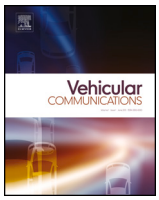
Vehicular Communications 49 (2024) 100834

Contents lists available at ScienceDirect

Vehicular Communications

journal homepage: www.elsevier.com/locate/vehcom



HCFAIUN: A novel hyperelliptic curve and fuzzy extractor-based authentication for secure data transmission in IoT-based UAV networks

Jatin Sharma, Pawan Singh Mehra *

Department of CSE, Delhi Technological University, Delhi, India

ARTICLE INFO

Keywords:
Internet of Things(IoT)
Unmanned Aerial Vehicle(UAV)
Authentication
Fuzzy Extractor(FE)
Hyperelliptic Curve Cryptography(HCC)

ABSTRACT

IoT-based UAV networks comprise interconnected UAVs outfitted with sensors and microcontrollers to simplify data exchange in environments such as smart cities. In light of open-access communication landscapes, IoT-based UAV networks could pose security challenges, encompassing authentication vulnerabilities and the inadvertent disclosure of location and other confidential information to unauthorised parties. Henceforth, we have proposed a lightweight and secure authentication protocol: Hyperelliptic Curve and Fuzzy Extractor based Authentication in IoT-based UAV networks (HCFAIUN) leveraging Hyperelliptic Curve Cryptography(HCC), Fuzzy Extractor (FE), XOR operations and hash functions. HCC's maximum key size is 80 bits, differing from the 160-bit requirement of the elliptic curve, making it apt for UAVs with limited resources. The proposed scheme utilises biometrics traits of users to avoid exposing data from stealing smart devices using FE. This protocol facilitates the mutual authentication of users and UAVs, allowing them to exchange a session key for secure communication. The Hyperelliptic Curve (HC) scalar multiplication protects the user's private key from attackers, even in public channels. The obfuscation identity of the user and UAVs generated through the hash function and timestamp makes the external user and UAV anonymous. The efficacy of this proposed framework is examined using the Scyther verification tool and Random oracle model-based formal analysis, and informal analysis is also discussed, which validates its robustness against well-known potential physical and logical attacks. The performance analysis shows that the HCFAIUN scheme has lower computation, communication, and storage costs, i.e., 3.832 ms and 1456 bits and 1128 bits, respectively, compared to existing schemes.

1. Introduction

IoT provides unprecedented benefits while introducing new concerns, particularly privacy and security. The word "things" in the context of IoT refers to intelligent objects connected over the Internet with computational, sensory, and actuation capabilities [1]. These intelligent gadgets are pervasive in many aspects of our lives, including typical IT-centric tools like smartphones and laptops and more lifestyle-oriented entities like smart lighting, linked appliances, and electronic personal assistants. A UAV is an unmanned aircraft that can be controlled remotely via a radio communication interface and has an inbuilt programme control unit [2]. With their versatility and simplicity of operation, IoT-based UAV networks eliminate the inherent risks of personal damage or loss. IoT-based UAV network architecture has three entities: external users (EU), Ground Control Station (GCS) and UAVs. In these networks, UAVs collect data and send it to a GCS. The GCS then sends out commands to control and watch over the drones through wireless connections [3]. IoT-based UAV networks are ubiquitously deployed across diverse sectors, with notable prominence in civil and military spheres. UAVs play crucial roles in studying the earth's structure, spraying crops, surveillance and monitoring during natural disasters [4–6]. The fifth-generation (5G) networks are the latest cellular technology with the major benefits in IoT. UAVs can support the network in various ways, such as 5G network slicing [7], acting as base stations or relays during emergencies, or collecting data from IoT devices using various data collection schemes such as graph and AI-based methods [8]. Thus, the UAV-assisted paradigm provides better coordination or fosters connectivity between the EU and GCS in case of network infrastructure flaws. EU, acting as data consumers, seek real-time access to the information the UAVs acquire via the public Internet. Using public wireless channels for data exchange between the GCS and users introduces vulnerabilities and security risks, including the potential for unauthorised information

* Corresponding author.
E-mail address: pawansinghmehra@gmail.com (P.S. Mehra).

https://doi.org/10.1016/j.vehcom.2024.100834
Received 8 April 2024; Received in revised form 2 July 2024; Accepted 20 July 2024
Available online 26 July 2024

disclosure. Given the sensitive and critical nature of the information being collected and exchanged between UAVs and the GCS, ensuring information security emerges as a pivotal and intricate challenge in IoT-based UAV networks. Unlike the previous studies that relied on protocols which are not lightweight for resource-constraints UAVs as they use pairing-based cryptography and chaotic maps. Therefore, we have developed a lightweight and safe authentication method called Hyperelliptic Curve and Fuzzy Extractor-based authentication in IoT-based UAV networks (HCFAIUN) employing HC, XOR operations and SHA-1 (Secure Hash Algorithm) hash functions. The maximum key size for HCC is 80 bits, as opposed to the elliptic curve's need of 160 bits, making it suitable for UAVs with limited resources. This protocol supports the mutual authentication of users and UAVs by allowing them to share a session key for safe interactions and prevent malicious activity of exposing sensitive data by generating biometric traits through FE. The proposed protocol adopts HC scalar multiplication to protect the private key from well-known assaults and identity obfuscation to keep the external user, GCS, and UAVs anonymous. This work excludes the requirement of Physical Unclonable Functions (PUF) [9], which are hardware modules for physical UAV attack prevention because the private key of a UAV is securely stored using a hash function, obfuscation identity, timestamp and random numbers, which prevent attackers from predicting session key between EU and UAVs. HCFAIUN scheme generates separate sessions by creating unique obfuscation identity of EU and UAVs. The storage overhead of the UAV is also minimised to 160 bits in the authentication stage as compared to existing schemes, thus eliminating its resource limitations. This paper analyses existing authentication systems in a tabular and graphical format utilising security characteristics followed by formal and informal security assessments using the recent Scyther verification tool, Random oracle model and cryptographic primitives, resulting in less computation, communication and storage overheads.

1.1. Motivation

In an IoT-based UAV network environment, consider the usual circumstance where the EU want quick access to real-time data directly from a specific UAV. This makes the situation pivotal for protecting the sensitive data and identity of the UAV from adversaries, as this can help them track a UAV by determining its geographical location. There are various security concerns like tampering attacks, replay attacks, man-in-the-middle (MITM) attacks, etc., while communicating among EU, UAVs and GCS. The impact of these security concerns can be seen in various scenarios like smart city environments, including search and rescue, package deliveries, ensuring safety and locating people in emergencies [9]. IoT-based UAVs use a lot of sensors like thermal and imaging sensors, which collect vital data during natural calamities and accidents [8,10]. This data aids in locating missing individuals and injured victims in adverse conditions, which can be intercepted by adversaries in between public channels instead of transferring the crucial information to the rescue team, resulting in delaying the monitoring scenario. In 2015, Nepal faced a natural disaster, an earthquake that put the lives of 2.8 million people in danger with physical property destruction, and UAVs played a crucial role in transferring the analysed destruction data to GCS. The rough landscape and damaged properties made it hard for UAVs to operate, resulting in frequent interruptions to the precision and reliable data transmitted to GCS. These disruptions stemmed from unauthorised data tampering and natural barriers, which affected the rescue operations by the Nepal army [11]. Therefore, the proposed work endeavours to bridge the existing discrepancies and address the security requirements for securing communication in IoT-based UAV networks. So, we have proposed the HCFAIUN protocol, a lightweight and secure authentication protocol based on the HC for IoT-based UAV networks.

1.2. Contribution

In this segment, we elaborate on the proposed HCFAIUN's primary research contributions, which are outlined subsequently.

- Presents a novel lightweight mutual authentication protocol, HCFAIUN, for secure communication in IoT-based UAV Networks by utilising HC.
- Utilises the HC with a maximum key size of 80 bits rather than the elliptic curve key size, i.e., 160 bits, which is useful for resource-constrained UAVs.
- Employs the FE mechanism to generate biometric traits of the user, such as a key which can be reproducible to prevent exposing data from stealing smart devices.
- Employs the HC scalar multiplication to make the private key secure and obfuscation identity to maintain the anonymity of the EU, GCS, and UAVs.
- Comparison with existing authentication methods using security parameters like mutual authentication, un-traceability, etc., in a tabular layout.
- Excludes the requirement of PUF hardware module on UAVs by utilising secure hash function, obfuscation identity, timestamp and random numbers time to prevent the physical attacks.
- The storage overhead of resource-constrained UAVs is reduced to 160 bits compared to established benchmark schemes during the login and authentication stage.
- The security of the HCFAIUN scheme was formally verified using the Scyther tool and Random Oracle Model, ensuring its resistance to various attacks. Informal security analyses have been conducted to underscore the scheme's resilience against potential attacks.
- According to a comparative study, HCFAIUN produces lower computational, communication and storage overheads than existing schemes.

1.3. Outline of the paper

The paper is organised as follows: Section 2 explores relevant literature. Section 3 conducts a preliminary study on the HC. Section 4 outlines the system model, followed by Section 5 detailing the proposed protocol. Security assessment is in Section 6, and Section 7 offers performance evaluation. Finally, Section 8 summarizes the concluding remarks and outlines future directions. Fig. 1 illustrates the hourglass structure of the paper.

2. Pertinent work

In this section, we unleash a powerhouse collection of vital research initiatives focused on ensuring authentication within IoT-based UAV networks. Despite the relatively small pool of existing literature in this field, this paper goes all out to encompass the most noteworthy advancements and a summary of pertinent work is represented in Table 1.

Wazid et al. (2019) [12] presented an identity verification method to facilitate remote user access to drone services, employing a three-factor authentication methodology. Their approach involves lightweight communication techniques, incorporating hash functions and exclusive-OR operators. However, it is crucial to mention that their approach does not have safeguards to protect against risks like insider attacks by trusted individuals or attempts at impersonation.

Srinivas et al. (2019) [13] proposed a security scheme, TCALAS, designed to protect UAVs. The purpose was to ensure that customers could not access UAV services without completing the registration process with the GCS. Additionally, all legitimate drones must be certified by the GCS. Such a credential shared by every drone and GCS facilitates safe communication with legal users. It also allows users to change passwords or biometric data without engaging GCS. However, it should be

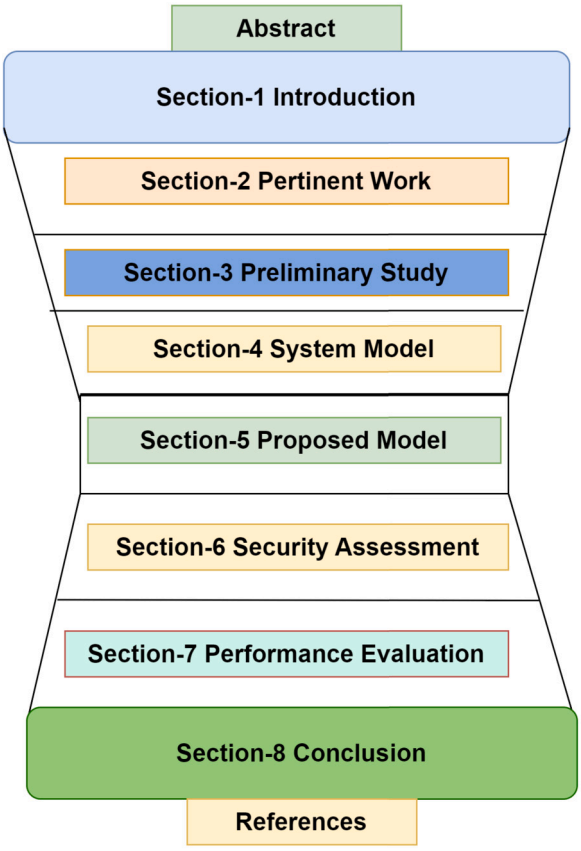


Fig. 1. Hourglass structure of the paper.

noted that this method is prone to security vulnerabilities like forgery, reply attacks, anonymity of the user and impersonation issues.

Tian et al. (2019) [14] presented an authentication method that prioritizes the privacy of the environment within which UAV systems operate. This architecture passionately embraces an efficient online/offline signature scheme to efficiently handle the finite resources in UAVs. The authors also look at a prospective means of authentication through edge computing that will reduce costs. In addition, they developed a full method that includes buffer pseudonyms and management for public key upgrades. This versatile approach has a dual role: keeping the identity, location, and flight paths of drones confidential while substantially strengthening the protective layers of privacy preservation. The security model proposed by them satisfies pre-defined security requirements and has better performance characteristics as well. However, it should be noted that this security strategy does not offer any dependable means for protecting dynamic privacy.

Ali et al. (2020) [15] identified flaws in Srinivas et al.'s approach, especially the lack of untraceability and vulnerability to stolen verifier attacks. Accordingly, Ali et al. developed another lightweight drone authentication system that employs symmetric key primitives and temporal credentials to enhance security. Nevertheless, this system is prone to counterfeiting, traitor, Denial of Service (DoS), and Smart card loss attacks and does not utilize perfect forward secrecy.

Alladi et al. (2020) [16] presented the mutual authentication of drones and users, even when operating across an unsecured communication channel, by combining hash functions and bitwise XOR operations. During the setup phase, a control server creates a master private key and other vital public system parameters. Subsequently, during the registration process, the users and drone contacts register on the control server and receive their secret keys. Finally, the users and drones are authenticated, generating the secure session key and verifying communication integrity between both sides. However, the scheme is weak against forgery, turn cloak, DoS, and smart card loss attacks, and it turned out that perfect forward secrecy is not used.

Ever [17] developed a framework for drone applications that used bilinear pairing and ECC to create a mobile sink, an approach that requires a fair amount of computation. However, it is crucial to ensure that Ever's protocol complies with the security model since the authors did not conduct formal security analysis and guaranteed untraceability and user privacy.

Feng et al. [18] address the problems of the centralized security paradigm and assert that such an approach cannot sufficiently secure identifiers across multiple domains due to one point of failure. These findings provided the foundation for a cross-domain authentication model built on blockchain technology. Especially, they introduce a threshold-multi-party signing mechanism in their design to implement the federated identity across separate domains. In addition, they implemented a smart contract to perform the verification mission of the drone from the other domain. However, this implementation will incur a much higher communication overhead.

