT_{SP_{TA}} = E_{K_{INT_{NUAV}}}(PRT_{SP_{TA}}, J_{SP_{TA}})$ and sends $EPRT_{SP_{TA}}$ to INT_{NUAV} through an open network.

(3) New UAV Private and Public Key Generation (UPBG): When INT_{UAV} receives ($EPRT_{SP_{TA}}$), it performs the following steps.
   (A) It recovers $(PRT_{SP_{TA}}, J_{SP_{TA}})$ as $(PRT_{SP_{TA}}, J_{SP_{TA}}) = D_{K_{INT_{UAV}}}(EPRT_{SP_{TA}})$.
   (B) INT_{NUAV} sets $(J_{SP_{TA}}, Q_{INT_{NUAV}})$ as his public key and $(PRT_{SP_{TA}}, P_{INT_{NUAV}})$ as his private key.

An illustration of the addition of the new device is given in Fig. 4.

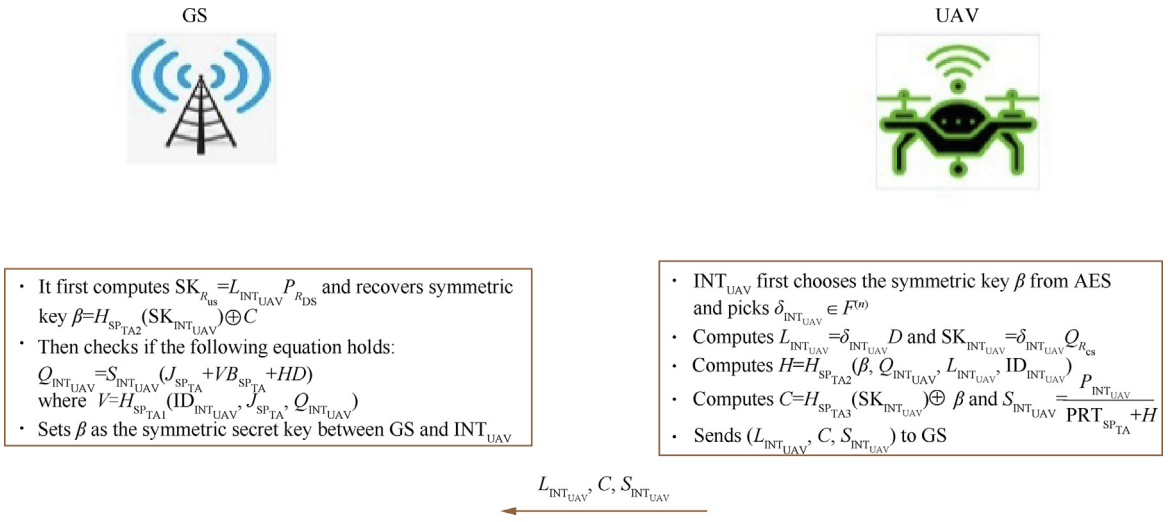

Fig. 3 Illustration of authentication and key agreement.

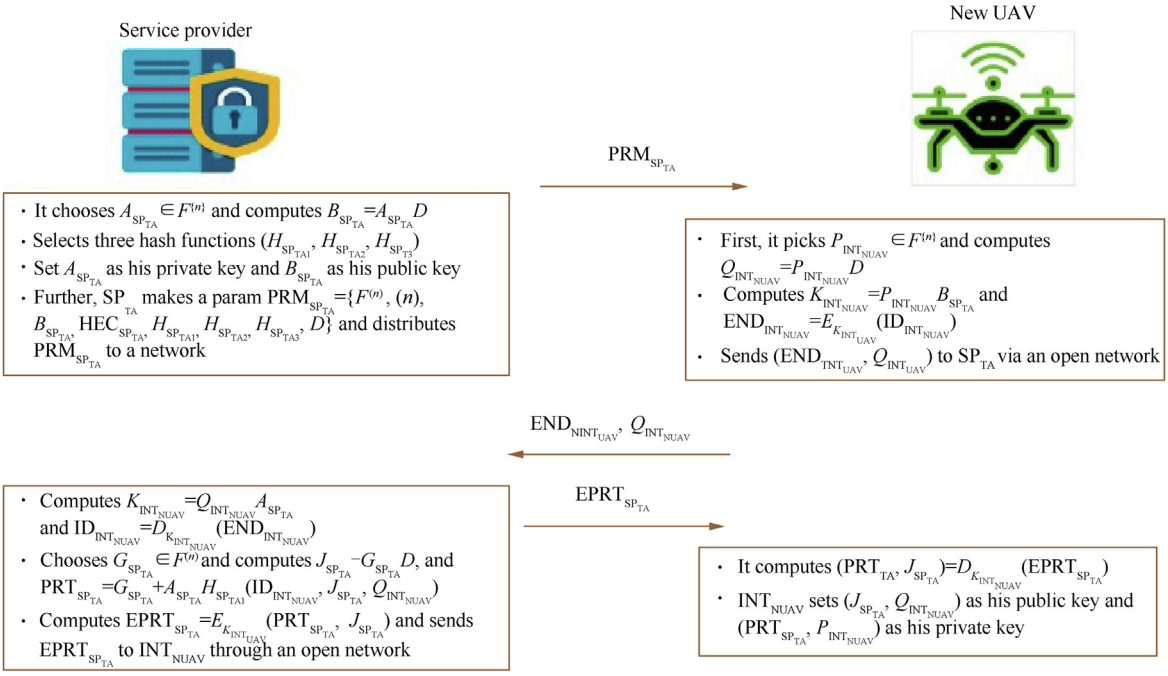

Fig. 4 Illustration of new device adding.

4.5. *Correctness*

The SP_{TA} can recover $K_{INT_{UAV}}$ by using the following computations.

$$K_{INT_{UAV}} = Q_{INT_{UAV}}A_{SP_{TA}} = P_{INT_{UAV}}DA_{SP_{TA}} = P_{INT_{UAV}}B_{SP_{TA}} = K_{INT_{UAV}}$$

The R_{GS} can prove $SK_{R_{GS}} = SK_{INT_{UAV}}$ by using the following computations.

$$SK_{R_{GS}} = SK_{INT_{UAV}} = L_{INT_{UAV}}P_{R_{GS}} = \delta_{INT_{UAV}}DP_{R_{GS}} = \delta_{INT_{UAV}}Q_{R_{GS}} = SK_{INT_{UAV}}$$

Using the following computations, the RGS can verify the triple $(LINTUAV, C, INTUAV)$ by verifying the triple.

$$
\begin{aligned}
Q_{\mathrm{INT_{UAV}}} &= S_{\mathrm{INT_{UAV}}}(J_{\mathrm{SP_{TA}}} + VB_{\mathrm{SP_{TA}}} + HD) \\
&= S_{\mathrm{INT_{UAV}}}(J_{\mathrm{SP_{TA}}} + VB_{\mathrm{SP_{TA}}} + HD\,) \\
&= \frac{P_{\mathrm{INT_{UAV}}}}{\mathrm{PRT_{SP_{TA}}} + H}(J_{\mathrm{SP_{TA}}} + VB_{\mathrm{SP_{TA}}} + HD) \\
&= \frac{P_{\mathrm{INT_{UAV}}}}{\mathrm{PRT_{SP_{TA}}} + H}(G_{\mathrm{SP_{TA}}}D + VB_{\mathrm{SP_{TA}}} + HD) \\
&= \frac{P_{\mathrm{INT_{UAV}}}}{\mathrm{PRT_{SP_{TA}}} + H}(G_{\mathrm{SP_{TA}}}D + VA_{\mathrm{SP_{TA}}}D + HD) \\
&= \frac{P_{\mathrm{INT_{UAV}}}D}{\mathrm{PRT_{SP_{TA}}} + H}(G_{\mathrm{SP_{TA}}} + A_{\mathrm{SP_{TA}}} \cdot H_{\mathrm{SP_{TA1}}} \\
&\quad (\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})A_{\mathrm{SP_{TA}}} + H\,) \\
&= \frac{P_{\mathrm{INT_{UAV}}}D}{\mathrm{PRT_{SP_{TA}}} + H}(\mathrm{PRT_{SP_{TA}}} + H) \\
&= P_{\mathrm{INT_{UAV}}}D = Q_{\mathrm{INT_{UAV}}}
\end{aligned}
$$

