

Received July 10, 2020, accepted August 16, 2020, date of publication August 25, 2020, date of current version September 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019367

# LAKE-IoD: Lightweight Authenticated Key Exchange Protocol for the Internet of Drone Environment

MUHAMMAD TANVEER<sup>1</sup>, AMJAD HUSSAIN ZAHID<sup>2</sup>, MUSHEER AHMAD<sup>3</sup>,  
ABDULLAH BAZ<sup>4</sup>, (Senior Member, IEEE), AND HOSAM ALHAKAMI<sup>5</sup>, (Member, IEEE)

<sup>1</sup>Telecommunications and Networking (TeleCoN) Research Laboratory, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan

<sup>2</sup>Department of Informatics and Systems, University of Management and Technology, Lahore 54700, Pakistan

<sup>3</sup>Department of Computer Engineering, Jamia Millia Islamia, New Delhi 110025, India

<sup>4</sup>Department of Computer Engineering, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia

<sup>5</sup>Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia

Corresponding author: Musheer Ahmad (musheer.cse@gmail.com)

This work was supported by the Deanship of Scientific Research at Umm Al-Qura University under Grant 19-COM-1-01-0015.

**ABSTRACT** A drone is an unmanned aerial vehicle, which is deployed in a particular Fly Zone (FZ), and used to collect crucial information from its surrounding environment to be transmitted to the server for further processing. Generally, a Mobile User (MU) is required to access the real-time information collected by the drone stationed in a specific FZ securely. Therefore, to ensure secure and reliable communications an Authenticated Key Exchange (AKE) protocol is imperative to the Internet of Drone (IoD) environment. An AKE scheme ensures only authentic MU to access IoD network resources. Upon successful authentication, MU and drone can set up a secret session key for secure communication in the future. This paper presents a novel Lightweight AKE Protocol for IoD Environment (LAKE-IoD), which first ensures the authenticity of MU and also renders session key establishment mechanism between MU and drone with the help of a server. LAKE-IoD is an AKE protocol, which is based on an authenticated encryption scheme AEGIS, hash function, and bit-wise XOR operation. Meticulous formal security verification by employing a software tool known as Scyther and informal security analysis demonstrates that LAKE-IoD is protected against different well-known active and passive security attacks. Additionally, Burrows-Abadi-Needham logic is applied to verify the logical completeness of LAKE-IoD. Furthermore, a comparison of LAKE-IoD with the related schemes shows that LAKE-IoD incurs less communication, computational and storage overhead.

**INDEX TERMS** Internet of Drone, authenticated key exchange, lightweight cryptography, unmanned aerial vehicles, security and privacy.

## I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) applications have observed outstanding growth in diverse fields along with the colossal demand of the Internet of Things (IoT). UAV can be employed in several applications, such as security surveillance system, traffic monitoring system in a smart city [1], disaster management, goods distribution, data collection, distributed processing, object detection and tracking, localization and mapping, environmental monitoring, health-care system, and rescue system [2]–[4]. Besides, the advancements presented by UAVs, these also have motivated the way

for the unification of UAVs, like smart drones within IoT domain. Drones are existing around for a long time; recently their uses within IoT realm have become a vital research topic [4].

Drones are a new form of the flying IoT objects acting as a sensing device. The synthesis of the smart drones and IoT domain is known as the Internet of Drone (IoD). IoD is a layered network control architecture devised especially to control the airspace by deploying drones technology and by establishing the coordination among the drones [5]. Fig. 1 shows a high-level architecture of the IoD system [6], [7], which is the interconnection of a Ground Station (GS), and smart drone deployed in the airspace. A drone is a fundamental component of IoD networks. The primary function

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Hui Yeh.

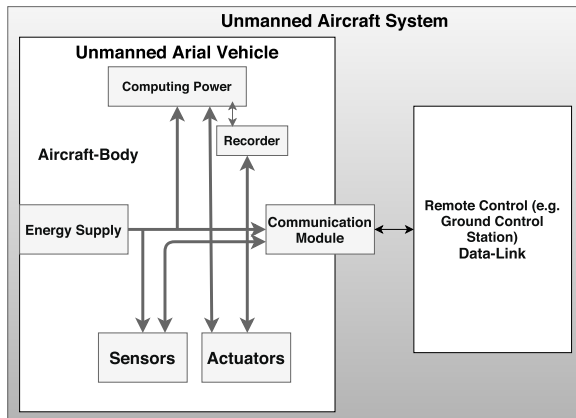


FIGURE 1. IoD system architecture.

of the drone is to collect the information from a specific Fly Zone (FZ) and transmit the collected information to GS. It is usually equipped with a communication module for transmission with GS, sensors used to collect the information, memory to store the data collected by the sensor, and also has computational capabilities and power resources [6], [7].

IoD is a new paradigm in wireless communication, which utilizes IoT technologies to accomplish its various critical operations. The cost-effective operational functionalities such as drone monitoring and control, trajectory planning, localization, authorization, and security and privacy are the prime requirements of IoD networks [8]–[10]. Irrespective of the advancements and plethora of solutions for drone communications, security and privacy in an IoD environment is still a major issue. IoD networks are resource constricted because a drone has limited computational, storage, and power resources. However, to enhance the lifetime of an IoD network, it is inevitable to devise a communication protocol that requires minimum resources [11]. Therefore, an efficient AKE protocol is necessary before utilizing a cryptographic encryption and decryption mechanism to ensure the secure and reliable transmission of information in an IoD network. This paper proposed a novel and lightweight AKE protocol for an IoD environment to ensure secure communication. The proposed scheme utilized a Lightweight Cryptographic (LWC) and Authenticated Encryption (AE) mechanism to ensure the confidentiality and integrity of the exchanged messages during the AKE phase. An AE encryption and decryption scheme can provide confidentiality and integrity simultaneously. LWC mechanism is suitable for the resource constricted environment.