Rajasekaran et al. (2022) [19] presented An anonymous mutual authentication and privacy-preserving scheme suitable for drones and users. The batch authentication technique in the proposed system was used to verify clusters of drones and keep significant data private. Even though the scheme preserves location privacy for verified drones, it fails to fully protect authorized drones from entities. In addition to bilinear pairing, the scheme is computationally expensive, which makes it unsuitable for UAVs with limited computing resources.

Tanveer et al. (2022) [3] offered the multi-UAV system with the authentication model, which is composed of the Elliptic Curve Cryptography (ECC), specially dedicated authenticated encryption algorithm (AE), and a hash function. The plan takes the form of a simplified and well-prepared series of seven stakes at a time approach. It first confirms the user's identification and then establishes a confidential key between the user and the drone, priming them for later secure connections. The execution of this task results in greater computing and communication costs.

Nyangaresi (2023) [20] proposed a provably secure authentication framework in UAVs, which is based on hash functions and quadratic residues. The protocol provides mutual authentication among operators and UAVs by addressing security concerns like physical and logical attacks. The provided scheme is not analysed using a formal analysis tool like Scyther. Moreover, the protocol suffers from high computation and communication costs.

Tanveer et al. (2024) [9] proposed a secure authentication and key agreement protocol for drone networks. The proposed protocol uses hash functions, physical unclonable Functions (PUF) and symmetric key cryptography. The authors have achieved mutual authentication to prevent various physical and logical attacks and successfully generated the session key, but their scheme comes with higher computation and communication overhead.

Mahmood et al. (2024) [21] presented a safe and secure mutual authentication protocol for drone environments. This protocol discussed the robust security functions to prevent physical attacks, masquerading, and privacy invasion. The protocol relies on chaotic maps (Chebyshev Polynomial) and hash functions. The scheme is prone to MITM, and DoS attacks but unsuitable for resource-constrained UAV environments due to higher computation and communication overhead.

Tanveer et al. (2024) [22] demonstrated a secure and anonymous authentication with a key agreement approach between end users and UAVs based on chaotic maps and symmetric encryption. It addresses security vulnerabilities such as privileged insider attacks and ground station server bypass attacks. The suggested technique employs a FE to leverage users' fingerprints as hidden information. Nevertheless, this approach lacks emphasis on security aspects such as DoS attacks, smart device attacks, and physical UAV attacks. Implementing this method results in elevated computation and communication costs.

Table 1
Summary of pertinent work.

Reference	Year	Schemes	Techniques	Limitations
Wazid et al. [12]	2019	Three-factor authentication methodology.	Ex-OR and secure hash function SHA-160.	Prone to privileged insider attacks and impersonation attempts.
Srinivas et al. [13]	2019	A lightweight authentication scheme incorporating temporal credentials for anonymity.	ECC, Ex-OR, and SHA-160.	Susceptible to forgery, replay, user anonymity and impersonation attacks.
Tian et al. [14]	2019	Efficient privacy-preserving authentication.	Signature method, Hash functions(SHA-160).	Prone to forgery, session key, spoofing attack, and physical attacks.
Ever [17]	2020	Secure authentication framework.	ECC and SHA-160.	It does not offer untraceability or user anonymity without formal security analysis.
Alladi et al. [16]	2020	Novel authentication scheme for UAVs.	Pseudo-random number generation (PRNG), ExOR, HMAC (SHA-1), Hash (SHA-1).	Shared secret key can be compromised.
Feng et al. [18]	2021	Blockchain-based Cross-Domain Authentication.	Hash, ExOR, Key generation centre.	High computation cost.
Ali et al. [15]	2020	Improved temporal credential-based anonymous lightweight authentication scheme.	Symmetric encryption, and SHA-160.	Fragile to forgery, turn cloak, DoS, and Smart card loss attack and does not utilize the perfect forward secrecy.
Tanveer et al. [3]	2022	Robust authenticated key management protocol.	AES-CBC-256 cypher, ECC, and SHA-256.	High computation and communication costs.
Rajasekaran et al. [19]	2022	Anonymous mutual authentication scheme.	Bilinear pairing cryptography, SHA-1.	Substantial computational burden due to bilinear pairing and exponential operation.
Nyangaresi [20]	2023	Provably secure authentication framework in UAVs.	Hash functions and quadratic residues.	High computation and communication cost with no formal verification through the tool.
Tanveer et al. [9]	2024	SAAF-IoD: Secure and Anonymous Authentication Framework for the Internet of Drones	Chaotic map, FE, hash function, and EX-OR.	High computation and communication costs.
Mahmood et al. [21]	2024	Chaotic map-based secure authentication scheme.	Chaotic map, hash function, and EX-OR.	Prone to DoS and MITM attacks with high computation and communication overhead.
Tanveer et al. [22]	2024	Secure and Anonymous Authentication scheme.	Chaotic map, FE, hash function, and EX-OR.	High computation and communication costs.

3. Preliminary study

This section discusses the preliminaries required for a better understanding of the proposed work.

3.1. Hyperelliptic curves

Hyperelliptic curves encompass a range of algebraic curves with various degrees of complexity, including elliptic curves [23–25]. Consequently, it is possible to perceive an elliptic curve as an HC with a genus of 2, 3 and so on [26–28].

HC is different from RSA, elliptic curve, and bilinear pairing due to its ability to maintain an equivalent level of security while utilizing smaller parameter sizes. Suppose we have a field called "E". The hyperelliptic curve, denoted as HC, with genus "n" over E, can be expressed using the following general equation (1):

$$HC : u^2 + p(x)u = q(x) \tag{1}$$

Here:
"$p(x)$" represents a polynomial of degree at most "n" over F.
"$q(x)$" is a monic polynomial of degree "$2n + 1$" over F.
These specifications are subject to certain additional conditions for a complete curve definition.

3.1.1. Computational hypotheses

- **Hypothesis Underlying the Hyperelliptic Curve Discrete Logarithm Problem (HCDLP)**
  In the context of HCDLP, we have adopted the following assumptions:
  - Let's denote a variable as "α", and it takes on values from the set $\{j \mid j$ is a positive integer greater than or equal to 1$\}$
  - The likelihood of successfully computing "α" from the equation $K = \alpha \cdot D$ is deemed to be exceedingly small.
- **Computational Assumption for the Diffie-Hellman Problem in Hyperelliptic Curve (HCDHP)**
  In the context of HCDHP (Hyperelliptic Curve Diffie-Hellman Problem), we establish the following assumptions:
  - We introduce two variables, ρ and Ω, and both are drawn from the set $\{j \mid j$ is a positive integer greater than or equal to 1$\}$.
  - To predict the variables ρ and Ω from the equation $K = \rho \cdot \Omega \cdot D$ is considered to be of negligible significance.

3.2. Fuzzy extractor

A fuzzy extractor is a cryptographic technology that is commonly applied in scenarios where biometric data, such as fingerprints or voiceprints, are used for user identification purposes. It includes two operations:

- The generation operation ($Gen(\cdot)$) receives biometrtic traits (BT_{EU}) as inputs and produces a secret biometric key γ_{EU} and helper data (hd). The equation for the generator function is $Gen(BT_{EU}) = (\gamma_{EU}, hd)$
- The reproduction operation, ($Rep(\cdot)$), accepts Biometric traits (BT^*_{EU}) and hd as inputs. It reproduces the biometric key on verifying the condition "$HamD(BT^*_{EU}, BT_{EU}) \leq t$", where $HamD$ signifies the Hamming distance $HamD$, and t denotes the threshold. The equation for the Reproduction function is $Rep(BT^*_{EU}, hd) = \gamma_{EU}$.

4. System model

This section elucidates two essential models, namely the threat and network model, which are integral to elucidating the functionality and applicability of the devised scheme.

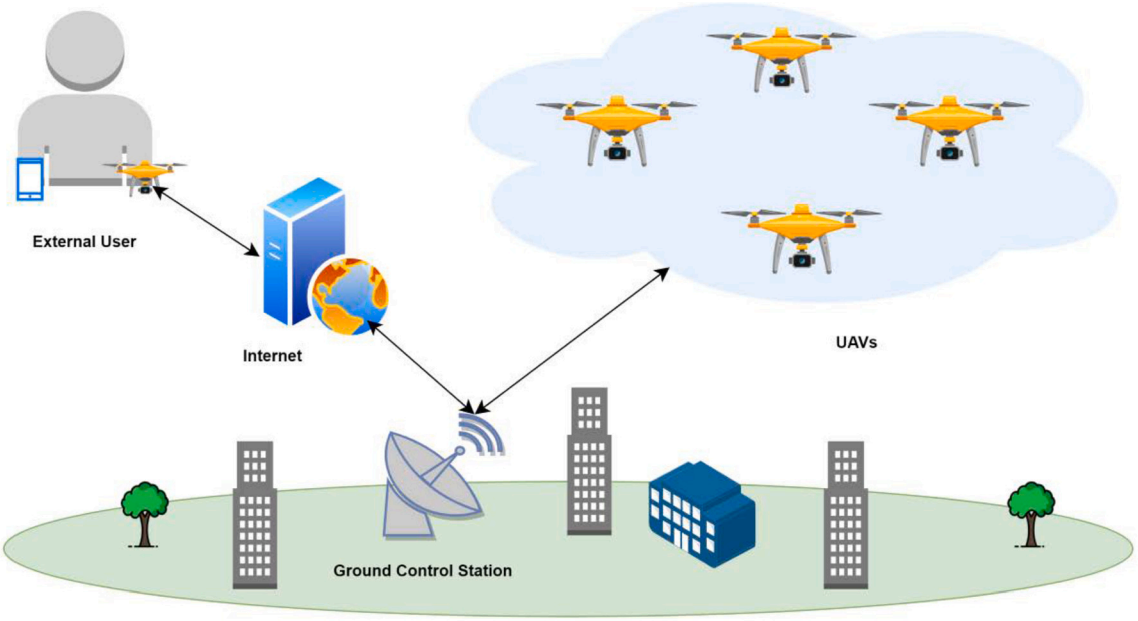


Fig. 2. Network model.

4.1. Network model

Within this architectural framework, the GCS serves as the trusted registration authority responsible for enrolling both drones and users. A drone operates within a designated airspace, collecting data from its immediate surroundings. The network architecture of the formulated framework is depicted in Fig. 2, featuring three key entities: the GCS, the EU, and the UAVs.

GCS: It is established as a trusted entity responsible for registering all users and drones. The GCS generates long-term secret keys for both EU and UAVs based on their respective identities.

EU: The user having a smart device gets his/her secret key from GCS in the registration phase. Before accessing and communicating with drones on the mission, he/she should be verified.

UAVs: Drones, during the registration phase, receive their secret keys from the GCS as well. Once the validity of the EU is confirmed, the UAV and EU establish a session key to ensure the security of their communication.

4.2. Threat model

Within the scope of this document, we examine two distinct threat models and a concise overview of each is presented herein:

4.2.1. DY threat model

In the IoT-based UAV Networks environment, we operate under the widely recognized DY (Dolev-Yao) threat model [22]. This model posits that any communication transmitted or received through vulnerable channels can be eavesdropped upon by an adversary denoted as'*A*.' Furthermore, *A* can tamper with the messages by deleting, modifying, or introducing spurious content into the communication stream. It is imperative to note that, following this model, the communication endpoints, in this case, UAVs, do not automatically command dependability on this connection.

4.2.2. CK - adversary model

We apply the complete CK adversary framework [22] to improve the resilience of our user identification approach, which outperforms the efficacy of the DY threat model typically used in current literature for user authentication techniques. Following the CK-adversary model, adversary'*A*' can gain session states and sensitive information, including secret keys and capabilities. It is also believed that this enemy will physically seize selected drones. A may acquire access to all private information on the seized UAVs by using power analysis attacks. UAV capture attacks highlight the concern under the circumstances.

4.3. Security objectives

Given the inherent properties of the authentication mechanism for IoT-based UAV networks. Our suggested scheme must meet the following security requirements to provide reliable and resilient communication. [29–33]

4.3.1. Mutual authentication

This pertains to the process where both users and drones mutually authenticate themselves before transmitting messages through the network channel, thereby verifying their respective identities, which are conveyed alongside the messages.

4.3.2. Privacy protection

Our proposed plan or scheme safeguards users' privacy by ensuring their identities remain confidential. Only authorized counterparts with whom the user has registered possess access to this information. In the event of an adversarial attempt to obtain such information, only encrypted data will be revealed.

4.3.3. Un-traceability

Our scheme guarantees the security of users and drones by providing un-traceability. Should an adversary endeavour to ascertain the locations of drones or users by intercepting the network channel, the untraceability feature will thwart such efforts.

4.3.4. Session key establishment

Following the effective implementation of the scheme, a session key is generated to facilitate further secure communication between users and drones. For other legitimate users not part of the ongoing session, even if they possess the session key, the adversary cannot access any information.

Table 2
Notation Table.

Notation	Description
ID_{EU}	Actual identity of external user
$Pass_{EU}$	Password of external user
BT_{EU}	Biometric Traits of external user
hd	Helper data to get biometric secret key
γ_{EU}	Biometric secret key
Pri_Key_{GCS}	Private key of GCS
Pub_Key_{GCS}	Public key of GCS
Pri_Key_{UAV}	Private key of UAV
D	Divisor of HC
ID_{UAV}	Actual identity of UAV
OID_{UAV}	Obfuscation identity of UAV
$hash(.)$	hash function
$\|\|$	Concatenation
$\oplus$	Bitwise exclusive OR function
TS_1, TS_2, TS_3	Timestamps
TS_{cur}	Current timestamp
τ	The maximum time threshold for message reception
R_1, R_2	Arbitrary numbers
$Sess\_Key_{EU \rightarrow UAV/UAV \rightarrow EU}$	Session key from one entity to other
M_1, M_2, M_3	Messages among entities

4.3.5. Resilience against diverse threats

To prevent data loss or unauthorized disclosure of user information, our scheme is designed to withstand various attacks, including impersonation attacks, MITM attacks, drone capture attempts, server impersonation, password and biometric modification, message modification, replay attacks, and known session key attacks.