The R_{GS} can recovers β by using the following computations.

$$
\begin{aligned}
\beta &= H_{\mathrm{SP_{TA3}}}(\mathrm{SK_{INT_{UAV}}}) \oplus C \\
&= H_{\mathrm{SP_{TA3}}}(\mathrm{SK_{INT_{UAV}}}) \oplus H_{\mathrm{SP_{TA3}}}(\mathrm{SK_{INT_{UAV}}}) \oplus \beta = \beta
\end{aligned}
$$

5. Security analysis using ROM

To carry out the provable security analysis of the proposed scheme, which makes use of a well-known method of formal security analysis called the ROM, in which the adversary ($A_{\mathrm{INT_{GS1}}}, A_{\mathrm{INT_{GS2}}}$) and challenger ($C_{\mathrm{INT_{GS}}}$) rely on two main challenging problems, which are as follows.

Hyper Elliptic Curve Discrete Logarithm Problem ($\mathrm{HEC}_{D_{\mathrm{lp}}}$): Suppose $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}}D$, where $Y_{\mathrm{INT_{GS}}} \in F^{(n)}$, hence finding the value $Y_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$ is called $\mathrm{HEC}_{D_{\mathrm{lp}}}$.

Hyper Elliptic Curve Diffie-Hellman Problem ($\mathrm{HEC}_{D_{\mathrm{PH}}}$): Suppose $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}}X_{\mathrm{INT_{GS}}}D$, where $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}} \in F^{(n)}$, hence finding the values $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$ is called $\mathrm{HEC}_{D_{\mathrm{PH}}}$.

In the following theorems, we will demonstrate that the proposed scheme is secure against adversaries of types 1 and 2 ($A_{\mathrm{INT_{GS1}}}, A_{\mathrm{INT_{GS2}}}$) regarding confidentiality and unforgeability.

Theorem 1. *In this theorem, we explain how the challenger $C_{\mathrm{INT_{GS}}}$ helps the adversary $A_{\mathrm{INT_{GS1}}}$ to get the solution for $\mathrm{HEC}_{D_{\mathrm{PH}}}$. Furthermore, $A_{\mathrm{INT_{GS1}}}$ can perform several hash queries ($Q_{\mathrm{INT_{GSi}}}$) like $H_{\mathrm{SP_{TA1}}}, H_{\mathrm{SP_{TA2}}}, H_{\mathrm{SP_{TA3}}}$, a sender query($Q_{\mathrm{INT_{GSS}}}$), then gets the following adversary $A_{\mathrm{INT_{GS1}}}$: Avat* $^{\mathrm{HEC}_{D_{\mathrm{PH}}}}{}_{A_{\mathrm{INT_{GS1}}}} = \frac{E_{\mathrm{INT_{GS1}}}}{Q_{\mathrm{INT_{GS1}}}Q_{\mathrm{INT_{GS2}}}}(1 - \frac{1}{Q_{\mathrm{INT_{GSS}}}+1})^{Q_{\mathrm{INT_{GSS}}}}(\frac{1}{Q_{\mathrm{INT_{GSS}}}+1})$.

Proof: Suppose $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}}X_{\mathrm{INT_{GS}}}D$, where $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}} \in F^{(n)}$. The task of $C_{\mathrm{INT_{GS}}}$ is to perform the services of helper for $A_{\mathrm{INT_{GS1}}}$ for finding the values $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$. Hence, the following steps in which $A_{\mathrm{INT_{GS1}}}$ and $C_{\mathrm{INT_{GS}}}$ are performed to get the solution for $\mathrm{HEC}_{D_{\mathrm{PH}}}$.

Setup: $C_{\mathrm{INT_{GS}}}$ generates $\mathrm{SP_{TA}}$, $B_{\mathrm{SP_{TA}}}$, and $\mathrm{PRM_{SP_{TA}}}$ and sends $\mathrm{PRM_{SP_{TA}}}$ to $A_{\mathrm{INT_{GS1}}}$.

$H_{\mathrm{SP_{TA1}}}$ Query : When $A_{\mathrm{INT_{GS1}}}$ ask a query with triple $(\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$, then $C_{\mathrm{INT_{GS}}}$ checks for a tuple $(\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}}, H_{1i})$ in a list $L_{H_{1i}}$, if it exists, then $C_{\mathrm{INT_{GS}}}$ returns with a value H_{1i} to. Otherwise, it picks H_{1i} randomly, sends it to $A_{\mathrm{INT_{GS1}}}$, and updates the list $L_{H_{2i}}$ with a new value such as $(\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}}, H_{1i})$.

$H_{\mathrm{SP_{TA2}}}$ Query : When $A_{\mathrm{INT_{GS1}}}$ ask a query with triple $(\beta, Q_{\mathrm{INT_{UAV}}}, L_{\mathrm{INT_{UAV}}}, \mathrm{ID_{INT_{UAV}}})$, then $C_{\mathrm{INT_{GS}}}$ checks for a tuple $(\beta, Q_{\mathrm{INT_{UAV}}}, L_{\mathrm{INT_{UAV}}}, \mathrm{ID_{INT_{UAV}}}, H_{2i})$ in a list $L_{H_{2i}}$, if it exists, then $C_{\mathrm{INT_{GS}}}$ returns with value H_{2i} to. Otherwise, it picks H_{2i} randomly, and sends it to $A_{\mathrm{INT_{GS1}}}$, and updates the list $L_{H_{2i}}$ with the new value $(\beta, Q_{\mathrm{INT_{UAV}}}, L_{\mathrm{INT_{UAV}}}, \mathrm{ID_{INT_{UAV}}}, H_{2i})$.

$H_{\mathrm{SP_{TA3}}}$ Query : When $A_{\mathrm{INT_{GS1}}}$ ask a query with $(\mathrm{SK_{INT_{UAV}}})$, then $C_{\mathrm{INT_{GS}}}$ checks for a tuple $(\mathrm{SK_{INT_{UAV}}}, H_{3i})$ in a list $L_{H_{3i}}$, if it exists, then $C_{\mathrm{INT_{GS}}}$ returns with value H_{3i} to. Otherwise, it picks H_{3i} randomly, and sends it to $A_{\mathrm{INT_{GS1}}}$, and update the list $L_{H_{3i}}$ with the new value $(\mathrm{SK_{INT_{UAV}}}, H_{3i})$.

PPKGN Query : When $A_{\mathrm{INT_{GS1}}}$ ask a query with $(\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, J_{\mathrm{SP_{TA}}})$, then $C_{\mathrm{INT_{GS}}}$ check for a tuple $(\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, J_{\mathrm{SP_{TA}}})$, in a list L_{PPKGN}, if it exists, then $C_{\mathrm{INT_{GS}}}$ returns with value $\mathrm{PRT_{SP_{TA}}}$ to. Otherwise it picks $\mathrm{PRT_{SP_{TA}}}, J_{\mathrm{SP_{TA}}}$ randomly, and sends it to $\mathrm{PRT_{SP_{TA}}}$ to $A_{\mathrm{INT_{GS1}}}$, and update the list L_{PPKGN} with the new value $(\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, J_{\mathrm{SP_{TA}}})$.