## A. RESEARCH CONTRIBUTION

The summary of the main contributions are listed as follows:

- In this paper, we devised a novel and lightweight Authenticated Key Exchange (AKE) protocol named as Lightweight AKE protocol for IoD Environment (LAKE-IoD). The proposed AKE scheme utilizes an AE algorithm AEGIS, a hash function (SHA-256), and

exclusive-OR operation. LAKE-IoD renders password update phase, revocation or reissue phase, and dynamic drone deployment phase.

- Informal security analysis shows that LAKE-IoD is secure. Furthermore, LAKE-IoD is analyzed formally by employing Burrows-Abadi-Needham (BAN) logic and by using automatic verification software tool Scyther, which shows that the proposed LAKE-IoD is logically complete and secure against the various security attacks, such as Man-in-the-Middle (MITM) attack and replay attack.
- Finally, LAKE-IoD is compared with the related existing AKE schemes in terms of computational, communication, and storage overheads. The comparisons illustrate that the proposed scheme incurs fewer overheads than the existing schemes.

## B. PAPER ORGANIZATION

The rest of the paper is organized as follows. Section II reports different relevant security schemes for the IoD environment. System models are presented in Section III and preliminaries are discussed in Section IV. The details of the devised LAKE-IoD scheme are described in Section V. Security analysis of LAKE-IoD is provided in Section VI. A detailed comparison of LAKE-IoD with the recent related schemes is presented in Section VII. Finally, the paper is concluded in Section VIII.

## II. RELATED WORK

In this section, various related user authentication schemes are discussed. Lin *et al.* [11] present a review on the security and privacy issues in the Internet of Drone (IoD) and discuss various applications of IoD in the next generation of communication technology. Wazid *et al.* [12] present a survey on the security requirements in the IoD environment and also analyze various security protocols suitable for the IoD environment. Wazid *et al.* [6] proposed user Authentication and Key Establishment (AKE) scheme for the IoD environment. The proposed scheme is lightweight and insecure against various well-known attacks. Srinivas *et al.* [7] an AKE scheme for IoD, which is insecure against impersonation and privilege insider attacks. Srinivas *et al.*'s scheme also does not scale well as demonstrated in [13]. Wazid *et al.* [6] presented a security solution based on convolution neural networks for the IoD environment.

Farash and Attari [14] presented an Elliptic Curve Cryptography (ECC) based AKE scheme for Session Initiation Protocol (SIP). Thereafter, Lu *et al.* [15] demonstrated that scheme presented by Farash *et al.* is insecure against the off-line-password guessing attack, and they presented an AKE scheme based on ECC to remove the shortcomings of Farash *et al.* Zhang *et al.* [16] presented an authentication strategy for SIP. However, the proposed scheme is vulnerable to various security attacks such as privileged-insider attack and Denial-of-Service (DoS) attack as pointed out in [17]. Kumari *et al.* [18] proposed an AKE scheme for the Multi-Server Environment (MSE) based on ECC. Feng *et al.* [19]

pointed out the scheme devised by Kumari *et al.* is insecure against the server-impersonation Attack and presented an authentication scheme for the MSE environment. Ali and Pal [20] devised an AKE based on ECC for MSE and thereafter, Wang *et al.* [21] demonstrated that the scheme proposed by Ali *et al.* can not withstand privileged insider attack, user/server impersonation attack, DoS attack, and fails to provide forward-secrecy. Challa *et al.* [22] devised an ECC based AKE scheme, which is unprotected against various security attacks.

Amin *et al.* [23] constructed an AKE scheme for the cloud computing-based IoT environment, which is lightweight and suitable for resource constricted devices. However, the strategy presented by Amin *et al.* can not withstand the impersonation attack and privileged insider attack as demonstrated in [24]. Das *et al.* [24] proposed an AKE scheme for the IoT environment, which utilized lightweight hash function and FE technique for the bio-metric verification and cannot withstand traceability attack. Hussain and Chaudhry [25] pointed out that the scheme proposed by DAS *et al.* is vulnerable to various security attacks such as traceability attack, stolen-verifier attack, stolen/lost smart-device attack, and also does not render forward secrecy. Moreover, Challa *et al.* [26] presented an AKE scheme based on ECC, which is not suitable for resource-limited devices because of high computational overhead. Additionally, Jia *et al.* [27] highlighted that the scheme presented by Challa *et al.* is insecure against the impersonation attack and also does not ensure the untraceability property. Sharma and Kalra [28] proposed an AKE scheme for the cloud-based IoT environment. However, Sharma *et al.*'s scheme is vulnerable to the privileged-insider attack. Tanveer *et al.* [29] proposed an AKE scheme for 6LoWPAN resource-limited devices, which utilizes an authenticated encryption scheme known as ASCON and hash function. However, this scheme cannot resist the tractability attack.

The existing studies and their shortcomings motivate us to work on their weakness. For this aim, we target to construct a novel security scheme called LAKE-IoD. The LAKE-IoD utilizes a Secure Hash Algorithm (SHA-256), an AE scheme known as AEGIS, which is LWC mechanism, and an FE for the bio-metric verification of a user.

### III. SYSTEM MODELS

We consider the following two models in designing the proposed LAKE-IoD.