5. Proposed protocol

The proposed protocol encompasses six stages under this section. The notations used in this scheme are presented in Table 2

5.1. Initialization stage

In this phase, the GCS, operating as a certificate authority, is responsible for generating the public parameters of this scheme along with the confidential key. GCS perform the following steps as mentioned below:

- **Step-1:** Selects a randomly generated numerical value private key, $Pri\_Key_{GCS} \in \{j \mid j$ is a positive integer greater than or equal to $1\}$.
- **Step-2:** The GCS Public Key is calculated as in the equation (2):

$$Pub\_Key_{GCS} = Pri\_Key_{GCS}.D \tag{2}$$

as D denotes the divisor on a HC.
- **Step-3:** The GCS uses the unidirectional cryptographic hash function '$hash(\cdot)$'. Finally, the ensemble of parameters $\{Pub\_Key_{GCS}, D, n = 280, hash(\cdot)\}$ is made public.

5.2. Registration stage

The GCS performs a complete offline registration process for all UAVs before deployment during this phase and is considered to be storage efficient for credentials. Furthermore, the GCS ensures that users' registrations are safe. The following explanation delves into the registration step.

5.2.1. UAV registration

The GCS enrols all UAVs before being deployed in a certain geographic region. The UAV enrollment process is explained in detail, and Fig. 3 represents the UAV registration process.

Step-1: The GCS selects a unique identification known as'ID_{UAV}' for each drone, after which a random number $'\alpha'$ is chosen from the set of natural numbers'N' to facilitate computation, and then proceeds to ascertain the corresponding obfuscation identity (OID_{UAV}) using the message digest MD_{UAV} and Pri_Key_{GCS} as in equations (3) and (4):

$$MD_{UAV} = hash(ID_{UAV} \parallel \alpha) \tag{3}$$

$$OID_{UAV} = hash(hash(ID_{UAV} \parallel Pri\_Key_{GCS}) \oplus MD_{UAV})) \tag{4}$$

Step-2: The GCS securely retains the identity'OID_{UAV}' within its proprietary database, establishing a permanent association of the pair (ID_{UAV}, OID_{UAV}) in the memory of the respective UAV.

5.2.2. External user registration

An EU is enrolled through a secure registration process with the GCS at this stage. The EU can access instantaneous data by a specified UAV flying inside a particular aerial zone upon completing the enrollment. The GCS and EU performs the following operations as mentioned below:

- **Step-1:** EU selects an exclusive identifier 'ID_{EU}' and a corresponding password '$PASS_{EU}$', after which EU engraves his/her biometric traits (BT_{EU}) such as iris and fingerprint into a sensor of smart device and a random number 'μ' is chosen from the set of natural numbers 'N' to facilitate the computation as given in equations (5) and (6):

$$Gen(BT_{EU}) = (\gamma_{EU}, hd) \tag{5}$$

$$Y_{EU} = hash(ID_{EU} \parallel PASS_{EU} \parallel \mu \parallel \gamma_{EU}) \tag{6}$$

- **Step-2:** Upon the reception of the message, GCS proceeds to calculate OID_{EU} and Z_{EU}, as in equations (7) and (8):

$$OID_{EU} = hash(ID_{EU} \parallel Pri\_Key_{GCS}) \tag{7}$$

$$Z_{EU} = hash(OID_{EU} \parallel Y_{EU}) \tag{8}$$

Subsequently, GCS records (ID_{EU}, OID_{EU}, Z_{EU}) within its database, and securely transmits (OID_{EU}, Z_{EU}) to EU via a protected communication channel.
- **Step-3:** Upon receiving the information from GCS, EU performs the computation as in equations (9) and (10)

$$Z_{EU}' = hash(OID_{EU} \parallel PASS_{EU} \parallel \gamma_{EU}) \oplus Z_{EU} \tag{9}$$

$$OID_{EU}' = hash(ID_{EU} || PASS_{EU}) \oplus OID_{EU} \tag{10}$$

In conclusion, EU archives the information μ, Z_{EU}', OID_{EU}' in its device's local memory, marking the completion of the enrolment phases and Fig. 4 represents the external user enrollment steps.

5.3. Login and authentication stage

An External user, denoted as EU, initiates the access and verification phase within the proposed approach to establish a secure communication channel and obtain authorization. This section offers a comprehensive elaboration on the intricacies of this particular stage.

- **Step-1:** EU is obligated to furnish their identification, denoted as ID_{EU}, and their respective password indicated as $PASS_{EU}$ with biometric traits as $BT_{EU}*$ into smart device prior to initiate the computation as in equations (11)–(14),

$$\gamma_{EU}* = Rep(BT_{EU}*, hd) \tag{11}$$

$$Y_{EU}^{S} = hash(hash(ID_{EU} \parallel \mu \parallel \gamma_{EU}*) \oplus hash(PASS_{EU} \parallel \mu \parallel \gamma_{EU}*)) \tag{12}$$

$$OID_{EU}s = hash(ID_{EU} \parallel Pub\_Key_{GCS}) \tag{13}$$

$$Z_{EU}^{S} = hash(OID_{EU}^{S} \parallel Y_{EU}^{S}). \tag{14}$$

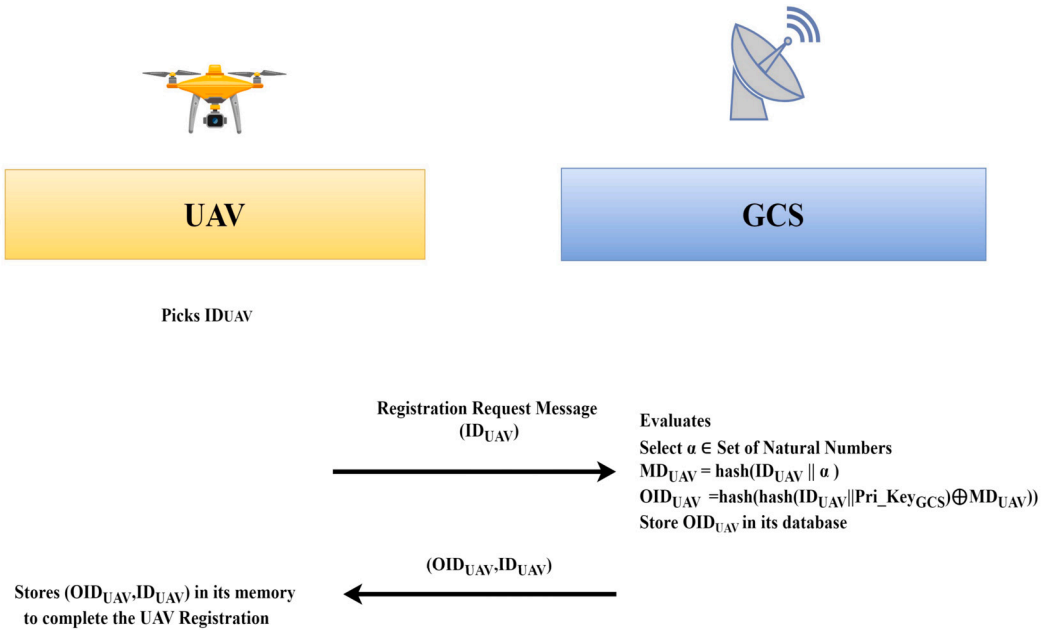


Fig. 3. UAV registration.

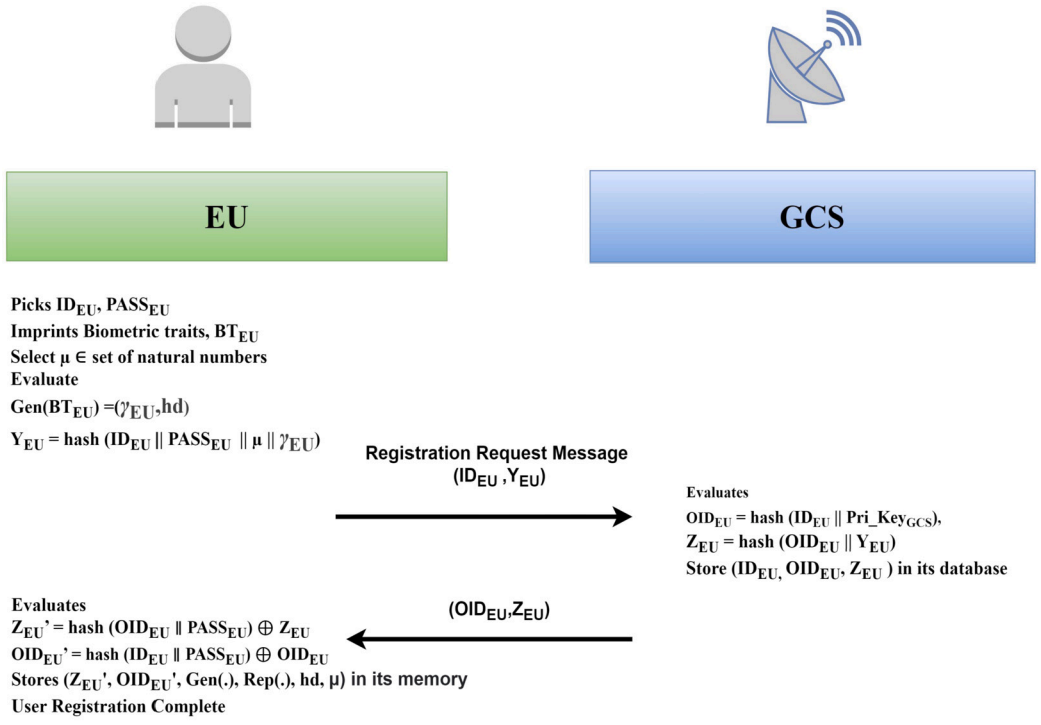


Fig. 4. External user registration.

Now prove (Z_{EU}^{S}? = Z_{EU}). In the event of unsuccessful verification, the procedure is promptly terminated. Otherwise, EU generates Pub_Key_{GCS} and a timestamp TS_1 to calculate the following equations (15)–(19):

$$Pub\_Key_{GCS} = Pri\_Key_{EU}.D \tag{15}$$

$$K_{EU} = Pri\_Key_{EU}.Pub\_Key_{GCS} \tag{16}$$

$$EU_1 = OID_{EU} \oplus hash(OID_{GCS} || TS_1) \tag{17}$$

$$EU_2 = OID_{UAV} \oplus hash(OID_{GCS} \parallel TS_1 \parallel K_{EU}) \tag{18}$$

$$EU_3 = hash(OID_{EU} \parallel OID_{GCS} \parallel OID_{UAV} \parallel K_{EU} \parallel TS_1) \tag{19}$$

Subsequently, the authentication message request, denoted as $M_1 = (EU_1, EU_2, EU_3, Pub\_Key_{EU}, TS_1)$ is transmitted over a public channel and is subject to subsequent analysis by the GCS.

- **Step-2:** Upon receipt of the authentication request message by the GCS, i.e. $M_1 = (EU_1, EU_2, EU_3, Pub\_Key_{EU}, TS_1)$, GCS initially assesses the validity of TS_1 through verification. ($|TS_{cur} - TS_1| \leq \tau$), where τ represents the threshold time receiving the information, and on successful verification, the GCS will calculate the following based on the current message time (TS_{cur}) as in equation (20).

$$K_{GCS} = Pub\_Key_{EU}.Pri\_Key_{GCS} \tag{20}$$

With this value as a starting point, GCS proceeds to perform the subsequent calculations as in equations (21)–(23):

$$OID_{EU}{}^{*} = EU_1 \oplus hash(OID_{GCS} \parallel TS_1) \tag{21}$$

$$OID_{UAV}{}^{*} = EU_2 \oplus hash(OID_{EU}{}^{*} \parallel TS_1 \parallel K_{GCS}) \tag{22}$$

$$EU_3{}^{*} = hash(OID_{EU}{}^{*} \parallel OID_{UAV}{}^{*} \parallel OID_{GCS}{}^{*} \parallel K_{GCS} \parallel TS_1) \tag{23}$$

GCS checks if the condition (EU_3? = $EU_3{}^{*}$) is valid. In the event of an invalid request, the GCS will decline the authentication request. If the request is valid, GCS can proceed with the EU authentication and subsequently execute the following equations (24)–(26).

$$B_1 = hash(OID_{UAV}{}^{*} \parallel TS_2) \oplus R_1 \tag{24}$$

$$B_2 = OID_{EU}{}^{*} \oplus hash(OID_{UAV}{}^{*} \parallel OID_{GCS}{}^{*} \parallel TS_2 \parallel R_1) \tag{25}$$

$$B_3 = hash(OID_{UAV}{}^{*} \parallel OID_{GCS} \parallel OID_{EU}{}^{*} \parallel TS_2 \parallel R_1) \tag{26}$$

Ultimately, GCS transmits message M_2 to the UAV via a publicly accessible communication channel, comprising elements B_1, B_2, B_3 and TS_2

- **Step-3:** UAV validates recent content by confirming that the $|TS_{cur} - TS_2| \leq \tau$ after receiving and If this validation is successful, the drone, denoted as a UAV, proceeds to initiate the following equations (27)–(29) computations:

$$R_1{}^{*} = B_1 \oplus hash(OID_{UAV} \parallel TS_2) \tag{27}$$

$$OID_{UAV}{}^{*} = B_2 \oplus hash(OID_{UAV} \parallel OID_{GCS} \parallel TS_2 \parallel R_1{}^{*}) \tag{28}$$

$$B_3{}^{*} = hash(OID_{UAV} \parallel OID_{GCS} \parallel OID_{EU}{}^{*} \parallel TS_2 \parallel R_1{}^{*}) \tag{29}$$

It additionally verifies if (B_3? = $B_3{}^{*}$) to establish the authenticity of GCS. In the event of failure, the session is promptly terminated. However, if the verification is successful, it generates a random number, denoted as R_2, based on the current timestamp, TS_3, before advancing to the subsequent stages as in equations (30)–(32).

$$U_1 = hash(OID'_{UAV}{}^{*} \parallel OID_{UAV} \parallel TS_3) \oplus R_2 \tag{30}$$

$$Sess\_Key_{UAV \to EU} = hash(OID_{UAV} \parallel OID_{GCS} \parallel OID'_{EU}{}^{*} \parallel TS_3 \parallel R_2) \tag{31}$$

$$AUT_N = hash(Sess\_Key_{UAV \to EU} \parallel TS_3) \tag{32}$$

Conclusively, the UAV directly transmits message M_3, comprising elements U_1, AUT_N, and TS_3, to the EU via a publicly accessible communication channel.