UPBG Query : When $A_{\mathrm{INT_{GS1}}}$ ask a query with $(\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$, then $C_{\mathrm{INT_{GS}}}$ check for a tuple $(\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$, in a list L_{UPBG}, if it exists, then $C_{\mathrm{INT_{GS}}}$ returns with value $(\mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$, Otherwise, it picks $(\mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$ randomly, and sends it to $(\mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$ to $A_{\mathrm{INT_{GS1}}}$, and update the list L_{UPBG} with the new value $(\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$.

In the case of PKI key generation, when $A_{\mathrm{INT_{GS1}}}$ asks a query with $(\mathrm{ID_{INT_{UAV}}}, Q_{R_{\mathrm{GS}}}, P_{R_{\mathrm{GS}}})$, then $C_{\mathrm{INT_{GS}}}$ checks for a tuple $(\mathrm{ID_{INT_{UAV}}}, Q_{R_{\mathrm{GS}}}, P_{R_{\mathrm{GS}}})$, in a list L_{UPBG}, if it exists, then $C_{\mathrm{INT_{GS}}}$ returns with value $(Q_{R_{\mathrm{GS}}}, P_{R_{\mathrm{GS}}})$ to $A_{\mathrm{INT_{GS1}}}$, otherwise it picks $(Q_{R_{\mathrm{GS}}}, P_{R_{\mathrm{GS}}})$ randomly, and sends it to $(Q_{R_{\mathrm{GS}}}, P_{R_{\mathrm{GS}}})$ to $A_{\mathrm{INT_{GS1}}}$, and update the list L_{UPBG} with the new value $(\mathrm{ID_{INT_{UAV}}}, Q_{R_{\mathrm{GS}}}, P_{R_{\mathrm{GS}}})$.

Public key Replaced Query : When $A_{\mathrm{INT_{GS1}}}$ provides the new pair of public key $(J_{\mathrm{SP_{TA}}}{}^{*}, Q_{\mathrm{INT_{UAV}}}{}^{*})$ to $C_{\mathrm{INT_{GS}}}$ and $C_{\mathrm{INT_{GS}}}$ substitutes $(J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$ with $(J_{\mathrm{SP_{TA}}}{}^{*}, Q_{\mathrm{INT_{UAV}}}{}^{*})$.

Sender Query : When $A_{\mathrm{INT_{GS1}}}$ sends a plaintext and identity to $C_{\mathrm{INT_{GS}}}$, and it will perform the following: $C_{\mathrm{INT_{GS}}}$ retrieves $\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}}, Q_{R_{\mathrm{GS}}}$ and generate the signcryption text $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})$ for $A_{\mathrm{INT_{GS1}}}$. $C_{\mathrm{INT_{GS}}}$ fail and abort the process if $c = 1$.

Receiver Query: When $A_{\mathrm{INT_{GS1}}}$ sends a plaintext and identity to $C_{\mathrm{INT_{GS,}}}$ and it will perform the following : $C_{\mathrm{INT_{GS}}}$ retrieves $\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}}, Q_{R_{\mathrm{GS,}}} P_{R_{\mathrm{GS}}}$ from the list and generates the receiver text $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})$ decryption and send plaintext to $A_{\mathrm{INT_{GS1}}}$. If $c = 1$, it first computes $\mathrm{SK}_{R_{\mathrm{GS}}} = L_{\mathrm{INT_{UAV}}} \cdot P_{R_{\mathrm{GS}}}$, recovers symmetric key β as $\beta = H_{\mathrm{SP_{TA3}}}(\mathrm{SK_{INT_{UAV}}}) \oplus C$, then check if the following equation holds: $Q_{\mathrm{INT_{UAV}}} = S_{\mathrm{INT_{UAV}}}(J_{\mathrm{SP_{TA}}} + VB_{\mathrm{SP_{TA}}} + HD)$, where $V = H_{\mathrm{SP_{TA1}}}(\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$.

Challenge phase: $A_{\mathrm{INT_{GS1}}}$ returns with two exact sizes but different nature plaintext and a test identity to $C_{\mathrm{INT_{GS}}}$. If $c = 0$, then it means that it has found $(\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$ in $L_{H_{1i}}$; otherwise, it randomly generates a challenge cipher text $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})^{*}$ and send it to $A_{\mathrm{INT_{GS1}}}$.

Guess phase: $A_{\mathrm{INT_{GS1}}}$ performs the queries once again, and $C_{\mathrm{INT_{GS}}}$ generates a random bit as c^*, then sends it to $A_{\mathrm{INT_{GS1}}}$. Then, it tries to guess the original decryption key; if he succeeds in this case, therefore he/she will be able to solve the following problem: $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}} X_{\mathrm{INT_{GS}}} D$, where $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}} \in F^{(n)}$, hence, the task of $C_{\mathrm{INT_{GS}}}$ which perform the services of helper for $A_{\mathrm{INT_{GS1}}}$ is to find the values $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$. Therefore, if $c = 1$, the probability will be $\frac{1}{Q_{\mathrm{INT_{GSS}}}+1}$; If the partial private was accessed, the probability will be $\frac{1}{Q_{\mathrm{INT_{GS1}}}}$, and to solve $\mathrm{HEC}_{D_{\mathrm{PH}}}$. Hence, the total advantages in which $A_{\mathrm{INT_{GS1}}}$ and $C_{\mathrm{INT_{GS}}}$ succeed as $\mathrm{Avat}^{\mathrm{HEC}_{D_{\mathrm{PH}}}}_{\mathrm{A_{INT_{GS1}}}} = \frac{E_{\mathrm{INT_{GS1}}}}{Q_{\mathrm{INT_{GS1}}} Q_{\mathrm{INT_{GS2}}}} (1 - \frac{1}{Q_{\mathrm{INT_{GSS}}}+1})^{Q_{\mathrm{INT_{GSS}}}} (\frac{1}{Q_{\mathrm{INT_{GSS}}}+1})$. The illustration of Theorem 1 is given in Fig. 5.

Theorem 2. *In this theorem, we explain how the challenger* $C_{\mathrm{INT_{GS}}}$ *helps the adversary* $A_{\mathrm{INT_{GS2}}}$ *to get the solution for* $\mathrm{HEC}_{D_{\mathrm{PH}}}$. *Furthermore,* $A_{\mathrm{INT_{GS2}}}$ *can perform several hash queries* $(Q_{\mathrm{INT_{GSi}}})$ *like* $H_{\mathrm{SP_{TA1}}}, H_{\mathrm{SP_{TA2}}}, H_{\mathrm{SP_{TA3}}}$, *a sender query*$(Q_{\mathrm{INT_{GSS}}})$, *then we can get the following adversary* $A_{\mathrm{INT_{GS2}}}$: $\mathrm{Avat}^{\mathrm{HEC}_{D_{\mathrm{PH}}}}_{\mathrm{A_{INT_{GS1}}}} = \frac{E_{\mathrm{INT_{GS1}}}}{Q_{\mathrm{INT_{GS1}}} Q_{\mathrm{INT_{GS2}}}} (1 - \frac{1}{Q_{\mathrm{INT_{GSS}}}+1})^{Q_{\mathrm{INT_{GSS}}}} (\frac{1}{Q_{\mathrm{INT_{GSS}}}+1})$.