#### A. NETWORK MODEL

For the remote user authentication, this paper considers the network model as shown in Fig. 2. According to the network model, the airspace is divided into multiple FZs and many drones can be deployed in a specific FZ to monitor a particular environment (airspace). The drone deployed in a particular FZ collects data or information from the surrounding environment and transmits the gathered information to the

Management Server (MS), which is stationed at the Ground Station (GS). The MS is used to store the data collected by the drone. It also stores the secret information related to the user, drone, and airspace. An internal user usually sits in the Control Room (CR) to monitor an IoD environment. Promising technologies such as 4G/5G cellular networks are used to provide wireless connectivity in a specific FZ. There is a wired connectivity between the GS and wireless access point. Generally, the External User (EU) requires to collect the real-time information from the drone instead of using buffered (stored) information at the MS. For instance, an ambulance driver requires to know the traffic condition on the roads to reach the destination (for example, a hospital) as soon as possible. To access the real-time information from a particular drone, an EU must register himself/herself with the MS. An EU and a drone require to authenticate with each other through MS. After authentication, both the drone and EU can establish the session-key (secret-key) to secure future communication.

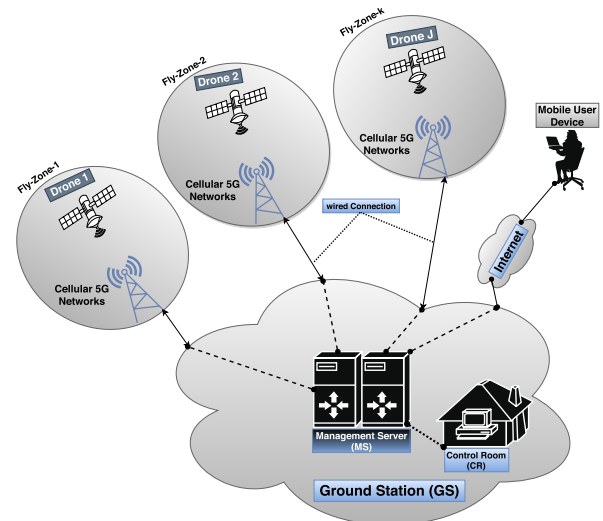


FIGURE 2. Internet of drone network model.

#### B. THREAT MODEL

We follow the widely accepted Dolev and Yao (DY) [30] threat model for the proposed scheme LAKE-IoD.

- 1) According to DY model, two entities (drone and EU) in the network can communicate using public (insecure) channel, and endpoint entities are trustworthy. Therefore, an adversary  $\mathcal{A}$  can capture or eavesdrop the communicated information or messages and can also forge or modify the exchanged messages.
- 2) The drone is usually deployed in a hostile or unattended environment. It is possible that  $\mathcal{A}$  can capture the drone and can also extract the secret information stored in the drone memory by employing the power analysis attack. However, the MS is considered as a secure entity in the proposed scheme and  $\mathcal{A}$  can not compromise the MS.

## IV. PRELIMINARIES

### A. FUZZY EXTRACTOR (FE)

In this paper, we employ Fuzzy Extractor (FE) [31] for the bio-metric ( $BU$ ) verification of the user. FE consists of the following two algorithms:

- 1)  $Gen(\cdot)$  : It is Bio-metric Key ( $BK$ ) generation algorithm. The  $Gen(\cdot)$  is a probabilistic algorithm. It takes  $BU$  as an input and generates  $BK \in [0, 1]^L$  and a Reproduction Parameter ( $RP$ ), where  $L$  is the length of the  $BK$ , that is,  $Gen(BU) = \{BK, RP\}$ .
- 2)  $Rep(\cdot)$  : It is a deterministic algorithm, which takes the noisy  $BU'$  and  $RP$  as input and recovers the  $BK$ . This implies that  $Rep(BU', RP) = BK$  provided the condition  $HD(BU, BU') \leq ET$  holds, where  $HD$  is the Hamming Distance between  $BU$  &  $BU'$ , and  $ET$  is the predefined Error Tolerance.

### B. AEGIS

AEGIS is a dedicated, lightweight, and high-performance Authenticated Encryption with Associative Data (AEAD) is an LWC mechanism. A brief description of AEGIS is given below:

- 1) The AEGIS was submitted to CAESAR competition and selected as the finalist candidates [32].
- 2) The AEGIS renders high security and speed of AEGIS is double as compared to Advance Encryption Standard (AES), i.e., ( $2 * AES$ ), 8 times of AES-CBC, and slightly faster than AES-CTR. The details of the recommended parameters for the AEGIS are given in [32], [33]. The AEGIS is appropriate for RFID tags and resource constricted IoT devices. It requires less computational resources as compared to AES and AES-GCM.
- 3) The AEGIS is an encryption algorithm, which generates output  $\langle CT, AUTH_{tag} \rangle$ , where  $CT$  is the ciphertext,  $AUTH_{tag}$  is the authentication parameter, by taking the plaintext  $PT$  as input. Logical operation of AEGIS can be expressed as  $CT = E_K\{\{IV, AD\}, PT\}$  and  $AUTH_{tag}$ , where  $K$  is the key,  $IV$  is the Initialization Vectors, and  $AD$  is the Associative Data.  $AUTH_{tag}$  is used to ensure the authenticity and integrity of  $AD$  and  $CT$ . In this paper, we employ AEGIS as the encryption/decryption algorithm.

## V. LAKE-IoD SCHEME

The proposed scheme LAKE-IoD comprises six phases, such as (i) Drone registration phase, (ii) User registration phase, (iii) User authentication and key exchange, (iv) Password and bio-metric update phase, (v) Revocation phase, (vi) Dynamic drone deployment phase. It is assumed that all the nodes in entities in an IoD environment are time-synchronized. Secure Hash Algorithm (SHA-256) utilized in the proposed scheme, which takes arbitrary input and generates a fixed-sized output. It is also assumed that all the entities in an IoD environment are time- synchronized. Table 1 presents the list of notations utilized in the proposed strategy. A detailed

TABLE 1. List of notations.