- **Step-4:** Following the reception of a message M_3, EU initiates the process by verifying the recent time through decision $|TS_{cur} - TS_3| <= \tau$. If the decision proves to be legitimate, EU proceeds to compute $R_2{}^{*}, AUT_N{}^{*}$, the session key, denoted as ($Sess\_Key_{EU \to UAV}$), in the following equations (33)–(35):

$$R_2{}^{*} = U_1 \oplus hash(OID_{EU} \parallel OID_{GCS} \parallel TS_3) \tag{33}$$

$$Sess\_Key_{EU \to UAV} = hash(OID_{UAV} \parallel OID_{GCS} \parallel OID_{EU} \parallel TS_3 \parallel R_2{}^{*}) \tag{34}$$

$$AUT_N{}^{*} = hash(Sess\_Key_{EU \to UAV} \parallel TS_3) \tag{35}$$

The EU performs an additional check to confirm the equivalence of $AUT_N{}^{*}$ and AUT_N. When these values match, it signifies the successful mutual authentication of the user EU and the UAV, and the calculated session key is stored for subsequent secure communication. Conversely, if ($AUT_N{}^{*}$) and (AUT_N) do not match, EU promptly terminates the session. Fig. 5 shows the steps involved in the login and authentication stage.

5.4. Password modification stage

In a secure authentication framework, a procedure for password modification should be accessible. This allows an authorized user, denoted as EU, utilizing a smart device, to replace the existing password $PASS_{EU}$ with a new one, referred to as $PASS_{EU}^{N}$ and biometric traits as BT_{EU}^{N}. To effect this change, the EU is required to carry out the following actions:

- **Step-1:** EU initiates the process by entering their login credentials, comprising identity ID_{EU}, password $PASS_{EU}$ and biometric traits as $BT_{EU}{}^{*}$. Subsequently, the smartphone device performs the ensuing computational tasks
as in equations (36), (38) and (39).

$$\gamma_{EU} = Rep(BT_{EU}{}^{*}, hd) \tag{36}$$

$$Y_{EU}^{S} = hash(hash(ID_{EU} \parallel \mu \parallel \gamma_{EU}) \oplus hash(PASS_{EU} \parallel \mu \parallel \gamma_{EU})) \tag{37}$$

$$OID_{EU}^{S} = hash(ID_{EU} \parallel Pub\_Key_{GCS}) \tag{38}$$

$$Z_{EU}^{S} = hash(OID_{EU}^{S} \parallel Y_{EU}^{S}) \tag{39}$$

The smart device subsequently verifies the validity of the condition ($Z_{EU}^{S} == Z_{EU}$), and if it proves invalid, the procedure is terminated. In contrast, when the condition proves valid, the device prompts the EU to furnish a fresh password as a requisite step in finalizing the process.

- **Step-2:** EU opts for a fresh password denoted as $PASS_{EU}^{N}$ with biometric traits as BT_{EU}^{N} and transmits it. Subsequently, the smart device performs the subsequent computations as in equations (41)–(43).

$$Gen(BT_{EU}^{N}) = (\gamma_{EU}^{N}, hd^{N}) \tag{40}$$

$$Y_{EU}^{N} = hash(hash(ID_{EU} \parallel \mu \parallel \gamma_{EU}^{N}) \oplus hash(PASS_{EU}^{N} \parallel \mu \parallel \gamma_{EU}^{N})) \tag{41}$$

$$OID_{EU} = hash(ID_{EU}^{N}, Pri\_Key_{GCS}) \tag{42}$$

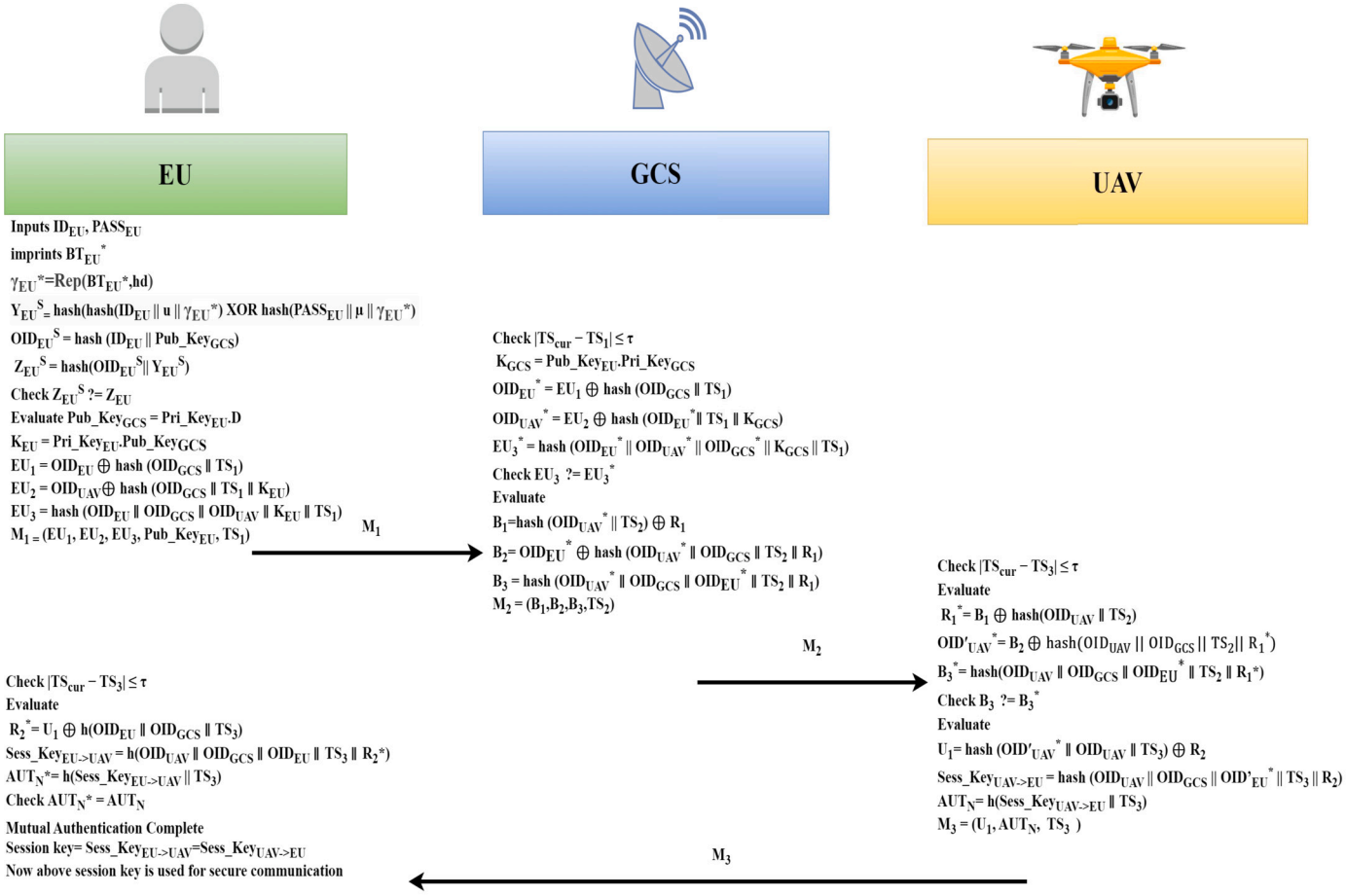


Fig. 5. Login and Authentication Stage.

$$Z_{EU}^N = hash(OID_{EU} \parallel Y_{EU}^N) \tag{43}$$

- **Step-3:** Ultimately, the EU substitutes Z_{EU}^N with Z_{EU} using a smart device. Finally, it is essential to remember that EU credentials may be subject to periodic revisions to improve the safety mechanism.

5.5. Withdrawal and renewal stage

The external user, denoted as EU, encounters the unfortunate circumstance of their smart device being lost or stolen, EU retains the capability to initiate a replacement procedure and diligently adhere to the prescribed instructions outlined below:

- **Step-1:** EU maintains his ID_{EU} identity while opting for $PASS_{EU}^N$ as his updated password with biometric traits (BT_{EU}^N). Subsequently, employing a randomly generated number μ', EU proceeds with the computation as in equations (44) and (45).

$$Gen(BT_{EU}^N) = (\gamma_{EU}^N, hd^N) \tag{44}$$

$$Y_{EU}^N = hash(hash(ID_{EU} \parallel \mu' \parallel BT_{EU}^N) \oplus hash(PASS_{EU}^N \parallel \mu' \parallel BT_{EU}^N)) \tag{45}$$

and conveys ID_{EU} and Y_{EU}^N to the GCS through a secure communication medium.

- **Step-2:** The GCS calculates OID_{EU} and Z_{EU}^N upon reception of the message in the manner as given in equations (46) and (47).

$$OID_{EU} = hash(ID_{EU} \parallel Pri\_Key_{GCS}) \tag{46}$$

$$Z_{EU}^N = hash(OID_{EU} \parallel Y_{EU}^S) \tag{47}$$

Subsequently, the GCS archives OID_{EU} and Z_{EU}^N within its storage and securely transmits the same data to the EU via a protected communication channel.

- **Step-3:** After receiving the data from the GCS, EU engages in the computation process as in equations (48) and (49)

$$Z_{EU}^N{}^* = hash(OID_{EU} \parallel Y_{EU}^N) \oplus Z_{EU}^N \tag{48}$$

$$OID_{EU}{}^* = hash(OID_{EU} \parallel PASS_{EU}^N) \oplus OID_{EU} \tag{49}$$

In conclusion, EU substitute Z_{EU} with $Z_{EU}^N{}^*$ and archives $Z_{EU}^N{}^*$, $OID_{EU}{}^*$ in its local device's memory. Furthermore, EU expunges Z'_{EU} from the device's memory, finalizing the revocation and re-issuance procedure.

5.6. Scalable UAV stage

In unforeseen incidents, such as situations involving depleted battery levels or UAV capture by a potential adversary, the prompt deployment of an alternative drone within the same operational airspace becomes paramount. In this regard, the proposal enables the inclusion of the latest aircraft into the infrastructure. This process closely mirrors the drone registration phase, and a more comprehensive delineation of this particular phase is presented below.

Step-1: To enable the utilization of a new UAV, which has not yet been registered, within a specific airspace, the GCS selects an individualized identity, ID_{UAV}^N and subsequently calculates the associated obfuscated identity as in equation (50):

$$OID_{UAV}^N = hash(ID_{UAV}^N \parallel Pri\_Key_{GCS}) \tag{50}$$

Step-2: The GCS archives ID^N_{UAV}, OID^N_{UAV} within the UAV storage before its deployment in the operational field, while also retaining ID^N_{UAV} in its current record.

6. Security assessment

This section examines the security aspects of the HCFAIUN scheme. To begin with, we demonstrate HCFAIUN's security by utilising the Scyther tool. Following this, we assess the security characteristics to confirm that HCFAIUN is strongly resistant to various potential attacks.

6.1. Formal security assessment

The proposed protocol HCFAIUN has been tested using the Scyther tool and Random Oracle model, which are the following:

6.1.1. Scyther tool-based security assessment

The proposed protocol HCFAIUN has been tested using the Scyther tool, which is written in the "Security Protocol Description Language (SPDL)" as described in works [34]. The primary aim of the simulation is to validate the security aspects related to authentication, confidentiality, and integrity. In this context, the Scyther tool characterizes the entities involved as roles within the proposed protocol. Scyther is instrumental in verifying, falsifying, and comprehensively analysing the security features of the protocol. Scyther serves as a mechanism for assessing the core security properties based on the assumption of perfect cryptography [9]. It's worth noting that when using the Scyther tool, potential attackers cannot execute security attacks on encrypted messages unless they possess the decryption key. Contrasting with the DY model presented by Dolev and Yao in 1983, where attackers had absolute control over communication entities, in the scenario involving the Scyther tool, adversaries are constrained from capturing, altering, or deleting transmissions across the network unless they can derive new information from their existing knowledge. The protocol was analyzed on a system configuring 12th Gen Intel(R) Core(TM) i5-1240P 1.70 GHz, 8 GB RAM, and a 64-bit Windows 11 operating system. The Scyther tool employs claims to articulate and define the security requisites. These claims encompass a range of criteria, including Nisynch, Secret, Niagree, Alive, and Weakagree.