Proof: Suppose $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}} X_{\mathrm{INT_{GS}}} D$, where $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}} \in F^{(n)}$, so, the task of $C_{\mathrm{INT_{GS}}}$ which performs the services of helper for $A_{\mathrm{INT_{GS1}}}$ is to find the values $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$. Therefore, the following are steps in which $A_{\mathrm{INT_{GS1}}}$ and $C_{\mathrm{INT_{GS}}}$ are working to get a solution for $\mathrm{HEC}_{D_{\mathrm{PH}}}$.

(1) Setup: $C_{\mathrm{INT_{GS}}}$ generates $\mathrm{SP_{TA}}$, $B_{\mathrm{SP_{TA}}}$, and $\mathrm{PRM_{SP_{TA}}}$ and sends $\mathrm{PRM_{SP_{TA}}}$ and $\mathrm{SP_{TA}}$ to $A_{\mathrm{INT_{GS2}}}$.
(2) $H_{\mathrm{SP_{TAi}}}$ Query: These queries are identical to those performed in Theorem 1.
(3) PPKGN Query : This query is identical to the one performed in Theorem 1.
(4) UPBG Query : This query is identical to the one performed in Theorem 1.
(5) Sender Query : When $A_{\mathrm{INT_{GS2}}}$ sends a plaintext and identity to $C_{\mathrm{INT_{GS}}}$, then the following steps will be performed.
(6) $C_{\mathrm{INT_{GS}}}$ retrieves $\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}}, Q_{R_{\mathrm{GS}}}$ and generates the signcryption text $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})$ for $A_{\mathrm{INT_{GS2}}}$.

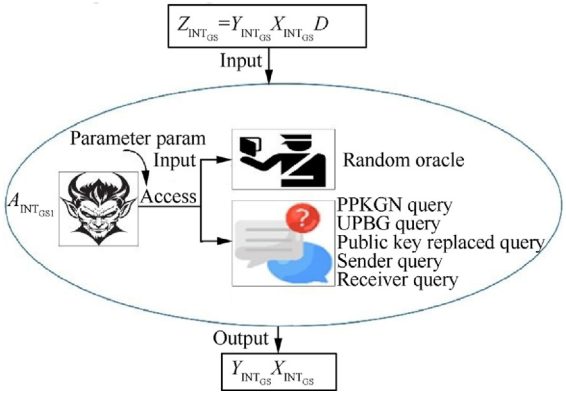

Fig. 5 Illustration of Theorem 1.

(7) $C_{\mathrm{INT_{GS}}}$ failed and aborted the process if $c = 1$.
(8) Reciever Query : When $A_{\mathrm{INT_{GS2}}}$ sends a plaintext and identity to $C_{\mathrm{INT_{GS}}}$ then it will perform the following steps.
(9) $C_{\mathrm{INT_{GS}}}$ retrieves $\mathrm{ID_{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}, P_{\mathrm{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}}, Q_{R_{\mathrm{GS}}}, P_{R_{\mathrm{GS}}}$ from the list and generates the plaintext from $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})$ and then sends the plaintext to $A_{\mathrm{INT_{GS2}}}$.
(10) If $c = 1$, then, it first computes $\mathrm{SK}_{R_{\mathrm{GS}}} = L_{\mathrm{INT_{UAV}}}$. Recovers symmetric key β as $\beta = H_{\mathrm{SP_{TA3}}}(\mathrm{SK_{INT_{UAV}}}) \oplus C$, then checks if the following equation holds: $Q_{\mathrm{INT_{UAV}}} = S_{\mathrm{INT_{UAV}}}(J_{\mathrm{SP_{TA}}} + VB_{\mathrm{SP_{TA}}} + HD)$, where $V = H_{\mathrm{SP_{TA1}}}(\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$.

Challenge phase: $A_{\mathrm{INT_{GS2}}}$ returns two plaintexts of the same size but distinct types, as well as a test identity to $C_{\mathrm{INT_{GS}}}$. If $c = 0$, it means that $(\mathrm{ID_{INT_{UAV}}}, J_{\mathrm{SP_{TA}}}, Q_{\mathrm{INT_{UAV}}})$ has been found in L_{Hli}; otherwise, it randomly generates a challenge cipher text $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})^*$ and sends it to $A_{\mathrm{INT_{GS2}}}$.

Guess phase: $A_{\mathrm{INT_{GS2}}}$ performs the queries once again, and $C_{\mathrm{INT_{GS}}}$ generates a random bit c^*, then sends it to $A_{\mathrm{INT_{GS2}}}$. Then, if he is successful, he must try to guess the original decryption key so that he can solve the following problem: $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}} X_{\mathrm{INT_{GS}}} D$, where $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}} \in F^{(n)}$, so, the task of $C_{\mathrm{INT_{GS}}}$ which perform the services of helper for $A_{\mathrm{INT_{GS2}}}$ is to find the values $Y_{\mathrm{INT_{GS}}}, X_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$. So, if $c = 1$, then the probability will be if the partial private is searched, then the probability will be $\frac{1}{Q_{\mathrm{INT_{GS1}}}}$, and to solve $\mathrm{HEC}_{D_{\mathrm{PH}}} \frac{1}{Q_{\mathrm{INT_{GS3}}}}$. So the total advantages in which $A_{\mathrm{INT_{GS2}}}$ and $C_{\mathrm{INT_{GS}}}$ succeed as $\mathrm{Avat}^{\mathrm{HEC}_{D_{\mathrm{PH}}}}_{\mathrm{A_{INT_{GS1}}}} = \frac{E_{\mathrm{INT_{GS1}}}}{Q_{\mathrm{INT_{GS1}}} Q_{\mathrm{INT_{GS2}}}} \cdot (1 - \frac{1}{Q_{\mathrm{INT_{GSS}}}+1})^{Q_{\mathrm{INT_{GSS}}}} (\frac{1}{Q_{\mathrm{INT_{GSS}}}+1})$. The illustration of Theorem 2 is given in Fig. 6.

Theorem 3. *In this theorem, we explain how the challenger* $C_{\mathrm{INT_{GS}}}$ *helps the adversary* $A_{\mathrm{INT_{GS1}}}$ *to get the solution for* $\mathrm{HEC}_{D_{\mathrm{lp}}}$. *Furthermore,* $A_{\mathrm{INT_{GS1}}}$ *can perform several hash queries* $(Q_{\mathrm{INT_{GSi}}})$ *like* $H_{\mathrm{SP_{TA1}}}, H_{\mathrm{SP_{TA2}}}, H_{\mathrm{SP_{TA3}}}$, *a sender query*$(Q_{\mathrm{INT_{GSS}}})$. *Then, we can get the following adversary* $\mathrm{A_{INT_{GS1}}}$: $\mathrm{Avat}^{\mathrm{HEC}_{D_{\mathrm{PH}}}}_{\mathrm{A_{INT_{GS1}}}} = \frac{E_{\mathrm{INT_{GS1}}}}{Q_{\mathrm{INT_{GS1}}}} (1 - \frac{1}{Q_{\mathrm{INT_{GSS}}}+1})^{Q_{\mathrm{INT_{GSS}}}}$

Proof: Suppose $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}} D$, where $Y_{\mathrm{INT_{GS}}} \in F^{(n)}$, so, the task of $C_{\mathrm{INT_{GS}}}$ which performs the services of helper for $A_{\mathrm{INT_{GS1}}}$ is to find the values $Y_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$. Therefore, the following are steps in which $A_{\mathrm{INT_{GS1}}}$ and $C_{\mathrm{INT_{GS}}}$ are working to get the solution for $\mathrm{HEC}_{D_{\mathrm{lp}}}$.