Symbol	Description
$ID_{MS}, SID_{MS}$	Real-identity and temporary-identity of the Management Server (MS), respectively
$ID_{MU_i}, SID_{MU_i}, SP_{MU_i}$	Real-identity, temporary-identity, and authentication parameter for the user, respectively
$MU_i, MD_i$	Mobile user, mobile device, respectively
$ID_{D_j}, SID_{D_j}, FID_k, SP_{D_j}$	Real-identity, temporary-identity, Fly Zone (FZ) identity, and secret parameter for the drone, respectively
$T_1, T_2, T_3$	Timestamps utilized during the authentication phase, respectively
$T_{ad}^1, T_{ad}^2, T_{ad}^3$ and $T^R$	Maximum time delay and message received time, respectively
$K_1, K_2, K_3$	Shared-secret Key used during the Authentication phase
$E_k(x), D_k(x)$	AEGIS encryption and decryption of message "x" using the secret-key "k", respectively.
$R_{MU_i}, R_{MS}, R_{D_j}$	Temporary random number used during authentication phase
$Gen(\cdot), Rep(\cdot)$	Fuzzy-extractor key generation and reproduction function, respectively
$BK_{MU_i}^{reg}, BK_{MU_i}^{LO}$	The bio-metric key during registration and login phase, respectively
$HW(x)$	Hamming Weight of x
$Rot(y, HW(x))$	Rotate left/right with respect to the $HW(x)$ .
$H(\cdot), \oplus, \parallel$	Cryptographic hash-function, bit-wise XOR, and concatenation, respectively

description of all phases of LAKE-IoD is presented as follows.

### A. DRONE REGISTRATION PHASE (DRP)

In this phase, the registration process for a drone  $D_j | j = 1, 2..N_j$  is discussed, where  $N_j$  is the total number of  $D_j$ . It is assumed that the airspace is divided into  $k$  number of Fly Zones (FZ). Each FZ is assigned with a unique Fly Zone Identity ( $FID_k | k = 1, 2..N_k$ ). It is necessary to register  $D_j$  with the Management Server (MS) before its deployment in a specific FZ. It is assumed that MS has a unique identity  $ID_{MS}$  and temporary identity  $SID_{MS}$ , which are known only to MS. The detailed process of  $D_j$  registration is given below.

- 1) Step DRP-1: MS assigns a unique identity  $ID_{D_j}$  and Fly Zone Identity  $FID_k$  to  $D_j$  before its deployment in a specif FZ.
- 2) Step DRP-2: MS computes the temporary identity  $SID_{D_j}$  for  $D_j$  by computing  $Q = H(ID_{MS} \parallel SID_{MS} \parallel R_{D_j})$ , where  $R_{D_j}$  is a random number of 128 bits,  $SID_{D_j} = Q_1 \oplus Q_2$ , where  $Q_1$  and  $Q_2$  are the two equal chunks (128-bits) of  $Q$ .
- 3) Step DRP-3: MS computes secret parameter  $SP_{D_j}$  for the drone  $D_j$  by computing  $SP_{D_j} = R_{D_j} \oplus Q_1$ .
- 4) Step DRP-4: Finally, MS stores the parameters  $\{ID_{D_j}, SID_{D_j}, SP_{D_j}, FID_k\}$  in the memory of  $D_j$ . MS also stores these credentials in its memory.

### B. MOBILE-USER REGISTRATION PHASE (MURP)

An  $MU_i$  requires to register with MS in IoD environment before accessing the services provided by the Zone Service Provider (ZSP) (ZSP is an organization, which monitors and maintains an IoD network). After successful registration, ZSP

allow an  $MU_i$  to acquire the real-time vital information from a specific drone deployed in a particular FZ. The details of an  $MU_i$  registration process are given as follows.

- 1) Step MURP-1:  $MU_i$  picks his/her unique identity  $ID_{MU_i}$ , and  $PW_{MU_i}$ .
- 2) Step MURP-2:  $MD_i$  selects a random number  $R_{IN}$  and computes  $RID_{MU_i} = H(ID_{MU_i} \parallel R_{IN})$  and sends  $RID_{MU_i}$  to MS through a secure channel.
- 3) Step MURP-3: After obtaining  $RID_{MU_i}$  from  $MD_i$ , MS selects timestamp  $T_{MS}$  of size 32-bits, picks master-key  $MSK_{MU_i}$  for  $MU_i$  and computes  $S_{ms} = H(T_{MS} \parallel RID_{MU_i} \parallel SID_{MS})$ ,  $SID_{MU_i} = S_{ms}^1 \oplus S_{ms}^2$ , where  $S_{ms}^1$  and  $S_{ms}^2$  are the two equal 128-bits chunks of  $S_{ms}$ . MS calculates the Secret Parameter ( $SP_{MU_i}$ ) for  $MU_i$  by computing  $AP = (ID_{MS} \parallel MSK_{MU_i} \parallel S_{ms}^2)$ ,  $SP_{MU_i} = AP_1 \oplus AP_2$ , where  $AP_1$  and  $AP_2$  are the two equal parts of AP.
- 4) Step MURP-4: MS sends tuple  $TU = \{SID_{MU_i}, SP_{MU_i}, SID_{MS}, SID_{D_j}\}$  to  $MD_i$  through a reliable channel.
- 5) Step MURP-5:  $MD_i$  receives  $TU$  from MS,  $MD_i$  calculates  $A_{11} = SID_{MS} \oplus SID_{MU_i} \oplus SP_{MU_i} \oplus SID_{D_j}$ ,  $(BK_{MU_i}^{reg}, RP^{reg}) = Gen(BU_{MU_i})$ ,  $A_2 = SID_{MU_i} \parallel SID_{MS} \oplus H(ID_{MU_i} \parallel BK_{MU_i}^{reg} \parallel PW_{MU_i})$ ,  $A_3 = SID_{D_j} \parallel SP_{MU_i} \oplus H(BK_{MU_i}^{reg} \parallel ID_{MU_i} \parallel PW_{MU_i})$ ,  $AUTH_{reg} = H(ID_{MU_i} \parallel BK_{MU_i}^{reg} \parallel PW_{MU_i} \parallel A_{11})$ , and deletes  $A_{11}$  from the memory.
- 6) Step MURP-6: Finally,  $MD_i$  stores the parameters  $\{A_2, A_3, AUTH_{reg}, Gen(), Rep(), RP^{reg}, ET\}$  in its memory.

