USAF-IoD: Ultralightweight and Secure Authenticated Key Agreement Framework for Internet of Drones Environment

Akhtar Badshah, Ghulam Abbas, *Senior Member, IEEE*, Muhammad Waqas, *Senior Member, IEEE*, Shanshan Tu, *Member, IEEE*, Ziaul Haq Abbas, Fazal Muhammad, and Sheng Chen, *Life Fellow, IEEE*

***Abstract*—The use of Internet of Drones (IoD) technology has surged across various domains such as logistics, surveying, industrial inspections, emergency response, security, infrastructure monitoring, crop management, and more. However, real-time communication with drones or Unmanned Aerial Vehicles (UAVs) in the IoD environment occurs over an insecure open channel, making it susceptible to various security and privacy vulnerabilities, including unauthorized access, data interception, denial of service attacks, and privacy concerns. Due to their unique characteristics, including long transmission distances, unstable communication environments, resource limitations, and the highly dynamic nature of UAVs, ensuring the security and privacy of IoD systems is of paramount importance for the success of IoD-based applications. Furthermore, drones are resource-constrained devices, and employing expensive security solutions is impractical, as it would significantly reduce the operational capacity of drones. In this paper, we present the design of an ultralightweight, secure, and robust user-authenticated key agreement framework for the IoD environment, named USAF-IoD. The proposed USAF-IoD is developed by incorporating authenticated encryption (ASCON), cryptographic hashing, XOR operations, and the use of physical unclonable functions (PUFs). PUFs are employed to enhance resistance against physical tampering attacks. The security analysis reveals that the proposed USAF-IoD meets the essential security requirements of the IoD environment. The comparative analysis further highlights the effectiveness of the proposed USAF-IoD, notably excelling in terms of security and functionality characteristics when compared to existing benchmark schemes, and showcasing competitive performance in computation, communication, and energy overheads.**

***Index Terms*—Internet of Drones, physical unclonable functions, user authentication, key agreement, security.**

I. Introduction

DRONES, originally developed for costly military applications, have found a growing presence in various commercial sectors in recent years. Their versatility has made them indispensable in applications such as logistics and distribution, surveying and mapping, industrial site inspections, emergency response, security monitoring, infrastructure and crop monitoring, and more [1], [2]. As industries continue to explore the potential of drones or unmanned aerial vehicles (UAVs), their applications continue to expand. Concurrently, the Internet of Things (IoT) technology has become more cost-effective and operationally efficient. Consequently, IoT is increasingly integrated into the deployment of drones in the commercial sector. These drones equipped with IoT capabilities, commonly known as the Internet of Drones (IoD), play a pivotal role, particularly in tasks that are expensive, risky, or impractical for human intervention [3].

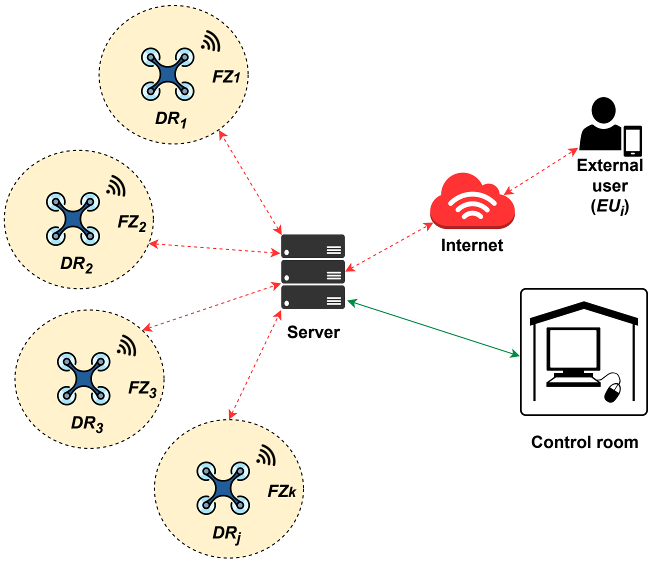

Fig. 1: Network architecture of IoD based system.

The architecture of the IoD-based network, as illustrated in Fig. 1, encompasses various entities, including external users, the control room (internal users), drones, and a server. Imagine a typical scenario where an external user requests access to the data collected by drones in a specific fly zone. It's crucial to recognize that the data collected by these drones are highly sensitive and could potentially be exploited as physical weapons if they fall into the wrong hands. Moreover, given the inherent openness of wireless networks, potential adversaries possess the capability to carry out a variety of attacks, such as eavesdropping, disruption, alteration, or replaying of aerial communications. Additionally, when we consider the unique characteristics of UAVs, such as extended transmission ranges, unpredictable communication conditions, resource limitations, and rapid dynamism, it becomes evident that ensuring the security and privacy of IoD systems is an absolute imperative for

A. Badshah is with the Department of Software Engineering, University of Malakand, Dir Lower, Pakistan (e-mail: akhtarbadshah@uom.edu.pk).

G. Abbas is with the Telecommunication and Networking (TeleCoN) Research Center, Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan (e-mail: abbasg@giki.edu.pk).

M. Waqas is with the School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, United Kingdom, and also with the School of Engineering, Edith Cowan University, Perth, 6007 WA, Australia (e-mail: engr.waqas2079@gmail.com).

S. Tu is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: sstu@bjut.edu.cn).

Z. H. Abbas is with Faculty of Electrical Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan (e-mail: ziaul.h.abbas@giki.edu.pk).

F. Muhammad is with the Department of Electrical Engineering, University of Engineering and Technology, Mardan 23200, Pakistan (e-mail: fazal.muhammad@uetmardan.edu.pk).

S. Chen is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: sqc@ecs.soton.ac.uk).

the success of IoD-driven applications [4], [5]. Consequently, safeguarding the devices involved and their communications within IoD environments is of paramount importance, a topic that has been extensively studied, as evidenced by numerous previous works, such as [6].

One of the key security solutions to address these challenges is the deployment of authenticated key agreement (AKA) schemes. AKA schemes authenticate the legitimacy of communicating entities (i.e., drones, users, and server) and establish a confidential session key before transmitting any sensitive data across an unsecured open channel [7]–[10]. However, it's worth noting that drones typically have limited computational capabilities, making it challenging to implement fully mature security solutions with high computational complexity. Furthermore, any security measures introduced should not negatively impact the performance of the drone. For instance, excessive battery usage in implementing security measures can significantly degrade the drone's operational capabilities. Moreover, there is a risk that an adversary may seize the drone and attempt to extract secret credentials stored in its memory. Therefore, tamper resistance is crucial to reduce the chances of compromising cryptographic security credentials. Additionally, some IoD applications demand privacy-preserving features to support entity anonymity, intractability, and non-linkability.

In light of the aforementioned challenges and considerations, this paper focuses on the development of a secure communication framework for users in the IoD environment. We leverage security and performance analysis to assess and quantify the trade-off between security robustness and performance. The primary contributions of this paper are summarized as follows.

- We propose an ultralightweight and secure user AKA framework for the IoD environment, called USAF-IoD. The proposed USAF-IoD utilizes an authenticated encryption primitive known as ASCON [11], cryptographic hash, and XOR operations in conjunction with a physical unclonable function (PUF). The PUF feature enables resistance to physical tampering attacks. USAF-IoD validates the authenticity of the user and the accessed remote drone, subsequently creating a confidential session key to facilitate secure communication.
- The proposed USAF-IoD is validated via formal security analysis using the widely accepted random-or-real (ROR) model to ensure session key security. Furthermore, informal security analysis demonstrates that our USAF-IoD effectively withstands numerous potential security attacks.
- Extensive comparative analysis highlights the effectiveness of the proposed USAF-IoD, notably excelling in terms of security and functionality characteristics when compared to existing benchmark schemes, and demonstrating competitive performance in computation, communication, and energy overheads.

The rest of the paper is organized as follows. Section II reviews the related works. The relevant background, including network basics, design objectives and threat models, as well as the essential preliminaries are discussed in Section III. The design of the proposed USAF-IoD is detailed in Section IV. A comprehensive security analysis is provided in Section V, and the comparative analysis is discussed in Section VI. The paper is concluded in Section VII.

II. Related Work

Various AKA schemes have been proposed recently to ensure effective and secure services in IoD environments. Nassi *et al.* [12] defined six key elements in the ecosystem of a conventional drone before suggesting a procedure to assess potential assaults and countermeasures. In [13], Wazid *et al.* introduced a lightweight user AKA scheme, highlighting that within the IoD environment, a user can access drone data directly only if they possess the appropriate authorization. The scheme of [13] only employs bitwise XOR operations, hash functions, and a fuzzy extractor. It cannot resist privileged-insider and impersonation attacks and also does not render the untraceability feature. In [14], Srinivas *et al.* introduced TCALAS, a lightweight three-factor anonymous user AKA scheme based on temporal credentials, designed specifically for IoD environments. Nevertheless, TCALAS does not ensure the untraceablity feature and also is not secured against impersonation attacks based on stolen verifiers, as shown in [15]. To address the limitations of TCALAS, Ali *et al.* [15] devised an improved version, which however is still vulnerable to server spoofing, forgery, and session key disclosure attacks.

Y. Ever presented a secure AKA framework for mobile sinks in IoD environments based on bilinear pairing in [16]. Nevertheless, the scheme [16] is vulnerable to impersonation and drone physical capture attacks and does not ensure perfect forward secrecy, as shown in [17]. Furthermore, the scheme [16] is inefficient, in terms of communication and computation overheads, due to the utilization of bilinear pairing cryptographic operations. Thus, it cannot ensure real-time services in resource-constrained IoD environments. Ali *et al.* [21] indicated that the scheme proposed in [17] is also vulnerable to impersonation, insider, and replay attacks and does not ensure mutual authentication among the participants. Similarly, the scheme proposed in [22] imposes high computation overheads due to bilinear pairing operations. Moreover, it is not resilient against impersonation attacks.

Wang *et al.* [23] introduced an ultra-fast authentication protocol utilizing extended chaos mapping for electric vehicle charging, targeting the challenge of slow authentication between vehicles and the grid, and claimed that their proposed protocol withstands various potential attacks. However, Chen *et al.* [24] identified vulnerabilities in their design, providing evidence that an attacker could easily acquire the session key.

Akram *et al.* [18] devised a drone-access protocol aimed at enhancing urban security monitoring. Nonetheless, subsequent research by the authors cited in [19] revealed vulnerabilities in their protocol, encompass a susceptibility to drone capture attacks and the risk of stolen-verifier attacks. Additionally, the protocol falls short in providing perfect forward secrecy.

