

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

# PASKE-IoD: Privacy-Protecting Authenticated Key Establishment for Internet of Drones

MUHAMMAD TANVEER<sup>1</sup>, ABD ULLAH KHAN<sup>2,3</sup>, (MEMBER, IEEE), HABIB SHAH<sup>4</sup>, SHEHZAD ASHRAF CHAUDHRY<sup>5</sup>, AND ALAMGIR NAUSHAD<sup>3</sup>

<sup>1</sup>Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan. (e-mail: tanveer123giki@gmail.com)

<sup>2</sup>High Speed Networks Lab, National Yang-Ming Chaio-Tung University, Hsinchu, Taiwan. (e-mail:abdullah.ee10@nycu.edu.tw)

<sup>3</sup>School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan.

<sup>4</sup>Department of Computer Science, College of Computer Science, King Khalid University, Abha 62529, Saudi Arabia. e-mail: (hurrahman@kku.edu.sa).

<sup>5</sup>Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey. (e-mail: ashraf.shahzad.ch@gmail.com)

Corresponding author: ABD ULLAH KHAN (e-mail: abdullah.ee10@nycu.edu.tw, akhan.dphd17seecs@seecs.edu.pk).

**ABSTRACT** Unmanned aerial vehicles/drones are considered an essential ingredient of traffic motoring systems in smart cities. Interconnected drones, also called the Internet of Drones (IoD), gather critical data from the environmental area of interest and transmit the data to a server located at the control room for further processing. This transmission occurs via wireless communication channels, which are exposed to various security risks. Besides this, an External User (EU) occasionally demands access to real-time information stored at a specific drone rather than retrieving data from the server, which requires an efficient Authenticated Session Key Establishment (ASKE) approach to ensure a reliable communication in IoD environment. In this article, we present a Privacy-Protecting ASKE scheme for IoD (PASKE-IoD). PASKE-IoD utilizes Authenticated Encryption (AE) primitive “ASCON”, and hash function “ASCON- hash”, to accomplish the ASKE phase. PASKE-IoD checks the EU’s authenticity before allowing him to access the IoD environment resources. Moreover, PASKE-IoD enables EUs and drones to communicate securely after establishing a session key. Meticulous informal security analysis and security verification are carried out using Scyther to demonstrate that PASKE-IoD is immune to numerous covert security attacks. In addition, Burrows-Abadi-Needham logic is utilized to corroborate the logical exactitude of PASKE-IoD. A comparative analysis is presented to illustrate that PASKE-IoD is efficient and renders more security features than the eminent ASKE scheme.

**INDEX TERMS** AEAD, Internet of Drones, Privacy, Unmanned Aerial Vehicles, Key Exchange

## I. INTRODUCTION

Internet of Things (IoT) is an emerging networking paradigm that facilitates daily life routines [1]–[3]. IoT connects different real-world wearable devices, vehicles, home, and office appliances, etc. [4], [5]. Connectivity among the IoT nodes is established through a private network or the public Internet [6], [7]. Recent technological advancements have given rise to an enhanced IoT network, namely, the Internet of Drones (IoD). In IoD, drones or Unmanned Aerial Vehicles (UAV) are utilized to enhance the versatility of the existing IoT networks [8]. UAVs are easy to deploy and troubleshoot, provide a swift response, and are capable of the Omni-direction movement, making them one of the most suitable solutions to assess their surrounding environment and

gather useful information. IoD has various applications, such as public safety, smart-city traffic monitoring, 3D-mapping, search & rescue, node tracking, agricultural, cinematography, and product delivery systems, disaster recovery [8]–[10].

IoD is considered a resource-constricted environment because the drones are limited in energy resources, computational capabilities, and storage capacity [11], [12]. In IoD, drones are deployed in an unattended environment, and the drones share information with other network entities using Public Communication Channels (PCCs). A PCC is vulnerable to various security threats. Security attacks on the IoD network can degrade the performance and interrupt the streamlined operations of the IoD network. So, It is imperative to thwart unauthorized information disclosure and

prevent illegitimate External Users (EU) from accessing the network resources. Therefore, an Authenticated Session Key Establishment (ASKE) is the essential requirement of IoD to revoke unauthorized EU access to the network resources and establish a secret Session Key (SK) to achieve information confidentiality.

Plenty of ASKE schemes have been proposed for IoT and IoD environments by employing symmetric and asymmetric cryptographic primitives. However, a large share of these schemes are not protected decently and are prone to various security attacks that include but are not limited to Stolen Smart Card (STSC), Privileged Insider (PRIN), Password Guessing (PAGU), User Impersonation (UIMP), and replay attacks, as presented in [13]–[15]. Apart from this, the ASKE schemes that utilize asymmetric cryptographic mechanisms are computationally infeasible, from computational standpoint, for the resource-limited small scale IoT devices and drones. Therefore, a lightweight and efficient ASKE scheme has become a decisive concern in the resource-limited IoD environment. This paper presents an ASKE scheme by applying Lightweight Cryptography (LWC) primitive known as ASCON [16], which is an Authenticated Encryption with Associative Data (AEAD) scheme. An LWC based AE scheme renders the functionality of data encryption and authentication simultaneously. Therefore, by employing AEAD mechanism, we propose a secure and efficient ASKE scheme for the IoD environment.

### A. RESEARCH CONTRIBUTIONS

To resolve the aforementioned issues, a novel efficient ASKE scheme, namely, Privacy-Protecting ASKE-IoD (PASKE-IoD), is presented with the following contributions.

- 1) An efficient ASKE scheme for the IoD environment is proposed. The scheme utilizes LWC-based AEAD primitive named as ASCON encryption along with ASCON-Hash and Exclusive-OR functions. PASKE-IoD ensures the authenticity of an EU before allowing access to the IoD network resources. Moreover, PASKE-IoD enables an EU and drone to set up an SK to accomplish indecipherable communication.
- 2) Informal security analysis is performed, and Scyther-based formal security verification is implemented, to demonstrate that PASKE-IoD is protected against malicious attacks. In particular, PASKE-IoD is effective against replay and Man-in-the-Middle (MAMI) attacks. The logical completeness of PASKE-IoD is confirmed using BAN logic.
- 3) A comparative study shows that PASKE-IoD yields enhanced security features at minimized communication overhead and computational costs compared to the eminent ASKE schemes.

### B. THE PAPER'S ORGANIZATION

The paper is distributed into various sections as follows. A brief overview of the existing leading ASKE schemes is presented in Section II. The assumed system model for the

proposed scheme is presented in Section III. The essential preliminaries are elaborated in Section IV. The proposed scheme with all its attributes is elaborated in Section V. The informal and formal security analyses associated with the proposed scheme are provided in Section VI. An in-depth performance analysis of the proposed scheme is given in Section VII. Finally, Section VIII presents the conclusion. A list of notations employed in PASKE-IoD is reported in Table 2.

## II. THE EXISTING WORK

In this section, the eminent and related ASKE schemes designed for IoT/IoD environments are surveyed. To this end, Chao Lin *et al.* [17] presented a detailed review of IoD applications and different security challenges associated with IoD networks. Additionally, they also described a security model for the IoD environment. Wazid *et al.* [18] presented an analysis of various ASKE schemes designed for IoD networks and security imperatives in the IoD environment. Similarly, the authors in [19] devised a resource-efficient ASKE scheme for IoD. The scheme utilizes a hash function and Exclusive-OR operation during the ASKE phase. Likewise, a lightweight ASKE protocol is proposed in [20] for IoD application. The scheme employs a symmetric encryption algorithm, hash function, and Exclusive-OR operations. Islam *et al.* [21] highlighted the limitations of the scheme presented by Wu *et al.* [22] in terms of non-protection against STSC, PRIN, and PAGU attacks and non-provisioning of anonymity and revocation mechanism. Similarly, a user ASKE scheme is presented in [23], which enables the user device to communicate securely after establishing the SK. Moreover, the security strength of the devised scheme is endorsed through AVISPA.

In addition to this, Xue *et al.* [24] proposed an ASKE scheme considering multi-server scenario. However, the devised scheme is prone to UIMP attack, PRIN attack, and PAGU attack, as demonstrated in [25], and additionally does not render User Anonymity (UA) and SK security. Similarly, an ASKE mechanism is presented in [26] for the smart-grid system. However, it is demonstrated by the authors in [27] that the scheme presented in [26] is not only prone to UIMP and MAMI attacks, but also cannot ensure the integrity of the communicated message. Furthermore, a novel ASKE scheme is devised by Mohammad Ali *et al.* in [28], which cannot stand against the replay, UIMP, and MAMI attacks, and cannot safeguard against Identity Guessing (IDGU) attack [29]. Turkanovic *et al.* [30] proposed an ASKE for Wireless Sensor Network (WSN), which is lightweight and less expensive from the standpoint of computational overhead and energy consumption. However, the scheme is unsafe against MAMI, STSC, and replay attacks. Furthermore, the scheme fails to provide UA [31]. Similarly, the authors in [32] proposed an ECC-based ASKE mechanism for the IoT environment, which is exposed to different types of pernicious attacks.

Proceeding in the same fashion, the authors in [33] also considered a multi-server environment and proposed

a lightweight ASKE mechanism for protection. Moreover, the authors also demonstrated the limitations associated with the scheme presented in [34] in the form of non-resistance against forgery-attack, replay attack, UIMP attack. Moreover, it is shown in [33] that the scheme presented in [34] fails to ensure mutual authentication. Similarly, the scheme presented for the IoT environment by F. Wu *et al.* in [35] is, though, computationally efficient, yet, it does not render resistance against STSC attack, DoS attack, and UIMP attack. Likewise, the ASKE mechanism presented by Tai *et al.* in [36] for IoT environment utilizes lightweight cryptography. Nevertheless, the scheme cannot protect perfectly against PAGU attack, PRIN attack, and STSC attack, and does not render UA and traceability security features, and does not provide the SK security, as pointed out in [13]. In the same fashion, the ECC-based ASKE mechanism presented by Challa *et al.* in [37] for IoT applications is computationally impracticable for the resource-limited devices and is insecure against UIMP attacks. Furthermore, the ASKE mechanism presented by Amin *et al.* in [25] is deemed to be a lightweight and efficient ASKE scheme in particular for IoT-based cloud computing applications. The scheme, however, cannot prevent UIMP and PRIN attacks. Similarly, the scheme presented by Wazid *et al.* in [14] for IoD applications requires communication and computational overheads. However, the presented scheme cannot meet the requirement of proper revocation or re-issue operations.

Jung *et al.* [38] come up with an efficient ASKE mechanism for WSN employing the hash function. However, the scheme cannot check tracing attacks, Ephemeral Secret Leakage (ESL) attack, UIMP attack, and does not ensure SK security [39]. In order to address the security limitations associated with the scheme presented in [38], Shin *et al.* [39] devised a user ASKE mechanism. The scheme of Shin *et al.* ably addresses most of the limitations of the scheme presented by Jung *et al.*, however, the computational cost incurred by the scheme of Shin *et al.* makes it computationally infeasible for IoT environment. Above this, the scheme of Shin *et al.* is also prone to ESL and de-synchronization attacks. The authentication scheme presented in [40] cannot prevent the de-synchronization and PRIN attacks. Gupta *et al.* [41] suggested a user ASKE mechanism to deal with the security of the wearable devices. However, the devised scheme is unprotected against impersonation and de-synchronization attacks and does not provide SK security, as illustrated in [42]. Additionally, Jangirala *et al.* presented a user ASKE mechanism for IoD environment [13], which is immune to various well-known attacks. However, the scheme cannot encompass all the security requirements of the IoD environment. Zhihan [43] used convolution neural network and presented a security solution for IoD, which is again not suitable to cover the security concern of the IoD environment completely. The authors in [44] presented an ASKE mechanism in order to protect 6LoWPAN networks. The scheme leverages ASCON and hash function for protecting the devices with 6LoWPAN. However, their scheme cannot achieve

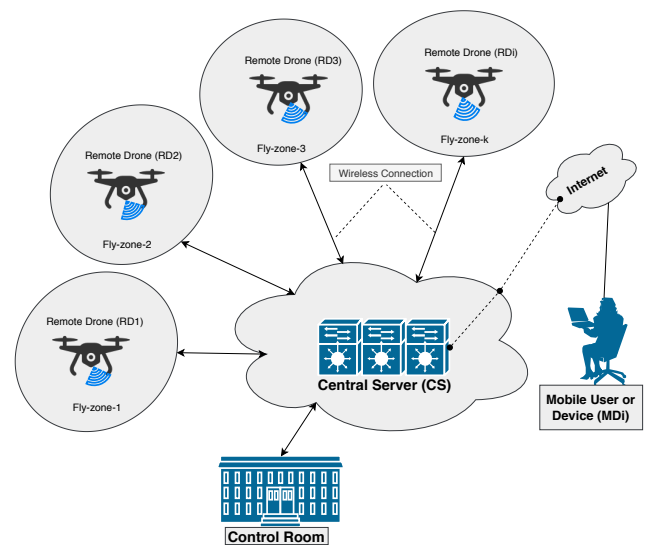


FIGURE 1: IoD network model [13], [14].

satisfactory performance against traceability attacks. In the same fashion, the scheme presented by Chen *et al.* in [45] is fallible to replay, DoS, STSC, PRIN, UIMP, PAGU, and also does not provide mutual authentication and anonymity features. Similarly, the ECC-based ASKE mechanism proposed by Wu *et al.* in [15] is insecure against replay, DoS, PAGU, and UIMP attacks. The scheme of Ref. [46] is unsafe against UIMP, PRIN, and STSC and also does not render SK security. Table 1 summarizes the security weaknesses of different ASKE for IoT and IoD environments.