### C. USER LOGIN & AUTHENTICATION PHASE (ULP)

This phase explains the AKE process between an  $MD_i$  and a  $D_j$  with the help of MS. In this phase all the entities utilize the public communication channel for AKE. Upon receiving the login request from  $MD_i$ , MS validates the validity of the receives message and also verifies the existence of an  $MD_i$  in its database. An  $MD_i$  has a list of  $D_j$  form which he/she is allowed to acquire the real-time data collected by  $D_j$ . The succeeding steps describe the details of ULP.

- 1) Step ULP-1: An  $MU_i$  inputs his/her real identity  $ID_{MU_i}$  and  $PW_{MU_i}$  on the available device login interface. He/she also imprints his/her bio-metric information  $BU_{MU_i}$  on the sensor available on the  $MD_i$ .
- 2) Step ULP-2: An  $MD_i$  calculates  $BK_{MU_i}^{LO} = Rep(BU_{MU_i}^{LO}, RP^{reg})$  provided the condition  $HD(BU_{MU_i}^{LO}, BU_{MU_i}^{reg}) \leq ET$  holds, where HD is the hamming distance between  $BU_{MU_i}^{LO}$  and  $BU_{MU_i}^{reg}$ . Additionally, an  $MU_i$  also computes  $(SID_{MU_i} \parallel SID_{MS}) = A_2 \oplus H(ID_{MU_i} \parallel BK_{MU_i}^{LO} \parallel PW_{MU_i})$  and  $(SID_{D_j} \parallel SP_{MU_i}) = A_3 \oplus H(BK_{MU_i}^{LO} \parallel ID_{MU_i} \parallel PW_{MU_i})$ . Furthermore, an  $MD_i$  calculates  $A_1^{LO} = SID_{MS} \oplus SID_{MU_i} \oplus SP_{MU_i} \oplus SID_{D_j}$ . To verify the login request, an  $MD_i$  computes  $AUTH_{LO} = H(ID_{MU_i} \parallel BK_{MU_i}^{LO} \parallel PW_{MU_i} \parallel A_{11}^{LO})$  and checks the condition  $AUTH_{LO} = AUTH_{reg}$ . If the condition holds,  $MD_i$  continues the authentication process. Otherwise,  $MD_i$  aborts the authentication process promptly.

- 3) Step ULP-3: After successful verification of the login parameters of  $MU_i$ ,  $MD_i$  picks timestamp  $T_1$  of 32 bit size, and random number  $R_{MU_i}$  of 128 bits.  $MD_i$  derives  $P_1 = R_{MU_i}$  and  $P_2 = SID_{D_j}$ , where  $P_1$  and  $P_2$  are the plaintext. Moreover,  $MD_i$  calculates  $A_1 = H(SID_{MU_i} \parallel SID_{MS} \parallel SP_{MU_i})$ ,  $K_1 = A_2 \oplus A_3$ , where  $A_2$  and  $A_3$  are the two equal 128-bits chunks of  $A_1$ . Furthermore,  $MD_i$  computes  $G = HW(SID_{MS})$ , where  $HW$  is the Hamming Weight,  $Z = (T_1 \parallel 0^l \parallel T_1 \parallel 1^l)$ , where  $l = 32$ ,  $ZZ = Rot(Z_1, G)$ ,  $A_4 = SID_{MU_i} \oplus ZZ \oplus SID_{MS}$ , and  $AD_1 = A_4$ . The AEGIS takes two parameter as input, which are secret key  $K_1$  of size 128-bits and Initialization Vector ( $IV$ ) of 128-bits. The  $IV$  is a public parameter. It is required to transmit  $IV$  with the communicated message. In the proposed scheme,  $IV$  can be computed as  $IV_1 = A_2 \oplus AD_1$ , which can be derived at the receiver side in the same way. Therefore, in the proposed scheme  $IV$  will not be transmitted with the exchanged messages to decrease the communication overhead. Furthermore,  $MD_i$  computes  $(C_1^{mu}, C_2^{mu}) = E_{K_1}(\{IV_1, AD_1\}, P_1, P_2)$ , and  $AUTH_{tag1}$  by using AEGIS encryption algorithm, where  $AD_1$  is the associative data. Finally,  $MD_i$  constructs the message  $M_1 : \langle T_1, A_4, C_1^{mu}, C_2^{mu}, AUTH_{tag1} \rangle$  and forwards  $M_1$  to MS through a public channel.
- 4) Step ULP-4: Upon receiving  $M_1$ , MS checks freshness of  $M_1$  by checking the condition  $T_{ad}^1 \geq |T^R - T_1|$ . If the condition holds, the received  $M_1$  is considered to be a fresh message. Otherwise, MS rejects  $M_1$ . MS computes  $G_1 = HW(SID_{MS})$ ,  $Z_1 = (T_1 \parallel 0^l \parallel T_1 \parallel 1^l)$ ,  $ZZ_1 = Rot(Z_1, G_1)$ ,  $SID_{MU_i} = A_4 \oplus ZZ_1 \oplus SID_{MS}$ , and verifies if  $SID_{MU_i}$  exists in its database or not. If  $SID_{MU_i}$  is found, MS retrieves  $SP_{MU_i}$  related to  $SID_{MU_i}$  from the database and continues the AKE process. Otherwise, MS aborts the AKE process promptly. Furthermore, MS computes  $A_5 = H(SID_{MU_i} \parallel SID_{MS} \parallel SP_{MU_i})$  and  $K_1 = A_6 \oplus A_7$ . MS picks  $A_4$  from the received  $M_1$  and calculates  $IV_2 = A_6 \oplus A_4$ , and  $AD_2 = A_4$ . Additionally, MS computes  $P_1, P_2 = D_{K_1}(\{IV_2, AD_2\}, C_1^{mu}, C_2^{mu})$ , and  $AUTH_{tag2}$  by using AEGIS decryption algorithm. To verify the authenticity of the received  $M_1$ , MS checks the condition  $AUTH_{tag1} = AUTH_{tag2}$ . If the condition does not hold, MS aborts the AKE process promptly. Otherwise, MS considers  $M_1$  as a valid message and continues the AKE process.
- 5) Step ULP-5: Moreover, MS picks  $T_2, R_{MS}$ , and computes  $P_3 = R_{MS} \oplus R_{MU_i}$ ,  $G_2 = HW(SID_{D_j})$ ,  $Z_2 = (T_2 \parallel 0^l \parallel T_2 \parallel 1^l)$ ,  $ZZ_2 = Rot(Z_2, G_2)$ ,  $A_9 = H(ID_{D_j} \parallel FID_k \parallel SP_{D_j})$ ,  $K_2 = A_{10} \oplus A_{11}$ ,  $A_{12} = SID_{MU_i} \oplus ZZ_2 \oplus SID_{D_j}$ ,  $AD_3 = A_{12}$ , and  $IV_3 = A_{10} \oplus AD_3$ . Additionally, MS calculates  $C_1^{ms} = E_{K_2}(\{IV_3, AD_3\}, P_3)$ , and  $AUTH_{tag3}$  by employing the encryption algorithm. Finally, MS constructs the message  $M_2 : \langle T_2, A_{12}, C_1^{ms}, AUTH_{tag3} \rangle$  and dispatches  $M_2$  to  $D_j$  through a public channel.