Tanveer *et al.* [25] devised a user AKA scheme for the IoD environment. This scheme employs hash functions, au-

TABLE I: Analysis of existing user AKA schemes tailored for the IoD

Scheme	Adopted Cryptographic Operations	Limitations/ Drawbacks
Wazid *et al.* 2019 [13]	Hash functions and fuzzy extractor	• Cannot resist stolen verifier, impersonation, and session key leakage attacks • Does not render untraceability feature
Srinivas *et al.* 2019 [14]	Hash functions and fuzzy extractor	• Cannot resist impersonation attacks based on stolen verifiers • Does not render untraceability feature
Ali *et al.* 2020 [15]	Biometric fuzzy extractor, symmetric encryption/decryption, and hash functions	• Vulnerable to server spoofing, forgery, and session key disclosure attacks
Ever 2020 [16]	Bilinear pairing, ECC, and hash functions	• Vulnerable to impersonation and drone physical capture attacks • Computational overhead is high
Tanveer *et al.* 2022 [25]	ECC, hash functions, and AEAD	• Cannot resist drone physical capture and impersonation attacks • Lacks session key verification trait
Bera *et al.* 2020 [27]	ECC and hash functions	• Vulnerable to impersonation, replay, MitM attacks • Computational overhead is high • Does not provide user anonymity
Chaudhry *et al.* 2021 [28]	hash functions and ECC	• Vulnerable to impersonation and ESL attacks • Does not provide untraceability feature
Yu *et al.* 2022 [29]	PUF, fuzzy extractor and hash functions	• Does not render untraceability feature

ECC: elliptic-curve cryptography; AEAD: authenticated encryption with associative data; ESL: ephemeral secret leakage; MitM: man-in-the-middle.

thenticated encryption with associative data (AEAD), and elliptic-curve cryptography (ECC). However, the scheme [25] cannot withstand drone physical capture and impersonation attacks and lacks the session key verification trait. The scheme proposed in [26] also cannot withstand impersonation and drone physical capture attacks. Subsequently, Bera *et al.* [27] designed a blockchain-based access control scheme for the IoDs environment. However, Chaudhry *et al.* [28] pointed out that the scheme [27] is vulnerable to impersonation, replay, and man-in-the-middle (MitM) attacks and also does not provide user anonymity. Chaudhry *et al.* [28] then suggested a certificate-based access control scheme for IoD setups to fix the security vulnerabilities in the scheme [27]. Unfortunately, the scheme [28] is still vulnerable to ephemeral secret leakage (ESL) and impersonation attacks and does not support the untraceability trait. A lightweight user AKA scheme was proposed by Yu *et al.* in [29], which reveals that a user in the IoD-based smart city environment can directly access data from a drone if the user is authorized to do so. The scheme [29] employs bitwise XOR operations, PUF, hash functions, and a fuzzy extractor. However, it fails to ensure user untraceability.

Table I summarizes the state-of-the-art user authentication schemes in the IoD environment, including cryptographic operations employed and their limitations.

Against this background, we devise an ultralightweight and secure AKA framework using ASCON and hash function alongside PUF for IoD environment to resolve the security and efficiency shortcomings of existing AKA schemes.

III. Background

This section provides a concise overview of the relevant background, encompassing network and threat models, design goals, and essential prerequisites. Table II summarizes the notations utilized in this paper.

A. Network Model

The architecture of the IoD-based network is illustrated in Fig. 1. In this IoD-based network scenario, numerous drones are placed in various geographic zones, which can transmit the data that they have collected to a server or control center. Consider a usage example where an external user EU_i, such as an ambulance, wishes to know the traffic situation in a specific city section. EU_i can acquire these details from the drones that are deployed in that geographic zone. EU_i is also linked to the server via the Internet. To access real-time information, a secure remote user authentication process is necessary when an external user EU_i wants to connect with and access a drone DR_j. With the assistance of the server, authentication between EU_i and DR_j takes place. Following a successful mutual authentication process, EU_i and DR_j negotiate a session key

TABLE II: Notations and abbreviations summary

Symbol	Description
AD	Associative data
CT, PT	Cipher text and plain text
ΔT	Message time delay limit
ID_i, PID_i	EU_i's unique identity and pseudo-identity
EU_i, MT_i	The ith external user and the mobile terminal of EU_i
$E(\cdot)/D(\cdot)$	Function for ASCON encryption/decryption
ID_{D_j}, PID_{D_j}	jth drone's unique identity and pseudo-identity
IV	Initialization vector
K	Encryption/decryption key
PW_i	Password associated with EU_i
$PUF(\cdot)$	Physical unclonable function
(C, R)	Challenge-response pair for $PUF(\cdot)$
rn, N	A 128-bit random number and a nonce
S	Server (trusted authority)
MAC	Message authentication code
SK	Session key
T_i	Timestamp
$\|$, $\oplus$, $h(\cdot)$	Concatenation, XOR, and hash-function
$\mathcal{A}$	Adversary

and begin secure encrypted communication using this key.

B. Threat Model

We follow the extensively adopted 'Dolev-Yao' (DY) threat model [30] in the design of the proposed USAF-IoD to describe the adversary capabilities and the pertinent perils to the IoD environment. In the DY model, communication between any two participants occurs through vulnerable open channels, and the adversary $\mathcal{A}$ possesses the capability to intercept, breach, delay, replay, alter, or erase the complete message or segments of the message. Furthermore, as drones may be deployed in a hostile environment and cannot be monitored $24/7$, it is possible that a drone is physically captured from the deployed zone. Then by employing power analysis (PA) attacks [31], the adversary $\mathcal{A}$ can extract the secret credential stored in the captured drone, which can be used to breach the IoD system's security. Similarly, when $\mathcal{A}$ gets hold the stolen or lost mobile terminal of an external user, it can use the PA attack attempting to compromise the security of the system, in order to stolen the user's secret credentials, i.e., the identity of the external user, password, and biometrics. If $\mathcal{A}$ successfully extracts these secret credentials, it can perform numerous potential security attacks, such as MitM, impersonation, and privileged-insider attacks on the system.

In addition to the capabilities of $\mathcal{A}$ under the DY model, we also consider 'Canetti and Krawczyk' (CK) adversary model [32], which is a widely-recognized *de facto* model. According to the CK-adversary model, $\mathcal{A}$ can compromise ephemeral information such as secret keys, session keys, and other session states. Therefore, it is crucial that even if the secret keys, session keys, and other session states are compromised in a specific session, this comprised information does not compromise the secrecy of other participants' secret credentials during communication. Hence, a user authentication protocol should be built under the CK-adversary model to maintain both backward and forward secrecy.

In addition, the server is considered to be a trusted entity in the IoD environment. It can be protected physically from $\mathcal{A}$ using a locking system, similar to the scenario illustrated in [33]. Therefore, it is assumed that $\mathcal{A}$ cannot compromise the server in the IoD environment.

C. Design Objectives

The design goals of the proposed USAF-IoD are as follows:

1) **Mutual authentication**: Both the accessed remote drone and the external user must authenticate each other, ensuring the credibility and trustworthiness of the involved entities.
2) **Session key agreement**: Following a successful mutual authentication, the external user and the accessed remote drone establish a confidential session key to secure all subsequent communications.
3) **Physical security**: The design must ensure the physical security of IoDs. If a drone is captured or a mobile device is stolen or lost, the security measures implemented can prevent the adversary from extracting the secret credentials stored in the memory of these devices.
4) **Forward security**: The confidentiality of the previous secret session key must remain intact, even in the event of a compromise of the current secret session key.
5) **Anonymity**: The real identity of the communicating entities, including the server, drones, and external users, must be protected.
6) **Untraceability**: The authenticated key agreement messages communicated among the involved entities, i.e., external users, server, and drones, must not be traceable by the adversary.
7) **Un-linkability**: The design must guarantee that multiple communications originating from the same source cannot be correlated. This means that the adversary is prevented from collecting sensitive parameters from various interactions of the same entity by ensuring that there is no correlation between the various interactions of the same entity.
8) **Robustness against potential attacks**: The network must exhibit the capacity to withstand various types of attacks, including replay, modification, MitM, and impersonation, to guarantee the security of the IoD environment.

D. Preliminary Knowledge

1) ASCON

The widely recognized AEAD symmetric cipher ASCON guarantees both authenticity and data confidentiality without the need for message authentication codes. ASCON is specifically designed for resource-constrained devices, offering online (encryption and decryption), inverse-free operations, and nonce-based encryption in a single pass.

The encryption function E within ASCON operates with a set of input parameters. A shared secret key K, associated data AD, a nonce N, and variable-length plaintext PT constitute these parameters. This process results in the generation of ciphertext CT, which matches the length of PT. Additionally, the function produces a message authentication code MAC, offering authentication for both the associated data AD and the plaintext PT. In the context of ASCON and authentication protocol design, the term 'AD' refers to supplementary information provided to the authentication algorithm. This information serves to provide additional context, ensuring the integrity of both the data and this associated context during authentication. This feature enhances the security and flexibility of the authentication protocol by allowing the verification of extra information alongside the core data:

$$(CT, MAC) = E_K(N, AD, PT). \tag{1}$$

The decryption function D accepts K, N, AD, CT, and MAC as input. It yields the plaintext PT if the MAC is successfully verified, or it triggers an error represented as $\perp$ if the MAC verification fails:

$$(PT \text{ or } \perp) = D_K(N, AD, CT, MAC). \tag{2}$$

2) Physical Unclonable Function

A PUF is an integrated circuit that accepts an input and produces an outcome based on its unique physical properties. Because during the manufacturing process, a small physical variation is placed on each integrated circuit, PUF can generally be considered an electronic identity, comparable to biometrics like hand geometry, iris, and palm prints.

A PUF generates a challenge-specific outcome, known as the response when an input query known as the challenge is passed into the device. The relationship between the challenge and the response pair (CRP) is as follows

$$PUF(C) \to R, \tag{3}$$

where C and R denote the function PUF's challenge and response, respectively. A PUF emanates two appealing qualities. Firstly, it is possible to reproduce confidential information via publicly accessible data. Secondly, an intrinsic ability to resist tampering can protect against various physical assaults.

Given an identical challenge, a PUF's response in a noisy setting may vary slightly. In other words, PUF is not by default noise-resistant, which could result in the inaccessibility of sensitive data, for instance, cryptographic keys, for crucial operations. Recent research [34] has looked into many noise-resistant and stable PUF designs that can achieve nearly 0% bit error rate, even in challenging conditions characterized by voltage fluctuations and high-temperature ranges. Therefore, in this paper we assume that drones and mobile terminals are equipped with an ideal and noise-resistant PUF.

IV. The Proposed USAF-IoD

This section introduces the envisioned USAF-IoD framework. Our approach relies on a preloaded key mechanism and integrates the secure hash technique (SHA-256) alongside symmetric authenticated encryption. Additionally, every entity in the IoD environment is time-synchronized. USAF-IoD consists of seven phases, and we now detail these seven phases.

A. Initialization Phase

In this process, the server or trusted authority, denoted as S, integrates the system's PUF obtained from a trusted source. The PUF's public parameters, including the challenge-response format, output size, and error correction mechanism, are published. Furthermore, S publishes the hash function $h(\cdot)$. In addition to this, S selects a secret master key K_S and a unique identity ID_S. It then computes the pseudo-identity PID_S as follows: $X_S = h(ID_S \parallel K_S)$ and $PID_S = X_S^a \oplus X_S^b$, where X_S^a and X_S^b represent two equal portions of X_S, each 128 bits in size. S securely stores these parameters in its secure database, which is considered impervious to be compromised by $\mathscr{A}$.

B. Pre-Deployment Phase

In this phase, a drone is registered in a specific flying zone before deployment. Server S is responsible for registering each drone in the IoD setting. To achieve this, S performs the following essential steps.