### III. SYSTEM MODEL

The subsequent models (Network & Threat model) are utilized in designing PASKE-IoD.

#### A. NETWORK MODEL

This paper considers IoD architecture, as shown in Fig.1 for the ASKE process, which consists of Remote Drones (RDs) deployed in specific FZ, EU, CR, and CS. In an IoD environment, RDs and GS are connected through wireless channels. An RD is equipped with various types of sensors, an actuator, a communication module, power resources (battery), and processing capabilities. An RD collects significant information from the different circumstances and sends the collected sensitive information to the Central Server (CS) stationed at GS. EU and GS communicate through the public Internet. In IoD, the EU is an external entity and requires collecting real-time information from RD instead of procuring the information stored at CS. CS is the only trusted object/entity in the deployed IoD network, which is used to keep secret information about EU and RD. The internal user at the CR monitors RDs and controls their activities by sending various command and control (C&C) information to RDs. Due to the wireless channel's open nature, many security threats (attacks) can arise and deteriorate the performance

TABLE 1: Summary of the various existing security schemes

Security Scheme	Year	Cryptographic Operation Applied	Shortcoming
Xue <i>et al.</i> [24]	2016	Exclusive-OR and SHA-160	The devised scheme is unsafe against UIMP attack, PRIN attack, and PAGU attack. It also does not provide UA and SK security.
Mohammad Ali <i>et al.</i> [28]	2016	Exclusive-OR and SHA-160	The scheme is unprotected against IDGU attack.
Jangirala <i>et al.</i> [34]	2017	Exclusive-OR and SHA-160	The scheme cannot withstand UIMP attack, replay attack, and forgery attack, and also fails to render the mutual authentication.
F. Wu <i>et al.</i> [35]	2017	Exclusive-OR and SHA-160	The designed scheme is unprotected against STSC attack, DoS attack, and UIMP attack.
Wu <i>et al.</i> [15]	2017	Exclusive-OR, ECC, and SHA-160	The scheme cannot withstand DoS, replay, PAGU, and UIMP attacks.
Jung <i>et al.</i> [38]	2017	Exclusive-OR and SHA-160	The scheme is unsafe against tracing attack, ESL attack, and UIMP attack, and also does not render SK security.
Amin <i>et al.</i> [25]	2018	Exclusive-OR and SHA-160	The scheme does not ensure resistance against UIMP attack and PRIN attack.
Chen <i>et al.</i> [45]	2018	Exclusive-OR, ECC, and SHA-160	The scheme is unsafe against STSC, PRIN, PAGU, UIMP, DoS, and replay attacks. It also does not render anonymity and mutual authentication features.
Das <i>et al.</i> [46]	2018	Exclusive-OR and SHA-160	The scheme cannot withstand STSC, PRIN, and UIMP attacks. It does not ensure SK's security.
Gupta <i>et al.</i> [41]	2019	Exclusive-OR and SHA-160	The scheme is unprotected against de-synchronization attack and UIMP attack, and also does not ensure SK security.
Shin <i>et al.</i> [39]	2019	Exclusive-OR and SHA-160	The scheme is insecure against ESL attack and de-synchronization attack.
Jangirala <i>et al.</i> [13]	2019	Exclusive-OR and SHA-160	The scheme is unprotected against STSC attack, UIMP attack, and PRIN attack. It also suffers from scalability issue.
Wazid <i>et al.</i> [14]	2019	Exclusive-OR and SHA-160	The scheme is unprotected against STSC attack, UIMP attack, and PRIN attack.

Note: SHA stands for Secure Hash Algorithm, ECC for Elliptic Curve Cryptography, MAMI for Man-in-the-Middle, and DoS for Denial-of-Service.

of IoD networks. Therefore, it is of grave importance to secure the communication among RD, CS, and EU to avoid severe security circumstances, such as illegal information disclosure, unauthorized access to the network resources in the IoD environment.

### B. THREAT MODEL

As a threat model, the well-known Dolev-Yao (DY) [47] [48], [49] threat-model is considered for PASKE-IoD. It is worth noting that intruders can capture and record the communicating messages of network entities in the IoD network under the DY model. Communication among the entities in the IoD network is public, and an intruder or adversary can update, delete, modify, or forge the captured message. RDs are usually stationed in an unattended environment, making their physical security challenging to guarantee. There is always a physical security threat in which an intruder or adversary can capture RDs and extract the secret information from their memory. The adversary can afterward utilize the confidential information extracted from seized RD to compromise the security of other protected RDs in the IoD environment.

Furthermore, an adversary is assumed to be able to obtain, from the lost or stolen mobile device of a user, the stored information in the device's memory, by applying the Power Analysis (PA) attack. By deriving the secret parameters successfully, the adversary may launch various malicious attacks that include but are not limited to privileged-insider, replay, and impersonation attacks. Equally important, it is taken for granted that CS is a trusted entity and cannot be compromised by an adversary in the IoD environment.

## IV. PRELIMINARIES

Here, the preliminaries employed for our proposed scheme are elaborated.

### A. ASCON

ASCON is an AEAD scheme, which has the attributes of being symmetric [16], [50]. Moreover, it is inverse free, requires a single pass, and provides an online block cipher. ASCON is therefore selected as the finalist candidate in Caesar competition [1], [51]–[53]. ASCON generates output tuple  $\{CT, AuPa\}$ . Mathematically, the encryption operation of ASCON can be represented as  $CT, AuPa = E_{S_k}(\{AD\}, PT)$ , and decryption process by  $PT, AuPa' = D_{S_k}(\{AD\}, CT)$  and  $AuPa$ , where  $AD$  is the Associative Data, and  $PT$  is Plaintext. ASCON  $S_k$  can be computed as  $S_k = k \parallel N \parallel IV$ , where  $k$  is pre-shared key,  $N$  is nonce (random number used once with a key), and  $IV$  is the initialization vector.

### B. FUZZY EXTRACTOR

This paper employs the Fuzzy Extractor (FE) [54] method for the bio-metric verification of EU. FE consist of two functions  $gen(\cdot)$  and  $rep(\cdot)$ .

- 1)  $gen(\cdot)$  : is a probabilistic function, which is used to generate secret bio-metric key by computing  $(k_{EU}, r_p) = gen(Bio_{EU})$  of length  $L$  bits.  $Bio_{EU}$  is the bio-metric information of EU,  $k_{EU}$  is the generated secret key for EU, and  $r_p$  is the public-reproduction parameter.



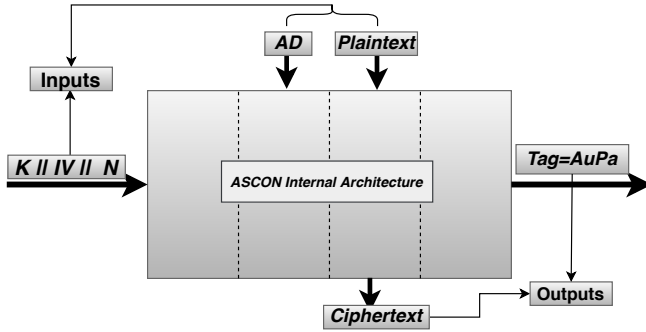


FIGURE 2: ASCON high level architecture.

TABLE 2: List of notations

Symbol	Description
$M_{Di}$ , $EU_i$ , and $CS$ ,	Mobile device and $i_{th}$ external user, and Central Server (CS), respectively
$TID_{CS}$ and $ID_{CS}$	Temporary and real identities of CS, respectively
$TID_{EU_i}$ , $ID_{EU_i}$ , AP	Temporary and real identities, and authentication parameter for the user, respectively
$ID_{RD_j}$ , $TID_{RD_j}$ , $ZID_k$	Real identity, temporary identity and FZ identity for the drone, respectively
$(T_{am1}, T_{am2}, T_{am3})$ , $(R_{am1}^v, R_{am3}^v)$	Timestamps and initialization vectors utilized during the user authentication phase, respectively
$AuPa_z$	Authentication parameter, where $z = 1, 2, 3, 4, 5, 6$ , which is used to check the authenticity of a message
$k_x$ and $N_x$	Key and Nonce, where $x = 1, 2, \dots, 9$ , respectively
$T_1^d, T_2^d, T_3^d$ , and $T^r$	Maximum allowed time delay at $CS$ , $RD_j$ , $M_{Di}$ , and message receive time at the receiver, respectively
$K_{am1}, K_{am2}, K_{am3}$	Initialization state ( $S_k$ ) during user authentication phase, respectively
$E_k(x1)$ , $D_k(x1)$	Encryption/decryption of message "x1" using the secret-key " $k$ ", respectively
$R_{se1}, R_{se2}, R_{se3}$	Temporary random number used during the drone and user authentication phase, respectively
$H(\cdot), \parallel, \oplus, gen(\cdot), rep(\cdot)$	ASCON-Hash function, concatenation, Exclusive-OR, fuzzy extractor-based key generation, and reproduction function, respectively

2)  $rep(\cdot)$  : is a deterministic function.  $rep(\cdot)$  takes  $EU$  bio-metric information  $Bio'_{EU}$  and  $r_p$  as the input and generates the original bio-metric key  $k_{EU}$ , while ensuring the condition  $HD(Bio_{EU}, Bio'_{EU}) \leq t$ , where  $HD$  is the Hamming Distance and  $t$  is the error tolerance threshold.

## V. THE PROPOSED PASKE-IOD SCHEME

The proposed PASKE-IoD is divided into the following six phases. The proposed PASKE-IoD utilizes the ASCON-Hash function that takes an arbitrary input length and produces 256 bits output. A detailed description of PASKE-IoD is given in the trailing sections.

### A. DRONE DEPLOYMENT PHASE (DDP)

This phase deals with the drone deployment in a specific FZ in an IoD environment. Each FZ has a unique identity  $ZID_k$ . It is supposed that  $CS$  has its distinct identity  $ID_{CS}$  and temporary identity  $TID_{CS}$ , which are known only to  $CS$ . The subsequent steps are necessitated to perform the registration of a  $RD_j$  with  $CS$ .

1) Step DDP-2:  $CS$  assigns a unique identity  $ID_{RD_j}$  and

a FZ identity  $ZID_k$  to the drone.

- 2) Step DDP-3:  $CS$  picks  $R_j$  and determines the temporary identity of  $RD_j$  by determining  $U = H(ID_{CS} \parallel R_j \parallel ZID_k \parallel ID_{RD_j})$ ,  $TID_{RD_j} = U_a \oplus U_b$ , where  $U_1$  and  $U_2$  are two same-sized parameters of  $U$ .
- 3) Step DDP-3:  $CS$  stores the parameters  $\{TID_{RD_j}, ID_{RD_j}, ZID_k\}$  in the memory of  $RD_j$ .

### B. USER REGISTRATION PHASE (URP)

Before obtaining the real-time information from a particular  $RD_j$  stationed in a FZ,  $EU_i$  requires registering with  $CS$ . For  $EU_i$  registration, subsequent steps are needed.

1) Step URP-1

$EU_i$  chooses its identity  $ID_{EU_i}$ , password  $PW_{EU_i}$ , and also generates a random number  $R_{ue}$ .  $EU_i$  imprints its bio-metric information  $Bio_{EU_i}$  at the interface available on  $M_{Di}$  and computes  $(k_{EU_i}^{reg}, r_p) = gen(Bio_{EU_i})$ ,  $AS_{reg} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i})$ , and  $SID_i = H(AS_{reg} \parallel R_{ue})$ . Furthermore,  $M_{Di}$  constructs a message  $M_{reg}^1 : \{SID_i\}$  and forwards  $M_{reg}^1$  to  $CS$  via a reliable channel.

2) Step URP-2

After receiving  $M_{reg}^1$  from  $M_{Di}$ ,  $CS$  picks timestamp  $T_{reg}$  and a master-key  $M_{ku}$  for  $EU_i$ . Additionally,  $CS$  computes  $G^{reg} = H(ID_{CS} \parallel SID_i \parallel T_{reg})$ ,  $TID_{EU_i} = G_1^{reg} \oplus G_2^{reg}$ . Moreover,  $CS$  calculates  $Z^{reg} = H(ZID_k \parallel M_{ku} \parallel ID_{CS})$  and authentication parameter  $AP = Z_1^{reg} \oplus Z_2^{reg}$ . Finally,  $CS$  fabricates a message  $M_{reg}^2 : \{TID_{CS}, TID_{EU_i}, TID_{RD_j}, AP\}$ , where  $TID_{CS}$ ,  $TID_{EU_i}$ , and  $TID_{RD_j}$  are the temporary identities of  $CS$ ,  $EU_i$ , and  $RD_j$ , respectively and dispatches  $M_{reg}^2$  to  $M_{Di}$  securely. Furthermore,  $CS$  stores  $\{TID_{CS}, TID_{EU_i}, TID_{RD_j}, AP\}$ .