- **Secret:** The objective is to establish confidentiality measures that facilitate secure communication between the two participating parties. From the figure, it is clear that claim(EU, Secret, Keuuav), claim(GCS, Secret,sk(GCS)), and claim(UAV, Secret, Kuaveu) represent the shared session key, which is confidential for secure communication, helps prevent session key attacks, and helps achieve mutual authentication.
- **Niagree:** The establishment of a non-injective agreement with a role concerning a set of data items can be achieved through the inclusion of the relevant signal claims like claim(EU, Niagree), claim(GCS, Niagree), and claim(UAV, Niagree), which prevents from tampering, smart device attack and provides mutual authentication.
- **Nisynch:** All processes related to data transmission and network sessions involving the entities must adhere rigorously to the security regulations delineated within the proposed protocol. It is paramount that all participating entities diligently uphold synchronization with their current operational states. From the figure, it is clear that claim(EU, Nisynch), claim(GCS, Nisynch), and claim(UAV, Nisynch) represent all entities that can send and receive all messages which will prevent replay and MITM attacks.
- **Alive:** The aim is to ensure a robust authentication process between the designated parties, focusing on enabling the execution of specific tasks by an intended communication partner. From the simulation results, it was found that claim(EU, Alive), claim(GCS, Alive), and claim(UAV, Alive) represent the trust among the entities, and each entity talks to the intended communicating partner.
- **Weakagree:** In professional terminology, one can assert that a protocol provides a form of weak agreement to an initiating party denoted as 'A' concerning another party, referred to as 'B,' when it ensures that whenever 'A' assumes the role of the initiator and successfully concludes a protocol session, ostensibly involving 'B' as the responder, it is implied that 'B' had been engaged in a prior execution of the same protocol, seemingly with 'A' as the initiator. In the domain of SPDL programming, which facilitates input provision to the Scyther tool, security assertions are appended to the conclusion of each role. These assertions serve as essential criteria enabling entities to assess whether the protocol has successfully passed the verification process as intended and whether the predefined security goals have been achieved. From the simulation result, it is clear that the claim(EU, Weakagree), claim(UAV, Weakagree), and claim(GCS, Weakagree) represent an impersonation attack that an adversary cannot perform. Scyther conducts a comprehensive assessment of security claims within the protocol. In identifying any security vulnerabilities or attacks, it presents a graphical representation of the security breach. To elaborate, Scyther specifically scrutinises secrecy and authentication aspects in the context of security protocols. The proposed framework has been implemented, encompassing three fundamental roles: EU, GCS, and UAVs. The system model of mutual authentication in IoT-based UAV Networks will satisfy the above-mentioned claims for security requirements using the simulation of the proposed protocol, as shown in Fig. 6. According to verification findings, all roles satisfy the requirements for being alive, Niagree, and Nisynch. Moreover, no attacks within the bounds were found, which means EU and UAV parameters, i.e. $Sess\_Key_{EU/UAV}$, Pri_Key_{EU}, and Pri_Key_{GCS} are secured from the attacker.

6.1.2. ROM based security assessment

The ROM model is a formal provable security analysis that validates the session key ($Sess\_Key_{EU \to UAV/UAV \to EU}$) security from Attacker A. This builds the groundwork for integrating the HCFAIUN with ROM. The model posits various queries such as Execute, Corrupt, Reveal and Test, which are required for adversary attack analysis as given in Table 3. The core terms associated with ROM are the following:

- **Random Oracle:** The selected one-way cryptographic function acts as a random oracle $hash(\cdot)$.
- **Participants:** Participants are the entities indulging in the communication, and the entities present in our protocols are EU, GCS and UAV. We denote the instances INS_1, INS_2, and INS_3 of EUi, GCS, and UAV as $\chi_{EU}^{INS_1}$, $\chi_{GCS}^{INS_2}$, $\chi_{UAV}^{INS_3}$ which act as oracles.
- **Partnerships:** EU and UAV will become partners if they retain a securely shared session key. The two instances $\chi_{EU}^{INS_1}$, $\chi_{UAV}^{INS_3}$ during the acceptance state can become partners if they own a common session key($Sess\_Key_{EU/UAV}$).
- **Freshness:** Freshness is achieved when Attacker (A) cannot leak the session key details maintained between $\chi_{EU}^{INS_1}$, $\chi_{UAV}^{INS_3}$
- **Attacker**: The adversary or attacker (A) model is mentioned in Section 4.2

Definition *(Semantic Security of Sess_Key)*. The foundation for the secrecy of $Sess\_Key$ shared between EU and UAV is the difficulty in discovering the real session key generated from an arbitrary number through an attacker (A). A has the advantage of violating the semantic security of HCFAIUN protocol by leakage of $Sess_{Key}$ information, which is described as:

$$Adv_A^{HCFAIUN} = |2.Pr[b' = b] - 1| \tag{51}$$

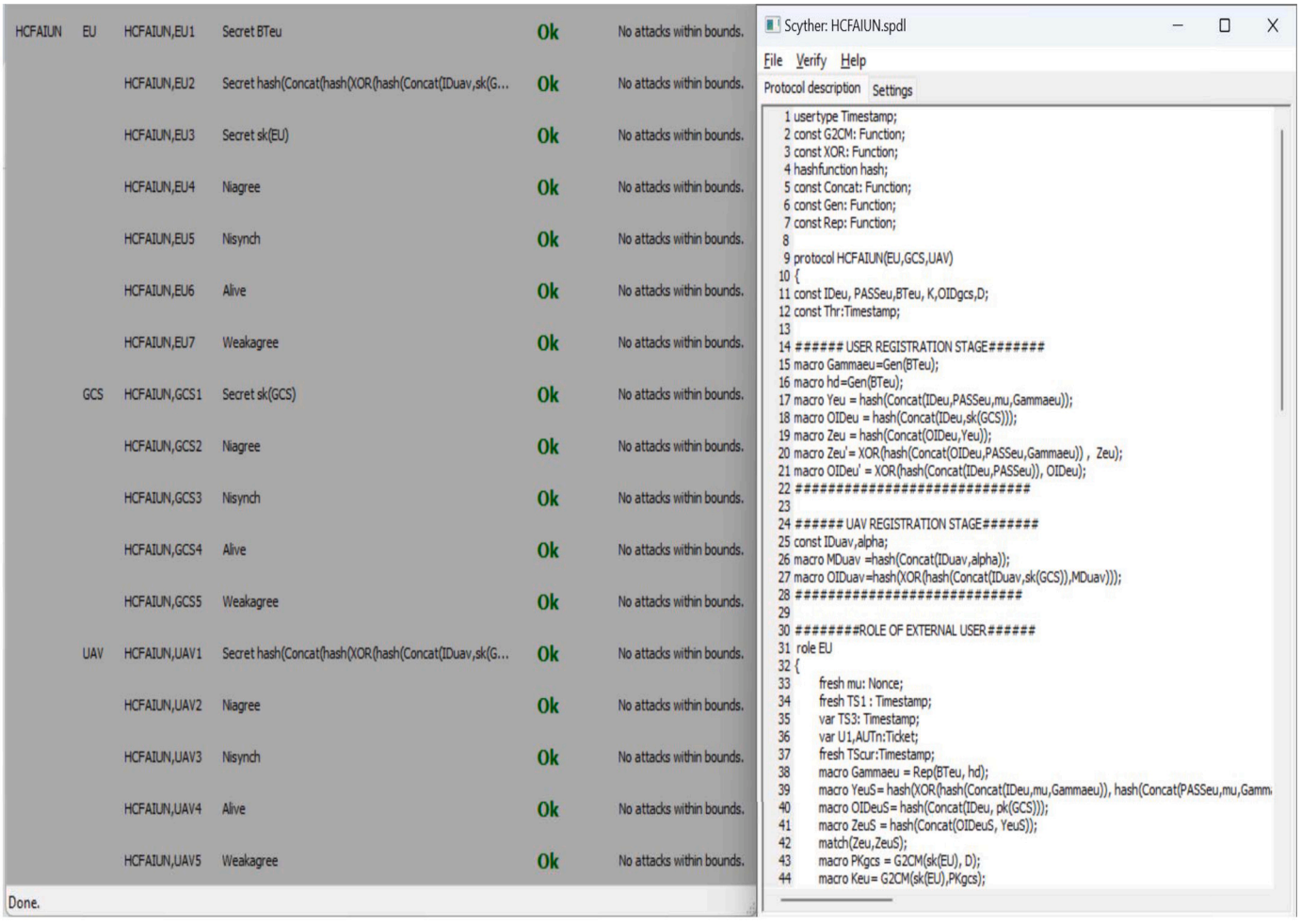


Fig. 6. Scyther tool results.

Table 3
ROM Queries.

Query	Description
Execute($\chi_{EU}^{INS_1}$,$\chi_{GCS}^{INS_2}$,$\chi_{UAV}^{INS_3}$)	Attacker A can gain the complete information of messages exchanged with the help of this query
Corrupt(χ^{INS_1})	Attacker A performs malicious activity by executing a Corrupt query to gather secret credentials of External user
Reveal(χ^{INS_1}, MSG)	Attacker A runs the reveal() query to obtain the Session key generated between UAV and external user.
Test(χ^{INS_1})	Attacker A runs this query to decide whether the session is actual or random.

where b and b' denote "correct" bits and "guess" bits, respectively, and Pr[b'=b] denotes the success probability.

Theorem 1. *Suppose a polynomial time (T_{poly}) in which attacker A attempts to gain $Sess\_Key$ information $Sess\_Key_{UAV\rightarrow EU} = Sess\_Key_{EU\rightarrow UAV}$ in login and authentication stage of HCFAIUN then the advantage is mentioned as:*

$$Adv_A^{HCFAIUN}(T_{poly}) \leq \frac{Hash_Q^2}{|Hash|} + 2Adv_A^{HCDLP}(T_{poly}) \quad (52)$$

Where terms like $Hash\_Q$ and $|Hash|$ denote the number of hash queries and the length of the one-way hash function with collision resistance, respectively. Moreover, $Adv_A^{HCDLP}(T_{poly})$ represents the advantage of attacker A to compromise the HCDLP security in T_{poly}.

Proof. We illustrate the proof of Theorem by the following three games such as ($Game_k^A | k = 0, 1, 2$), which are played by Attacker A such that in each game, if A can predict the random bit b in $Game_k^A$ correctly, then it wins the game with event $Success_{Game_k}^A$. The probability of attacker A winning the $Game_k^A$ is given by the $Adv_{A,Game_k}^{HCFAIUN} = Pr[Success_{Game_k}^A]$. The demonstration of the three games played by attacker A is as follows: $Game_0^A$: Attacker A plays the game and performs an actual attack by taking a random bit b;then from the definition of semantic security, it is given as:

$$Adv_A^{HCFAIUN}(T_{poly}) = |2Adv_{A,Game_k}^{HCFAIUN} - 1| \quad (53)$$

$Game_1^A$: In this Game, Attacker A can gain the complete information of messages such as $M_1 = (EU_1, EU_2, EU_3, Pub\_Key_{EU}, TS_1), M_2 = (B_1, B_2, B_3, TS_2)$ and $M_3 = (U_1, AUT_N, TS_3)$ from eavesdropping attack by queries like Execute query during Login and Authentication Stage. Attacker A runs the reveal() and test() query to obtain Session key, $Sess\_Key_{UAV\rightarrow EU} = hash(OID_{UAV} \parallel OID_{GCS} \parallel OID'_{EU}{}^* \parallel TS_3 \parallel R_2) = Sess\_Key_{EU\rightarrow UAV}$ which is generated between the UAV and the user. The HCFAIUN protocol uses a one-way collision-resistant hash function which protects anonymous id such as $OID_{UAV}, OID_{GCS}, OID'_{EU}$ along with a timestamp and random nonce, R2*. Thus, an eavesdropping attack does not impact the leakage

of the shared session key, which makes $Game_0^A$ and $Game_1^A$ identical, which is shown below:

$$Adv_{A,Game_0}^{HCFAIUN} = Adv_{A,Game_1}^{HCFAIUN} \tag{54}$$

$Game_2^A$: In this scenario, Attacker A performs malicious activity by executing a Corrupt query to gather secret credentials of EU such as ID_{EU} and $PASS_{EU}$. Moreover, if the attacker gains the secrete credentials stored in memory, then predicting $M_1 = (EU_1, EU_2, EU_3, Pub\_Key_{EU}, TS_1)$ is difficult due to the one-way hash function with collision resistance and HCDLP computation hardness to predict Pri_Key_{EU}. Additionally, $Game_1^A$ and $Game_2^A$ are hard to distinguish if hash queries and computational HCDLP are ignored. Utilizing the birthday paradox to find the hash collision along with the advantage of computing HCDLP is given as:

$$\begin{aligned} &|Adv_{A,Game_1}^{HCFAIUN} - Adv_{A,Game_2}^{HCFAIUN}| \\ &\leq \frac{Hash_Q^2}{2|Hash|} + 2Adv_A^{HCDLP}(T_{poly}) \end{aligned} \tag{55}$$

Attacker A, on completing all games $(Game_k^A | k = 0, 1, 2)$, does not gain any valid bit to win the game. Thus, we obtain the equation:

$$|Adv_{A,Game_2}^{HCFAIUN}| \leq \frac{1}{2} \tag{56}$$

From equation (53) and (54), we get

$$\frac{1}{2} Adv_A^{HCFAIUN}(T_{poly}) = |Adv_{A,Game_0}^{HCFAIUN} - \frac{1}{2}| \tag{57}$$

From the triangular inequality result and equations (53)–(55), we get,

$$\begin{aligned} \frac{1}{2} Adv_A^{HCFAIUN}(T_{poly}) &= |Adv_{A,Game_0}^{HCFAIUN} - Adv_{A,Game_2}^{HCFAIUN}| \\ &= |Adv_{A,Game_1}^{HCFAIUN} - Adv_{A,Game_2}^{HCFAIUN}| \\ &\leq \frac{Hash_Q^2}{2|Hash|} + Adv_A^{HCDLP}(T_{poly}) \end{aligned} \tag{58}$$

The final equation can be obtained by multiplying the equation (58) on both sides by 2.

$$Adv_A^{HCFAIUN}(T_{poly}) \leq \frac{Hash_Q^2}{|Hash|} + 2Adv_A^{HCDLP}(T_{poly}) \tag{59}$$

6.2. Informal security assessment

The protocol's safety management is commonly subjected to informal validation by analyzing it for vulnerability to numerous kinds of attacks. Its effectiveness and security assuredness are guaranteed. This section is aimed at conducting a complete security assessment of HCFAIUN, where well-known attacks are selected to showcase the defense capabilities for the system.

6.2.1. Replay attack

In the login and authentication stage, all of the messages M_1, M_2, and M_3 contain some enciphered messages such as (EU_1) in M_1 and M_1 carries timestamp TS_1. The entity GCS will verify the timestamp TS_1 with the timestamp contained in the enciphered message (EU_1) on receiving the message from the EU. If the matching fails, the recipient will quickly learn that the message has undergone unauthorized modifications by the attacker A. The same scenario is followed for message M_2. Therefore, our protocol remains resilient in the face of replay attacks.