Step DR-1: S chooses a distinct identity ID_{D_j} and a random number rn_{D_j}. Next, S uses its own secret key K_S to compute $X_j = h(ID_{D_j} \parallel K_S)$ and $PID_{D_j} = X_j^a \oplus X_j^b$, where X_j^a and X_j^b are the two equal portions of X_j each of 128 bits in size. Next, S securely transmits $\{ID_{D_j}, PID_{D_j}, rn_{D_j}\}$ to drone D_j via a private channel.

Step DR-2: Drone D_j obtains the parameters $\{ID_{D_j}, PID_{D_j}, rn_{D_j}\}$ from S. D_j picks a challenge parameter C_{D_j} and computes the corresponding response parameter R_{D_j} as $R_{D_j} = PUF(C_{D_j})$. Next, D_j picks a secret key K_{D_j} and computes $Y_j = h(ID_{D_j} \parallel R_{D_j})$, $K_j = Y_j^a \oplus Y_j^b$, $PT_{D_j} = (PID_{D_j} \parallel K_{D_j})$, and $(CT_{D_j}, MAC_{D_j}) = E_{K_j}\big(rn_{D_j}, rn_{D_j}, PT_{D_j}\big)$, where Y_j^a and Y_j^b are the two equal portions of Y_j each of 128 bits in size. D_j stores the credentials $\{ID_{D_j}, C_{D_j}, rn_{D_j}, CT_{D_j}, MAC_{D_j}, PUF(\cdot)\}$ in its memory and forwards the parameter K_{D_j} to S via a private channel.

Step DR-3: After acquiring the parameter K_{D_j} from D_j, S computes $(CT_j, MAC_j) = E_{K_S}(rn_{D_j}, rn_{D_j}, PT_j)$, where $PT_j = K_{D_j}$ and keeps the parameters $\{PID_{D_j}, rn_{D_j}, CT_j, MAC_j\}$ in its database.

C. User Registration Phase

Before being able to acquire real-time data from a particular drone D_j, external user EU_i must register with server S through the user registration (UR) procedure. S provides secret credentials and a list of drones from which EU_i can obtain real-time information. S follows the steps below to complete the UR procedure.

Step UR-1: EU_i picks an identity ID_i and securely transmits the registration request message $< ID_i >$ to S. Upon obtaining the registration request, S selects a secret key K_i and a random number rn_i and computes EU_i's pseudo-identity PID_i as $Z_i = h(ID_i \parallel K_S)$ and $PID_i = Z_i^a \oplus Z_i^b$, where K_S is the secret key of S and Z_i^a and Z_i^b are the two equal parts of Z_i each of 128 bits. Further, S picks a random number rn_i and a secret key K_i and then set the associative data $AD_i = rn_i$, nonce $N_i = rn_i$, and plaintext $PT_i = (PID_{D_j} \parallel K_i)$. Moreover, S computes $(CT_i,\ MAC_i) = E_{K_S}(N_i,\ AD_i,\ PT_i)$ by utilizing ASCON encryption function. S then constructs the registration response message $<PID_i,\ PID_S,\ PID_{D_j},\ K_i,\ rn_i>$ and dispatches it to EU_i securely.

Step UR-2: EU_i, after obtaining $<PID_i,\ PID_S,\ PID_{D_j},\ K_i,\ rn_i>$, picks a challenge parameter C_i and computes the response R_i of PUF as $R_i = PUF(C_i)$. EU_i chooses a password PW_i and computes $Z = h(R_i \parallel PW_i \parallel ID_i)$

External user $\boldsymbol{EU_i}$ / Mobile terminal $\boldsymbol{MT_i}$	**Server $\boldsymbol{S}$**
Pick identity ID_i. $\xrightarrow{<ID_i>}$ (Via secure channel)	Pick $rn_i,\ K_i,\ K_S$. Compute $Z_i = h(ID_i \parallel K_S)$, $PID_i = Z_i^a \oplus Z_i^b$, $PT_i = (PID_{D_j} \parallel K_i)$, $N_i = rn_i,\ AD_i = rn_i$, $(CT_i,\ MAC_i) = E_{K_S}(N_i,\ AD_i,\ PT_i)$. $\xleftarrow{<PID_i,\ PID_S,\ PID_{D_j},\ K_i,\ rn_i>}$ (Via secure channel)
Pick PW_i, C_i. Compute $R_i = PUF(C_i)$, $Z = h(R_i \parallel PW_i \parallel ID_i)$, $K = (Z^a \oplus Z^b)$, $N = rn_i,\ AD = rn_i$, $PT = (PID_i \parallel PID_S \parallel PID_{D_j} \parallel K_i)$, $(CT,\ MAC) = E_K(N,\ AD,\ PT)$. Store $\{CT,\ MAC,\ rn_i,\ C_i,\ PUF(\cdot)\}$ in MT_i.	Store $\{ID_i,\ PID_i,\ CT_i,\ MAC_i,\ rn_i\}$ in S.

Fig. 2: Illustration of the user registration procedure.

and $K = (Z^a \oplus Z^b)$, where Z^a and Z^b are the two equal parts of Z each of 128 bits. Moreover, EU_i set the associative data $AD = rn_i$, nonce $N = rn_i$, and plaintext $PT = (PID_i \parallel PID_S \parallel PID_{D_j} \parallel K_i)$ and computes $(CT,\ MAC) = E_K(N,\ AD,\ PT)$ by utilizing ASCON encryption function.

Step UR-3: Finally, EU_i stores $\{CT,\ MAC,\ rn_i,\ C_i,\ PUF(\cdot)\}$ in MT_i. Furthermore, S also keeps the credentials $\{ID_i,\ PID_i,\ CT_i,\ MAC_i,\ rn_i\}$ in its database.

Fig. 2 summarizes the user registration procedure.

D. User Login Phase

To complete the login procedure, EU_i must perform the following steps:

Step LG-1: EU_i inputs their identity ID_i and password PW_i^l into the user interface provided on MT_i.

Step LG-2: MT_i then retrieves C_i, rn_i, CT, MAC from its memory and computes $R_i = PUF(C_i)$, $X_1 = h(R_i \parallel PW_i^l \parallel ID_i)$, and $K^l = X_1^a \oplus X_1^b$. Next, MT_i sets the associative data $AD = rn_i$ and nonce $N = rn_i$ and computes $(PT' \text{ or } \perp) = D_{K^l}(N, AD, CT, MAC)$ using ASCON decryption function. If the verification of the MAC fails, it triggers an error message, and the login procedure is terminated instantly. Otherwise it retrieves the plaintext $PT' = \{PID_i \parallel PID_S \parallel PID_{D_j} \parallel K_i\}$.

E. Authenticated Key Agreement Phase

In the user login phase, EU_i successfully logins in by submitting its secret credentials to MT_i. Following this local authentication, MT_i transmits the AKA authentication request message to S for additional EU_i / MT_i validation. To ensure future secure communication, S will aid EU_i and D_j in setting up a secret session key. In order to complete this phase, the following steps are crucial.

Step AKA-1: MT_i chooses two random numbers, n_1 and n_2, each of size 128 bits, and current timestamp T_1 of 32 bits. Next, MT_i computes $X_2 = (PID_i \parallel n_2) \oplus h(PID_S \parallel T_1)$. It further sets the associative data, nonce, and plaintext as $AD_1 = rn_i$, $N_1 = rn_i \oplus n_2$ and $PT_1 = (PID_{D_j} \parallel n_1)$, respectively, and computes $(CT_1,\ MAC_1) = E_{K_i}(N_1,\ AD_1, PT_1)$ using ASCON encryption function. MT_i then transmits an authentication request message $M_1 =< X_2,\ CT_1,\ MAC_1,\ T_1 >$ to S via insecure channel.

Step AKA-2: After obtaining the message M_1 at time T_1' from EU_i, S first verifies the freshness of T_1 by checking the condition $|T_1 - T_1'| \overset{?}{\leq} \Delta T$. If verified, S proceeds to compute $(PID_i \parallel n_2) = X_2 \oplus h(PID_S \parallel T_1)$, and fetches ID_i, CT_i, MAC_i, and rn_i related to PID_i. Moreover, S computes $(PT_i' \text{ or } \perp) = D_{K_S}(rn_i, rn_i, CT_i, MAC_i)$ using ASCON decryption function. If the verification of MAC_i fails, it triggers an error message; else it retrieves the plaintext $PT_i' = \{PID_{D_j} \parallel K_i\}$.

Step AKA-3: After retrieving the parameters PID_{D_j} and K_i, S sets the associated data $AD_2 = rn_i$ and nonce $N_2 = rn_i \oplus n_2$. Next, S computes $(PT_1' \text{ or } \perp) = D_{K_i}(N_2, AD_2, CT_1, MAC_1)$ using ASCON decryption function. If the verification of MAC_1 fails, it triggers an error message; else it retrieves the plaintext $PT_1' = \{PID_{D_j} \parallel n_1\}$.

Step AKA-4: After retrieving the parameters PID_{D_j} and n_1, S verifies the presence of PID_{D_j} within the authorized drone list for EU_i. If validated, it proceeds to retrieve the parameters rn_{D_j}, CT_j, and MAC_j corresponding to PID_{D_j}. Subsequently, S computes $(PT_j' \text{ or } \perp) = D_{K_S}(rn_{D_j}, rn_{D_j}, CT_j,\ MAC_j)$ using the ASCON decryption function. In the event of a failed MAC_j verification, an error message is triggered; otherwise, S retrieves the plaintext $PT_j' = K_{D_j}$.

Step AKA-5: Next, S generates current timestamp T_2 and sets $N_3 = rn_{D_j} \oplus T_2$, $AD_3 = rn_{D_j}$, and $PT_2 = (PID_i \parallel n_1 \parallel n_2)$. S further computes $(CT_2, MAC_2) = E_{K_{D_j}}(N_3, AD_3, PT_2)$ using ASCON encryption function. S then transmits message $M_2 =< CT_2, MAC_2, T_1, T_2 >$ to D_j via insecure channel.

Step AKA-6: After obtaining the message M_2 at time T_2' from S, D_j first verifies condition $|T_2 - T_2'| \overset{?}{\leq} \Delta T$. If so, D_j retrieves the parameters $C_{D_j}, rn_{D_j}, CT_{D_j}, MAC_{D_j}$ and ID_{D_j} from its memory. Next, D_j computes $R_{D_j} = PUF(C_{D_j})$, $X_3 = h(R_{D_j} \parallel ID_{D_j})$ and $K_j = X_3^a \oplus X_3^b$. D_j further computes $(PT_{D_j}' \text{ or } \perp) = D_{K_j}(rn_{D_j}, rn_{D_j}, CT_{D_j}, MAC_{D_j})$ using ASCON decryption function. If the verification of MAC_{D_j} fails, it triggers an error message; else it retrieves the plaintext $PT_{D_j}' = \{PID_{D_j} \parallel K_{D_j}\}$.