Mobile-User/Mobile-Device $MU_i/MD_i$	Management Server MS	Drone $D_j$
$\langle A_2, A_3, AUTH_{reg}, Gen(), Rep(), R^{pres}, ET \rangle$	$\langle (SID_{MU_i}, SP_{MU_i}), (ID_{D_j}, SID_{D_j}, SP_{D_j}, FID_k) \rangle$	$\langle (SID_{D_j}, ID_{D_j}, SP_{D_j}, FID_k) \rangle$
<p>Inputs <math>ID_{MU_i}, PW_{MU_i}</math> and imprints <math>BU_{MU_i}^{LO}</math>,  computes <math>BK_{MU_i}^{LO} = Rep(BU_{MU_i}^{LO}, R^{pres})</math>,  <math>(SID_{MU_i} \parallel SID_{MS}) = A_2 \oplus H(ID_{MU_i} \parallel BK_{MU_i}^{LO} \parallel PW_{MU_i})</math>,  <math>(SID_{D_j} \parallel SP_{MU_i}) = A_3 \oplus H(BK_{MU_i}^{LO} \parallel ID_{MU_i} \parallel PW_{MU_i})</math>,  <math>A_1^{LO} = SID_{MS} \oplus SID_{MU_i} \oplus SP_{MU_i} \oplus SID_{D_j}</math>,  <math>AUTH_{LO} = H(ID_{MU_i} \parallel BK_{MU_i}^{LO} \parallel PW_{MU_i} \parallel A_1^{LO})</math>,  Checks if <math>AUTH_{LO} = AUTH_{reg}^?</math>, if so,  picks timestamp <math>T_1</math> and random number <math>R_{MU_i}</math>,  derives <math>P_1 = R_{MU_i}, P_2 = SID_{D_j}</math>,  computes <math>A_1 = H(SID_{MU_i} \parallel SID_{MS} \parallel SP_{MU_i})</math>,  <math>K_1 = A_2 \oplus A_3, G = HW(SID_{MS})</math>,  <math>Z = (T_1 \parallel 0^l \parallel T_1 \parallel 1^l), ZZ = Rot(Z, G)</math>,  <math>A_4 = SID_{MU_i} \oplus ZZ \oplus SID_{MS}</math>,  <math>AD_1 = A_4, IV_1 = A_2 \oplus AD_1</math>  <math>(C_1^{ms}, C_2^{ms}) = E_{K_1}\{IV_1, AD_1\}, P_1, P_2</math>, and <math>AUTH_{tag1}</math>.</p> <p><math>M_1: \{T_1, A_4, C_1^{ms}, C_2^{ms}, AUTH_{tag1}\}</math>  via an open channel to MS</p>	<p>Checks if <math>T_{ad}^1 \geq  T^R - T_1 </math>. If so,  picks <math>T_1</math> from <math>M_1</math>, and computes <math>G_1 = HW(SID_{MS})</math>,  <math>Z_1 = (T_1 \parallel 0^l \parallel T_1 \parallel 1^l), ZZ_1 = Rot(Z_1, G_1)</math>,  <math>SID_{MU_i} = A_4 \oplus ZZ_1 \oplus SID_{MS}</math>,  checks if <math>SID_{MU_i}</math> exists in the database, if yes,  retrieves <math>SP_{MU_i}</math> related to <math>SID_{MU_i}</math>,  computes <math>A_5 = H(SID_{MU_i} \parallel SID_{MS} \parallel SP_{MU_i})</math>,  <math>K_1 = A_6 \oplus A_7</math>, and picks <math>A_4</math> from <math>M_1</math>,  computes <math>AD_2 = A_4, IV_2 = A_6 \oplus A_4</math>,  <math>P_1, P_2 = D_{K_1}\{IV_2, AD_2\}, C_1^{ms}, C_2^{ms}</math>, and <math>AUTH_{tag2}</math>,  checks if <math>AUTH_{tag1} = AUTH_{tag2}^?</math>, if so,  picks <math>T_2, R_{MS}</math>, and computes <math>P_3 = R_{MS} \oplus R_{MU_i}</math>,  <math>G_2 = HW(SID_{D_j}), Z_2 = (T_2 \parallel 0^l \parallel T_2 \parallel 1^l)</math>,  <math>ZZ_2 = Rot(Z_2, G_2), A_9 = H(ID_{D_j} \parallel FID_k \parallel SP_{D_j})</math>,  <math>K_2 = A_{10} \oplus A_{11}, A_{12} = SID_{MU_i} \oplus ZZ_2 \oplus SID_{D_j}</math>,  calculates <math>AD_3 = A_{12}, IV_3 = A_{10} \oplus AD_3</math>,  <math>C_1^{ms} = E_{K_2}\{IV_3, AD_3\}, P_3</math>, and <math>AUTH_{tag3}</math></p> <p><math>M_2: \{T_2, A_{12}, C_1^{ms}, AUTH_{tag3}\}</math>  via an open channel to <math>D_j</math></p>	<p>Checks if <math>T_{ad}^2 \geq  T^R - T_2 </math>, if so,  computes <math>G_3 = HW(SID_{D_j})</math>,  <math>Z_3 = (T_2 \parallel 0^l \parallel T_2 \parallel 1^l), ZZ_3 = Rot(Z_3, G_3)</math>,  <math>A_{13} = H(ID_{D_j} \parallel FID_k \parallel SP_{D_j})</math>,  <math>K_2 = A_{14} \oplus A_{15}, SID_{MU_i} = A_{12} \oplus Z_3 \oplus SID_{D_j}</math>,  picks <math>A_{12}</math> from <math>M_2</math>,  computes <math>AD_4 = A_{12}, IV_4 = A_{14} \oplus AD_4</math>,  <math>P_3 = D_{K_2}\{IV_4, AD_4\}, C_1^{ms}</math>, and <math>AUTH_{tag4}</math>,  checks if <math>AUTH_{tag4} = AUTH_{tag3}^?</math>, if so,  retrieves <math>P_3 = R_{MU_i} \oplus R_{MS}</math> from <math>C_1^{ms}</math>,  picks <math>T_3, R_{D_j}</math>, and computes <math>P_4 = R_{D_j} \oplus FID_k \oplus P_3</math>,  <math>A_{16} = H(SID_{D_j} \parallel SID_{MU_i} \parallel T_3), K_3 = A_{17} \oplus A_{18}</math>,  computes <math>SK_X = H(SID_{D_j} \parallel P_4 \parallel SID_{MU_i} \parallel T_3)</math>,  <math>A_{19} = SK_X^1 \oplus SK_X^2 \oplus SID_{MU_i}</math>,  <math>AD_5 = A_{19}, IV_5 = A_{17} \oplus AD_5</math>,  <math>C_1^d = E_{K_3}\{IV_5, AD_5\}, P_4</math>, and <math>AUTH_{tag5}</math>,</p> <p><math>M_3: \{T_3, A_{19}, C_1^d, AUTH_{tag5}\}</math>  via an open channel to <math>MD_i</math></p>
<p>Checks if <math>T_{ad}^3 \geq  T^R - T_3 </math>, if so,  computes <math>A_{20} = H(SID_{D_j} \parallel SID_{MU_i} \parallel T_3)</math>,  <math>K_3 = A_{21} \oplus A_{22}</math>,  picks <math>A_{19}</math> from <math>M_3</math> and computes <math>AD_6 = A_{19}</math>,  <math>IV_6 = A_{21} \oplus AD_6, P_4 = D_{K_3}\{IV_6, AD_6\}, C_1^d</math>, and  <math>AUTH_{tag6}</math>,  checks if <math>AUTH_{tag6} = AUTH_{tag5}^?</math>, if so,  retrieves <math>P_4 = R_{D_j} \oplus FID_k \oplus P_3</math>,  computes <math>SK_Y = H(SID_{D_j} \parallel P_4 \parallel SID_{MU_i} \parallel T_3)</math>,  <math>A_{23} = SK_Y^1 \oplus SK_Y^2 \oplus SID_{MU_i}</math>,  checks if <math>A_{19} = A_{23}^?</math>, if so, <math>SK_X (= SK_Y)</math>.</p> <p><math>SK_X (= SK_Y) = H(SID_{D_j} \parallel R_{D_j} \oplus FID_k \oplus P_3 \parallel SID_{MU_i} \parallel T_3)</math></p>		