(1) Setup: $C_{\mathrm{INT_{GS}}}$ generates $\mathrm{SP_{TA}}$, $B_{\mathrm{SP_{TA}}}$, and $\mathrm{PRM_{SP_{TA}}}$ and send $\mathrm{PRM_{SP_{TA}}}$ to $A_{\mathrm{INT_{GS1}}}$.
(2) Queries: All queries are identical to those performed in Theorem 1.

Forgery: After performing the above queries, $A_{\mathrm{INT_{GS1}}}$ can produce two sets of ciphertext $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})^*$ and $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})^{**}$. To do this, he must have the partial private key and the sender and receiver private keys. The probability analysis thus proceeds as follows: If $c = 1$, then the probability will be $\frac{1}{Q_{\mathrm{INT_{GS1}}}}$, and to solve $\mathrm{HEC}_{D_{\mathrm{PH}}} \frac{1}{Q_{\mathrm{INT_{GS3}}}}$, there-

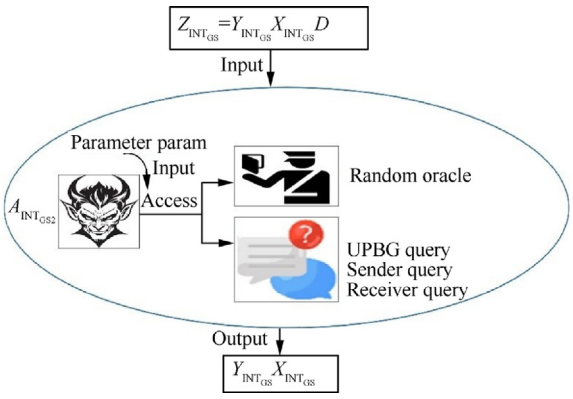

Fig. 6 Illustration of Theorem 2.

fore, the total advantages in which $A_{\mathrm{INT_{GS1}}}$ and $C_{\mathrm{INT_{GS}}}$ succeed as $\mathrm{Avat}^{\mathrm{HEC_{D_{PH}}}}_{\mathrm{A_{INT_{GS1}}}} = \frac{1}{Q_{\mathrm{INT_{GS1}}}}(1-\frac{1}{Q_{\mathrm{INT_{GSS}}}+1})^{Q_{\mathrm{INT_{GSS}}}}$.

Theorem 4. *In this theorem, we explain how the challenger* $C_{\mathrm{INT_{GS}}}$ *helps the adversary* $A_{\mathrm{INT_{GS2}}}$ *to get the solution for* $\mathrm{HEC}_{D_{\mathrm{lp}}}$. *Furthermore,* $A_{\mathrm{INT_{GS2}}}$ *can perform several hash queries* $(Q_{\mathrm{INT_{GSi}}})$ *like* $H_{\mathrm{SP_{TA1}}}, H_{\mathrm{SP_{TA2}}}, H_{\mathrm{SP_{TA3}}}$, *a sender query*$(Q_{\mathrm{INT_{GSS}}})$, *then we can get the following adversary* $A_{\mathrm{INT_{GS2}}}$: $\mathrm{Avat}^{\mathrm{HEC_{D_{PH}}}}_{\mathrm{A_{INT_{GS1}}}} = \frac{E_{\mathrm{INT_{GS1}}}}{Q_{\mathrm{INT_{GS1}}}}(1-\frac{1}{Q_{\mathrm{INT_{GSS}}}+1})^{Q_{\mathrm{INT_{GSS}}}}$.

Proof: Suppose $Z_{\mathrm{INT_{GS}}} = Y_{\mathrm{INT_{GS}}}D$, where $Y_{\mathrm{INT_{GS}}} \in F^{(n)}$, so, the task of $C_{\mathrm{INT_{GS}}}$ which performs the services of helper for $A_{\mathrm{INT_{GS2}}}$ and is to find the values $Y_{\mathrm{INT_{GS}}}$ from $Z_{\mathrm{INT_{GS}}}$. To get the solution for $\mathrm{HEC}_{D_{\mathrm{lp}}}$, $A_{\mathrm{INT_{GS2}}}$ and $C_{\mathrm{INT_{GS}}}$ are worked out in the following phases.

(1) Setup: $C_{\mathrm{INT_{GS}}}$ generates $\mathrm{SP_{TA}}$, $\mathrm{B_{SP_{TA}}}$, and $\mathrm{PRM_{SP_{TA}}}$ and send $\mathrm{PRM_{SP_{TA}}}$ and $\mathrm{SP_{TA}}$ to $A_{\mathrm{INT_{GS2}}}$.
(2) Queries: All queries are identical to those performed in Theorem 2.

Forgery: After performing the above queries, $A_{\mathrm{INT_{GS1}}}$ can produce two sets of ciphertext $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})^{*}$ and $(L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}})^{**}$. For this purpose, he/she must get the sender and receiver's partial private and private keys. The probability analysis thus proceeds as follows: If $c = 1$, then the probability will be if the partial private key accesses, then the probability will be $\frac{1}{Q_{\mathrm{INT_{GS1}}}}$, and to solve $\mathrm{HEC}_{D_{\mathrm{PH}}}\frac{1}{Q_{\mathrm{INT_{GS3}}}}$, therefore, the total advantages in which $A_{\mathrm{INT_{GS1}}}$ and $C_{\mathrm{INT_{GS}}}$ success will be as

$$\mathrm{Avat}^{\mathrm{HEC_{D_{PH}}}}_{\mathrm{A_{INT_{GS1}}}} = \frac{1}{Q_{\mathrm{INT_{GS1}}}}\left(1-\frac{1}{Q_{\mathrm{INT_{GSS}}}+1}\right)^{Q_{\mathrm{INT_{GSS}}}} \quad (1)$$

6. Security analysis using ROR model

We employed the popular "Real-Or-Random" (ROR) oracle model to demonstrate that the proposed scheme has semantic security (secret key security). In this model, we only examine how the proposed scheme's secret key will be protected from (ξ = Type 1 ($A_{\mathrm{INT_{GS1}}}$) and Type 2 ($A_{\mathrm{INT_{GS2}}}$) attackers.

Suppose ξ wants to intercept the communication between nodes N_1 and N_2. We further suppose that the instances of 1 and N_2 are represented as $N = N_1$ and N_2.

(1) Execute query: In this query, ξ can capture the communication between N_1 and N_2.
(2) Send a query: In this query, can communicate with N and get a response accordingly.
(3) Reveal query: In this query, ξ attempts to construct or reveal the session secret key of N_1 and N_2.
(4) Test query: In this query, ξ demands N for the secret session key ($\mathscr{G}$) and replies with an arbitrary bit $\mathscr{J}$.

Moreover, we consider the publicly accessible hash function for the random oracle. Consequently, utilizing the following Theorem 3, we can implement the security resistance against the proposed scheme's secret key.

Theorem 5. *Suppose* ξ *wants to break secret session key* ($\mathscr{G}$) *security between* N_1 *and* N_2 *utilizing polynomial time* ($\mathfrak{T}$). Therefore, *the breaking advantages of* ξ *are as follows:*

$$\xi^{\mathrm{Advantage}}(\mathfrak{T}) \le \frac{\theta \mathrm{H}}{|\mathrm{hash}|} + 2\xi^{\mathrm{HECDLP}}(\mathfrak{T}) \quad (2)$$

where $|\mathrm{hash}|, \theta\mathrm{H}$, $\xi^{\mathrm{HECDLP}}(\mathfrak{T})$ denotes space for a hash function, requested hash queries, and un-ignored efforts of ξ to get the solution of HECDLP. The following games illustrate the solution to Theorem 5.