Step AKA-7: After retrieving the parameters PID_{D_j} and K_{D_j}, D_j sets $N_4 = rn_{D_j} \oplus T_2$ and $AD_4 = rn_{D_j}$, and then computes $(PT_2' \text{ or } \perp) = D_{K_{D_j}}(N_4, AD_4, CT_2, MAC_2)$, using ASCON decryption function. If the verification of MAC_2 fails, it triggers an error message; otherwise it retrieves the plaintext $PT_2' = (PID_i \parallel n_1 \parallel n_2)$.

Step AKA-8: Next, D_j generates current timestamp T_3 and random number n_3, and computes $X_4 = h(n_1 \parallel n_3 \parallel T_1 \parallel T_3)$, $K_1 = X_4^a \oplus X_4^b$, $X_5 = (K_1 \parallel n_3) \oplus h(PID_i \parallel n_1 \parallel T_1)$ and $PT_3 = h(PID_{D_j} \parallel PID_i \parallel n_1 \parallel n_2 \parallel n_3 \parallel (T_1 \oplus T_2 \oplus T_3))$. Furthermore, D_j computes $(CT_3, MAC_3) = E_{K_1}(n_2, n_3, PT_3)$ using ASCON encryption function, and stores $SK_{D_j U_i} = CT_3$ as session key. Finally, D_j transmits message $M_3 =< X_5, MAC_3, T_2, T_3 >$ to EU_i via insecure channel.

Step AKA-9: After obtaining the message M_3 at time T_3'

from D_j, EU_i first verifies condition $|T_3 - T'_3| \overset{?}{\leq} \Delta T$. If verified, EU_i computes $(K_1 \parallel n_3) = X_5 \oplus h(PID_i \parallel n_1 \parallel T_1)$, $PT'_3 = h(PID_{D_j} \parallel PID_i \parallel n_1 \parallel n_2 \parallel n_3 \parallel (T_1 \oplus T_2 \oplus T_3))$ and $(CT'_3,\ MAC'_3) = E_{K_1}(n_2, n_3, PT'_3)$. EU_i then checks $MAC_3 \overset{?}{=} MAC'_3$. If it holds, it stores $SK_{U_iD_j} = CT'_3$ as a session key.

Both EU_i and D_j store the same session key $SK_{U_iD_j}(= SK_{D_jU_i})$ for their future secure communication.

Fig. 3 summarizes the login and AKA phases of the proposed USAF-IoD.

F. Password Reset Phase

When required, a valid EU_i can alter its password for security reason anytime locally without involving the server. EU_i completes the following steps to reset the password.

Step PRP-1: EU_i first enters its identity ID_i and old password PW_i^{old} to the mobile terminal MT_i.

Step PRP-2: MT_i then retrieves C_i, rn_i, CT, MAC from its memory and computes $R_i = PUF(C_i)$, $X_1 = h(ID_i \parallel$

External user EU_i / Mobile terminal MT_i	**Server**	**Drone D_j**
$\{CT,\ MAC,\ rn_i,\ C_i,\ PUF(\cdot)\}$	$\{ID_i,\ PID_i,\ CT_i,\ MAC_i,\ rn_i\}$, $\{PID_{D_j},\ rn_{D_j},\ CT_j,\ MAC_j\}$	$\{ID_{D_j},\ C_{D_j},\ rn_{D_j},\ PUF(\cdot),\ CT_{D_j},\ MAC_{D_j}\}$
LG-1: Inputs $ID_i,\ PW_i^l$. **LG-2**: Retrieve C_i, rn_i, CT, MAC. Compute $R_i = PUF(C_i)$, $X_1 = h(R_i \parallel PW_i^l \parallel ID_i)$, $K^l = X_1^a \oplus X_1^b$, $N = rn_i,\ AD = rn_i$, $(PT'$ or $\perp) = D_{K^l}(N, AD, CT, MAC)$. If MAC verification fails: $\perp$, Else: $PT' = \{PID_i \parallel PID_S \parallel PID_{D_j} \parallel K_i\}$.		
AKA-1: Pick $n_1,\ n_2,\ T_1$. Compute $X_2 = (PID_i \parallel n_2) \oplus h(PID_S \parallel T_1)$, $N_1 = rn_i \oplus n_2,\ AD_1 = rn_i$, $PT_1 = (PID_{D_j} \parallel n_1)$, $(CT_1, MAC_1) = E_{K_i}(N_1,\ AD_1, PT_1)$. $\xrightarrow{M_1 = <X_2,\ CT_1,\ MAC_1,\ T_1>}$ (Via insecure open channel)	**AKA-2**: Check $|T_1 - T'_1| \overset{?}{\leq} \Delta T$ If not, abort. Compute $(PID_i \parallel n_2) = X_2 \oplus h(PID_S \parallel T_1)$. Check if PID_i exists in the database. If so, retrieve the parameters $ID_i,\ CT_i,\ MAC_i,\ rn_i$ corresponding to PID_i and compute $(PT'_i$ or $\perp) = D_{K_S}(rn_i,\ rn_i,\ CT_i,\ MAC_i)$. If verification of MAC_i fails: error $\perp$, Else: retrieve $PT'_i = \{PID_{D_j} \parallel K_i\}$.	
	AKA-3: Compute $N_2 = rn_i \oplus n_2,\ AD_2 = rn_i$, $(PT'_1$ or $\perp) = D_{K_i}(N_2, AD_2, CT_1, MAC_1)$. If verification of MAC_1 fails: error $\perp$, Else: retrieve $PT'_1 = \{PID_{D_j} \parallel n_1\}$. **AKA-4**: Check if PID_{D_j} is within the authorized drone list for EU_i. If so, retrieve the parameters rn_{D_j}, CT_j, and MAC_j corresponding to PID_{D_j} and compute $(PT'_j$ or $\perp) = D_{K_S}(rn_{D_j}, rn_{D_j}, CT_j, MAC_j)$, If verification of MAC_j fails: error $\perp$, Else: retrieve $PT'_j = \{K_{D_j}\}$. **AKA-5**: Pick T_2. Compute $N_3 = rn_{D_j} \oplus T_2,\ AD_3 = rn_{D_j}$, $PT_2 = (PID_i \parallel n_1 \parallel n_2)$, $(CT_2, MAC_2) = E_{K_{D_j}}(N_3, AD_3, PT_2)$. $\xrightarrow{M_2 = <CT_2,\ MAC_2,\ T_1,\ T_2>}$ (Via insecure open channel)	**AKA-6**: Check $|T_2 - T'_2| \overset{?}{\leq} \Delta T$ If not, abort. Retrieve $C_{D_j},\ rn_{D_j},\ ID_{D_j}$. Compute $R_{D_j} = PUF(C_{D_j})$, $X_3 = h(R_{D_j} \parallel ID_{D_j})$, $K_j = X_3^a \oplus X_3^b$, $(PT'_{D_j}$ or $\perp) = D_{K_j}(rn_{D_j}, rn_{D_j}, CT_{D_j}, MAC_{D_j})$, If verification of MAC_{D_j} fails: error $\perp$, Else: retrieves $PT'_{D_j} = \{PID_{D_j} \parallel\ K_{D_j}\}$. **AKA-7**: Compute $N_4 = rn_{D_j} \oplus T_2,\ AD_4 = rn_{D_j}$, $(PT'_2$ or $\perp) = D_{K_{D_j}}(N_4,\ AD_4,\ CT_2,\ MAC_2)$, If verification of MAC_2 fails: error $\perp$, Else: retrieves $PT'_2 = (PID_i \parallel n_1 \parallel n_2)$. **AKA-8**: Pick $T_3,\ n_3$. Compute $X_4 = h(n_1 \parallel n_3 \parallel T_1 \parallel T_3)$, $K_1 = X_4^a \oplus X_4^b$, $X_5 = (K_1 \parallel n_3) \oplus h(PID_i \parallel n_1 \parallel T_1)$, $PT_3 = h(PID_{D_j} \parallel PID_i \parallel n_1 \parallel n_2 \parallel n_3 \parallel (T_1 \oplus T_2 \oplus T_3))$, $(CT_3,\ MAC_3) = E_{K_1}(n_2,\ n_3,\ PT_3)$. Store $SK_{D_jU_i} = CT_3$. $\xleftarrow{M_3 = <X_5,\ MAC_3,\ T_2,\ T_3>}$ (to EU_i Via insecure open channel)
AKA-9: Check $|T_3 - T'_3| \overset{?}{\leq} \Delta T$ If not, abort. Compute $(K_1 \parallel n_3) = X_5 \oplus h(PID_i \parallel n_1 \parallel T_1)$, $PT'_3 = h(PID_{D_j} \parallel PID_i \parallel n_1 \parallel n_2 \parallel n_3 \parallel (T_1 \oplus T_2 \oplus T_3))$, $(CT'_3,\ MAC'_3) = E_{K_1}(n_2,\ n_3,\ PT'_3)$. Check $MAC_3 \overset{?}{=} MAC'_3$ If not, abort. Store $SK_{U_iD_j} = CT'_3$.		
EU_i and D_j store the session key $SK_{U_iD_j} = (SK_{D_jU_i})$ for future secure communication.		

Fig. 3: Login and authenticated key agreement protocol.

$PW_i^{old} \parallel R_i$) and $K^l = X_1^a \oplus X_1^b$. Next, MT_i sets the associative data $AD = rn_i$ and nonce $N = rn_i$, and computes $(PT'$ or $\perp) = D_{K^l}(N, AD, CT, MAC)$ using ASCON decryption function. If the verification of MAC fails, it triggers an error message and terminates the password change procedure instantly; else it prompts for new password PW_i^{new}.

Step PRP-3: EU_i enters new password PW_i^{new} in MT_i, after confirming the password change request.

Step PRP-4: MT_i computes $R_i = PUF(C_i)$, $Z^{new} = h(R_i \parallel PW_i^{new} \parallel ID_i)$ and $K^{new} = (Z^{new^a} \oplus Z^{new^b})$, and sets $AD = rn_i$, $N = rn_i$ and plaintext $PT = (PID_i \parallel PID_S \parallel PID_{D_j} \parallel K_i)$. Next, MT_i computes $(CT^{new}, MAC^{new}) = E_K^{new}(N, AD, PT)$ by utilizing ASCON encryption function. Finally, MT_i updates the parameters $\{CT^{new}, MAC^{new}, rn_i, C_i, PUF(\cdot)\}$ in the memory.

G. *Revocation Phase*

In the event that a legitimate EU_i loses their mobile terminal MT_i, the server S has the capability to issue and register a new mobile terminal MT_i^{new} for EU_i. To get new MT_i^{new}, EU_i requires to recall its old identity ID_i, and S performs the following steps to issue a new mobile terminal to EU_i.

Step RVP-1 : EU_i selects its old identity ID_i and forwards it to S. S calculates EU_i's pseudo-identity PID_i as $Z_i = h(ID_i \parallel K_S)$ and $PID_i = Z_i^a \oplus Z_i^b$, where K_S is the secret key of S. Furthermore, S searches PID_i in its database. If a matching record is found, S deletes the record associated with PID_i and forwards a new registration request message to EU_i.

Step RVP-2:

Upon receiving the new registration message from S, EU_i selects a fresh and unique identity ID_i^{new} and securely sends the registration request message $< ID_i^{new} >$ to S. The rest procedure is the same as described in Subsection IV-C.