FIGURE 3. LAKE-IoD login and authentication phase.

- 6) Step ULP-6: After receiving  $M_2$  from MS,  $D_j$  verifies the condition  $T_{ad}^2 \geq |T^R - T_2|$ . If the condition does not hold,  $M_2$  is considered to be outdated message. Otherwise,  $D_j$  calculates  $G_3 = HW(SID_{D_j}), Z_3 = (T_2 \parallel 0^l \parallel T_2 \parallel 1^l), ZZ_3 = Rot(Z_3, G_3), A_{13} = H(ID_{D_j} \parallel FID_k \parallel SP_{D_j}), K_2 = A_{14} \oplus A_{15}$ , and  $SID_{MU_i} = A_{12} \oplus ZZ_3 \oplus SID_{D_j}$ .  $D_j$  picks the  $A_{12}$  from the received message  $M_2$  and computes  $AD_4 = A_{12}$ , and  $IV_4 = A_{14} \oplus AD_4$ . Additionally,  $D_j$  calculates  $P_3 = D_{K_2}\{\{IV_4, AD_4\}, C_1^{ms}\}$ , and  $AUTH_{tag4}$  by using AEGIS decryption algorithm. To establish the authenticity of the received message  $M_2$ , MS validates the condition  $AUTH_{tag4} = AUTH_{tag3}$ . If the condition does not hold,  $D_j$  rejects the message and aborts the AKE process. Otherwise,  $D_j$  retrieves  $P_3 = R_{MU_i} \oplus R_{MS}$  from  $C_1^{ms}$ , which is received with  $M_2$ .
- 7) Step ULP-7:  $D_j$  picks  $T_3, R_{D_j}$ , and computes  $P_4 = R_{D_j} \oplus FID_k \oplus P_3, A_{16} = H(SID_{D_j} \parallel SID_{MU_i} \parallel T_3)$ , and  $K_3 = A_{17} \oplus A_{18}$ . To secure the communication between  $D_j$  and  $MD_i$ ,  $D_j$  computes the session-key by