Game 1. It resembles a real approach that can be implemented in the ROR model, allowing us to obtain the following results.

$$\xi^{\mathrm{Advantage}}(\mathfrak{T}) \le |2\xi^{\mathrm{HECDLP}}(\mathrm{Game\,1}) - 1| \quad (3)$$

Game 2. In this Game 2, all the conversations $((\varphi, \omega)$ and $(C))$ between N_1 and N_2 intercepts by ξ, then it can apply to execute, reveal, and test query in the intention for recovering secret session key (β) in real or random form. The secret key is encrypted as, i.e., $= H_{\mathrm{SP_{TA3}}}(\mathrm{SK_{INT_{UAV}}}) \oplus \beta$. Here, $\mathrm{SK_{INT_{UAV}}} = \delta_{\mathrm{INT_{UAV}}}Q_{R_{\mathrm{GS}}}$, however, struggle to find $\delta_{\mathrm{INT_{UAV}}}$, which is challenging due to the difficulty of finding a solution for the hyperelliptic curve discrete logarithm problem. Hence, in the following Eq. (4), we represent the indistinguishability of Game 1 and Game 2.

$$(\mathrm{Game\,1}) = \xi^{\mathrm{Advantage}}(\mathrm{Game\,2}) \quad (4)$$

Game 3. In this Game 3, we will focus on the collision resistance property of the hash function during conversations $L_{\mathrm{INT_{UAV}}}, C, S_{\mathrm{INT_{UAV}}}$ between N_1 and. All the messages are protected through the hash function and HECDLP; for example, $S_{\mathrm{INT_{UAV}}} = \frac{P_{\mathrm{INT_{UAV}}}}{\mathrm{PRT_{SP_{TA}}}+H}$. If ξ attempts to generate a forge one, then he/she must extract $P_{\mathrm{INT_{UAV}}}, \mathrm{PRT_{SP_{TA}}}$, and H, that are normally equals to process two-time hyperelliptic curve discrete logarithm problem and same hash value, which is difficult. So, the advantages of solving HECDLP of ξ denoted by $\xi^{\mathrm{HECDLP}}(\mathfrak{T})$ and using the concept of birthday paradox, the probability for hash query denoted by $\frac{\theta \mathrm{H}}{2|\mathrm{hash}|}$, so we get the following outcomes.

$$\xi^{\text{Advantage}}(\text{Game 2}) - \xi^{\text{Advantage}}(\text{Game 3}) \leq \frac{\theta H}{2|\text{hash}|} + \xi^{\text{HECDLP}}(\Upsilon) \tag{5}$$

When ξ can perform all the expected queries, it can get the following results for guessing arbitrary bits $\mathscr{J}$.

$$\xi^{\text{Advantage}}(\text{Game 2}) = \frac{1}{2} \tag{6}$$

We can get the following results from Eqs. (3) and (4).

$$\begin{aligned} \frac{1}{2}\xi^{\text{HECDLP}}(\Upsilon) &= \left| \xi^{\text{Advantage}}(\text{Game 1}) - \frac{1}{2} \right| \\ &= \left| \xi^{\text{Advantage}}(\text{Game 2}) - \frac{1}{2} \right| \end{aligned} \tag{7}$$

We can get the following results From Eqs. (6) and (7).

$$\frac{1}{2}^{\text{HECDLP}}(\Upsilon) = 667em| \xi^{\text{Advantage}}(\text{Game 2}) - \xi^{\text{Advantage}}(\text{Game 3}) \tag{8}$$

We can conclude the following from Eqs. (4) and (8).

$$\frac{1}{2}\xi^{\text{HECDLP}}(\Upsilon) = \frac{\theta H}{2|\text{hash}|} + \xi^{\text{HECDLP}}(\Upsilon) \tag{9}$$

Multiplying 2 by both sides of the Eq. (9) yields the following results.

$$\xi^{\text{HECDLP}}(\Upsilon) = \frac{\theta H}{|\text{hash}|} + 2\xi^{\text{HECDLP}}(\Upsilon)$$

7. Informal security analysis

The proposed scheme provides the following security requirements.

(1) Unforgeability: Our scheme provides the unforgeability of signature. In an event when an attacker wants to generate a forged signature $S_{\text{INT}_{\text{UAV}}} = \frac{P_{\text{INT}_{\text{UAV}}}}{\text{PRT}_{\text{SP}_{\text{TA}}} + H} \bmod(n)$ in authentication phase, for this, he/she need two unknown values $\text{PRT}_{\text{SP}_{\text{TA}}}$ which is only known to UAV, and it is mathematically not feasible to solve the Hard problem like hyperelliptic curve discrete logarithm problem. From the above discussions, it is proved that the proposed scheme provides the unforgeability of a signature.
(2) Secret Key Security: Our scheme provides the secret key security. In an event when an attacker wants to generate a secret key $(\text{SK}_{\text{INT}_{\text{UAV}}} = \delta_{\text{INT}_{\text{UAV}}} Q_{R_{\text{GS}}})$ in authentication phase, for this, he/she needs two unknown values $\delta_{\text{INT}_{\text{UAV}}}$ which is only known to UAV, and it is mathematically not feasible to solve the hard problem like HECDLP. From the above discussions, it is proved that the proposed scheme provides the secret key security.
(3) Resist against Reply Attack: The proposed scheme is safeguarded from reply attack. In the communication of proxy signature the UAV sends $(L_{\text{INT}_{\text{UAV}}}, C, S_{\text{INT}_{\text{UAV}}}, T_{\text{im}})$ as a signature tuple, where T_{im} represents a time stamp. When $(L_{\text{INT}_{\text{UAV}}}, C, S_{\text{INT}_{\text{UAV}}}, T_{\text{im}})$ is received to R_{GS}, it can check the validity of T_{im}, if it is valid, it can verify the signature.
(4) Resist against Impersonation Attacks: The proposed scheme is safeguarded from impersonation attacks. In CL-Key Generation phase, the UAV, which is acting as the initiator/sender that could be represented as $(\text{INT}_{\text{UAV}})$, picks $P_{\text{INT}_{\text{UAV}}} \in F^{(n)}$, compute $Q_{\text{INT}_{\text{UAV}}} = P_{\text{INT}_{\text{UAV}}} D$, $K_{\text{INT}_{\text{UAV}}} = P_{\text{INT}_{\text{UAV}}} B_{\text{SP}_{\text{TA}}}$, $\text{END}_{\text{INT}_{\text{UAV}}} = E_{\text{K}_{\text{INT}_{\text{UAV}}}}(\text{ID}_{\text{INT}_{\text{UAV}}})$, sends $(\text{END}_{\text{INT}_{\text{UAV}}}, Q_{\text{INT}_{\text{UAV}}})$ to SP_{TA} via an open network, and stores the tuple for $(\text{END}_{\text{INT}_{\text{UAV}}}, Q_{\text{INT}_{\text{UAV}}})$ in his database for future use. When an attacker uses some identity for extracting the partial public and private key, then it could fail because SP_{TA} has the records of all previous identities.
(5) Man-In-the-Middle Attack: In our proposed scheme, the attacker will not be able to impersonate the legitimate user if the attacker wants to launch a Man-in-the-Middle attack by accessing the identity of the initiator/sender, which could be represented as $(\text{INT}_{\text{UAV}})$, then he first decrypt $\text{END}_{\text{INT}_{\text{UAV}}}$ as $\text{ID}_{\text{INT}_{\text{UAV}}} = D_{\text{K}_{\text{INT}_{\text{UAV}}}}(\text{END}_{\text{INT}_{\text{UAV}}})$ and this process the attacker needs to compute $K_{\text{INT}_{\text{UAV}}} = P_{\text{INT}_{\text{UAV}}} B_{\text{SP}_{\text{TA}}}$, which is not feasible for him to extract the private number $P_{\text{INT}_{\text{UAV}}}$ from $Q_{\text{INT}_{\text{UAV}}} = P_{\text{INT}_{\text{UAV}}} D$ that leads to solving a hard problem called hyperelliptic curve discrete logarithm problem.