Step RVP-3: EU_i stores $\{CT^{new}, MAC^{new}, rn_i^{new}, C_i^{new}, PUF(\cdot)\}$ in MT_i^{new}. S keeps the credentials $\{ID_i^{new}, PID_i^{new}, CT_i^{new}, MAC_i^{new}, rn_i^{new}\}$ in its database.

V. Security Analysis

This section first performs an informal security analysis of USAF-IoD to demonstrate its resilience to numerous potential security attacks. Subsequently, we utilize the ROR model to conduct a rigorous, formal security evaluation, specifically focusing on the session key's security.

A. *Informal Security Analysis*

Within this subsection, we illustrate the resilience of USAF-IoD by evaluating its resistance to the following significant potential security threats.

1) *Replay Attack*

In our USAF-IoD, the communicated messages $M_1 =< X_2, CT_1, MAC_1, T_1 >$, $M_2 =< CT_2, MAC_2, T_1, T_2 >$, and $M_3 =< X_5, MAC_3, T_2, T_3 >$ utilize fresh timestamps T_1, T_2, and T_3. After receiving M_1, S checks $|T_1 - T_1'| \stackrel{?}{\leq} \Delta T$. If not, it aborts. Similarly, D_j and EU_i check the conditions $|T_2 - T_2'| \stackrel{?}{\leq} \Delta T$ and $|T_3 - T_3'| \stackrel{?}{\leq} \Delta T$ to assure the freshness of M_2 and M_3, respectively. The recipient of the message considers the obtained message to be authentic if it is received within the time delay threshold. Otherwise the AKA process is aborted. Therefore, USAF-IoD is robust against replay attacks.

2) *MitM Attack*

During the AKA procedure, $\mathscr{A}$ can capture and forge the transmitted messages M_1, M_2 and M_3. MitM attack constructs the forged messages and attempts to fool other entities to believe that the message forged by $\mathscr{A}$ is valid. For instance, consider that $\mathscr{A}$ tries to forge $M_1 =< X_2, CT_1, MAC_1, T_1 >$. However, without knowing the secret credentials, such as $PID_i, PID_S, rn_i, PID_{D_j}$ and K_i, it is difficult for $\mathscr{A}$ to produce a valid M_1. Likewise, forging M_2 and M_3 is also difficult for $\mathscr{A}$. Thus $\mathscr{A}$ cannot launch MitM attacks successfully, and therefore our USAF-IoD is resistant to MitM attacks.

3) *Impersonation Attack*

According to the analysis of Subsection V-A2, $\mathscr{A}$ is unable to produce a legitimate AKA request message, $M_1 =< X_2, CT_1, MAC_1, T_1 >$, on behalf of EU_i, without being aware of the secrete parameters $PID_i, PID_S, rn_i, PID_{D_j}$ and K_i to carry out user impersonation attack. Likewise, $\mathscr{A}$ cannot launch a server impersonation attack without knowing the secret credentials K_{D_j}, rn_{D_j} and PID_i. Similarly, $\mathscr{A}$ cannot construct the response message $M_3 =< X_5, MAC_3, T_2, T_3 >$ to carry out a drone impersonation attack without knowing the secret credentials PID_i, n_1, n_2 and n_3. The proposed USAF-IoD is hence resistant to users, server and drones impersonation attacks.

4) *Captured Drone Attack*

Drones may be deployed in a hostile environment and cannot be monitored 24/7. Therefore, it is possible that a drone is physically captured from its deployed zone. Suppose that $\mathscr{A}$ has successfully seized a legitimate drone D_j that is currently deployed in a flying zone. $\mathscr{A}$ may try to retrieve the secret data kept in D_j's memory, such as $ID_{D_j}, rn_{D_j}, CT_{D_j}$ and MAC_{D_j}, using PA attacks. To retrieve the embedded CRP (C_{D_j}, R_{D_j}) in the PUF of D_j, however, $\mathscr{A}$ has to probe or modify the integrated circuit of the captured drone D_j. But this effort will permanently alter the small physical changes in the integrated circuit and destroys the PUF. Therefore, even if $\mathscr{A}$ can obtain $ID_{D_j}, rn_{D_j}, CT_{D_j}$ and MAC_{D_j} successfully, it cannot recover the valid CRP (C_{D_j}, R_{D_j}). Hence, USAF-IoD is immune and resilient to captured drone attacks.

5) *Stolen Mobile Device Attack*

Assume that adversary $\mathscr{A}$ has obtained the stolen or lost mobile terminal MT_i of legitimate external user EU_i. $\mathscr{A}$ can extract the data $\{CT, MAC, rn_i, C_i, PUF(\cdot)\}$ stored in MT_i's memory utilizing PA attacks. After retrieving this information, $\mathscr{A}$ tries to extract the encrypted secret credentials, such as PID_i, PID_S, PID_{D_j} and K_i. To obtains these parameters, $\mathscr{A}$ needs to guess ID_i and PW_i accurately. As a result, $\mathscr{A}$'s ability to accurately anticipate both ID_i and PW_i is computationally near impossible. Furthermore, retrieving the embedded CRP (C_i, R_i) from the PUF is impossible for $\mathscr{A}$, as discussed in Subsection V-A4. Therefore, the proposed USAF-IoD is safe from attacks using stolen mobile devices.

6) Anonymity and Untraceability

Anonymity and untraceability are essential characteristics of an AKA scheme. Three messages, i.e., $M_1 =< X_2, CT_1, MAC_1, T_1 >$, $M_2 =< CT_2, MAC_2, T_1, T_2 >$ and $M_3 =< X_5, MAC_3, T_2, T_3 >$, are exchanged in order to complete the AKA procedure. It is difficult for $\mathscr{A}$ to obtain the real identities of the external user, server, and drone in the AKA procedure by seizing these exchanged messages. Furthermore, fresh timestamps and random numbers are employed in these exchanged messages. Consequently, the exchanged messages are distinct and random in every session. Therefore, $\mathscr{A}$ cannot correlate the captured messages of two different AKA sessions. Thus, USAF-IoD provides both anonymity and untraceability features.

7) ESL Attack

The session key, which is established between EU_i and D_j in our USAF-IoD, is calculated as $PT_3' = h(PID_{D_j} \parallel PID_i \parallel n_1 \parallel n_2 \parallel n_3 \parallel (T_1 \oplus T_2 \oplus T_3))$ and $(CT_3', MAC_3') = E_{K_1}(n_2, n_3, PT_3')$. $SK_{U_iD_j}(= SK_{D_jU_i}) = CT_3'$ if and only if MAC_3 is verified as discussed in Subsection IV-E. $SK_{U_iD_j}(= SK_{D_jU_i})$ is composed utilizing both long-term secrets (LTS), PID_i and PID_j, as well as short-term secrets (STS), $n_1, n_2, n_3, T_1, T_2, T_3$ and K_1. $\mathscr{A}$ cannot compromise the session key $SK_{U_iD_j}(= SK_{D_jU_i})$ without knowing LTS even if $\mathscr{A}$ compromises the STS using the session hijacking attacks. Similarly, even if $\mathscr{A}$ compromises LTS, it cannot access the session key $SK_{U_iD_j}(= SK_{D_jU_i})$ without first knowing STS. Therefore, $\mathscr{A}$ must know both LTS and STS to compromise the session key's security, which is computationally near impossible. Thus, the proposed USAF-IoD is resilient against ESL attacks.

8) Privileged-Insider/Offline Password Guessing Attacks

Assume that adversary $\mathscr{A}$, who is a privileged insider user, e.g., a user inside the control room, is aware of the registration information ID_i sent by EU_i to S during the user registration phase. After the registration process is completed, suppose that $\mathscr{A}$ has stolen the registered user EU_i's mobile terminal MT_i. Utilizing the PA attacks, $\mathscr{A}$ can then retrieve the crucial data, i.e., CT, MAC and rn_i, stored in MT_i's memory. However, retrieving the embedded CRP (C_i, R_i) from the PUF is impossible for $\mathscr{A}$. Moreover, guessing a correct high-entropy password is relatively hard. Therefore, the proposed USAF-IoD is resilient to attacks from privileged-insiders and offline password guessing.

9) DoS Attack

If a legitimate external user EU_i enters an incorrect ID_i and/or PW_i^l during the login or password update phases of the proposed USAF-IoD, it is locally verified by evaluating MAC using ASCON decryption function (Step **LG-2** of Subsection IV-D) and the local verification will fail. Only after successful local verification, EU_i can send the AKA request message to server S. Likewise, the password update only happens when the old password is successfully verified during the password update phase. Therefore, our USAF-IoD is resilient against denial-of-service (DoS) attacks.

TABLE III: Queries and their descriptions

Query	Description
$Send(\mho^t,\, msg)$	This query allows $\mathscr{A}$ to send a message msg to $\mho^t$ and receive the response.
$Execute(\mho_{EU}^{t_1},\, \mho_S^{t_2},\, \mho_D^{t_3})$	This query simulates an eavesdropping attack on messages exchanged among participants via an insecure open channel.
$CorruptMT(\mho_{EU}^{t_1})$	This query enables $\mathscr{A}$ to access secret parameters from the compromised user terminal $\mho_{EU}^{t_1}$.
$CorruptD(\mho_D^{t_2})$	By employing this query, $\mathscr{A}$ can obtain secret parameters from the captured drone $\mho_D^{t_2}$.
$Reveal(\mho^t)$	Based on this query, $\mathscr{A}$ reveals the current SK generated by its partner to the $\mathscr{A}$.
$Test(\mho^t)$	In this query, $\mathscr{A}$ asks $\mho^t$ for the SK, and $\mho^t$ responds probabilistically with an unbiased coin flip outcome c.

B. Formal Security Analysis Via ROR Model

We employ the ROR model to evaluate the security of the SK in the proposed USAF-IoD against both active and passive adversary $\mathscr{A}$, as detailed in Theorem 1. Prior to establishing the SK security for USAF-IoD, we present a concise overview of the ROR concepts. In the proposed USAF-IoD framework, there are three primary participants: the external user $\mho_{EU}^{t_1}$, the server $\mho_S^{t_2}$, and the drone $\mho_D^{t_3}$. Here, $\mho_{EU}^{t_1}$, $\mho_S^{t_2}$, and $\mho_D^{t_3}$ represent instances corresponding to the t_1^{th} external user (EU_i), the t_2^{th} server (S), and the t_3^{th} drone (D_j), respectively. To facilitate our formal security analysis, we utilize Table III, which provides various queries, such as '$Reveal()$', '$Send()$', '$Execute()$', '$Corrupt()$', and '$Test()$'. In addition to these queries, we make use of the 'PUF function $PUF()$' and a 'collision-resistant one-way hash function $Hash$' as random oracles in our analysis.

Definition 1. *Assuming that $\mathscr{A}$ operates within a polynomial-time frame of t_p and sends at most QR queries to an encryption/decryption oracle with a length of L_{ED}, the advantage in the context of the 'online chosen ciphertext attack' (OCCA3) by $\mathscr{A}$ can be expressed as follows:*

$$Adv_{\phi,\mathscr{A}}^{OCCA3}(QR, L_{ED}, t_p) \leq Adv_{\phi}^{OPRP-CPA}(QR, L_{ED}, t_p) + Adv_{\phi}^{INT-CT}(QR, L_{ED}, t_p), \quad (4)$$

where $Adv_{\phi}^{OPRP-CPA}(QR, L_{ED}, t_p)$ denotes $\mathscr{A}$'s advantage in the 'online pseudo-random permutation chosen-plaintext' attack, and $Adv_{\phi}^{INT-CT}(QR, L_{ED}, t_p)$ represents $\mathscr{A}$'s advantage in ensuring the integrity of the ciphertext.