6.2.2. Man in the middle attack

If the hacker (A) tries to intercept and modify the contents of messages such as $M_1, M_2, and M_3$ to generate a false breach by disguising himself as a real participant. This illicit activity will be pointless because the attacker cannot emit authentication tokens or validate them. They cannot delay or forge messages due to fresh timestamps and unidirectional cryptographic $hash(\cdot)$, which guarantees the authenticity and validity of messages. As a result, the suggested strategy displays resistance against MITM assaults.

6.2.3. Impersonation attack

6.2.3.1. Protection for UAV If a malicious entity attempts to assume the identity of a registered UAV, denoted as UAV, they must generate authentic messages denoted as $AUT_N = hash(Sess\_Key_{UAV} \parallel TS_3)$ and successfully transmit them to the intended recipient, EU. However, it's important to note that AUT_N encapsulates the session key $Sess\_Key_{UAV}$, which remains beyond the attacker's reach and Upon receipt of the message AUT_N, EU proceeds to compute $AUT_N{}^*$ and subsequently compares it with AUT_N to assess their validity. Consequently, the EU can distinguish between an attacker posing as a drone and a genuine registered drone. This capability underscores the security of the proposed scheme against drone impersonation attacks.

6.2.3.2. Protection for user Under the information provided during the 2nd stage of the access and verification process, the GCS verifies the identity of user EU by calculating $EU_3{}^*$ and subsequently comparing it to the EU_3 value received from EU. In an attempt to impersonate U_i, one method available to an attacker involves the creation of legitimate messages in the form of $EU_3 = hash(OID_{EU} \parallel OID_{GCS} \parallel OID_{UAV} \parallel K_{EU} \parallel TS_1)$, which are then transmitted to the GCS. It is important to note that the attacker can generate their timestamp (TSattacker), but they lack access to the confidential parameters, which encompass K_{EU} and GCS's private key (Pri_Key_{GCS}). These confidential parameters remain beyond the reach of the adversary. Consequently, the adversary cannot generate a legal EU_3, thereby enabling GCS to discern the impostor from the legitimate user.

6.2.3.3. Protection for GCS In this assault, the attacker A acts as an authentic enrolled GCS and intercepts the authentication message M_2 between the GCS and the UAV. The attacker may construct modified or fraudulent communications by obtaining crucial data from the GCS to demonstrate his legitimacy. The attacker constructs legitimate information M_2 by generating slot TS_2 & a new arbitrary numeral R1. Due to a lack of knowledge regarding OID_{UAV}, OID_{GCS}, and OID_{EU}, the attacker cannot compute B_1, B_2, B_3, or alter M_3. As a result, it ensures that the attacker cannot fabricate or alter the confidential message of the GCS in polynomial time. Thus, the HCFAIUN protocol remains resilient to GCS impersonation attempts.

6.2.4. Session key attack

The session key derived as $Sess\_Key_{UAV \to EU} = hash(OID_{UAV} \parallel OID_{GCS} \parallel OID'_{EU}{}^* \parallel TS_3 \parallel R_2)$ incorporates unique random numbers specific to the ongoing session. Including the trapdoor hash function ensures that the intruder (A) cannot extract arbitrary numerals such as R_2 from the session key. Consequently, even if an attacker gains possession of a previous session key, they are precluded from obtaining access to the current session key. This characteristic underscores the HCFAIUN protocol resilience against known session key attacks.

6.2.5. DoS attack

During the access phase or a password update operation, if an enrolled EU provides an incorrect ID_{EU} and $PASS_{EU}$, a local validation process is employed, which includes verifying the condition $Y_{prev} = Y_{updated}$. Once this validation is completed, the external user EU login request is relayed to the GCS. Additionally, password updates are exclusively allowed when the old password is verified successfully during the update procedure. As a result, the HCFAIUN protocol exhibits resilience against DoS attacks of this nature.

6.2.6. Smart device attack

If an attacker A steals or loses the smart device belonging to a registered user EU. It is possible to extract all information Z'_{EU}, OID'_{EU}

stored in the device's memory through power analysis attacks. Where $Z'_{EU} = hash(OID_{EU} \parallel Y_{EU}) \oplus Z_{EU}$ and $OID'_{EU} = hash(ID_{EU} \parallel PASS_{EU}) \oplus OID_{EU}$. Despite this knowledge, the attacker is unable to reliably deduce OID_{EU} and $PASS_{EU}$ from the collected data without the safe factor Y_{EU}. Moreover, applying a unidirectional trapdoor hashing (SHA-1) averts the intruder from concurrently recovering the confidential details. Nevertheless, the assailant remains incapable of obtaining the secret parameters of the EU. As a result, the HCFAIUN protocol is resilient against assaults involving the theft of mobile devices.

6.2.7. Physical UAV attack

As previously mentioned, a potential attacker's physical seizure of a UAV is a legitimate concern. Let us consider a scenario in which an attacker has successfully taken control of a UAV and gained access to all stored credentials and communication data, specifically $\{ID_{UAV}, OID_{UAV}\}$. It is important to note that the Pri_Key_{GCS} is securely stored within a unidirectional hash (SHA-1), effectively protecting any malicious actor by calculating the subsequently shared key without requisite information of the arbitrary value (R_1*) and obfuscation identity (OID_{GCS}, OID_{EU}). The confidential information differs for each individual deployed UAV, so attacker A cannot produce shared keys for UAVs and the EU. Consequently, the HCFAIUN demonstrates robustness against physical drone capture attacks.

6.2.8. Tampering attack

An intruder A may manipulate the authentication and response details. We employ a unidirectional trapdoor hash method (SHA-1) to mitigate this risk and safeguard against unauthorized alterations. It is important to note that the transmitted message EU_3 includes the recipient's (sender's) secret key K_{EU}. The GCS can distinguish any modifications to the message with the help of equation $EU_3 = EU_3$*. Additionally, the user EU can detect any changes to the authentication component by verifying the equation $AUT = AUT_N$*. Consequently, the HCFAIUN protocol maintains its resilience against tampering attacks.

6.2.9. Password and biometric modification attack

In this attack, attacker A can gather sensitive credentials of EU such as Z'_{EU}, OID'_{EU}, $Gen(\cdot)$, $Rep(\cdot)$, hd, μ by capturing the smart device. The purpose of A in this attack is to modify or update the $PASS_{EU}$ and biometric traits (BT_{EU}) of EU. To achieve this, A randomly chooses sensitive credentials such as ID^A_{EU}, $PASS^A_{EU}$ and BT^A_{EU}, and evaluate the following computations: $Gen(BT^A_{EU}) = (\gamma^A_{EU}, hd)$, $Y^A_{EU} = hash(ID^A_{EU} \parallel PASS^A_{EU} \parallel \mu \parallel \gamma^A_{EU})$, $Z^A_{EU}{}' = hash(OID_{EU} \parallel PASS_{EU} \parallel \gamma^A_{EU}) \oplus Z_{EU}$, $OID^A_{EU}{}' = hash(ID^A_{EU} \parallel PASS^A_{EU}) \oplus OID_{EU}$. To complete the evaluation of the above parameters, it is difficult for attacker A to guess secret credentials of EU like OID_{EU}, Z_{EU}, ID_{EU}, $PASS_{EU}$ and BT_{EU}. Thus, guessing the password and modifying the smart device's information is impossible for the attacker.

6.2.10. Un-traceability

To guarantee that every participant's message is unique, at the authentication step, the random nonce, R_1, R_2, and the current timestamp TS_{cur} for each session are selected at random. The enemy adversary cannot connect the communications the GCS, UAV, and EU sent. Likewise, it's hard to track down the sender. In addition, a secure unidirectional hash function contains or conceals genuine identities. (ID, OID). Thus, the HCFAIUN approach may be used to create un-traceability.

6.2.11. Privacy preserving

To safeguard privacy and anonymity, the proposed protocol must ensure that an attacker cannot retrieve actual identities once our system is operational. Our protocol can safeguard the confidentiality of all sent and received communications, including M_1, M_2, and M_3. Moreover, these messages are generated using new periods and random integers. This provides a significant benefit in that it is harder for attackers to obtain sensitive information and actual identities from users, GCS, and drones. The HCFAIUN system, as a result, guarantees anonymity and privacy.

6.2.12. Accuracy and validity

Data Accuracy and Validity are the assurance that no attacker may alter the information that is sent, and if they do, the system will detect and report the alteration. First of all, attacker A find it difficult to infer the matching session key ($Sess\_Key_{UAV/EU}$) based on the HCDLP. Second, nodes use the one-way hash function to conduct an integrity check following each phase's message exchange. As a result, the HCFAIUN system has significantly more integrity maintenance security.

6.2.13. Forward secrecy

The Forward secrecy attribute ensures that the session key from the prior communication does not leak due to the compromised persistent key. Under the HCFAIUN method, participants must generate a new key ($Sess\_Key$) for every session. This new session key ($Sess\_Key$) must contain a random number that makes it difficult for attacker A to calculate or predict. Furthermore, the protocol incorporates a timestamp TS that checks current sessions. Therefore, the gathered secret key is irrelevant to the attacker in case of attempting to breach earlier sessions, indicating that our protocol guarantees absolute forward secrecy.

6.2.14. Authentication and key agreement

The common session key $Sess\_Key_{UAV \rightarrow EU} = hash(OID_{UAV} \parallel OID_{GCS} \parallel OID'_{EU}{}^* \parallel TS_3 \parallel R_2) = Sess\_Key_{EU \rightarrow UAV}$ is calculated among UAV and EU for secure communication which will be possible when mutual authentication is done.

7. Performance evaluation

This section conducts an exhaustive performance analysis, comparing HCFAIUN with other pertinent schemes.

7.1. Computation cost

In the mutual authentication phase, we evaluate the computing costs of the suggested method and previous work [3,19,17,22]. According to [3,17,22], $T_H, T_{ECM}, T_{HCM}, T_{FE}, T_P, T_{CM}, T_{AC}, T_{ENC/DEC}, T_E$ denote the hash function with 0.027 ms time for GCS and 0.06 ms for EU and UAV, ECC multiplication with 0.56 ms duration for GCS and 1.27 ms for EU and UAV, HC multiplication with 0.48 ms time, FE with 1.27 ms time for EU and UAV, Bilinear pairing with 5.6 ms operation time for EU and UAV and 3.61 ms operation time for GCS, chaotic map with 0.512 ms for GCS and 0.98 ms operation time for EU and UAV, AEGIS with computation time of 0.415 ms, encryption operation time of 0.5 ms for EU and UAV and 0.19 ms for GCS. Compared with previous work, our proposed work shows less computation cost, i.e. 3.832 ms, with high security against logical and physical attacks in IoT-based UAV networks, illustrated in Table 4 and Fig. 7.

7.2. Communication cost

To showcase the efficacy compared to prevailing methodologies [3,19,17,22], we assess the communication expenditures incurred by diverse entities involved in the login and authentication phases. This analysis focuses on transmitting messages among the participants during these stages. To gauge communication expenses, we posit that the sizes for the hash function (SHA-1), timestamp, HC, elliptic curve point, identity, and random number are 160, 32, 80, 160, 160, and 160 bits, respectively. To transmit message, $M_1 = (EU_1, EU_2, EU_3, Pub\_Key_{EU}, TS_1)$ the cost will be (160+160+160+80+32 = 592 bits). Similarly for $M_2 = (B_1, B_2, B_3, TS_2)$, cost will be (160+160+160+32 = 512) and to transmit $M_3 = (U_1, AUT_N, TS_3)$, the cost will be (160+160+32 = 352 bits).

Table 4
Comparative study of computation costs.

Schemes	Ever [17]	Tanveer et al. [3]	Rajasekaran et al. [19]	Tanveer et al. [22]	Proposed HCFAIUN
EU Side	$3T_H + 2T_P$(11.38)	$6T_H + 3T_{AC} + 3T_{ECM} + T_{FE}$(6.685)	$4T_E + T_P + T_H$(8.06)	$6T_H + 3T_{CM} + T_{FE}$(6.57)	$10T_H + 2T_{HCM} + T_{FE}$(2.83)
GCS Side	$9T_H + 2T_P + 4T_{ECM}$(13.683)	$2T_H + T_{ECM} + 3T_{AC}$(0.824)	-	$2T_H + T_{CM} + T_{ENC}$(0.756)	$6T_H + T_{HCM}$(0.642)
UAV Side	$5T_H + 2T_P$(11.335)	$3T_H + 2T_{ECM} + 2T_{AC}$(3.451)	$3T_E + 2T_P + T_H$(13.06)	$4T_H + T_{CM} + 2T_E$(2.22)	$6T_H$(0.36)
Total (ms)	36.398 ms	10.96 ms	21.12 ms	9.54 ms	3.832 ms

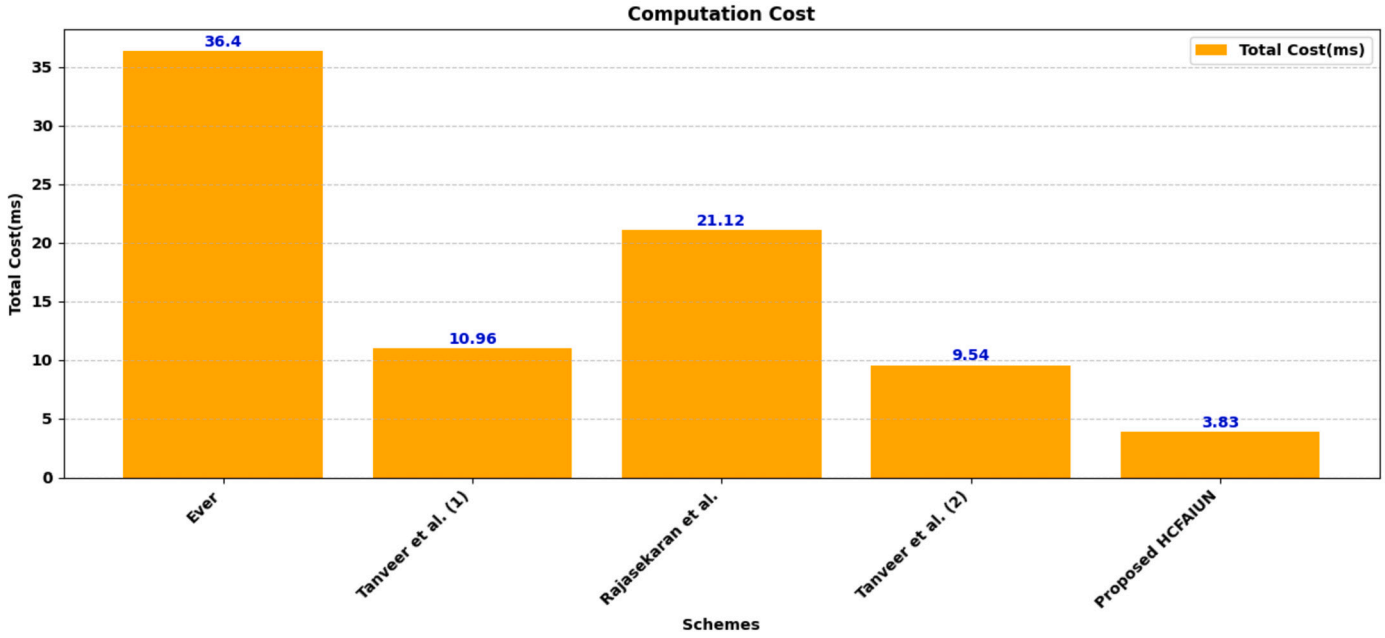


Fig. 7. Comparison of computation costs.