3) Step URP-3

Upon receiving  $M_{reg}^2$  from  $CS$ ,  $M_{Di}$  calculates  $Q = H(PW_{EU_i} \parallel ID_{EU_i} \parallel (0000))$ . Moreover,  $EU_i$  determines  $P_1 = (TID_{CS} \oplus TID_{EU_i} \oplus TID_{RD_j} \oplus AP)$ ,  $P_2 = (TID_{CS} \parallel TID_{EU_i}) \oplus AS_{reg} \oplus Q$ , and  $P_3 = Q \oplus (TID_{RD_j} \parallel AP) \oplus AS_{reg}$ . Furthermore,  $EU_i$  computes  $AuPa_{reg} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i} \parallel P_1)$ . Finally,  $M_{Di}$  stores the parameters  $\{P_2, P_3, AuPa_{reg}, gen(\cdot), rep(\cdot), r_p, t\}$  in its own memory and removes  $P_1$  from the memory.

The summary of the user registration process as shown in Fig.4. Fig.3 illustrates the parameters stored in  $M_{Di}$ ,  $CS$ , and  $RD_j$  during deployment phase.

### C. USER LOGIN AND AUTHENTICATION PHASE (ULAP)

This phase validates the user's authenticity by verifying the secret login credentials stored on  $CS$  and  $M_{Di}$ . After receiving the login request,  $CS$  and  $RD_j$  validate the authenticity of  $EU_i$ . It is assumed that  $EU_i$  has a list of  $RD_j$  from where  $EU_i$  is granted to obtain the real-time data accumulated by  $RD_j$ . The subsequent steps outline the details of ULAP.

External User's $M_{Di}$
$\{P_2, P_3, AuPa_{reg}, gen(\cdot), rep(\cdot), r_p, t\}$
Control Server $CS$
$\langle (TID_{EU_i}, AP), (ID_{RD_j}, TID_{RD_j}, ZID_k) \rangle$
Remote Drone $RD_j$ deployed in a specific FZ
$\langle (TID_{RD_j}, ID_{RD_j}, ZID_k) \rangle$

FIGURE 3: Parameters stored during the pre-deployment phase.

External User $EU_i$	Central Server $CS$
Inputs $ID_{EU_i}$ , $PW_{EU_i}$ , and $R_{ue}$ , imprints bio-metric $Bio_{EU_i}$ , $(k_{EU_i}^{reg}, r_p) = gen(Bio_{EU_i})$ , $AS_{reg} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i})$ , $SID_i = H(AS_{reg} \parallel R_{ue})$ , $M_{reg}^1 \{SID_i\}$  Upon receiving $M_{reg}^2$ , calculates $Q = H(PW_{EU_i} \parallel ID_{EU_i} \parallel (0000))$ , $P_1 = (TID_{CS} \oplus TID_{EU_i} \oplus TID_{RD_j} \oplus AP)$ , $P_2 = (TID_{CS} \parallel TID_{EU_i}) \oplus AS_{reg} \oplus Q$ , $P_3 = Q \oplus (TID_{RD_j} \parallel AP) \oplus AS_{reg}$ , $AuPa_{reg} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i} \parallel P_1)$ , stores $\{P_2, P_3, AuPa_{reg}, gen(\cdot), rep(\cdot), r_p, t\}$ .	picks $T_{reg}$ and $M_{ku}$ , computes $G_{reg}^{reg} = H(ID_{CS} \parallel SID_i \parallel T_{reg})$ , $TID_{EU_i} = G_1^{reg} \oplus G_2^{reg}$ , $Z^{reg} = H(ZID_{RD_j} \parallel M_{ku} \parallel ID_{CS})$ , $AP = Z_1^{reg} \oplus Z_2^{reg}$ , $M_{reg}^2: \{TID_{CS}, TID_{EU_i}, TID_{RD_j}, AP\}$  stores $\{TID_{CS}, TID_{EU_i}, TID_{RD_j}, AP\}$

FIGURE 4: User registration process.

## 1) Step ULAP-1

$EU_i$  inputs the login secret credential, such as  $ID_{EU_i}$ ,  $PW_{EU_i}$ , and imprints  $Bio_{EU_i}^b$  at bio-metric sensor of  $M_{Di}$ .  $M_{Di}$  computes  $k_{EU_i}^b = rep(Bio_{EU_i}^b, r_p)$  provided the condition  $HD(Bio_{EU_i}^b, Bio_{EU_i}^b) \leq t$  holds. Moreover,  $M_{Di}$  calculates  $AS_{lo} = H(PW_{EU_i} \parallel k_{EU_i}^b \parallel ID_{EU_i})$ ,  $Q_{lo} = H(PW_{EU_i} \parallel ID_{EU_i} \parallel (0000))$ . In addition,  $M_{Di}$  derives the secret parameters, which are used in the ASKE process as  $P_2 \oplus AS_{lo} \oplus Q_{lo} = (TID_{CS} \parallel TID_{EU_i})$  and  $P_3 \oplus Q \oplus AS_{reg} = (TID_{RD_j} \parallel AP)$ . Finally, to validate the local authentication of  $EU_i$ ,  $M_{Di}$  determines  $P_{lo} = (TID_{CS} \oplus TID_{EU_i} \oplus TID_{RD_j} \oplus AP)$  and  $AuPa_{lo} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i} \parallel P_{lo})$ .  $M_{Di}$  verifies the condition  $AuPa_{lo} = AuPa_{reg}$ . If the condition is true, the login attempt will be successful and  $M_{Di}$  continues the ASKE process. Otherwise,  $M_{Di}$  terminates the login process. Moreover,  $M_{Di}$  retrieves the credentials  $\{TID_{CS}, TID_{EU_i}, TID_{RD_j}, AP\}$ . To generate a ASKE request message,  $M_{Di}$  picks timestamp  $T_{am1}$ , two random numbers  $R_{am1}^{iv}$ ,  $R_{se1}$ , where the size of  $T_{am1}$ ,  $R_{am1}^{iv}$ ,  $R_{se1}$  is 32 bits, 64 bits, and 128 bits, respectively. Additionally,  $M_{Di}$  determines  $P_6 = R_{se}$ ,  $P_7 = TID_{RD_j}$ ,  $X_n = H(TID_{CS} \parallel R_{am1}^{iv} \parallel T_{am1})$ , and  $TID_{CS}^n = X_n^1 \oplus X_n^2$ , where  $X_n^1$  and  $X_n^2$  are two same-sized parameters of  $X_n$  each of size 128 bits. Furthermore,  $M_{Di}$  computes  $X_2 = TID_{CS}^n \oplus TID_{EU_i}$ ,  $N_3 = X_2$ ,  $k_3 = AP$ ,  $K_{am1} = (k_3 \parallel N_3)$ , and  $AD_5 = X_2$ , where the size of both  $k_3$  and  $N_3$  is 128 bits and Associative Data of size 128 bits. Finally,  $M_{Di}$  by using ASCON, calculates  $(CT_6, CT_7)$ ,  $AuPa_1 = EK_{am1}\{AD_5, PT_6, PT_7\}$ , where  $AuPa_1$  is the authentica-

tion parameter and fabricates a message  $M_{am1}: \{T_{am1}, X_2, CT_6, CT_7, AuPa_1, R_{am1}^{iv}\}$ , and sends  $M_{am1}$  to  $CS$  via an open channel.

## 2) Step ULAP-2

Upon receiving  $M_{am1}$  from  $EU_i$ ,  $CS$  ensures the freshness of  $M_{am1}$  by verifying the condition  $T_3^d \geq |T^r - T_{am1}|$ , where  $T_3^d$  maximum allowed delay and  $T^r$  is the message received time.  $CS$  picks  $T_{am1}$  and  $R_{am1}^{iv}$  from the received  $M_{am1}$  and computes  $X_n = H(TID_{CS} \parallel R_{am1}^{iv} \parallel T_{am1})$ ,  $TID_{CS}^n = X_n^1 \oplus X_n^2$ , and  $TID_{EU_i} = TID_{CS} \oplus X_2$ . Moreover,  $CS$  checks if  $TID_{EU_i}$  exists in its own database. If  $TID_{EU_i}$  is found in its own database,  $CS$  retrieves  $AP$  related to  $TID_{EU_i}$ . Additionally,  $CS$  computes  $N_4 = TID_{CS}^n \oplus TID_{EU_i}$ ,  $k_4 = AP$ ,  $K_{am1} = (k_4 \parallel N_4)$ , and  $AD_6 = X_2$ , which is Associative Data of size 128 bits. In addition,  $CS$  by using ASCON determines  $(PT_6, PT_7)$ ,  $AuPa_2 = D_{K_{am1}}\{\{AD_6\}, CT_6, CT_7\}$ . Furthermore, to check the authenticity of the received message,  $CS$  checks the condition  $AuPa_1 = AuPa_2$ . If the condition is true,  $CS$  extracts  $R_{se1}$  and  $TID_{RD_j}$  from decryption process. Otherwise,  $CS$  terminates the ASKE process. Upon successful verification of  $EU_i$ ,  $CS$  retrieve  $ID_{RD_j}$  and  $ZID_k$  from its databases corresponding to  $TID_{RD_j}$ .

## 3) Step ULAP-3

$CS$  picks timestamp  $T_{am2}$ , two random numbers  $R_{se2}$  and  $R_{am2}^{iv}$  and computes  $X_3 = TID_{RD_j} \oplus R_{se2}$ ,  $PT_8 = R_{se1} \oplus AP$ , where  $PT_8$  is the plaintext.  $CS$  also computes  $U = H(ID_{RD_j} \parallel ZID_k \parallel T_4 \parallel R_{am2}^{iv})$  and splits  $U$  into two similar-sized parameters  $N_5$  and  $k_5$  each of size 128 bits. Moreover,  $CS$  calculates  $K_{am2} = (k_5 \parallel N_5)$ . Furthermore,  $CS$  by employing ASCON, calculates  $AD_7 = X_3$ ,  $(CT_8, AuPa_3) = EK_{am2}\{\{AD_7\}, PT_8\}$ . Finally,  $CS$  fabricates a message  $M_{am2}: \{T_{am2}, X_3, CT_8, AuPa_3, R_{am2}^{iv}\}$  and dispatches  $M_{am2}$  to  $RD_j$  via the public communication channel.

## 4) Step ULAP-4

After receiving  $M_{am2}$ ,  $RD_j$  verifies the freshness of  $M_{am2}$  by verifying the condition  $T_4^d \geq |T^r - T_{am1}|$ . If the condition is true,  $RD_j$  continues the ASKE process. Otherwise,  $RD_j$  rejects  $M_{am2}$  and aborts the ASKE process. In addition,  $RD_j$  determines  $R_{se2} = TID_{RD_j} \oplus X_3$ ,  $AD_8 = X_3$ ,  $U_1 = H(ID_{RD_j} \parallel ZID_k \parallel T_{am2})$ , and divides  $U_1$  into two similar-sized parameters each of 128 bits, namely, nonce  $N_6$  and key  $k_6$ . Furthermore,  $RD_j$  calculates  $K_{am2} = (k_6 \parallel N_6)$  and by using ASCON computes  $(PT_8, AuPa_3) = D_{K_{am2}}\{\{AD_8\}, CT_8\}$ . Finally, to verify the authenticity of received  $M_{am2}$ ,  $RD_j$  verifies the condition  $AuPa_3 = AuPa_4$ . If the condition is true, decryption process reveals the plaintext, i.e.,  $P_8 = (R_{se1} \oplus AP)$ . If the condition is not true,  $RD_j$  aborts the ASKE process.