Theorem 1. *Let $\mathscr{A}$ be an adversary operating against USAF-IoD in polynomial time t_p, and $Adv_{\mathscr{A}}^{USAF-IoD}(t_p)$ signify its advantage in obtaining the session key created between external user EU_i and drone D_j during the AKA phase to break the semantic security of USAF-IoD in time t_p. Then*

$$Adv_{\mathscr{A}}^{USAF-IoD}(t_p) \leq \frac{Q_h^2}{|Hash|} + \frac{Q_p^2}{|PUF|} + \frac{2 \cdot Q_s}{|Dict|} + 2 \cdot Adv_{ASCON,\mathscr{A}}^{OCCA3}(QR, L_{ED}, t_p), \quad (5)$$

*where $Q_s, Q_h, Q_p, Dict, Hash$ and PUF denote send queries, hash queries, PUF queries, password dictionary, output range of Hash and key length of PUF, respectively, while $Adv_{ASCON,\mathscr{A}}^{OCCA3}(QR,\, L_{ED}, t_p)$ signifies advantage of $\mathscr{A}$ in breaching the security of an online AEAD scheme (ASCON) (**Definition** 1).*

Proof. $\mathcal{A}$ plays a series of six games $\{Game_i : 0 \leq i \leq 5\}$ to breach the SK security. Let SUC_i signify the success probability in which $\mathcal{A}$ wins game $Game_i$ in t_p. The specifics of each game are outlined below.

$Game_0$: This game simulates a real attack by $\mathcal{A}$ against USAF-IoD. The decision is made by flipping an unbiased coin and, therefore, we have

$$\text{Adv}_{\mathcal{A}}^{USAF-IoD}(t_p) = |2 \cdot \text{Prob}[SUC_0] - 1|. \tag{6}$$

$Game_1$: This game represents an eavesdropping attack against USAF-IoD in which $\mathcal{A}$ eavesdrops on the transmitted messages among EU_i, S and D_j during the AKA phase. Then $\mathcal{A}$ runs $Execute(\mho_{EU}^{t_1}, \mho_{S}^{t_2}, \mho_{D}^{t_3})$ query, followed by $Test$ and $Reveal$ queries to confirm the validity of $SK_{U_iD_j} = SK_{D_jU_i}$. It's important to note that the SK between EU_i and D_j is calculated as $PT_3' = h(PID_{D_j} \parallel PID_i \parallel n_1 \parallel n_2 \parallel n_3 \parallel (T_1 \oplus T_2 \oplus T_3))$ and $(CT_3', MAC_3') = E_{K_1}(n_2, n_3, PT_3')$, and $SK_{U_iD_j}(= SK_{D_jU_i}) = CT_3'$ if and only if MAC_3 is verified as discussed in Subsection IV-E. Since $SK_{U_iD_j}(= SK_{D_jU_i})$ is composed utilizing both LTS and STS, computing the SK is computationally very difficult for $\mathcal{A}$, and the probability of winning $Game_1$ remains the same as in $Game_0$. Thus, the indistinguishability of $Game_0$ and $Game_1$ renders

$$\text{Prob}[SUC_1] = \text{Prob}[SUC_0]. \tag{7}$$

$Game_2$: By simulating $Hash$ and $Send$ queries, $\mathcal{A}$ attempts to launch an active attack in this game. $\mathcal{A}$ employs multiple $Hash$ queries to detect SHA-256 collisions. Considering that the transmitted messages contain timestamps and random numbers, the probability of a collision happening during the execution of the $Send$ query is extremely low. Consequently, $\mathcal{A}$'s attempt to retrieve the secret parameters becomes unfeasible. Therefore, employing the birthday paradox, we obtain

$$|\text{Prob}[SUC_2] - \text{Prob}[SUC_1]| \leq \frac{Q_h^2}{2|Hash|}. \tag{8}$$

$Game_3$: This game is an extension of $Game_2$ that simulates PUF query $PUF()$. It is worth noting that the PUF in D_j and MT_i are secure, and hence

$$|\text{Prob}[SUC_3] - \text{Prob}[SUC_2]| \leq \frac{Q_p^2}{2|PUF|}. \tag{9}$$

$Game_4$: This game mimics stolen/lost MT and password guessing attacks. By utilizing $CorruptMT(\mho_{EU}^{t_1})$ query, $\mathcal{A}$ obtains $\{CT, MAC, rn_i, C_i, PUF(\cdot)\}$ from a stolen/lost MT_i. Subsequently, $\mathcal{A}$ attempts to extract the encrypted secret credentials, including PID_i, PID_S, PID_{D_j} and K_i. $\mathcal{A}$ must correctly determine both ID_i and PW_i within a limited number of guesses from $Dict$ in order to win this game and, therefore,

$$|\text{Prob}[SUC_4] - \text{Prob}[SUC_3]| \leq \frac{Q_s}{|Dict|}. \tag{10}$$

$Game_5$: Finally, in $Game_5$, $\mathcal{A}$ initiates an active attack by intercepting transmitted messages, including $M_1 =< X_2, CT_1, MAC_1, T_1 >$, $M_2 =< CT_2, MAC_2, T_1, T_2 >$ and $M_3 =< X_5, MAC_3, T_2, T_3 >$. $\mathcal{A}$ seeks to obtain the secret credentials necessary for constructing the SK after capturing these messages. However, the secret credentials are encrypted using ASCON, rendering $\mathcal{A}$ incapable of decrypting the secured data. Therefore, according to **Definition** 1, we can conclude the following

$$|\text{Prob}[SUC_5] - \text{Prob}[SUC_4]| \leq \text{Adv}_{ASCON,\mathcal{A}}^{OCCA3}(QR, L_{ED}, t_p). \tag{11}$$

TABLE IV: Approximated execution time for various primitives (in milliseconds) [35], [36]

↓Primitive/ Device →	User terminal/ Drone	Server
T_a: ASCON	0.370	0.0351
T_{ea}: ECC point addition	0.124	0.006
T_{em}: ECC point multiplication	2.850	0.780
$T_{fe} \approx T_{em}$: Fuzzy extractor	2.850	0.780
T_{puf}: $PUF(\cdot)$	0.4 μs	-
T_h: Hash function	0.345	0.039
T_{se}/T_{sd}: Symmetric encryption/decryption	0.391	0.02

After finishing all the games, $\mathcal{A}$ performs a $Test$ query. Additionally, the semantic security of SK is decided by flipping a fair coin, and as a result

$$\text{Prob}[SUC_5] = \frac{1}{2}. \tag{12}$$

Thus, from (6), we have

$$\frac{1}{2}\text{Adv}_{\mathcal{A}}^{USAF-IoD}(t_p) = \left|\text{Prob}[SUC_0] - \frac{1}{2}\right|. \tag{13}$$

Using (12) and (13) as well as noting (7), we obtain

$$\begin{aligned} \frac{1}{2}\text{Adv}_{\mathcal{A}}^{USAF-IoD}(t_p) &= |\text{Prob}[SUC_0] - \text{Prob}[SUC_5]| \\ &= |\text{Prob}[SUC_1] - \text{Prob}[SUC_5]|. \end{aligned} \tag{14}$$

Applying the well-known triangular inequality to (14) yields

$$\begin{aligned} \frac{1}{2}\text{Adv}_{\mathcal{A}}^{USAF-IoD}(t_p) \leq\ & |\text{Prob}[SUC_1] - \text{Prob}[SUC_2]| \\ &+ |\text{Prob}[SUC_2] - \text{Prob}[SUC_3]| \\ &+ |\text{Prob}[SUC_3] - \text{Prob}[SUC_4]| \\ &+ |\text{Prob}[SUC_4] - \text{Prob}[SUC_5]|. \end{aligned} \tag{15}$$

Substituting (8), (9), (10) and (11) into (15) leads to

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{USAF-IoD}(t_p) \leq\ & \frac{Q_h^2}{|Hash|} + \frac{Q_p^2}{|PUF|} + \frac{2 \cdot Q_s}{|Dict|} \\ &+ 2 \cdot \text{Adv}_{ASCON,\mathcal{A}}^{OCCA3}(QR,\ L_{ED},\ t_p), \end{aligned} \tag{16}$$

namely, (5). This completes the proof. ■

VI. Comparative Analysis

This section provides a comparative analysis of the proposed USAF-IoD and the existing state-of-the-art schemes, Wazid *et al.* [13], Srinivas *et al.* [14], Ali *et al.* [15], Nikooghadam *et al.* [17], Akram *et al.* [18], Tanveer *et al.* [25], and Yu *et al.* [29], in terms of computation overheads, communication overheads, energy overheads, and security and functionality features. For the sake of fairness, the comparison will not include any pairing-based AKA scheme because the computation overhead introduced by bilinear-pairing is significantly greater than other cryptographic primitives.

A. Computation Overhead Comparison

The computational overheads of the proposed USAF-IoD and other state-of-the-art benchmark schemes are calculated using experimental results presented in [35], [36]. Table IV provides execution times for various cryptographic opera-

TABLE V: Comparative analysis: computational overheads

Scheme	User	Server	Drone	TE (ms)
Wazid *et al.* [13]	$T_{fe} + 16T_h (\approx 8.890)$	$8T_h (\approx 0.312)$	$7T_h (\approx 2.415)$	$T_{fe} + 31T_h (\approx 11.617)$
Srinivas *et al.* [14]	$T_{fe} + 14T_h (\approx 7.83)$	$9T_h (\approx 0.351)$	$7T_h (\approx 2.415)$	$T_{fe} + 30T_h (\approx 10.596)$
Ali *et al.* [15]	$T_{fe} + 10T_h (\approx 6.3)$	$3T_{se}/T_{sd} + 7T_h (\approx 0.333)$	$7T_h (\approx 2.415)$	$T_{fe} + 3T_{se}/T_{sd} + 24T_h \approx 9.048$
Nikooghadam *et al.* [17]	$6T_h + 2T_{em} (\approx 3.27)$	$8T_h (\approx 0.312)$	$5T_h + 2T_{em} (\approx 3.615)$	$19T_h + 4T_{em} (\approx 7.197)$
Akram *et al.* [18]	$9T_h (\approx 3.105)$	$7T_h + 2T_{se} (\approx 0.313)$	$7T_h (\approx 2.415)$	$23T_h + 2T_{se} (\approx 5.833)$
Tanveer *et al.* [25]	$T_{fe} + 6T_h + 3T_{em} + 3T_a (\approx 14.58)$	$2T_h + T_{em} + 3T_a (\approx 0.9633)$	$3T_h + 2T_{em} + 2T_a (\approx 7.475)$	$T_{fe} + 11T_h + 6T_{em} + 8T_a (\approx 23.0183)$
Yu *et al.* [29]	$T_{puf} + T_{fe} + 12T_h (\approx 6.99)$	$9T_h (\approx 0.351)$	$T_{puf} + T_{fe} + 8T_h (\approx 5.6104)$	$2T_{puf} + 2T_{fe} + 29T_h \approx 12.9514$
Proposed USAF-IoD	$T_{puf} + 4T_h + 3T_a (\approx 2.4904)$	$T_h + 4T_a (\approx 0.1794)$	$T_{puf} + 4T_h + 3T_a (\approx 2.4904)$	$2T_{puf} + 9T_h + 10T_a (\approx 5.1602)$

TE (ms): Estimated overall execution time in milliseconds.