Table 5
Comparative study of communication cost.

Schemes	Ever et al. [17]	Tanveer et al. [3]	Rajasekaran et al. [19]	Tanveer et al. [22]	Proposed HCFAIUN
Total Cost (bits)	1920	1856	1184	1664	1456
Messages	3	3	2	3	3

Therefore, the total cost will be 1456 bits, which is less than the existing ones with three messages exchanged as illustrated in Table 5 and Figs. 8-9.

7.3. Storage cost

The proposed protocol provides insights into storage overhead complexities and demonstrates a comparison with existing schemes [3,19,17,22] during the login and authentication stage, as shown in Table 6 and Fig. 10. As UAVs are resource-constrained devices which often carry limited storage on board. Therefore, a reduction of storage costs is necessary. In the HCFAIUN protocol, the cost required to store the information $\{Z_{EU}, OID_{EU}, Gen(\cdot), Rep(\cdot), hd, \mu\}$,$\{OID_{UAV}\}$ and $\{OID_{EU}, OID_{UAV}\}$ at EU, UAV and GCS are $\{160+160+160+8+160\} = 648$ bits, 160 bits and $\{160+160\} = 320$ bits, respectively. Therefore, the total storage overhead in this scheme is 1128 bits, which is lower than existing schemes.

7.4. Comparative analysis of security features

The proposed HCFAIUN scheme is provably secure and lightweight as compared to the previous schemes [3,19,17,22] and the security features comparison is shown in Table 7. In the case of Tanveer et al. [3], the scheme is prone to various attacks, such as session key attacks, physical UAV attacks and tampering attacks with the lack of integrity and forward secrecy. This proposed scheme is secured against user and device impersonation attacks without discussing its effect on UAVs and GCS. Regarding Rajasekran [19], the scheme is vulnerable to session key attacks, DoS attacks, smart device attacks, physical UAV attacks, tampering attacks and biometric modification attacks. Additionally, this scheme lacks forward secrecy and dynamic device addition and no formal security analysis is provided for the session key. As per Ever [17], the discussed scheme is prone to an MITM attack, impersonation attack at the user and GCS side, DoS attack, smart device attack, tampering attack, password and biometric modification with weakness against un-traceability, privacy-preserving, integrity, forward secrecy, dynamic

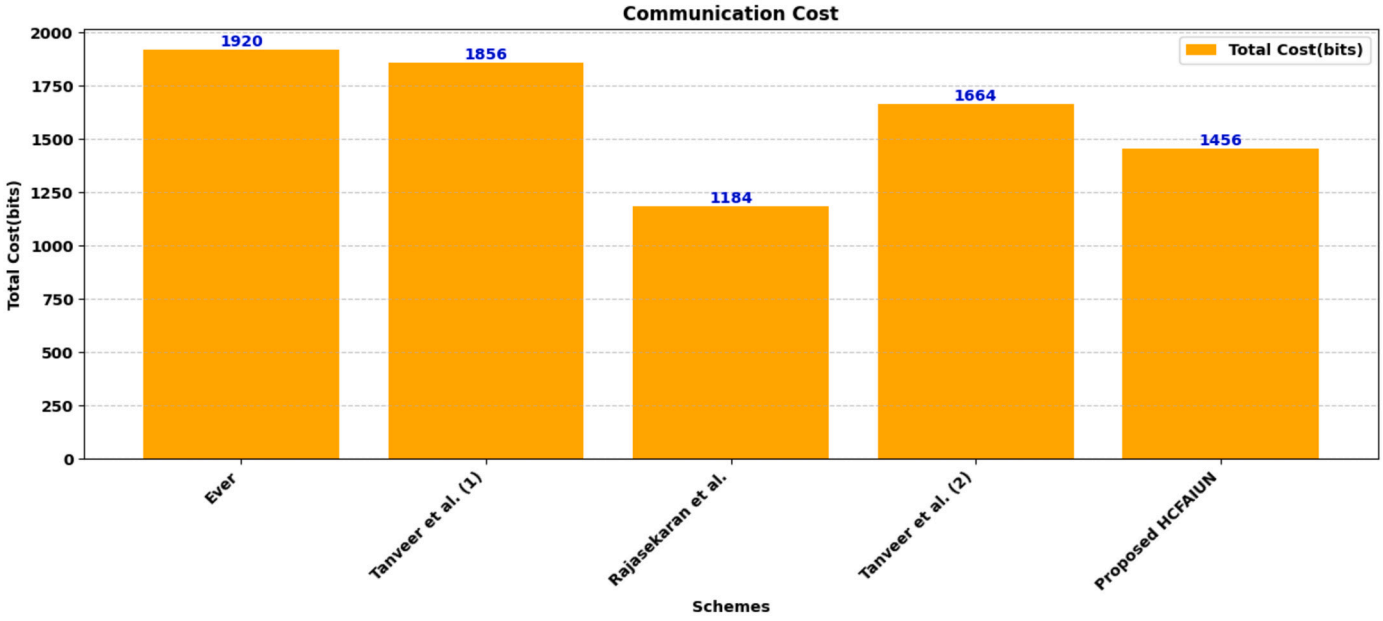


Fig. 8. Comparison of communication costs.

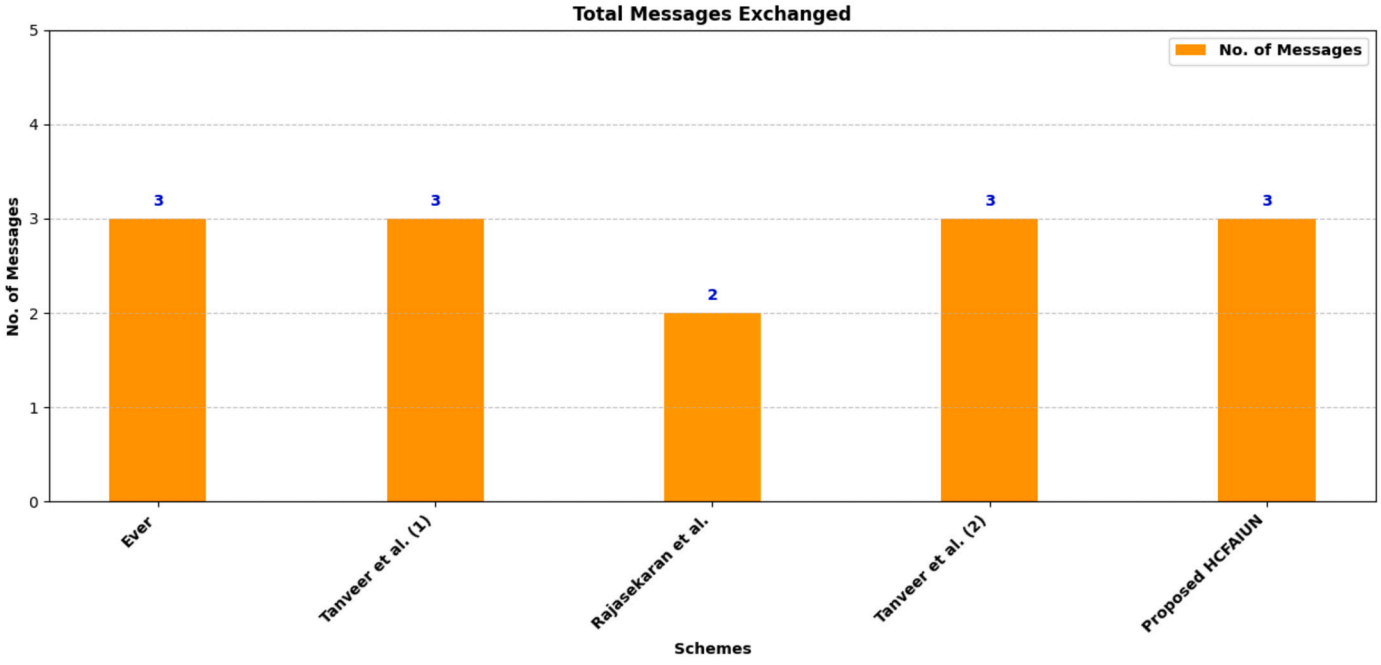


Fig. 9. Comparison of total messages exchanged.

Table 6
Comparative study of storage costs.

Schemes	Ever [17]	Tanveer et al. [3]	Rajasekaran et al. [19]	Tanveer et al. [22]	Proposed HCFAIUN
EU Side	160 bits	536 bits	1152 bits	824 bits	648 bits
GCS Side	128 bits	592 bits	-	608 bits	320 bits
UAV Side	1184 bits	256 bits	1046 bits	416 bits	160 bits
Total (bits)	1472 bits	1384 bits	2198 bits	1848 bits	1128 bits

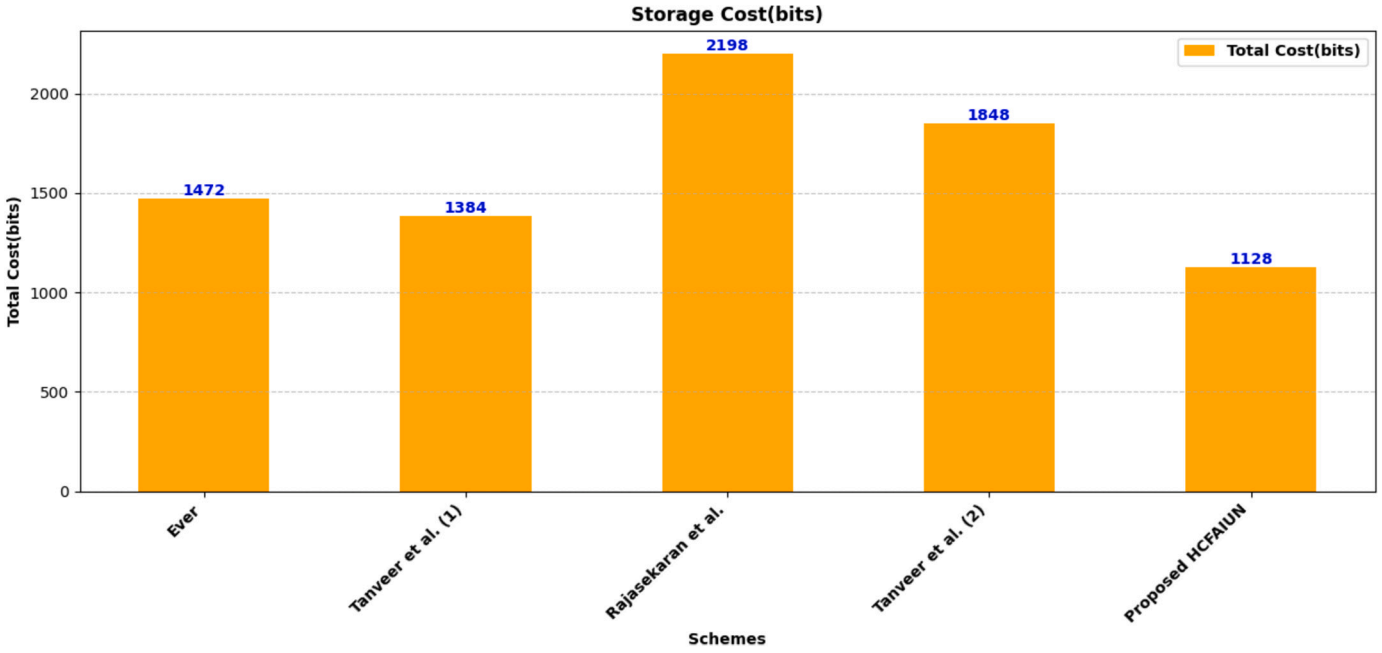


Fig. 10. Comparison of storage costs.

Table 7
Comparative study of security features.