External User/ Mobile Device $EU_i/M_{Di}$	Control Server $CS$	Drone $RD_j$
input $ID_{EU_i}$ , $PW_{EU_i}$ and imprints $Bio_{EU_i}^b$ , computes $k_{EU_i}^b = rep(Bio_{EU_i}^b, r_p)$ , $AS_{lo} = H(PW_{EU_i} \parallel k_{EU_i}^b \parallel ID_{EU_i})$ , $Q_{lo} = H(PW_{EU_i} \parallel ID_{EU_i} \parallel (0000))$ , $P_2 \oplus AS_{lo} \oplus Q_{lo} = (TID_{CS} \parallel TID_{EU_i})$ , $P_3 \oplus Q \oplus AS_{reg} = (TID_{RD_j} \parallel AP)$ , $P_{lo} = (TID_{CS} \oplus TID_{EU_i} \oplus TID_{RD_j} \oplus AP)$ , $AuPa_{lo} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i} \parallel P_{lo})$ , verifies $AuPa_{lo} = AuPa_{reg}$ , if so, picks $T_{am1}$ , $R_{am1}^{iv}$ , $R_{se1}$ , and computes $P_6 = R_{se1}$ , $P_7 = TID_{RD_j}$ , $X_n = H(TID_{CS} \parallel R_{am1}^{iv} \parallel T_{am1})$ , $TID_{CS}^n = X_n^1 \oplus X_n^2$ , $X_2 = TID_{CS}^n \oplus TID_{EU_i}$ , $N_3 = X_2$ , $AD_5 = X_2$ , $k_3 = AP$ , $K_{am1} = (k_3 \parallel N_3)$ , $(CT_6, CT_7, AuPa_1) = E_{K_{am1}}\{\{AD_5\}, PT_6, PT_7\}$ $\xrightarrow{\text{via an open channel}} M_{am1}:\{T_{am1}, X_2, CT_6, CT_7, AuPa_1, R_{am1}^{iv}\}$	checks if $T_3^d \geq  T^r - T_{am1} $ , If so, picks $T_{am1}$ and $R_{am1}^{iv}$ from the received $M_{am1}$ , computes $X_n = H(TID_{CS} \parallel R_{am1}^{iv} \parallel T_{am1})$ , $TID_{CS}^n = X_n^1 \oplus X_n^2$ , $TID_{EU_i} = TID_{CS}^n \oplus X_2$ , checks if $TID_{EU_i}$ exist in database or not, retrieves $AP$ related to $TID_{EU_i}$ , calculates $N_4 = TID_{CS}^n \oplus TID_{EU_i}$ , $k_4 = AP$ , $K_{am1} = (k_4 \parallel N_4)$ , $AD_6 = X_2$ , $PT_6, (PT_7, AuPa_2) = D_{K_{am1}}\{\{AD_6\}, CT_6, CT_7\}$ , checks the condition $AuPa_1 = AuPa_2$ ?, if so, obtains $R_{se1}$ and $TID_{RD_j}$ , retrieves $ID_{RD_j}$ , $ZID_k$ related to $TID_{RD_j}$ , picks $T_{am2}$ , $R_{se2}$ , and $R_{am2}^{iv}$ , computes $X_3 = TID_{RD_j} \oplus R_{se2}$ , $PT_8 = R_{se1} \oplus AP$ , $U = H(ID_{RD_j} \parallel ZID_k \parallel T_4 \parallel R_{am2}^{iv})$ , $K_{am2} = (k_5 \parallel N_5)$ , $AD_7 = X_3$ , $(CT_8, AuPa_3) = E_{K_{am2}}\{\{AD_7\}, PT_8\}$ . $\xrightarrow{\text{via an open channel}} M_{am2}:\{T_{am2}, X_3, CT_8, AuPa_3, R_{am2}^{iv}\}$	checks if $T_4^d \geq  T^r - T_{am2} $ , if so, computes $R_{se2} = TID_{RD_j} \oplus X_3$ , $AD_8 = X_3$ , $U_1 = H(ID_{RD_j} \parallel ZID_k \parallel T_{am2})$ , $K_{am2} = (k_6 \parallel N_6)$ , $PT_8 = D_{K_{am2}}\{\{AD_8\}, CT_8\}$ , and $AuPa_3$ , checks if $AuPa_3 = AuPa_4$ ?, if so, retrieves $PT_8 = R_{se1} \oplus AP$ , and picks $T_{am3}$ , $R_{se3}$ , $R_{am3}^{iv}$ , computes $PT_9 = R_{se3} \oplus ZID_k \oplus R_{se2}$ , $U_2 = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel T_{am3})$ , $AD_9 = R_{am3}^{iv} \parallel R_{am3}^{iv}$ , $K_{am3} = (k_6 \parallel N_6)$ , $(CT_9, AuPa_5) = E_{K_{am3}}\{\{AD_9\}, PT_9\}$ . computes $SK_{d-u} = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel PT_9 \parallel T_{am3})$ $\xleftarrow{\text{via an open channel}} M_{am3}:\{T_{am3}, CT_9, AuPa_5, R_{am3}^{iv}\}$
checks if $T_5^d \geq  T^r - T_{am3} $ , if so, computes $R_{se1} \oplus AP$ , $AD_{10} = R_{am3}^{iv}$ , $U_3 = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel T_5)$ , $K_{am3} = (k_7 \parallel N_7)$ , $(PT_9, AuPa_6) = D_{K_{am3}}\{\{AD_{10}\}, CT_9\}$ , checks $AuPa_5 = AuPa_6$ ?, if so, retrieves $PT_9$ , computes $SK_{u-d} = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel PT_9 \parallel T_{am3})$	$SK_{u-d} = (SK_{d-u}) = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel PT_9 \parallel T_{am3})$	

FIGURE 5: PASKE-IoD user ASKE phase.

## 5) Step ULAP-5

$RD_j$  picks timestamp  $T_{am3}$ , two random numbers  $R_{se3}$ ,  $R_{am3}^{iv}$ , and computes  $PT_9 = (R_{se3} \oplus ZID_k \oplus R_{se2})$ ,  $U_2 = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel T_{am3})$  and divides  $U_2$  into two similar-sized parameters  $N_6$  and  $k_6$ , where  $N_6$  is the nonce and  $k_6$  is the key. Moreover,  $RD_j$  calculates  $AD_9 = R_{am3}^{iv} \parallel R_{am3}^{iv}$ ,  $K_{am3} = (k_6 \parallel N_6)$ . Finally,  $RD_j$  computes  $(CT_9, AuPa_5) = E_{K_{am3}}\{\{AD_9\}, PT_9\}$ . Additionally,  $RD_j$  constructs a message  $M_{am3}:\{T_{am3}, CT_9, AuPa_5, R_{am3}^{iv}\}$  and sends  $M_{am3}$  to  $M_{Di}$  via an open channel. In addition, to secure the future communications between  $RD_j$  and  $M_{Di}$ ,  $RD_j$  computes SK as  $SK_{d-u} = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel PT_9 \parallel T_{am3})$

## 6) Step ULAP-7

After receiving  $M_{am3}$  from  $RD_j$ ,  $M_{Di}$  verifies the freshness of  $M_{am3}$  by verifying the condition  $T_3^d \geq |T^r - T_{am3}|$ . If the condition holds,  $M_{Di}$  continues the ASKE process. Otherwise,  $M_{Di}$  rejects the received  $M_{am3}$  and aborts the ASKE process. In addition,  $M_{Di}$  computes  $R_{se1} \oplus AP$ ,  $AD_{10} = R_{am3}^{iv}$ ,  $U_3 = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel T_{am3})$ ,  $AS_f = (k_7 \parallel N_7)$ , where  $N_7$  is nonce and  $k_7$  is key,

which are two same-sized parameters of  $U_3$ , and  $K_{am3} = (k_7 \parallel N_7)$ . Moreover,  $M_{Di}$  also computes  $(PT_9, AuPa_6) = D_{K_{am3}}\{\{AD_{10}\}, CT_9\}$  by using ASCON. Furthermore,  $M_{Di}$  checks the legitimacy of  $M_{am3}$  by checking the condition  $AuPa_5 = AuPa_6$ . If the condition is true,  $M_{Di}$  retrieves  $PT_9$  from the decryption process. To secure the communication between  $M_{Di}$  and  $RD_j$ ,  $M_{Di}$  computes SK as  $SK_{u-d} = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel PT_9 \parallel T_5)$ . The summary of the user login and ASKE phase as shown in Fig. 5.

## D. USER BIO-METRIC/PASSWORD UPDATE PHASE (UBPU)

It is important to note that the bio-metric information of  $EU_i$  remains unchanged. However, to achieve the maximum security,  $EU_i$  required to update his/her password periodically. In this phase, the new bio-metric information considered the same as the old bio-metric information.  $EU_i$  required to execute the following steps to update both bio-metric and password.

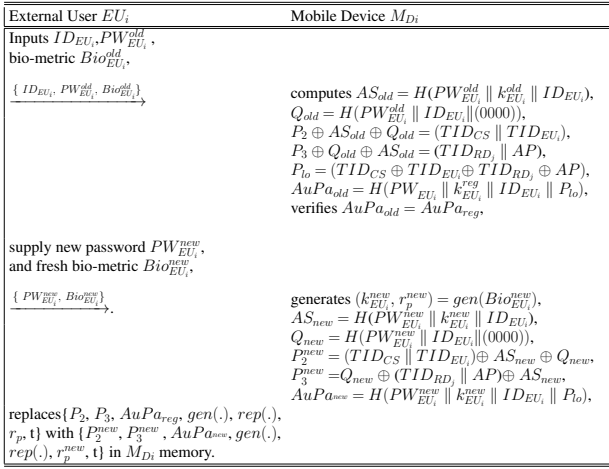


FIGURE 6: User bio-metric/password update phase.

### 1) Step UBPU-1

$EU_i$  inputs its secret parameters, such as  $ID_{EU_i}$ ,  $PW_{EU_i}^{old}$  and imprints bio-metric information  $Bio_{EU_i}^{old}$  at smart  $M_{Di}$ . Upon receiving the secret parameters,  $M_{Di}$  computes  $k_{EU_i}^{old} = rep(Bio_{EU_i}^{old}, r_p)$ , both old and fresh bio-metric information are same. Moreover, to accomplish the bio-metric and password change phase,  $M_{Di}$  computes  $AS_{old} = H(PW_{EU_i}^{old} \parallel k_{EU_i}^{old} \parallel ID_{EU_i})$ ,  $Q_{old} = H(PW_{EU_i}^{old} \parallel ID_{EU_i} \parallel (0000))$ ,  $P_2 \oplus AS_{old} \oplus Q_{old} = (TID_{CS} \parallel TID_{EU_i})$ ,  $P_3 \oplus Q_{old} \oplus AS_{old} = (TID_{RD_j} \parallel AP)$ , and  $P_{lo} = (TID_{CS} \oplus TID_{EU_i} \oplus TID_{RD_j} \oplus AP)$ . Finally,  $M_{Di}$  determines  $AuPa_{old} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i} \parallel P_{lo})$  and verifies the condition  $AuPa_{old} = AuPa_{reg}$ . If the condition is true,  $M_{Di}$  sends a notification message to  $EU_i$  to select new secret parameters, such as password and bio-metric information.

### 2) Step UBPU-2

After receiving the notification message from  $M_{Di}$ ,  $EU_i$  picks its new password  $PW_{EU_i}^{new}$  and  $Bio_{EU_i}^{new}$ . Upon procuring the new inputs from  $EU_i$ ,  $M_{Di}$  by using  $FE$  calculates new bio-metric key as  $(k_{EU_i}^{new}, r_p^{new}) = gen(Bio_{EU_i}^{new})$ . In addition,  $M_{Di}$  calculates  $AS_{new} = H(PW_{EU_i}^{new} \parallel k_{EU_i}^{new} \parallel ID_{EU_i})$  and  $Q_{new} = H(PW_{EU_i}^{new} \parallel ID_{EU_i} \parallel (0000))$ . In addition,  $M_{Di}$  computes  $P_2^{new} = (TID_{CS} \parallel TID_{EU_i}) \oplus AS_{new} \oplus Q_{new}$ ,  $P_3^{new} = Q_{new} \oplus (TID_{RD_j} \parallel AP) \oplus AS_{new}$ , and  $AuPa_{new} = H(PW_{EU_i}^{new} \parallel k_{EU_i}^{new} \parallel ID_{EU_i} \parallel P_{lo})$ , where  $P_{lo} = (TID_{CS} \oplus TID_{EU_i} \oplus TID_{RD_j} \oplus AP)$ . Finally,  $M_{Di}$  stores the parameters  $\{P_2^{new}, P_3^{new}, AuPa_{new}, gen(\cdot), rep(\cdot), r_p^{new}, t\}$  in its own memory. Fig. 6 shows summary of the user bio-metric/password update phase.

## E. REISSUE OR REVOCATION PHASE

If  $M_{Di}$  of a legitimate  $EU_i$  somehow lost or stolen,  $EU_i$  gets a new  $M_{Di}^n$  and accomplishes the Reissue or Revocation Phase (RRP) as follows.

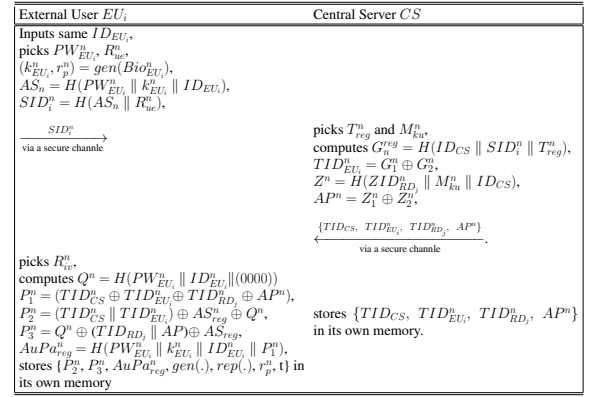


FIGURE 7: Reissue and revocation phase.