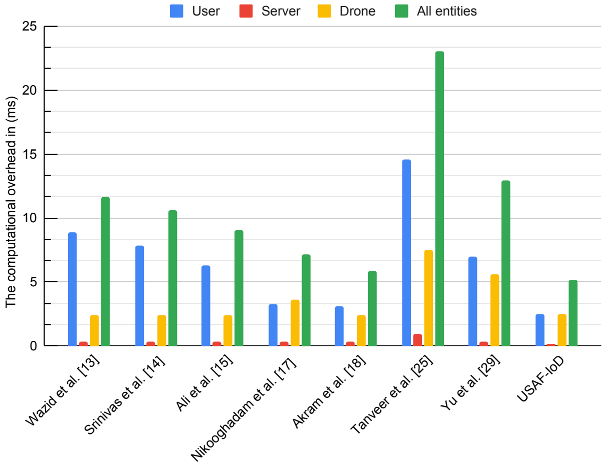

Fig. 4: Comparison of computation overhead.

TABLE VI: Comparative analysis: communication overheads

Scheme	No. of transmitted messages	TO (bits)
Wazid *et al.* [13]	3	1696
Srinivas *et al.* [14]	3	1536
Ali *et al.* [15]	3	1696
Nikooghadam *et al.* [17]	3	2336
Akram *et al.* [18]	3	2176
Tanveer *et al.* [25]	3	1856
Yu *et al.* [29]	4	2048
Proposed USAF-IoD	3	1696

TO (bits): Total communication overhead (bits).

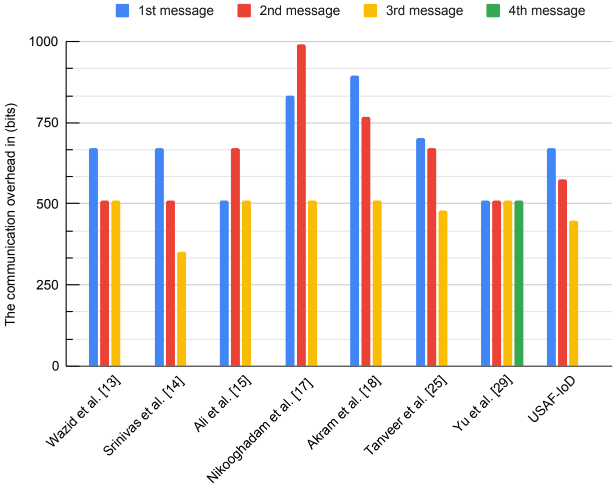

Fig. 5: Comparison of communication overhead.

tions on different platforms. Specifically, we represent the time requirements for the following operations as follows: ASCON encryption/decryption as T_a, ECC point multiplication as T_{em}, ECC point addition as T_{ea}, fuzzy extractor as T_{fe}, $PUF(\cdot)$ as T_{puf}, hash function as T_h, and symmetric encryption/decryption function as T_{se}/T_{sd}. We also assume that ASCON and AEGIS AEAD primitives require the same time for their executions. Furthermore, user terminal/drone is considered as a resource-restricted device, utilizing the setting: Raspberry PI-3 (R-PI3), Ubuntu 16.04 LTS, OS 64- bits, 1.2 GHz Quad-core processor, and RAM 1 GiB. Conversely, the server is considered as a resource-rich device, utilizing the setting: Intel® Core $^{\text{TM}}$ i7-6700 CPU@3.4GHz; RAM@8 GiB; Ubuntu 16.04 LTS, and OS 64-bit. Based on Table IV, we have computed the computational overheads of the seven schemes and compared the results obtained in Fig. 4 and Table V. Our proposed USAF-IoD scheme stands out for its superior computational efficiency during the AKA phase. It exhibits the lowest computational overhead when compared to other benchmark AKA schemes. This underscores the efficiency and effectiveness of our proposed approach in minimizing computational overheads, a crucial consideration in the resource-constrained environment of IoD systems. However, it's worth noting that our proposed USAF-IoD incurs a slightly higher computational overhead at the drone side when compared to specific schemes, namely Wazid *et al.* [13], Srinivas *et al.* [14], Ali *et al.* [15], Nikooghadam *et al.* [17], and Akram *et al.* [18]. This marginal increase in computational overhead is attributed to the enhanced security and advanced functionality features that our scheme offers. As depicted in Table VIII, these features contribute to a more robust and secure IoD system, ultimately justifying the minimal additional computational load.

B. Communication Overhead Comparison

We next compare the communication overheads of the proposed USAF-IoD and the other five benchmark schemes during the login and AKA phases. It is assumed that all identities, timestamps, random nonces (numbers), ECC points, and hash values require 128 bits, 32 bits, 128 bits, 320 bits and 256 bits, respectively. Furthermore, the ASCON/AEGIS key and authentication parameters are 128 bits each. During the login and AKA phases, USAF-IoD transmits three messages, $M_1 =<$

TABLE VII: Comparative analysis: energy overheads

Scheme	**Computational Energy (mJ)**	**Transmission Energy (mJ)**	**Total Energy (mJ)**
Wazid *et al.* [13]	$(0.021 * 2688)/1000 = 0.0564$	$(0.66 * 512 + 0.29 * 512)/1000 = 0.486$	0.5424
Srinivas *et al.* [14]	$(0.021 * 3712)/1000 = 0.0780$	$(0.66 * 512 + 0.29 * 352)/1000 = 0.440$	0.5180
Ali *et al.* [15]	$(0.021 * 3712)/1000 = 0.0780$	$(0.66 * 512 + 0.29 * 672)/1000 = 0.533$	0.6110
Nikooghadam *et al.* [17]	$2 * 8.8 + (0.021 * 1952)/1000 = 17.678$	$(0.66 * 512 + 0.29 * 992)/1000 = 0.626$	18.304
Akram *et al.* [18]	$(0.021 * 4480)/1000 = 0.0941$	$(0.66 * 512 + 0.29 * 768)/1000 = 0.561$	0.6551
Tanveer *et al.* [25]	$(104 * 52)/1000 + 2 * 8.8 + (0.021 * 1696)/1000 = 23.049$	$(0.66 * 480 + 0.29 * 672)/1000 = 0.512$	23.561
Yu *et al.* [29]	$2.5 + 8.8 + (0.021 * 4160)/1000 = 11.3874$	$(0.66 * 512 + 0.29 * 512)/1000 = 0.486$	11.8734
Proposed USAF-IoD	$(23 * 104)/1000 + 2.5 + (0.021 * 1600)/1000 = 4.925$	$(0.66 * 448 + 0.29 * 576)/1000 = 0.462$	5.388

$X_2, CT_1, MAC_1, T_1 >$, $M_2 =< CT_2, MAC_2, T_1, T_2 >$ and $M_3 =< X_5, MAC_3, T_2, T_3 >$ with lengths M_1: $256 + 256 + 128 + 32 = 672$ bits, M_2: $384 + 128 + 32 + 32 = 576$ bits, and M_3: $256 + 128 + 32 + 32 = 448$ bits, respectively. The cumulative communication overhead of USAF-IoD while executing the AKA procedure is therefore $672 + 576 + 448 = 1696$ bits. The AKA schemes proposed by Wazid *et al.* [13], Srinivas *et al.* [14], Ali *et al.* [15], Nikooghadam *et al.* [17], Akram *et al.* [18], Tanveer *et al.* [25], and Yu *etal.* [29] are examined in terms of their communication overheads. These schemes require 1696 bits, 1536 bits, 1696 bits, 2336 bits, 2176 bits, 1856 bits, and 2048 bits, respectively, for the transmission of messages during the login and AKA phases. Table VI and Fig. 5 provide a comparative analysis of communication overheads for the USAF-IoD and the seven schemes. Notably, the scheme proposed by Srinivas *et al.* [14] imposes the lowest communication overhead. On the other hand, our USAF-IoD, along with the schemes proposed by Wazid *et al.* [13] and Ali *et al.* [15], exhibits an equivalent minimal communication overhead. This marginal increase in communication overhead or equal overhead can be attributed to the enhanced security and advanced functionality features that our scheme offers. As depicted in Table VIII, these features contribute to a more robust and secure IoD system, ultimately justifying the minimal additional communication load.

TABLE VIII: Comparative analysis: security and functionality features

Scheme	$\mathcal{SF}_1$	$\mathcal{SF}_2$	$\mathcal{SF}_3$	$\mathcal{SF}_4$	$\mathcal{SF}_5$	$\mathcal{SF}_6$	$\mathcal{SF}_7$	$\mathcal{SF}_8$	$\mathcal{SF}_9$	$\mathcal{SF}_{10}$	$\mathcal{SF}_{11}$
Wazid *et al.* [13]	✓	✓	×	✓	✓	✓	×	✓	×	✓	×
Srinivas *et al.* [14]	✓	✓	×	✓	✓	✓	×	✓	✓	✓	✓
Ali *et al.* [15]	✓	✓	×	✓	✓	✓	✓	✓	×	✓	✓
Nikooghadam *et al.* [17]	×	✓	×	✓	✓	✓	✓	✓	×	✓	✓
Akram *et al.* [18]	✓	✓	✓	×	✓	✓	✓	×	✓	✓	✓
Tanveer *et al.* [25]	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓
Yu *et al.* [29]	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
Proposed USAF-IoD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

C. Energy Overhead Comparison

We conducted an analysis of the energy consumption of UAVs, taking into account their potential energy constraints. There are two primary contributors to energy consumption: data transmission and cryptographic operations. According to the scheme in [4], transmitting and receiving one bit of data consumes 0.66 μJ and 0.29 μJ, respectively. Additionally, as stated in [4], the microprocessor incurs an energy cost of 8.8 mJ for multiplication operations, 0.021 μJ per bit for hash operations, and 2.5 mJ for PUF operations. For ASCON, as reported in [20], the energy cost is 23 μJ/byte. Using this data, we calculated the overall computational and transmission energy. The energy overhead comparison results are presented in Table VII.

In our proposed USAF-IoD, the UAV consumes 5.388 mJ of energy in the AKA phase. However, it is noteworthy that Nikooghadam *et al.* [17], Tanveer *et al.* [25], and Yu *et al.* [29] scheme stands out with the highest energy consumption among all schemes. On the other hand, schemes proposed by Wazid *et al.* [13], Srinivas *et al.* [14], Ali *et al.* [15], and Akram *et al.* [18] are energy-efficient designs. Nevertheless, it is important to note that the higher energy consumption of our proposed USAF-IoD is well justified, as it provides enhanced security and additional functionality features as depicted in Table VIII.