8. Performance comparison

This section is devoted to the performance comparison of the proposed scheme with existing equivalents schemes, such as those of which were suggested by Wang et al.[17], Tao et al. [18], Liu et al. [19], Pan et al. [20], Aithekar et al. [21], and Xie et al. [22] in terms of computation, communication costs, and security requirements. The results of both sets of comparisons are then presented in graphs and tables to provide a better understanding of the viability of the proposed scheme. Tables 3 and 4 present a comparison of computation costs, Table 5 outlines communication costs, and Table 6 highlights security requirements.

8.1. Computation cost

The comparison of the computation cost is presented in Tables 3 and 4. These tables include performance based on the operations of computation cost expressed in major operations and milliseconds, respectively, for the proposed scheme and the methods presented by Wang et al.[17], Tao et al. [18], Liu et al. [19], Pan et al. [20], Aithekar et al. [21], and Xie et al. [22]. The experiment was conducted to compute the time required for a single major operation. The results indicate that INT_{TBP} needs 1.02 ms, INT_{TEXP} requires 0.84 ms, INT_{TESM} needs 0.64 ms, and INT_{THSM} requires 0.36 ms.[24] The experiment was carried out employing both hardware and software configurations. Specifically, the system model was a MacBook Pro-2015 equipped with 16 GB 1 600 MHz DDR3 RAM, a 2.2 GHz quad-core Intel Core i7 processor, and 512 GB of storage. The employed operating system was macOS Big Sur 11.4. Furthermore, the investigation employed iFogSim 1.0, libg2hec 2.1, NTL 5.5, MIRACL 7.0.0, PBC 1.0, and GMP 6.2.1. The fog client application is

Table 3 Comparison of computation cost with primary operations.

Scheme	Sender	Receiver	Total
Wang et al. [17]	$6INT_{TBP} + 7INT_{TEXP}$	$6INT_{TBP} + 2INT_{TEXP}$	$12INT_{TBP} + 9INT_{TEXP}$
Tao et al. [18]	$5INT_{TBP} + 3INT_{TEXP}$	$3INT_{TBP} + 2INT_{TEXP}$	$8INT_{TBP} + 5INT_{TEXP}$
Liu et al. [19]	$4INT_{TESM}$	$4INT_{TESM}$	$8INT_{TESM}$
Pan et al. [20]	$2INT_{TEXP}$	$2INT_{TBP} + 3INT_{TEXP}$	$2INT_{TBP} + 5INT_{TEXP}$
Aithekar et al. [21]	$3INT_{TESM}$	$3INT_{TESM}$	$6INT_{TESM}$
Xie et al. [22]	$3INT_{TBP} + 2INT_{TEXP}$	$2INT_{TBP} + 1INT_{TEXP}$	$5INT_{TBP} + 3INT_{TEXP}$
Proposed	$2INT_{THSM}$	$4INT_{THSM}$	$6INT_{THSM}$

Note: INT_{TBP}: Time needs for a single bilinear pairing operation, INT_{TSM} : Time needs for single multiplication operation on bilinear pairing, INT_{TESM} : Time needs for single multiplication operation on EC, INT_{THSM} : Time needs for single multiplication operation on HEC, INT_{TEXP} : exponential operation

Table 4 Comparison of computation cost (in ms).

Scheme	Sender	Receiver	Total
Wang et al. [17]	12	7.8	19.8
Tao et al. [18]	7.62	4.74	12.36
Liu et al. [19]	2.56	2.56	5.12
Pan et al. [20]	1.68	4.56	6.24
Aithekar et al. [21]	1.92	1.92	3.84
Xie et al. [22]	4.74	2.88	7.62
Proposed	0.72	1.44	2.16

Table 5 Comparison of communication cost with major operations.

Scheme	Ciphertext size	Ciphertext size (bits)
Wang et al. [17]	$10\|INT_G\| + 2\|m\|$	10,496
Tao et al. [18]	$8\|INT_G\| + 4\|m\|$	8704
Liu et al. [19]	$16\|INT_q\|$	2560
Pan et al. [20]	$5\|INT_G\|$	5120
Aithekar et al. [21]	$4\|INT_q\| + \|m\|$	768
Xie et al. [22]	$5\|INT_G\|$	5120
Our Scheme	$2\|INT_n\| + \|m\|$	320

Note: INT_G is the parameter size belongs to the bilinear group and $INT_G = 1024$, m is a message and we assume its size is 128 bits, INT_q : the parameter size belongs to elliptic curve group and $INT_q = 160$ bits, and $INT_n$is the parameter size belongs to hyperelliptic curve group and $INT_n = 80$ bits.

developed with the aid of the iFogSim simulator. The cost of the main operation, hyperelliptic curve divisor multiplication (HDM), is computed using the application libg2hec, which is accessible on GitHub. The libg2hec library provides a divisor group operation on the Jacobian of a hypothetical genus 2-hyperelliptic curve. The NTL library of V. Shoup is required to operate the G2HEC library efficiently. The G2HEC library's results were evaluated with NTL 5.5 on macOS Mac fat 11.4 x86_64. The Multi-Precision Integer and Rational Arithmetic C Library (MIRACL) is used to test the runtime of the basic cryptographic operations up to 1000 times to measure the performance of the proposed scheme. With the aid of the pairing-based cryptography library PBC, bilinear pairing modular exponential (INT_{TEXP}) and bilinear pairing operation (INT_{TBP}) costs can be calculated. PBC is built on the GMP library, which handles all pairing-based point multiplication and bilinear pairing modular exponential operations. With type A bilinear pairing over an elliptic curve with $p = 512$ bits and $q = 160$ bits, we can attain RSA-1024-level security. ECC can use either the Secp160r1 curve over a finite field or the genus two hyperelliptic curve over a finite field with the following values: $|F^{(n)}|g \log_2(n) \approx 2^{80}$ to obtain the same level of security. In comparison with the existing schemes proposed by Wang et al.[17], Tao et al. [18], Liu et al. [19], Pan et al. [20], Aithekar et al. [21], and Xie et al. [22]. Table 3 presents the data com-

Table 6 Comparison of security requirements.

Scheme	Unforgeability	Security of Secret Key	Resistance against replay attack	Resistance against Impersonation attack	Resistance against Man in The Middle attack
Wang et al. [17]	Yes	Yes	No	No	No
Tao et al. [18]	Yes	Yes	No	No	No
Liu et al. [19]	Yes	Yes	No	No	No
Pan et al. [20]	Yes	Yes	No	No	No
Aithekar et al. [21]	Yes	Yes	No	No	No
Xie et al. [22]	Yes	Yes	No	No	No
proposed	Yes	Yes	Yes	Yes	Yes

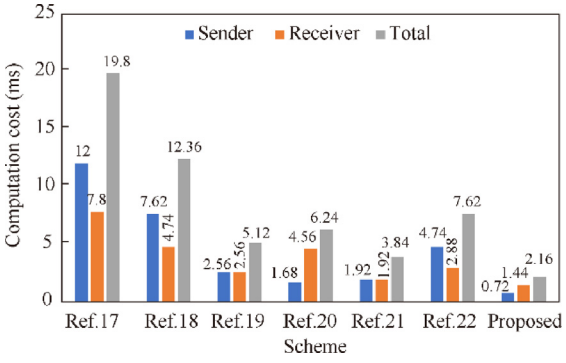

Fig. 7 Comparison of computation cost.

paring computation costs, further visualized in Fig. 7 and detailed in milliseconds in Table 4. The results show that the proposed scheme has a lower computation cost.