### 1) Step RRP-1

$EU_i$  needs to maintain same identity  $ID_{EU_i}$ .  $EU_i$  picks a new password  $PW_{EU_i}^n$ , random number  $R_{ue}^n$ , and  $EU_i$  imprints fresh/new bio-metric information  $Bio_{EU_i}^n$  and computes  $(k_{EU_i}^n, r_p^n) = gen(Bio_{EU_i}^n)$ ,  $AS_n = H(PW_{EU_i}^n \parallel k_{EU_i}^n \parallel ID_{EU_i})$ , and  $SID_i^n = H(AS_n \parallel R_{ue}^n)$ . Furthermore, the  $M_{Di}$  a message  $M_{reg}^n : \langle SID_i^n \rangle$  and dispatches  $M_{reg}^n$  to  $CS$  through a secure channel.

### 2) Step RRP-3

$CS$  picks timestamp  $T_{reg}^n$  and a new master-key  $M_{ku}^n$ .  $CS$  computes  $G_{reg}^n = H(ID_{CS} \parallel SID_i^n \parallel T_{reg}^n)$ ,  $TID_{EU_i}^n = G_1^n \oplus G_2^n$ ,  $Z^n = H(ZID_{RD_j}^n \parallel M_{ku}^n \parallel ID_{CS})$ , and  $AP^n = Z_1^n \oplus Z_2^n$ . Furthermore,  $CS$  dispatches a message  $M_{reg}^n : \{TID_{CS}, TID_{EU_i}^n, TID_{RD_j}^n, AP^n\}$  to  $M_{Di}$  via public channel.

### 3) Step RRP-3

Upon receiving  $M_{reg}^n$  from  $CS$ ,  $M_{Di}$  calculates  $Q^n = H(PW_{EU_i}^n \parallel ID_{EU_i}^n \parallel (0000))$ . Moreover,  $EU_i$  determines  $P_1^n = (TID_{CS} \oplus TID_{EU_i}^n \oplus TID_{RD_j}^n \oplus AP^n)$ ,  $P_2^n = (TID_{CS} \parallel TID_{EU_i}^n) \oplus AS_{reg}^n \oplus Q^n$ , and  $P_3^n = Q^n \oplus (TID_{RD_j} \parallel AP) \oplus AS_{reg}^n$ . Furthermore,  $EU_i$  computes  $AuPa_{reg}^n = H(PW_{EU_i}^n \parallel k_{EU_i}^n \parallel ID_{EU_i}^n \parallel P_1^n)$ . Finally,  $M_{Di}$  stores the parameters  $\{P_2^n, P_3^n, AuPa_{reg}^n, gen(\cdot), rep(\cdot), r_p^n, t\}$  in its own memory and removes  $P_1^n$  from the memory. The summary of the reissue and revocation phase is presented in the Fig. 7.

## F. DYNAMIC DRONE ADDITION PHASE (DDAP)

To deploy, a new remote drone  $RD_i^n$  in a specific FZ,  $CS$  executes the following necessary steps.

### 1) Step DDAP-1

$CS$  assigns a new unique  $ID_{RD_j}^n$  and a particular FZ identity  $ZID_k^n$  to the drone  $RD_i^n$  before its deployment.  $CS$  selects  $R_j^n$  and computes  $U^n = H(ID_{CS} \parallel R_j^n \parallel ZID_k^n \parallel ID_{RD_j}^n)$ ,  $TID_{RD_j}^n = U_1^n \oplus U_2^n$ , where  $U_1^n$  and  $U_2^n$  are two same-sized parameters of  $U^n$ .



## 2) Step DDAP-2

$CS$  stores the parameters  $\{ID_{RD_j}^n, TID_{RD_j}^n, ZID_k^n\}$  in the memory of  $RD_j^n$ .

## VI. SECURITY ANALYSIS

In this section, we present both the formal and informal security analysis of PASKE-IoD.

### A. INFORMAL SECURITY ANALYSIS

The trailing analysis demonstrates that PASKE-IoD is protected against different malicious attacks, such as replay, privilege insider, and impersonation, ensuring user's anonymity and untraceability.

#### 1) User Device Capture Attack

Suppose an adversary  $\mathcal{A}$  somehow gets/steals the Mobile Device  $M_{Di}$  of the user  $EU_i$  and extracts the parameters  $\{P_2, P_3, AuPa_{reg}, gen(\cdot), rep(\cdot), r_p, t\}$  stored on  $M_{Di}$  using PA attack [55]. To guess the valid  $PW_{EU_i}$  of  $EU_i$ ,  $\mathcal{A}$  requires to compute  $k_{EU_i}^A = rep(Bio_{EU_i}^A, r_p)$ ,  $AS_A = H(PW_{EU_i}^A \parallel k_{EU_i}^A \parallel ID_{EU_i}^A)$ ,  $Q_A = H(PW_{EU_i}^A \parallel ID_{EU_i}^A \parallel (0000))$ ,  $P_2^A \oplus AS_A \oplus Q_A = (TID_{CS} \parallel TID_{EU_i})$ ,  $P_3^A \oplus Q_A \oplus AS_{reg}^A = (TID_{RD_j} \parallel AP)$ ,  $P_A = (TID_{CS} \oplus TID_{EU_i} \oplus TID_{RD_j} \oplus AP)$ ,  $AuPa_A = H(PW_{EU_i}^A \parallel k_{EU_i}^A \parallel ID_{EU_i}^A \parallel P_A)$ , and verifies  $AuPa_A = AuPa_{reg}$ . However, it is infeasible for  $\mathcal{A}$  to compute these computation without knowing valid secret parameters, such as  $PW_{EU_i}$ ,  $Bio_{EU_i}$ , and  $ID_{EU_i}$ , which are known only to  $EU_i$ . Therefore, it is hard for  $\mathcal{A}$  to guess the password of  $EU_i$ . Thus, PASKE-IoD is resilient against the off-line PAGU attack.

#### 2) IDGU Attack

During  $EU_i$  registration phase,  $EU_i$  sends a registration message  $M_{reg}^1 : \langle SID_i \rangle$ , where  $SID_i = H(AS_{reg} \parallel R_{ue})$  via a reliable channel to  $CS$ , where  $AS_{reg} = H(PW_{EU_i} \parallel k_{EU_i}^{reg} \parallel ID_{EU_i})$  and  $R_{ue}$  is a random number.  $\mathcal{A}$  cannot get any significant information about  $EU_i$ 's secret parameters. Let  $\mathcal{A}$  obtains the lost  $M_{Di}$  of  $EU_i$  and procure information, i.e.,  $\{P_2, P_3, AuPa_{reg}, gen(\cdot), rep(\cdot), r_p, t\}$ , which are stored in the  $M_{Di}$ 's memory by employing PA attack. From the extracted information, it is hard for  $\mathcal{A}$  to get a valid  $ID_{EU_i}$  of  $EU_i$ . Therefore, PASKE-IoD is protected against IDGU attack.

#### 3) Anonymity/Un-traceability

According to DY [47] threat model,  $\mathcal{A}$  can intercept the messages, such as  $M_{am1} : \{T_{am1}, X_2, CT_6, CT_7, AuPa_1, R_{am1}^{iv}\}$ ,  $M_{am2} : \{T_{am2}, X_3, CT_8, AuPa_3, R_{am2}^{iv}\}$ , and  $M_{am3} : \{T_{am3}, CT_9, AuPa_5, R_{am3}^{iv}\}$ , which are communicated during the ASKE phase of PASKE-IoD. From these messages, it hard for  $\mathcal{A}$  to determine the user identity  $ID_{EU_i}$ , because the real identity  $ID_{EU_i}$  is known only to  $EU_i$  and only the pseudo identity  $TID_{EU_i}$  is used in communication. Therefore, PASKE-IoD ensures  $EU_i$ 's anonymity. Moreover, the generation of ciphertext  $CT_6, CT_7, CT_8$ , and  $CT_9$  by the

encryption algorithm incorporates the fresh random numbers  $R_{se1}$ ,  $R_{se2}$ , and  $R_{se3}$ . Furthermore, nonces are involved in the encryption process introduces more randomness in  $M_{am1}$ ,  $M_{am2}$ , and  $M_{am3}$ . Therefore, it is hard for  $\mathcal{A}$  to correlate the communicated messages from the current and previous ASKE process. Hence, PASKE-IoD ensures user untraceability.

#### 4) Replay Attack

Suppose during the login & ASKE phase,  $\mathcal{A}$  intercepts  $M_{am1}$ ,  $M_{am2}$ , and  $M_{am3}$  to execute the replay attack by replaying the intercepted message. However, the communicated messages  $M_{am1}$ ,  $M_{am2}$ , and  $M_{am3}$  incorporates latest timestamp and fresh random numbers. After receiving the message, the first step is to verify the freshness of the received message by checking if the received message within the allowed maximum delay limit. Furthermore, all exchanged messages are validated by verifying the conditions  $AuPa_1 = AuPa_2$ ,  $AuPa_3 = AuPa_4$ , and  $AuPa_5 = AuPa_6$  for  $M_{am1}$ ,  $M_{am2}$ , and  $M_{am3}$ , respectively. If the condition is not true hold for a specific message, the received message will be rejected. In this way, the replay attack is detected in PASKE-IoD.

#### 5) STSC Attack

Assume the adversary  $\mathcal{A}$  has got the lost/stolen  $M_{Di}$  of  $EU_i$  and attempts to modify the password and bio metric information of  $EU_i$ , so that  $\mathcal{A}$  can get access to IoD environment. However,  $\mathcal{A}$  can retrieve the information  $\{P_2, P_3, AuPa_{reg}, gen(\cdot), rep(\cdot), r_p, t\}$  stored in the memory of  $M_{Di}$  by applying PA. Based on the discussion in Section VI-A1, it is impractical for  $\mathcal{A}$  to procure any important information from the smart capture device. Hence, PASKE-IoD is resistant to STSC attacks.

#### 6) DoS Attack

In the proposed scheme, during the login & ASKE phase, an  $EU_i$  enters the valid parameters, such as  $ID_{EU_i}$ ,  $Bio_{EU_i}$ , and  $PW_{EU_i}$ , the authenticity of the entered parameters are validated by checking the condition  $AuPa_{reg} = AuPa_{reg}^{lo}$  locally at  $M_{Di}$ . The login request will be sent to  $CS$  only after the successful verification of the login credentials by  $M_{Di}$ . If the condition is not true, the login process will be aborted. In this way, it is possible to prevent  $EU_i$  from sending a large number of login requests to  $CS$ . Hence, PASKE-IoD is resistant against DoS attack.

#### 7) UIMP Attack

Suppose an adversary  $\mathcal{A}$  attempts to impersonate as a legitimate  $EU_i$  in IoD communication environment. To make a legitimate authentication request message,  $\mathcal{A}$  can generate  $M_{am1}' : \{T_{am1}', X_2', CT_6', CT_7', AuPa_{am1}', R_{am1}'^{iv}\}$  by picking a timestamp  $T_{am1}'$  and  $R_{se1}'$  on behalf of  $EU_i$ . However, without knowing the secret parameters, such as  $AP$ ,  $TID_{CS}$ ,  $TID_{EU_i}$ , and  $TID_{RD_j}$ ,  $\mathcal{A}$  cannot construct a valid  $M_{am1}$  on

behalf of  $EU_i$ . Therefore, PASKE-IoD is resistance against UIMP attack.

#### 8) CS Impersonation Attack

To generate this attack, assume  $\mathcal{A}$  picks timestamp  $T'_{am2}$ , and random number  $R'_{se2}$ .  $\mathcal{A}$  generates a bogus message  $M'_{am2}:\{T'_{am2}, X'_3, CT'_8, AuPa'_{am3}, R'_{am2}\}$  and transmits the generated  $M'_{am2}$  to the drone  $RD_j$ , to make  $RD_j$  believe  $M'_{am2}$  is from a legitimate  $CS$ . However, without knowing valid secret parameters, such as  $TID_{CS}$ ,  $ID_{RD_j}$ ,  $TID_{RD_j}$ , and  $ZID_k$ , it is hard for  $\mathcal{A}$  to construct valid  $M_{am2}$ . Therefore, the proposed scheme is secure against  $CS$  impersonation attack.

#### 9) Drone Impersonation Attack

In this case, the adversary  $\mathcal{A}$  tries to generate a fake message  $M'_{am3}:\{T'_{am3}, CT'_9, AuPa'_{au5}, R'_{am3}\}$  by generating  $R'_{se3}$  and timestamp  $T'_{am3}$  on behalf of drone  $RD_j$  and transmit  $M'_{am3}$  to  $EU_i$ . However, without knowing the secret parameters, such as  $ID_{RD_j}$ ,  $TID_{RD_j}$ , and  $P_8 = R_{se1} \oplus AP$ , it is hard for  $\mathcal{A}$  to construct a valid  $M_{am3}$ . Therefore, PASKE-IoD is secure against the drone impersonation attack.