Features	Tanveer et al. [3]	Rajasekaran et al. [19]	Ever et al. [17]	Tanveer et al. [22]	Proposed HCFAIUN
SF1	✓	✓	✓	✓	✓
SF2	✓	✓	×	✓	✓
SF3	∂	✓	∂	✓	✓
SF4	×	×	×	✓	✓
SF5	✓	×	×	×	✓
SF6	✓	×	×	×	✓
SF7	×	×	✓	×	✓
SF8	×	×	×	×	✓
SF9	✓	×	×	✓	✓
SF10	✓	✓	×	✓	✓
SF11	✓	✓	×	×	✓
SF12	×	✓	×	✓	✓
SF13	×	×	×	✓	✓
SF14	✓	✓	✓	✓	✓
SF15	✓	×	×	×	✓
SF16	✓	×	×	✓	✓
SF17	✓	✓	✓	✓	✓

Note: ✓ - Discussion on security features, **x**- Not discussed ∂-partial information, SF1-Replay Attack, SF2-MITM attack, SF3-Impersonation attack (EU, GCS, UAV), SF4-Session Key Attack, SF5-DoS Attack, SF6-Smart Device Attack, SF7-Physical UAV Attack, SF8-Tampering Attack, SF9-Password and Biometric Modification Attack, SF10-Un-traceability, SF11- Privacy preserving, SF12-Integrity, SF13-Forward secrecy, SF14-Mutual Authentication and key agreement, SF15-dynamic device addition, SF16-formal security analysis, SF17-informal security analysis.

device addition and no formal security assessment. In the case of Tanveer et al., [22], the scheme offers protection against various attacks such as replay and MITM attacks, but it is weak against DoS attacks, Smart Device attacks, physical UAV attacks, tampering attacks, privacy-preserving and dynamic device addition. Most of the schemes are based on a high-cost chaotic map, bilinear pairing cryptography, and a genus-1 elliptic curve without a strong key generation strategy. Consequently, our proposed scheme demonstrates enhanced security and functional attributes compared to previous schemes by utilising an HC scalar multiplication and FE to keep the private key secure and generate strong keys.

8. Conclusion

UAVs have inherent privacy and security vulnerabilities due to their lack of comprehensive security mechanisms. These difficulties arise due to aerial vehicles' reliance on wireless connectivity and limited computer capabilities. The primary objective is to improve the security of EU and UAV communications while reducing computational, communication and storage costs in smart city environments during emergency situations. This study offers a Hyperelliptic curve and Fuzzy extractor-based authentication in IoT-based UAV networks that is effective and safe as per security requirements, leveraging HCC to satisfy the criterion of decreased computation and communication costs. The proposed protocol improves the elliptic curve using lower parameters and key sizes. Unlike typical bilinear pairing-based cryptography with exponential operations and elliptic curves, which require a 160-bit key size, HCC only requires a maximum of 80 bits, making it ideal for resource-constrained UAVs. This protocol also employs an FE mechanism to generate biometric traits of the user, such as a key which can be reproducible to prevent exposing data from stealing smart devices. The lower storage overhead in HCFAIUN eliminates the resource limitation of UAVs. Security and

performance evaluations using Scyther formal and informal analyses, including comparative analyses, show that our proposed solution is secure. The real-world application of this protocol is to assist smart cities in rescue, package deliveries, and predicting traffic behaviour by firefighting service vehicles and ambulance drivers without any large computation and communication delay. HCFAIUN protocol can protect sensitive information like property destruction of people during natural disasters from physical and logical attacks on UAVs. Besides its advantages, the HCFAIUN scheme is based on fixed UAV topology in smart city scenarios. It can be vulnerable to quantum-based attacks, as quantum computers can solve complex HCDLP in seconds. In the future, our objective is to evaluate the effectiveness of our methodology in real-world conditions with dynamic UAV topologies consideration. This work can be enhanced by utilising a post-quantum-based cryptosystem.

CRediT authorship contribution statement

Jatin Sharma: Writing – original draft, Visualization, Validation, Methodology, Formal analysis. **Pawan Singh Mehra:** Writing – original draft, Supervision, Formal analysis, Conceptualization.

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.

Data availability

No data was used for the research described in the article.

References

[1] R. Randhawa, M. Verma, A comprehensive review on recent advancements in the field of Internet of things, its challenges and future scope, Optoelectron. Instrum. Data Process. 59 (1) (2023) 137–147, https://doi.org/10.3103/S8756699023010156, https://link.springer.com/10.3103/S8756699023010156.

[2] J. Sharma, P.S. Mehra, Secure communication in iot-based uav networks: a systematic survey, Int. Things 23 (2023) 100883, https://doi.org/10.1016/j.iot.2023.100883, https://www.sciencedirect.com/science/article/pii/S2542660523002068.

[3] M. Tanveer, A.U. Khan, N. Kumar, M.M. Hassan, RAMP-IoD: a robust authenticated key management protocol for the Internet of drones, IEEE Int. Things J. 9 (2) (2022) 1339–1353, https://doi.org/10.1109/JIOT.2021.3084946, https://ieeexplore.ieee.org/document/9446966/.

[4] S.A.H. Mohsan, N.Q.H. Othman, Y. Li, M.H. Alsharif, M.A. Khan, Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends, Intell. Serv. Robot. (Jan. 2023), https://doi.org/10.1007/s11370-022-00452-4, https://link.springer.com/10.1007/s11370-022-00452-4.

[5] D. Chawla, P.S. Mehra, A survey on quantum computing for Internet of Things security, in: International Conference on Machine Learning and Data Engineering, Proc. Comput. Sci. 218 (2023) 2191–2200, https://doi.org/10.1016/j.procs.2023.01.195, https://www.sciencedirect.com/science/article/pii/S1877050923001953.

[6] D. He, S. Chan, M. Guizani, Communication security of unmanned aerial vehicles, IEEE Wirel. Commun. 24 (4) (2017) 134–139, https://doi.org/10.1109/MWC.2016.1600073WC, http://ieeexplore.ieee.org/document/7792372/.

[7] D. Chawla, P.S. Mehra, A roadmap from classical cryptography to post-quantum resistant cryptography for 5g-enabled iot: challenges, opportunities and solutions, Int. Things 24 (2023) 100950, https://doi.org/10.1016/j.iot.2023.100950, https://www.sciencedirect.com/science/article/pii/S2542660523002731.

[8] K. Messaoudi, O.S. Oubbati, A. Rachedi, A. Lakas, T. Bendouma, N. Chaib, A survey of UAV-based data collection: challenges, solutions and future perspectives, J. Netw. Comput. Appl. 216 (2023) 103670, https://doi.org/10.1016/j.jnca.2023.103670, https://linkinghub.elsevier.com/retrieve/pii/S1084804523000899.

[9] M. Tanveer, A. Aldosary, N. Kumar, S.A. Aldossari, SEAF-IoD: secure and efficient user authentication framework for the Internet of Drones, Comput. Netw. 247 (2024) 110449, https://doi.org/10.1016/j.comnet.2024.110449, https://linkinghub.elsevier.com/retrieve/pii/S1389128624002810.

[10] A. Khan, M.M.A. Khan, M.A. Javeed, M.U. Farooq, A. Akram, C. Wang, Multilevel privacy controlling scheme to protect behavior pattern in smart IoT environment, Wirel. Commun. Mob. Comput. 2021 (2021) 1–17, https://doi.org/10.1155/2021/9915408, https://www.hindawi.com/journals/wcmc/2021/9915408/.

[11] S. Ogunbunmi, Y. Chen, E. Blasch, G. Chen, A survey on reputation systems for UAV networks, Drones 8 (6) (2024) 253, https://doi.org/10.3390/drones8060253, https://www.mdpi.com/2504-446X/8/6/253.

[12] M. Wazid, A.K. Das, N. Kumar, A.V. Vasilakos, J.J.P.C. Rodrigues, Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of Drones deployment, IEEE Int. Things J. 6 (2) (2019) 3572–3584, https://doi.org/10.1109/JIOT.2018.2888821, https://ieeexplore.ieee.org/document/8581510/.

[13] J. Srinivas, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, TCALAS: temporal credential-based anonymous lightweight authentication scheme for Internet of Drones environment, IEEE Trans. Veh. Technol. 68 (7) (2019) 6903–6916, https://doi.org/10.1109/TVT.2019.2911672, https://ieeexplore.ieee.org/document/8693567/.

[14] Y. Tian, J. Yuan, H. Song, Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones, J. Inf. Secur. Appl. 48 (2019) 102354, https://doi.org/10.1016/j.jisa.2019.06.010, https://linkinghub.elsevier.com/retrieve/pii/S2214212618307038.

[15] Z. Ali, S.A. Chaudhry, M.S. Ramzan, F. Al-Turjman, Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles, IEEE Access 8 (2020) 43711–43724, https://doi.org/10.1109/ACCESS.2020.2977817, https://ieeexplore.ieee.org/document/9020140/.

[16] T. Alladi, Naren, G. Bansal, V. Chamola, M. Guizani, SecAuthUAV: a novel authentication scheme for UAV-ground station and UAV-UAV communication, IEEE Trans. Veh. Technol. 69 (12) (2020) 15068–15077, https://doi.org/10.1109/TVT.2020.3033060, https://ieeexplore.ieee.org/document/9237145/.

[17] Y. Kirsal Ever, A secure authentication scheme framework for mobile-sinks used in the Internet of Drones applications, Comput. Commun. 155 (2020) 143–149, https://doi.org/10.1016/j.comcom.2020.03.009, https://linkinghub.elsevier.com/retrieve/pii/S014036641930790X.

[18] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, K.-K.R. Choo, Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of Drones, IEEE Int. Things J. 9 (8) (2022) 6224–6238, https://doi.org/10.1109/JIOT.2021.3113321, https://ieeexplore.ieee.org/document/9540762/.

[19] A.S. Rajasekaran, A. Maria, F. Al-Turjman, C. Altrjman, L. Mostarda, Anonymous mutual and batch authentication with location privacy of UAV in FANET, Drones 6 (1) (2022) 14, https://doi.org/10.3390/drones6010014, https://www.mdpi.com/2504-446X/6/1/14.

[20] V.O. Nyangaresi, Provably secure authentication protocol for traffic exchanges in unmanned aerial vehicles, High-Confidence Comput. 3 (4) (2023) 100154, https://doi.org/10.1016/j.hcc.2023.100154, https://linkinghub.elsevier.com/retrieve/pii/S2667295223000521.

[21] K. Mahmood, Z. Ghaffar, M. Farooq, K. Yahya, A.K. Das, S.A. Chaudhry, A security enhanced chaotic-map-based authentication protocol for Internet of Drones, IEEE Int. Things J. 11 (12) (2024) 22301–22309, https://doi.org/10.1109/JIOT.2024.3379930, https://ieeexplore.ieee.org/document/10477269/.

[22] M. Tanveer, H. Alasmary, N. Kumar, A. Nayak, SAAF-IoD: secure and anonymous authentication framework for the Internet of Drones, IEEE Trans. Veh. Technol. 73 (1) (2024) 232–244, https://doi.org/10.1109/TVT.2023.3306813, https://ieeexplore.ieee.org/document/10229199/.

[23] L.C. Washington, Elliptic Curves: Number Theory and Cryptography, 2nd edition, Discrete Mathematics and Its Applications, Chapman & Hall/CRC, Boca Raton, FL, 2008, oCLC: ocn192045762.

[24] S. Zhang, Y. Liu, Z. Han, Z. Yang, A lightweight authentication protocol for UAVs based on ECC scheme, Drones 7 (5) (2023) 315, https://doi.org/10.3390/drones7050315, https://www.mdpi.com/2504-446X/7/5/315.

[25] G. Bansal, B. Sikdar, S-MAPS: scalable mutual authentication protocol for dynamic UAV swarms, IEEE Trans. Veh. Technol. 70 (11) (2021) 12088–12100, https://doi.org/10.1109/TVT.2021.3116163, https://ieeexplore.ieee.org/document/9551779/.

[26] S. Ullah, Z. Jiangbin, M.T. Hussain, N. Din, F. Ullah, M.U. Farooq, A perspective trend of hyperelliptic curve cryptosystem for lighted weighted environments, J. Inf. Secur. Appl. 70 (2022) 103346, https://doi.org/10.1016/j.jisa.2022.103346, https://linkinghub.elsevier.com/retrieve/pii/S2214212622001910.

[27] S. Hussain, I. Ullah, H. Khattak, M. Adnan, S. Kumari, S.S. Ullah, M.A. Khan, S.J. Khattak, A lightweight and formally secure certificate based signcryption with proxy re-encryption (CBSRE) for Internet of Things enabled smart grid, IEEE Access 8 (2020) 93230–93248, https://doi.org/10.1109/ACCESS.2020.2994988, https://ieeexplore.ieee.org/document/9094323/.

[28] R. Alimoradi, A study of hyperelliptic curves in cryptography, Int. J. Comput. Netw. Inf. Secur. 8 (8) (2016) 67–72, https://doi.org/10.5815/ijcnis.2016.08.08, http://www.mecs-press.org/ijcnis/ijcnis-v8-n8/v8n8-8.html.

[29] M.U.F. Qaisar, W. Yuan, P. Bellavista, S.A. Chaudhry, A. Ahmed, M. Imran, Reliable and resilient communication in duty cycled software defined wireless sensor networks, in: 2023 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, Rome, Italy, 2023, pp. 397–402, https://ieeexplore.ieee.org/document/10283622/.

[30] D. Chawla, P.S. Mehra, Qaka: a novel quantum authentication and key agreement (qaka) protocol using quantum entanglement for secure communication among iot devices, Trans. Emerg. Telecommun. Technol. 35 (3) (2024) e4957, https://doi.org/10.1002/ett.4957, https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.4957, https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4957.

[31] C. Pu, A. Wall, K.-K.R. Choo, I. Ahmed, S. Lim, A lightweight and privacy-preserving mutual authentication and key agreement protocol for Internet of drones environment, IEEE Int. Things J. 9 (12) (2022) 9918–9933, https://doi.org/10.1109/JIOT.2022.3163367, https://ieeexplore.ieee.org/document/9745033/.

[32] Y. Zhang, D. He, L. Li, B. Chen, A lightweight authentication and key agreement scheme for Internet of Drones, Comput. Commun. 154 (2020) 455–464, https://doi.org/10.1016/j.comcom.2020.02.067, https://linkinghub.elsevier.com/retrieve/pii/S0140366419319358.

[33] D. Chawla, P.S. Mehra, Qsmah: a novel quantum-based secure cryptosystem using mutual authentication for healthcare in the Internet of Things, Int. Things 24 (2023) 100949, https://doi.org/10.1016/j.iot.2023.100949, https://www.sciencedirect.com/science/article/pii/S254266052300272X.

[34] C. Cremers, Scyther tool, https://people.cispa.io/cas.cremers/scyther/index.html, 2014.