8.2. Communication cost

In this subsection of communication cost, the fundamental operations of communication cost for the proposed scheme and other comparable schemes proposed by Wang et al.[17], Tao et al. [18], Liu et al. [19], Pan et al. [20], Aithekar et al. [21], and Xie et al. [22]. Table 5 presents the data comparing communication costs, further visualized in Fig. 8. The proposed scheme has a lower communication cost than the schemes proposed by Wang et al.[17], Tao et al. [18], Liu et al. [19], Pan et al. [20], Aithekar et al. [21], and Xie et al. [22]. This comparison demonstrates the feasibility of the proposed scheme.

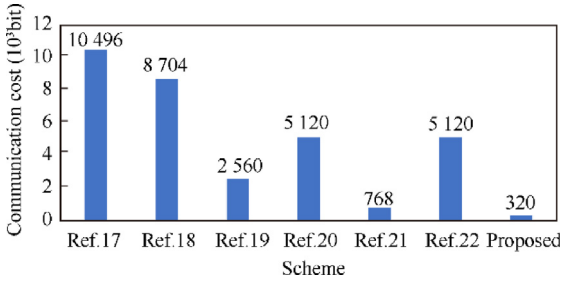

Fig. 8 Comparison of communication cost.

9. Conclusions

UAVs, typically not designed with security, are susceptible to fundamental privacy and security concerns. These concerns are because UAVs usually communicate through open wireless channels and have limited computing resources. The primary objective of developing a cryptographic scheme is to secure the communications of UAVs while concurrently reducing the cost of computation and communication. In this article, we proposed an efficient and secure cross-domain AKA scheme based on HECC to satisfy the criteria of low computation and communication costs. The ECC has been further improved by introducing the HECC, which has smaller parameters and key sizes. In contrast to the ECC, which requires a key length of 160 bits, the key size is restricted to a maximum of 80 bits, making it an excellent choice for UAVs with limited onboard computing resources. The formal security analysis conducted with the ROM and ROR models, along with the comparative analysis in computation and communication costs, conclusively demonstrated the improved security and efficiency of the proposed scheme compared to its equivalent existing schemes.

CRediT authorship contribution statement

Muhammad Asghar KHAN: Writing – review & editing, Writing – original draft, Validation, Formal analysis, Conceptualization. **Insaf ULLAH:** Writing – review & editing, Validation, Methodology, Conceptualization. **Haralambos MOURATIDIS:** Writing – review & editing, Supervision, Formal analysis. **Abdulmajeed ALQHATANI:** Writing – review & editing, Writing – original draft, Resources, Project administration. **Pascal LORENZ:** Software, Investigation, Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Khan MA, Kumar N, Mohsan SAH, et al. Swarm of UAVs for network management in 6G: a technical review. *IEEE Trans Netw Serv Manag* 2023;**20**(1):741–61.
2. Mozaffari M, Saad W, Bennis M, et al. A tutorial on UAVs for wireless networks: applications, challenges, and open problems. arXiv preprint: 1803.00680; 2018.
3. Hayat S, Yanmaz E, Muzaffar R. Survey on unmanned aerial vehicle networks for civil applications: a communications viewpoint. *IEEE Commun Surv Tutor* 2016;**18**(4):2624–61.
4. He DJ, Chan S, Guizani M. Communication security of unmanned aerial vehicles. *IEEE Wirel Commun* 2017;**24**(4):134–9.
5. Ullah I, Khan MA, Abdullah AM, et al. A conditional privacy preserving generalized ring signcryption scheme for micro aerial vehicles. *Micromachines* 2022;**13**(11):1926.
6. Ko Y, Kim J, Duguma DG, et al. Drone secure communication protocol for future sensitive applications in military zone. *Sensors* 2021;**21**(6):2057.
7. Khan MA, Ullah I, Alkhalifah A, et al. A provable and privacy-preserving authentication scheme for UAV-enabled intelligent transportation systems. *IEEE Trans Ind Inform* 2022;**18**(5):3416–25.
8. Leela Krishna CG, Murphy RR. A review on cybersecurity vulnerabilities for unmanned aerial vehicles. *2017 IEEE international symposium on safety, security and rescue robotics (SSRR)*. Piscataway: IEEE Press; 2017.
9. Khan MA, Alzahrani BA, Barnawi A, et al. A resource friendly authentication scheme for space–air–ground–sea integrated Maritime Communication Network. *Ocean Eng* 2022;**250**:110894.
10. Ullah S, Marcenaro L, Rinner B. Secure smart cameras by aggregate-signcryption with decryption fairness for multi-receiver IoT applications. *Sensors* 2019;**19**(2):327.
11. Bansal U, Kar J, Ali I, et al. ID-CEPPA: identity-based computationally efficient privacy-preserving authentication scheme for vehicle-to-vehicle communications. *J Syst Archit* 2022;**123**:102387.
12. Al-Riyami SS, Paterson KG. Certificateless public key cryptography. *Advances in cryptology - ASIACRYPT 2003*. Berlin, Heidelberg: Springer; 2003. p. 452–73.
13. Suárez-Albela M, Fraga-Lamas P, Fernández-Caramés TM. A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices. *Sensors* 2018;**18**(11):3868.
14. Naresh VS, Sivaranjani R, Murthy NVES. Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor networks. *Int J Commun Syst* 2018;**31**(15):e3763.
15. Wazid M, Das AK, Kumar N, et al. Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment. *IEEE Internet Things J* 2019;**6**(2):3572–84.
16. Ali Z, Chaudhry SA, Ramzan MS, et al. Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles. *IEEE Access* 2020;**8**:43711–24.
17. Wang CF, Liu C, Niu SF, et al. An authenticated key agreement protocol for cross-domain based on heterogeneous signcryption scheme. *2017 13th international wireless communications and mobile computing conference (IWCMC)*. Piscataway: IEEE Press; 2017.
18. Tao FS, Shi T, Li SJ. Provably secure cross-domain authentication key agreement protocol based on heterogeneous signcryption scheme. *2020 IEEE 4th information technology, networking, electronic and automation control conference (ITNEC)*. Piscataway: IEEE Press; 2020.
19. Liu XX, Ma WP. CDAKA: A provably-secure heterogeneous cross-domain authenticated key agreement protocol with symptoms-matching in TMIS. *J Med Syst* 2018;**42**(8):1–15.
20. Pan XY, Jin YQ, Li FG. An efficient heterogeneous authenticated key agreement scheme for unmanned aerial vehicles. *J Syst Archit* 2023;**136**:102821.
21. Aithekar A, Gupta P, Chaudhary D. A secure and efficient heterogeneous ID-based signcryption for unmanned aerial vehicular networking system. *Secur Priv* 2024;**7**(5):e389.
22. Xie MY, Chang Z, Li HW, et al. BASUV: a blockchain-enabled UAV authentication scheme for Internet of vehicles. *IEEE Trans Inf Forensics Secur* 2024;**19**:9055–69.
23. Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. *Proceedings of the 1st ACM conference on computer and communications security*. 1993.
24. Javed S, Khan MA, Abdullah AM, et al. An efficient authentication scheme using blockchain as a certificate authority for the Internet of drones. *Drones* 2022;**6**(10):264.