#### 10) MAMI Attack

During the login and ASKE phase,  $\mathcal{A}$  after intercepting exchanged message, such as  $M_{am1}$ ,  $M_{am2}$ , and  $M_{am3}$  attempts to modify the captured messages to make believe the receiving entities that these messages generated by a valid entity in IoD environment. To execute this attack,  $\mathcal{A}$  can capture and forge  $M_{am1}:\{T_{am1}, X_2, CT_6, CT_7, AuPa_1, R_{am1}^{iv}\}$ . However, without knowing the secret parameters, such as  $TID_{RD_j}$ ,  $TID_{CS}$ ,  $TID_{EU_i}$ , and  $R_{se1}$ , it is difficult for  $\mathcal{A}$  to modify  $M_{am1}$ . Furthermore, in the same way, it is impractical for  $\mathcal{A}$  to forge  $M_{am2}:\{T_{am2}, X_3, CT_8, AuPa_3, R_{am2}^{iv}\}$ , and  $M_{am3}:\{T_{am3}, CT_9, AuPa_5, R_{am3}^{iv}\}$  due the secret parameters, which are known to a specific entity in IoD environment. Thus, PASKE-IoD is secure against the MAMI attack.

#### 11) Drone Capture Attack

According to the threat model defined in Section III-B, the adversary  $\mathcal{A}$  can capture  $RD_j$  because they are deployed in hostile environment.  $\mathcal{A}$  can extract the secret parameters, such as  $ID_{RD_j}$ ,  $TID_{RD_j}$ , and  $ZID_k$  stored in  $RD_j$ 's memory by employing PA attack.  $CS$  calculates  $TID_{RD_j} = U_a \oplus U_b$ , which is unique for all deployed  $RD_j$ s in the IoD environment. After capturing a  $RD_j$ ,  $\mathcal{A}$  can compromised the security of captured  $RD_j$ . However,  $\mathcal{A}$  will be unable to breach the security of other non-compromised  $RD_j$  by using the extracted information from the compromised  $RD_j$ . In this way, PASKE-IoD is resilient against the drone capture attack.

#### 12) Mutual Authentication (MA)

Mutual Authentication of PASKE-IoD illustrated in the following steps.

TABLE 3: BAN logic notations

Feature	Description
$\frac{S}{H}$	If statement $S$ is true then statement $H$ is also true
$E \models M$	$E$ believes statement $M$ is true
$E \mid \sim M$	$E$ once said $M$
$E \triangleleft M$	$E$ sees $M$
$E \stackrel{k}{\leftrightarrow} P$	$k$ is a shared-secret between $E$ and $P$
$\#(M)$	$M$ is fresh.
$\{M\}_k$	Statement $M$ is encrypted with the secret key $k$
$\langle M \rangle Y$	Statement $M$ is combine with statement $Y$
$E \Rightarrow M$	$E$ has jurisdiction over $M$

- 1)  $M_{Di} \rightarrow CS: M_{am1}:\{T_{am1}, X_2, CT_6, CT_7, AuPa_1, R_{am1}^{iv}\}$ :  $CS$  checks the  $TID_{EU_i}$  existence in its database and also checks the condition  $AuPa_1 = AuPa_2$  to validate authenticity of  $M_{am1}$  received from  $EU_i$ . If it is true,  $CS$  considers  $M_{am1}$  received from a legitimate  $EU_i$  and  $CS$  also extracts  $R_{se1}$  from the received ciphertext.
- 2)  $CS \rightarrow RD_j: M_{am2}:\{T_{am2}, X_3, CT_8, AuPa_3, R_{am2}^{iv}\}$ :  $RD_j$  computes  $R_{se2} = TID_{CS} \oplus TID_{RD_j} \oplus X_3$  and also checks the condition  $AuPa_3 = AuPa_4$  to ensure the authenticity of the received message. If it is true,  $RD_j$  considers  $M_{am2}$  generated by a legitimate  $CS$ . In addition to this,  $RD_j$  extracts  $P_8 = R_{se1} \oplus AP$ .
- 3)  $RD_j \rightarrow M_{Di}: M_{am3}:\{T_{am3}, CT_9, AuPa_5, R_{am3}^{iv}\}$ :  $EU_i$  checks the condition  $AuPa_5 = AuPa_6$  to verify  $M_{am3}$  received from the legitimate  $RD_j$ . If it is true,  $M_{Di}$  believe that  $M_{am3}$  is from a legitimate  $RD_j$ .  $M_{Di}$  extracts  $P_9$  from  $CT_9$ .

From the above discussion, it is clear that the proposed PASKE-IoD ensures the mutual authentication and after achieving MA, the entities  $EU_i$  and  $RD_j$  can set up a SK  $SK_{u-d}(= SK_{d-u}) = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel PT_9 \parallel T_{am3})$  with the help of  $CS$  for securing the future communications.

#### 13) Ephemeral Secret Leakage (ESL) Attack

SK is constructed as  $SK_{u-d}(= SK_{d-u}) = H(TID_{RD_j} \parallel R_{se1} \oplus AP \parallel PT_9 \parallel T_{am3})$  in the proposed PASKE-IoD, which incorporates both the temporary secret credential (ephemeral secrets) and long-term secret parameters. It is imperative for the attacker to simultaneously guess both ephemeral and long-term secrets to compromise the constructed SK.

#### B. MA VERIFICATION USING BAN LOGIC

The BAN logic is employed to determine the logic exactitude of PASKE-IoD. BAN logic is the logic of belief and action. The objective of applying BAN logic is to investigate whether the security protocol's expected results can be reached by ascertaining the beliefs of each authorized entity associated with the ASKE process. Table 3 presents the list of notation/symbols employed in the BAN logic and Table 4 demonstrates the BAN deduction rules.

TABLE 4: BAN logic inference rules

Notation	Description
Message-Meaning-Rule (MMR)	$\frac{E  \equiv E \xrightarrow{SK} P, E \triangleleft \{M\}_k}{E  \equiv P  \sim M}$
Jurisdiction-Rule (JR)	$\frac{E  \equiv P \Rightarrow M, E  \equiv P  \equiv M}{E  \equiv M}$
Belief-Rule (BR)	$\frac{E  \equiv (M, Y)}{E  \equiv M}$
Nonce-Verification-Rule (NVR)	$\frac{E  \equiv \#(M), E  \equiv P  \equiv M}{E  \equiv P  \equiv M}$
Freshness-Rule (FR)	$\frac{E  \equiv \#(M)}{M  \equiv \#(M, Y)}$

### 1) Initial Assumptions

We consider the following assumption at the beginning of the proposed scheme PASKE-IoD, to verify the mutual authentication of PASKE-IoD.

- AS-1:  $M_{Di} | \equiv \#T_{am1}, \#T_{am3}, \#R_{se1}$
- AS-2:  $M_{Di} | \equiv TID_{EU_i}$
- AS-3:  $M_{Di} | \equiv TID_{CS}$
- AS-4:  $M_{Di} | \equiv TID_{RD_j}$
- AS-5:  $M_{Di} | \equiv AP$
- AS-6:  $M_{Di} | \equiv (M_{Di} \xrightarrow{K_{am3}} RD_j)$
- AS-7:  $M_{Di} | \equiv RD_j \Rightarrow (RD_j \xrightarrow{SK} M_{Di})$
- AS-8:  $M_{Di} | \equiv RD_j \Rightarrow RD_j | \sim P_9$
- AS-9:  $M_{Di} | \equiv (M_{Di} \xrightarrow{K_{am1}} CS)$
- AS-10:  $CS | \equiv \#T_{am1}, \#T_{am2}, \#R_{se1}, \#R_{se2}$
- AS-11:  $CS | \equiv TID_{EU_i}$
- AS-12:  $CS | \equiv TID_{CS}$
- AS-13:  $CS | \equiv TID_{RD_j}$
- AS-14:  $CS | \equiv AP$
- AS-15:  $CS | \equiv RD_j$
- AS-16:  $CS | \equiv ZID_K$
- AS-17:  $CS | \equiv (CS \xrightarrow{K_{am1}} M_{Di})$
- AS-18:  $CS | \equiv (CS \xrightarrow{K_{am2}} RD_j)$
- AS-19:  $RD_j | \equiv CS \Rightarrow CS | \sim P_2$
- AS-20:  $RD_j | \equiv \#T_{am2}, \#T_{am3}$
- AS-21:  $RD_j | \equiv \#R_{se2}, \#R_{se3}$
- AS-22:  $RD_j | \equiv ID_{RD_j}$
- AS-23:  $RD_j | \equiv TID_{CS}$
- AS-24:  $RD_j | \equiv TID_{RD_j}$
- AS-25:  $RD_j | \equiv ZID_k$
- AS-26:  $RD_j | \equiv (RD_j \xrightarrow{K_{am2}} CS)$
- AS-27:  $RD_j | \equiv (RD_j \xrightarrow{K_{am3}} M_{Di})$

### 2) Idealized Form

The idealized form of messages  $M_{am1}$ ,  $M_{am1}$ , and  $M_{am1}$  exchanged during the execution of PASKE-IoD protocol is given as follows.

- INF-1:  $\{T_{am1}, X_2, R_{se1}, TID_{RD_j}\}_{(M_{Di} \xrightarrow{K_{am1}} CS)}$
- INF-2:  $\{T_{am2}, R_{se2}, P_2\}_{(CS \xrightarrow{K_{am2}} RD_j)}$
- INF-3:  $\{T_{am3}, P_9, (RD_j \xrightarrow{SK} M_{Di})\}_{(RD_j \xrightarrow{K_{am3}} M_{Di})}$

### 3) Goals

We need to achieve the following goals, to ensure the mutual authentication between CS,  $RD_j$ , and  $M_{Di}$ .

- Goal-1:  $RD_j | \equiv (RD_j \xrightarrow{SK} M_{Di})$
- Goal-2:  $M_{Di} | \equiv (M_{Di} \xrightarrow{SK} M_{Di})$

### 4) Formal Verification

We verify the MA feature of PASKE-IoD formally by employing the fundamental BAN logic precept and deduction rules specified in Table 3 and Table 4, respectively. In addition, we consider the following assumptions. The detailed steps are provided below.

- FVri-1: From INF-1, by employing the AS-10, AS-17, and MMR, we get.

$$\frac{CS| \equiv (CS \xrightarrow{K_{am1}} M_{Di}), CS \triangleleft \{T_{am1}, X_2, R_{se1}, TID_{RD_j}\}}{CS| \equiv M_{Di} | \sim \{T_{am1}, X_2, R_{se1}, TID_{RD_j}\}} \xrightarrow{(M_{Di} \xrightarrow{K_{am1}} CS)}$$

- FVri-2: By using AS-10 and FR, we can obtain.

$$\frac{CS| \equiv \#T_{am1}}{CS| \equiv \#(T_{am1}, X_2, R_{se1}, TID_{RD_j})}$$

- FVri-3: From FVri-1, FVri-2, and by using NVR, we obtain.

$$\frac{CS| \equiv \#(T_{am1}, X_2, R_{se1}, TID_{RD_j}), CS \triangleleft \{T_{am1}, X_2, R_{se1}, TID_{RD_j}\}}{CS| \equiv M_{Di} | \equiv (T_{am1}, X_2, R_{se1}, TID_{RD_j})}$$

- FVri-4: Form INF-2, by using AS-19, AS-20, AS-21, AS-26, and MMR, we obtain.

$$\frac{RD_j| \equiv (RD_j \xrightarrow{K_{am2}} CS), RD_j \triangleleft \{T_{am2}, R_{se2}, P_2\}}{RD_j| \equiv CS | \sim \{T_{am2}, R_{se2}, P_2\}} \xrightarrow{(RD_j \xrightarrow{K_{am2}} CS)} \xrightarrow{(M_{Di} \xrightarrow{K_{am1}} CS)}$$

- FVri-5: By employing AS-20, AS-21, and by using FR, we get.

$$\frac{RD_j| \equiv \#T_{am1}}{RD_j| \equiv \#(T_{am2}, R_{se2}, P_2)}$$

- FVri-6: From FVri-4, FVri-5, and by using NVR, we achieve.

$$\frac{RD_j| \equiv \#(T_{am2}, R_{se2}, P_2), RD_j \triangleleft \{T_{am2}, R_{se2}, P_2\}}{RD_j| \equiv CS | \equiv (T_{am2}, R_{se2}, P_2)}$$

- FVri-7: From FVri-4, FVri-5, FVri-6, by applying AS-19, and by using NVR, we get  $RD_j | \equiv R_{se1} \oplus AP$ .
- FVri-8: Using FVri-7, and by using AS-19, AS-20, AS-21, AS-23, AS-24, and AS-26, Goal-1 can be achieved.

$$RD_j | \equiv (RD_j \xrightarrow{SK} M_{Di})$$

- FVri-9: From INF-3, by using AS-1, AS-6, AS-7, and AS-8, and by applying MMR, we get.

$$\frac{M_{Di}| \equiv (M_{Di} \xrightarrow{K_{am3}} RD_j), M_{Di} \triangleleft \{T_{am3}, P_9, (RD_j \xrightarrow{SK} M_{Di})\}}{M_{Di}| \equiv RD_j | \sim \{T_{am3}, P_9, (RD_j \xrightarrow{SK} M_{Di})\}} \xrightarrow{(M_{Di} \xrightarrow{K_{am3}} RD_j)}$$

- FVri-10: Using AS-1 and by using FR, we obtain.

$$\frac{M_{Di}| \equiv \#T_{am3}}{M_{Di}| \equiv \#(T_{am3}, P_9, (RD_j \xrightarrow{SK} M_{Di}))}$$

Claim	Status	Comments
PASKE_IoD_EU Secret H(TIDRD,XOR(Rse1,AP),XOR(Rse2,ZIDK,Rse3),Ta...	OK Verified	No attacks.
PASKE_IoD_EU2 Alive	OK	No attacks within bounds.
PASKE_IoD_EU3 Niagree	OK	No attacks within bounds.
PASKE_IoD_EU4 Nisynch	OK	No attacks within bounds.
CS PASKE_IoD_CS1 Secret TIDCS	OK Verified	No attacks.
PASKE_IoD_CS2 Alive	OK Verified	No attacks.
PASKE_IoD_CS3 Weakagree	OK Verified	No attacks.
PASKE_IoD_CS4 Niagree	OK Verified	No attacks.
PASKE_IoD_CS5 Nisynch	OK Verified	No attacks.
RD PASKE_IoD_RD1 Secret H(TIDRD,XOR(Rse1,AP),XOR(Rse2,ZIDK,Rse3),Ta...	OK Verified	No attacks.
PASKE_IoD_RD2 Alive	OK Verified	No attacks.
PASKE_IoD_RD3 Weakagree	OK Verified	No attacks.
PASKE_IoD_RD4 Niagree	OK Verified	No attacks.
PASKE_IoD_RD5 Nisynch	OK Verified	No attacks.