D. Security and Functionality Features Comparison

Table VIII compares the proposed USAF-IoD and the seven existing state-of-the-art schemes: Wazid *et al.* [13], Srinivas *et al.* [14], Ali *et al.* [15], Nikooghadam *et al.* [17], Akram *et al.* [18], Tanveer *et al.* [25], and Yu *etal.* [29]. This comparison is based on a set of eleven security and functionality features: $\mathcal{SF}_1$: replay attack; $\mathcal{SF}_2$: MitM attack; $\mathcal{SF}_3$: impersonation attacks; $\mathcal{SF}_4$: captured drone attack; $\mathcal{SF}_5$: stolen mobile device attack; $\mathcal{SF}_6$: anonymity preservation; $\mathcal{SF}_7$: untraceability preservation; $\mathcal{SF}_8$: ESL attack; $\mathcal{SF}_9$: privileged-insider attack; $\mathcal{SF}_{10}$: offline password guessing attack; and $\mathcal{SF}_{11}$: DoS attack.

In Table VIII, '✓' denotes the fulfillment of a specific functionality feature or withstanding against a specific security attack, whereas '×' indicates the non-provision of some functionality feature or insecurity against some attack. Table VIII shows that only the proposed USAF-IoD provides all the relevant and mandatory security and functionality features, whereas the benchmark schemes lack one or more functionality features or cannot resist one or more security attacks.

E. Critical Discussion

Compared to the cutting-edge schemes described in the literature, our proposed USAF-IoD is novel in four aspects. Firstly, USAF-IoD employs lightweight cryptographic primitives, including ASCON, hash functions, and XOR operations alongside PUF, to perform mutual authentication and create a secure session key between external user and drone in the IoD environment. This approach not only enhances security but also introduces additional functionality. Consequently, it offers a low-energy-consumption security solution that extends the operational lifespan of resource-limited drones. Secondly, adversary may physically capture a drone or steal a smart device in the IoD setting and employs PA attacks to retrieve secret credentials kept in the captured or stolen devices' memory. Many state-of-the-art schemes fail to protect against such physical attacks. In order to protect the data kept in drones or smart devices, USAF-IoD utilizes PUF as a tamper-resistant module to defend against assaults that are both physical tempering and software-based attacks. Thirdly, USAF-IoD provides conditional privacy-preserving so that only trusted server can reveal the genuine identity of drones and external users. USAF-IoD creates a different pseudonym for drone and external user for each session. Finally, many contemporary IoD security schemes consider insufficiently many security and functionality characteristics, and most significantly, they have certain intrinsic susceptibilities. By contrast, the security of USAF-IoD has been carefully assessed via informal and formal security analysis, which proves that USAF-IoD is a secure scheme for the IoD environment.

To summarize, various authentication schemes have been designed for IoD and similar resource-limited environments in recent years. However, a PUF and ASCON (AEAD scheme) based scheme for mutual authentication and key agreement that protects user privacy has yet to receive much attention. In our envisaged IoD environment, the participating entities, such as mobile terminals and drones, must be PUF-enabled in order to deploy the proposed USAF-IoD scheme.

VII. Conclusions

In this paper, we have proposed USAF-IoD, an ultra-lightweight and secure authenticated key agreement framework for the Internet of Drones environment based on ASCON, cryptographic hash functions, and XOR operations alongside PUF to resist physical tempering attacks. A comprehensive security analysis, encompassing both informal and formal techniques, demonstrated the resilience of USAF-IoD against various potential security threats. With its ultra-lightweight and highly reliable design, USAF-IoD surpassed several benchmark schemes in terms of communication, computation, and energy consumption overheads, all while preserving essential security and functionality features. In contrast, some competing schemes lacked one or more of these critical attributes, highlighting the exceptional performance of our proposed solution. This made USAF-IoD a robust choice. Furthermore, the ultralightweight nature and robustness of our proposed design made it suitable for deployment in various resource-constrained environments beyond the Internet of Drones.

Acknowledgment

This work is partly supported by the Beijing Natural Science Foundation (No. 4212015), China Ministry of Education - China Mobile Scientific Research Foundation (No. MCM20200102).

References

[1] J. A. Besada, *et al.*, "Drones-as-a-service: A management architecture to provide mission planning, resource brokerage and operation support for fleets of drones," in *Proc. PerCom 2019 Workshops* (Kyoto, Japan), Mar. 11-15, 2019, pp. 931–936.

[2] Y. Yang, *et al.*, "AoI optimization for UAV-aided MEC networks under channel access attacks: A game theoretic viewpoint," in *Proc. ICC 2022* (Seoul, Republic of Korea), May 16-20, 2022, pp. 1-6.

[3] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[4] X. Ren, *et al.*, "A novel access and handover authentication scheme in UAV-Aided satellite-terrestrial integration networks enabling 5G," *IEEE Trans. Netw. Serv. Manag.,* early access, pp. 1–20, Feb. 2023.

[5] S. A. Khowaja, *et al.*, "A secure data sharing scheme in community segmented vehicular social networks for 6G," *IEEE Trans. Ind. Inform.,* vol. 19, no. 1, pp. 890–899, Jan. 2023.

[6] W. Yang, *et al.*, "A review on security issues and solutions of the Internet of drones," *IEEE Open J. Computer Society*, vol. 3, pp. 96–110, Jun. 2022.

[7] C. Lin, *et al.*, "Security and privacy for the Internet of drones: Challenges and solutions," *IEEE Communi. Mag.*, vol. 56, no. 1, pp. 64–69, Jan. 2018.

[8] A. Badshah, *et al.*, "AAKE-BIVT: Anonymous authenticated key exchange scheme for blockchain-enabled Internet of vehicles in smart transportation," *IEEE Trans. Intell. Transp. Syst.*, early access, pp. 1–17, Nov. 2022. DOI:10.1109/TITS.2022.3220624.

[9] A. Badshah, *et al.*, "LAKE-BSG: Lightweight authenticated key exchange scheme for blockchain-enabled smart grids," *Sustain. Energy Technol. Assess.*, vol. 52, Art. no. 102248, pp. 1–13, Aug. 2022.

[10] M. Yahuza, *et al.*, "Internet of drones security and privacy issues: Taxonomy and open challenges," *IEEE Access*, vol. 9, pp. 57243–57270, Apr. 2021.

[11] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1.2. Submission to NIST," Sep. 27, 2019. https://ascon.iaik.tugraz.at, accessed: 2022-12-20.

[12] B. Nassi, *et al.*, "SoK: Security and privacy in the age of commercial drones," in *Proc. 2021 IEEE Symp. Security Privacy* (San Francisco, CA, USA), May 24-27, 2021, pp. 1434–1451.

[13] M. Wazid, *et al.*, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[14] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[15] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. AI-Turjman, "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–43724, Mar. 2020.

[16] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the Internet of Drones applications," *Computer Communi.*, vol. 155, pp. 143–149, Mar. 2020.

[17] M. Nikooghadam, H. Amintoosi, S. K. H. Islam, and M. F. Moghadam, "A provably secure and lightweight authentication scheme for Internet of Drones for smart city surveillance," *J. Syst. Architecture*, vol. 115, Article ID 101955, pp. 1–16, May 2021.

[18] M. W. Akram, *et al.*, "A secure and lightweight Drones-access protocol for smart city surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19634–19643, Oct. 2022.

[19] S. Liu and C. -M. Chen, "Comments on "A secure and lightweight Drones-access protocol for smart city surveillance,"," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25054–25058, Dec. 2022.

[20] J. Kaur, A. C. Canto, M. M. Kermani, and R. Azarderakhsh, "A comprehensive survey on the implementations, attacks, and countermeasures of the current NIST lightweight cryptography standard," *arXiv preprint* arXiv:2304.06222, 2023.

[21] Z. Ali, *et al.*, "TC-PSLAP: Temporal credential-based provably secure and lightweight authentication protocol for IoT-enabled drone environments," *Security and Communi. Netw.*, vol. 2021, Article ID 9919460, pp. 1–10, Dec. 2021

[22] M. Nikravan and A. Reza, "A multi-factor user authentication and key agreement protocol based on bilinear pairing for the Internet of things," *Wireless Pers. Communi.*, vol. 111, no. 1, pp. 463–494, Mar. 2020.

[23] W. Wang, Z. Han, M. Alazab, T. R. Gadekallu, X. Zhou, and C. Su, "Ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps," *IEEE Trans. Ind. Appl.*, vol. 58, no. 5, pp. 5616–5623, Sept.-Oct. 2022.

[24] C. -M. Chen, Y. Hao and T. -Y. Wu, "Discussion of “Ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps”," *IEEE Trans. Ind. Appl.*, vol. 59, no. 2, pp. 2091–2092, Mar.-Apr. 2023.

[25] M. Tanveer, A. U. Khan, N. Kumar, and M. M. Hassan, “RAMP-IoD: A robust authenticated key management protocol for the Internet of drones,” *IEEE Internet of Things J.*, vol. 9, no. 2, pp. 1339–1353, Jan. 2022.

[26] M. Tanveer, *et al.*, “RUAM-IoD: A robust user authentication mechanism for the Internet of drones,” *IEEE Access,* vol. 10, pp. 19836–19851, Feb. 2022.

[27] B. Bera, D. Chattaraj, and A. K. Das, “Designing secure blockchain-based access control scheme in IoT-enabled Internet of drones deployment,” *Computer Communi.*, vol. 153, pp. 229–249, Mar. 2020.

[28] S. A. Chaudhry, *et al.*, “GCACS-IoD: A certificate-based generic access control scheme for Internet of drones,” *Computer Netw.*, vol. 191, Article ID 107999, pp. 1–11, May 2021.

[29] S. Yu, A. K. Das, Y. Park, and P. Lorenz, “SLAP-IoD: Secure and lightweight authentication protocol using physical unclonable functions for Internet of drones in smart city environments,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10374–10388, Oct. 2022.

[30] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Trans. Inf. Theory,* vol. 29, no. 2, pp. 198–208, Mar. 1983.

[31] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Examining smart-card security under the threat of power analysis attacks,” *IEEE Trans. Computers*, vol. 51, no. 5, pp. 541–552, May 2002.

[32] R. Canetti and H. Krawczyk, “Universally composable notions of key exchange and secure channels,” in *Proc. EUROCRYPT 2002* (Amsterdam, The Netherlands), Apr. 28-May 2, 2002, pp. 337–351.

[33] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, “Cloud centric authentication for wearable healthcare monitoring system,” *IEEE Trans. Dependable Secure Computing*, vol. 17, no. 5, pp. 942–956, Sep.-Oct. 2020.

[34] K.-H. Chuang, *et al.*, “A physically unclonable function using soft oxide breakdown featuring 0% native BER and 51.8 fJ/bit in 40-nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2765–2776, Oct. 2019.

[35] M. Tanveer, *et al.*, “REAS-TMIS: Resource-efficient authentication scheme for telecare medical information system,” *IEEE Access*, vol. 10, pp. 23008–23021, Feb. 2022.

[36] T. Alladi, *et al.*, “SecAuthUAV: A novel authentication scheme for UAV-ground station and UAV-UAV communication,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15068–15077, Dec. 2020.