FIGURE 8: Simulation results of Scyther

- FVri-11: From FVri-9 and FVri-10, and by applying NVR, we get.

$$\frac{M_{Di} \models \#(T_{am3}, P_9, (RD_j \xrightarrow{SK} M_{Di})), M_{Di} \models (T_{am3}, P_9, (RD_j \xrightarrow{SK} M_{Di}))}{M_{Di} \models RD_j \models \#(T_{am3}, P_9, (RD_j \xrightarrow{SK} M_{Di}))}$$

- FVri-12: From FVri-9, FVri-10, FVri-11, and by applying AS-15, and NVR, we get  $RD_j \models P_9$ .
- FVri-13: Using FVri-12, by using AS-3, AS-4, AS-8, and AS-6, Goal-2 can be achieved.

$$M_{Di} \models (M_{Di} \xrightarrow{SK} RD_j)$$

From FVri-8 and FVri-13, it is clear that  $M_{Di}$  and  $RD_j$  authenticate with each other through CS.

### C. SECURITY ANALYSIS USING SCYTHYER

Scyther is a software tool used to validate the resiliency of the proposed security protocol against various security attacks. In addition, Scyther explicates the security vulnerability in the tested security protocol. Thus, we employed the Scyther tool to validate the security of the proposed ARAP-SG. Scyther uses the security protocol description language (SPDL) for the implementation of security protocol. SPDL is a python-like language. We coded ARAP-SG using the SPDL language.

In the SPDL script, we have defined three roles, such as  $EU_i$ , CS, and  $RD_j$ . Each role has some manually defined claims and some automatically generated roles. Manually specified claim for  $EU_i$  is  $claim(EU, Secret, SNK)$  and  $RD_j$  is  $claim(RD, Secret, SNK)$ , which are validated by the Scyther, as shown in Fig. 8. Moreover, the claims for the role  $EU_i$ , such as  $claim(EU, Alive)$ ,  $claim(EU, Nisynch)$ , and  $claim(EU, Niagree)$  are validated by Scyther. Similarly, same type of claims are also validated by Scyther for role  $RD_j$ , as demonstrated in Fig. 8.

## VII. PERFORMANCE EVALUATION

This section presents a detailed comparison among PASKE-IoD and other related schemes, such as Wazid *et al.* [14] and Jangirala *et al.* [13] in terms of Security Features (SF), storage, communication, and computational overheads.

TABLE 5: Setting Parameters

Cryptographic Primitive	Size (bits)
Hash Function (SHA-1)	160
ASCON-encryption	128
ASCON-Hash	256
Fuzzy Extractors	128
All identities	128
Timestamp	32
Nonce	128
Key	128
Random number	128

### A. PRACTICAL IMPLEMENTATION

The proposed PASKE-IoD is implemented using on the system with Intel(R) Dual-core(R) CPU @ 2.5GHz, Ubuntu (64 bits) operating system, and RAM 4 GB. PASKE-IoD is coded in python3 and socket programming with parameters setting as shown in Table 5. In addition, we utilized a python-based cryptographic "PyCryptodome" library for the implementation of Wazid *et al.* [14] and Jangirala *et al.* [13] ASKE schemes.

Although the proposed PASKE-IoD renders the protection against various security risks under TM presented in Section III-B. However, some covert attacks may occur during the execution of PASKE-IoD. Thus, to evaluate PASKE-IoD's performance, it is assumed that an adversary effectuates an attack during the ASKE phase execution of PASKE-IoD. We executed PASKE-IoD for 500 times and computed the total time for 500 runs as  $T_{500} = \sum_{x=1}^{x=500} (T_x)$ . If the numbers of successful attacks effectuated by an adversary to stop the execution of PASKE-IoD are increasing, PASKE-IoD takes a longer time to complete the ASKE phase. Total time required by PASKE-IoD to complete its execution under the success probability of an attack is computed as

$$T_{exe} = \frac{T_{500}}{500 \times (1 - \text{Attack Success Probability})}, \quad (1)$$

where  $T_{exe}$  denotes the time required during ASKE phase with unknown success probability. The average time required by PASKE-IoD 2.89 ms after 500 runs. Moreover, the average time required by Wazid *et al.* [14] and Jangirala *et al.* [13] is 3.95 ms and 3.86 ms, respectively, as shown in Fig. 9. Fig. 10 illustrates time consumption comparison during the ASKE phase of the proposed PASKE-IoD and related ASKE schemes.

#### 1) Computational Overhead Comparison

This section demonstrates the computation overhead required by PASKE-IoD and related ASKE mechanism. We denote the  $T_{ash}$ ,  $T_{ase}$ , and  $T_{sha}$  as the computation time of ASCON-Hash, ASCON encryption/decryption, and hash function, respectively. Computational cost of ASCON-Hash, ASCON encryption/decryption, and hash function is  $T_{ash} \approx 0.05\text{ms}$ ,  $T_{ase} \approx 0.04\text{ms}$ , and  $T_{sha} \approx 0.06\text{ms}$ , respectively. Total computational overhead of PASKE-IoD, the scheme of Wazid



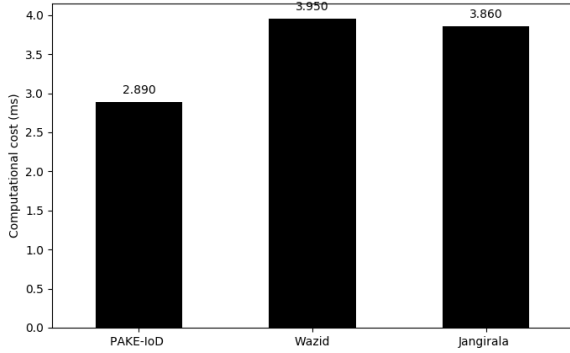


FIGURE 9: Average time required to complete the ASKE process

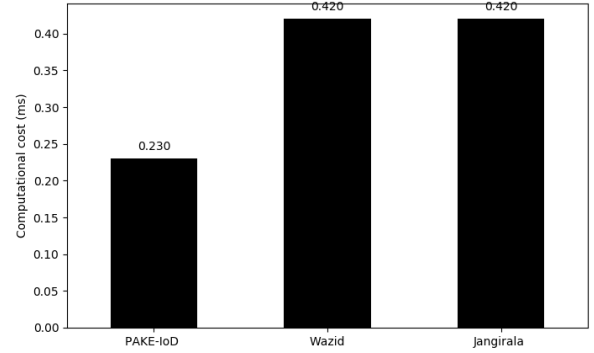


FIGURE 11: Computational overhead at  $RD_j$

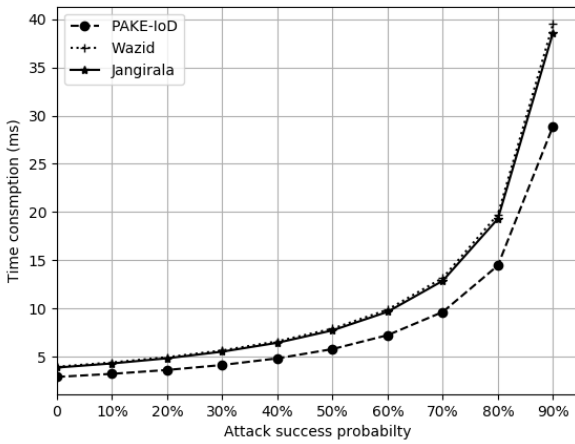


FIGURE 10: Time consumption with attack success probability

*et al.* [14], Jangirala *et al.* [13] is  $11T_{ash} + 6T_{ase} + T_{Bio} \approx 2.740$  ms,  $31T_{sha} + T_{Bio} \approx 3.810$  ms, and  $30T_{sha} + T_{Bio} \approx 3.750$  ms, respectively. The proposed PASKE-IoD requires less computation overhead as compared to other related ASKE schemes as shown in Table 6. Furthermore, PASKE-IoD, Wazid *et al.* [14], Jangirala *et al.* [13],  $7T_{sha} \approx 0.42$  ms,  $7T_{sha} \approx 0.42$ ms, and  $3T_{ash} + 2T_{ase} \approx 0.230$ ms require computational overhead at the drone/sensor side, respectively. Fig.11 shows that PASKE-IoD has less computation overhead at drone side than other related ASKE schemes, as shown in .

## B. SECURITY FEATURES COMPARISON

AA juxtaposition of security characteristics rendered by PASKE-IoD and other relevant ASKE schemes is presented in this section. It is evident from Table 7 that the scheme of Wazid *et al.* [14] is unprotected against  $SF2$ ,  $SF4$ , and  $SF7$  and Jangirala *et al.* [13] is insecure against  $SF2$ ,  $SF4$ , and  $SF7$ . However, PASKE-IoD renders better security features

as compared to the related ASKE schemes.

## C. COMMUNICATION OVERHEAD COMPARISON

This section deals with another significant performance parameter, namely communication overhead, to evaluate the efficiency of PASKE-IoD. To calculate the communication overhead of PASKE-IoD, we consider the parameters setting presented in Table 5. PASKE-IoD exchanged three messages during the ASKE process, such as  $M_{am1}:\{T_3, X_2, CT_6, CT_7, AuPa_{us}, R_{iv}\} = 608$  bits,  $M_{am2}:\{T_4, X_3, CT_8, AuPa_{si2}, R_{iv4}\} = 480$  bits, and  $M_{am3}:\{T_5, CT_9, AuPa_{du}, R_{iv5}\} = 352$  bits. Cumulative communication overhead while accomplishing the ASKE process of PASKE-IoD is  $\sum_{x=1}^3 |M_{aux}| = (608 + 480 + 352) = 1440$  bits. Contrarily, the scheme of Wazid *et al.* [14], Jangirala *et al.* [13], require 1696 bits and 1536 bits, respectively. The detailed description of the exchange messages of PASKE-IoD and related schemes while accomplishing the ASKE phase is given in Table 8, which clarifies that PASKE-IoD needs lower communication overhead in juxtaposition with the existing ASKE schemes.

## D. STORAGE OVERHEAD COMPARISON

The proposed PASKE-IoD requires to store  $\{P_2, P_3, AuPa_{reg}, gen(\cdot), rep(\cdot), r_p, t\} = 944$  bits,  $\{(TID_{EU_i}, AP), (ID_{RD_j}, TID_{RD_j}, ZID_k)\} = 640$  bits, and  $\{ID_{RD_j}, TID_{RD_j}, ZID_k\} = 384$  bits in the memory of  $EU_i$ ,  $CS$ , and  $RD_j$ , respectively. Total storage overhead of PASKE-IoD is 1968 bits. Furthermore, the scheme of that the scheme of Wazid *et al.* [14], Jangirala *et al.* [13], require storing 3242 bits, 2888 bits, respectively. Moreover, PASKE-IoD requires less storage cost as compared to related eminent schemes devised for the IoD environment.

## VIII. CONCLUSION

In this paper, we have designed a novel authentication scheme for the IoD environment called PASKE-IoD. The proposed PASKE-IoD is a three-factor ASKE mechanism, which enables the users to communicate securely, through

TABLE 6: Comparison of Computational Overhead

ASKE Scheme	$EU_i$ Side	$CS$ Side	$RD_j$ Side	Total Time
Wazid <i>et al.</i> [14]	$16T_{sha} + T_{Bio}$	$8T_{sha}$	$7T_{sha}$	$31T_{sha} + T_{Bio} \approx 3.810$ ms
Jangirala <i>et al.</i> [13]	$14T_{sha} + T_{Bio}$	$9T_{sha}$	$7T_{sha}$	$30T_{sha} + T_{Bio} \approx 3.750$ ms
PASKE-IoD	$6T_{ash} + 2T_{ase} + T_{Bio}$	$2T_{ash} + 2T_{ase}$	$3T_{ash} + 2T_{ase}$	$11T_{ash} + 7T_{ase} + T_{Bio} \approx 2.740$ ms

TABLE 7: Security feature comparison

$SF$	Wazid <i>et al.</i> [14]	Jangirala <i>et al.</i> [13]	PASKE-IoD
$SF_{\infty}$	✓	✓	✓
$SF_{\in}$	×	×	✓
$SF_{\exists}$	✓	✓	✓
$SF_{\Delta}$	×	×	✓
$SF_{\nabla}$	✓	✓	✓
$SF_{/}$	✓	✓	✓
$SF_{\downarrow}$	×	×	✓
$SF_{\forall}$	✓	✓	✓
$SF_{\exists}$	✓	✓	✓
$SF_{\infty/}$	✓	✓	✓
$SF_{\infty\infty}$	✓	✓	✓
$SF_{\infty\in}$	✓	✓	✓

: Note  $SF_{\infty}$ : Password/bio-metric update phase;  $SF_{\in}$ : Stolen smart device attack;  $SF_{\exists}$ : Password guessing attack;  $SF_{\Delta}$ : Privileged-insider attack;  $SF_{\nabla}$ : User anonymity/untraceability;  $SF_{/}$ : Impersonation attacks;  $SF_{\downarrow}$ : DoS attack;  $SF_{\forall}$ : Replay-attack;  $SF_{\exists}$ : MAMI attack;  $SF_{\infty/}$ : ESL attack;  $SF_{\infty\infty}$ : Sensor/drone capture attack;  $SF_{\infty\in}$ : Identity guessing attack; ✓: indicates feature is supported; ×: indicates not supported feature;

TABLE 8: Communication Overhead

Scheme	Messages exchanged during ASKE	Total (bits)
Wazid <i>et al.</i> [14]	$EU_i/U_i \xrightarrow{672} CS/GW \xrightarrow{512} RD_j/SN_j \xrightarrow{512} EU_i/U_i$	<b>1696</b>
Jangirala <i>et al.</i> [13]	$EU_i/U_i \xrightarrow{672} CS/CS \xrightarrow{512} RD_j/SN_j \xrightarrow{352} EU_i/U_i$	<b>1536</b>
PASKE-IoD	$EU_i/U_i \xrightarrow{608} MS/GW \xrightarrow{480} RD_j/SN_j \xrightarrow{352} EU_i/U_i$	<b>1440</b>

the public communication channel, with the network entities such as drones. To this end, PASKE-IoD utilizes LWC-based AE scheme known as ASCON along with hash function to accomplish the ASKE process. Meticulous formal and informal security analysis of PASKE-IoD and comprehensive comparative analysis show that PASKE-IoD is efficient than the existing security schemes devised for the IoD environment. Moreover, it is shown that PASKE-IoD provides better security and incurs less communication and computation overhead on the resource-limited devices in the IoD environment.

## REFERENCES

- [1] 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 Journal*, pp. 1–1, 2021.
- [2] M. Saqib, B. Jasra, and A. Hassan Moon, "A lightweight three factor authentication framework for IOT based critical applications," *Journal of King Saud University - Computer and Information Sciences*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1319157821002019>
- [3] A. U. Khan, G. Abbas, Z. H. Abbas, M. Tanveer, S. Ullah, and A. Naushad, "Hbip: A hybrid underlay-interweave mode CRN for the future 5G-based internet of things," *IEEE Access*, vol. 8, pp. 63 403–63 420, 2020.
- [4] G. Abbas, A. U. Khan, Z. H. Abbas, M. Bilal, K. S. Kwak, and H. Song, "FMCP: Flexible multiparameter-based channel prediction and ranking for cr-enabled massive iot," *IEEE Internet of Things Journal*, pp. 1–1, 2021.
- [5] A. U. Khan, G. Abbas, Z. H. Abbas, M. Waqas, S. Tu, and A. Naushad, "Service completion probability enhancement and fairness for sus using hybrid mode crns," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
- [6] C. Li and B. Palanisamy, "Privacy in Internet of Things: From principles to technologies," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 488–505, Feb 2019.
- [7] A. U. Khan, G. Abbas, Z. H. Abbas, W. U. Khan, and M. Waqas, "Spectrum utilization efficiency in CRNs with hybrid spectrum access and channel reservation: A comprehensive analysis under prioritized traffic," *Future Generation Computer Systems*, vol. 125, pp. 726–742, 2021.
- [8] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "UAV assistance paradigm: State-of-the-art in applications and challenges," *Journal of Network and Computer Applications*, p. 102706, 2020.
- [9] G. Tuna, B. Nefzi, and G. Conte, "Unmanned aerial vehicle-aided communications system for disaster recovery," *Journal of Network and Computer Applications*, vol. 41, pp. 27–36, 2014.
- [10] W. Zafar and B. M. Khan, "A reliable, delay bounded and less complex communication protocol for multicloud FANETs," *Digital Communications and Networks*, vol. 3, no. 1, pp. 30–38, 2017.
- [11] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered public safety IoT: A uav-assisted adaptive-learning approach towards energy efficiency," *Journal of Network and Computer Applications*, vol. 123, pp. 69–79, 2018.
- [12] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled

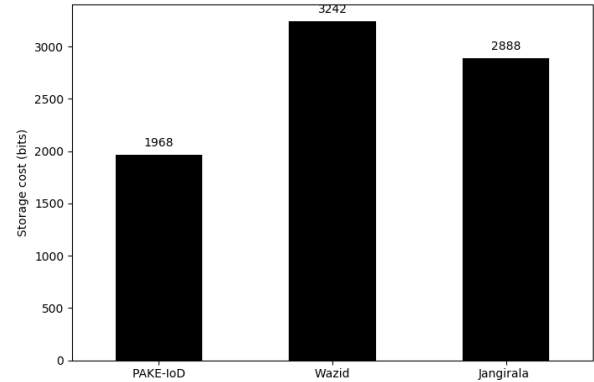


FIGURE 12: Storage cost comparison

- UAV: Network architecture and routing algorithm," *IEEE Internet of Things Journal*.
- [13] S. Jangirala, A. K. Das, N. Kumar, and J. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of Drones environment," *IEEE Transactions on Vehicular Technology*, 2019.
  - [14] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of Drones deployment," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3572–3584, 2018.
  - [15] F. Wu, L. Xu, S. Kumari, and X. Li, "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on Internet of Things security," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 1, pp. 101–116, 2017.
  - [16] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schl  fer, "ASCON v1.2," Submission to the CAESAR Competition, 2016.
  - [17] C. Lin, D. He, N. Kumar, K. R. Choo, A. Vinel, and X. Huang, "Security and Privacy for the Internet of Drones: Challenges and Solutions," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 64–69, Jan 2018.
  - [18] M. Wazid, A. K. Das, and J.-H. Lee, "Authentication protocols for the Internet of Drones: Taxonomy, analysis and future directions," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2018.
  - [19] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for Internet of Drones," *Computer Communications*, 2020.
  - [20] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the Internet of Drones applications," *Computer Communications*, 2020.
  - [21] S. H. Islam and G. Biswas, "Cryptanalysis and improvement of a password-based user authentication scheme for the integrated EPR information system," *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 2, pp. 211–221, 2015.
  - [22] Z.-Y. Wu, Y. Chung, F. Lai, and T.-S. Chen, "A password-based user authentication scheme for the integrated EPR information system," *Journal of medical systems*, vol. 36, no. 2, pp. 631–638, 2012.
  - [23] A. K. Das and A. Goswami, "A robust anonymous biometric-based remote user authentication scheme using smart cards," *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 2, pp. 193–210, 2015.
  - [24] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.
  - [25] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.
  - [26] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1900–1910, 2016.
  - [27] Y. Chen, J.-F. Mart  nez, P. Castillejo, and L. L  pez, "An anonymous authentication and key establish scheme for smart grid: FAuth," *Energies*, vol. 10, no. 9, p. 1354, 2017.
  - [28] A. Mohammadali, M. Sayad Haghighi, M. H. Tadayon, and A. Mohammadi-Nodooshan, "A novel identity-based key establishment method for advanced metering infrastructure in smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2834–2842, July 2018.
  - [29] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4349–4359, July 2019.
  - [30] M. Turkanovi  , B. Brumen, and M. H  lbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.
  - [31] M. S. Farash, M. Turkanovi  , S. Kumari, and M. H  lbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, 2016.
  - [32] M. F. Ayub, K. Mahmood, S. Kumari, A. K. Sangaiah et al., "Lightweight authentication protocol for e-health clouds in IoT based applications through 5G technology," *Digital Communications and Networks*, 2020.
  - [33] S. S. Sahoo, S. Mohanty, and B. Majhi, "An improved and secure two-factor dynamic ID based authenticated key agreement scheme for multi-server environment," *Wireless Personal Communications*, vol. 101, no. 3, pp. 1307–1333, 2018.
  - [34] S. Jangirala, S. Mukhopadhyay, and A. K. Das, "A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards," *Wireless Personal Communications*, vol. 95, no. 3, pp. 2735–2767, 2017.
  - [35] F. Wu, X. Li, L. Xu, S. Kumari, M. Karupiah, and J. Shen, "A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server," *Computers & Electrical Engineering*, vol. 63, pp. 168–181, 2017.
  - [36] W.-L. Tai, Y.-F. Chang, and W.-H. Li, "An iot notion-based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks," *Journal of Information Security and Applications*, vol. 34, pp. 133–141, 2017.
  - [37] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E.-J. Yoon, and K.-Y. Yoo, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.
  - [38] J. Jung, J. Moon, D. Lee, and D. Won, "Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks," *Sensors*, vol. 17, no. 3, p. 644, 2017.
  - [39] S. Shin and T. Kwon, "A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes," *Sensors*, vol. 19, no. 9, p. 2012, 2019.
  - [40] M. Tanveer, G. Abbas, and Z. H. Abbas, "LAS-6LE: A lightweight authentication scheme for 6lowpan environments," in 2020 14th International Conference on Open Source Systems and Technologies (ICOSST). IEEE, 2020, pp. 1–6.
  - [41] A. Gupta, M. Tripathi, T. J. Shaikh, and A. Sharma, "A lightweight anonymous user authentication and key establishment scheme for wearable devices," *Computer Networks*, vol. 149, pp. 29–42, 2019.
  - [42] M. Hassan, K. Mansoor, S. Tahir, and W. Iqbal, "Enhanced lightweight cloud-assisted mutual authentication scheme for wearable devices," in 2019 International Conference on Applied and Engineering Mathematics (ICAEM). IEEE, 2019, pp. 62–67.
  - [43] Z. Lv, "The security of Internet of Drones," *Computer Communications*, vol. 148, pp. 208–214, 2019.
  - [44] M. Tanveer, G. Abbas, Z. H. Abbas, M. Waqas, F. Muhammad, and S. Kim, "S6AE: Securing 6LoWPAN using authenticated encryption scheme," *Sensors*, vol. 20, no. 9, p. 2707, 2020.
  - [45] Y. Chen, L. L  pez, J.-F. Mart  nez, and P. Castillejo, "A lightweight privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: Lightpriauth," *Journal of Sensors*, vol. 2018, 2018.
  - [46] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.
  - [47] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
  - [48] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "LAKE-IoD: Lightweight authenticated key exchange protocol for the internet of drone environment," *IEEE Access*, vol. 8, pp. 155 645–155 659, 2020.
  - [49] M. Tanveer, A. U. Khan, N. Kumar, A. Naushad, and S. A. Chaudhry, "A robust access control protocol for the smart grid systems," *IEEE Internet of Things Journal*, pp. 1–1, 2021.
  - [50] M. Tanveer, G. Abbas, Z. H. Abbas, M. Bilal, A. Mukherjee, and K. S. Kwak, "LAKE-6SH: Lightweight user authenticated key exchange for 6LoWPAN-based smart homes," *IEEE Internet of Things Journal*, pp. 1–1, 2021.
  - [51] H. Gro  , E. Wenger, C. Dobraunig, and C. Ehrenh  fer, "Suit up!—made-to-measure hardware implementations of ascon," in 2015 Euromicro Conference on Digital System Design. IEEE, 2015, pp. 645–652.
  - [52] W. Diehl, A. Abdulgadir, F. Farahmand, J.-P. Kaps, and K. Gaj, "Comparison of cost of protection against differential power analysis of selected authenticated ciphers," *Cryptography*, vol. 2, no. 3, p. 26, 2018.
  - [53] A. Adomnicai, J. J. Fournier, and L. Masson, "Masking the lightweight authenticated ciphers ACORN and ASCON in software," *IACR Cryptology ePrint Archive*, vol. 2018, p. 708, 2018.
  - [54] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2884–2895, 2017.

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

...