- computing  $SK_X = H(SID_{D_j} \parallel P_4 \parallel SID_{MU_i} \parallel T_3)$ . Moreover,  $D_j$  calculates  $A_{19} = SK_X^1 \oplus SK_X^2 \oplus SID_{MU_i}$ ,  $AD_5 = A_{19}$ , and  $IV_5 = A_{17} \oplus AD_5$ . Additionally,  $D_j$  calculates  $C_1^d = E_{K_3}\{\{IV_5, AD_5\}, P_4\}$ , and  $AUTH_{tag5}$  by using AEGIS. Finally,  $D_j$  constructs the message  $M_3 : \{T_3, A_{19}, C_1^d, AUTH_{tag5}\}$  and sends  $M_3$  to  $MD_i$ .
- 8) Step ULP-8: After receiving the message  $M_3$  from  $D_j$ ,  $MD_i$  checks the freshness of  $M_3$  by checking the condition  $T_{ad}^3 \geq |T^R - T_3|$ . If the condition holds,  $MD_i$  computes  $A_{20} = H(SID_{D_j} \parallel SID_{MU_i} \parallel T_3)$  and  $K_3 = A_{21} \oplus A_{22}$ .  $MD_i$  picks  $A_{19}$  from the received message  $M_3$  and calculates  $AD_6 = A_{19}$  and  $IV_6 = A_{21} \oplus AD_6$ . Additionally,  $MU_i$  computes  $P_4 = D_{K_3}\{\{IV_6, AD_6\}, C_1^d\}$ , and  $AUTH_{tag6}$  by using AEGIS decryption process.  $MD_i$  verifies if the condition  $AUTH_{tag6} = AUTH_{tag5}$  holds. If so,  $MD_i$  retrieves  $P_4 = R_{D_j} \oplus FID_k \oplus P_3$  from  $C_1^d$ . To secure the communication between  $MD_i$  and  $D_j$ ,  $MD_i$  computes the session-key by computing  $SK_Y = H(SID_{D_j} \parallel P_4 \parallel SID_{MU_i} \parallel T_3)$ . Finally,  $MD_i$  computes  $A_{23} = SK_Y^1 \oplus SK_Y^2 \oplus SID_{MU_i}$  and checks the

condition  $A_{19} = A_{23}$ . If the condition holds, it indicates that  $SK_X$  computed at  $D_j$  and  $SK_Y$  computed at  $MD_i$  are same.

The summary of AKE process is shown in the Fig. 3.

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

A legitimate registered  $MU_i$  with an  $MD_i$  is required to execute the following steps to update the password  $PW_{MU_i}$  and  $BU_{MU_i}$  information of  $MU_i$ .  $BU_{MU_i}$  of  $MU_i$  remains unchanged and old bio-metric information is considered as new or fresh. However, to strengthen the security of the system, it is imperative to update  $MU_i$ 's password frequently. In this paper, we insinuate updating both  $PW_{MU_i}$  and  $BU_{MU_i}$  of  $MU_i$ .

- 1) Step PUP-1:  $MU_i$  enters his/her  $ID_{MU_i}$ , old password  $PW_{MU_i}^o$ , imprints old  $BU_{MU_i}^o$ , and calculates the following operations, such as computes  $BK_{MU_i}^o = Rep(BU_{MU_i}^o, RP^{reg})$  to reproduce the bio-metric key,  $(SID_{MU_i} \parallel SID_{MS}) = A_2 \oplus H(ID_{MU_i} \parallel BK_{MU_i}^o \parallel PW_{MU_i}^o)$ ,  $(SID_{D_j} \parallel SP_{MU_i}) = A_3 \oplus H(BK_{MU_i}^o \parallel ID_{MU_i} \parallel PW_{MU_i}^o)$ , and retrieves  $SID_{MU_i}$ ,  $SID_{MS}$ ,  $SID_{D_j}$ , and  $SP_{MU_i}$ . Furthermore,  $MD_i$  computes  $A_{11}^o = SID_{MS} \oplus SID_{MU_i} \oplus SP_{MU_i} \oplus SID_{D_j}$ , and  $AUTH_o = H(ID_{MU_i} \parallel BK_{MU_i}^o \parallel PW_{MU_i}^o \parallel A_{11}^o)$ .  $MU_i$  checks the condition  $AUTH_o = AUTH_{reg}$ . If the condition holds, the  $MU_i$  continues the PUP. Otherwise,  $MU_i$  terminates the PUP promptly.
- 2) Step PUP-2:  $MU_i$  enters his/her  $ID_{MU_i}$ , new password  $PW_{MU_i}^{ne}$ , imprints new/fresh bio-metric information  $BU_{MU_i}^{ne}$  (both the old and new bio-metric information are same), and calculates the following operations, such as computes  $BK_{MU_i}^{ne} = Rep(BU_{MU_i}^{ne}, RP^{ne})$  to reproduce the bio-metric key,  $A_2^{ne} = (SID_{MU_i} \parallel SID_{MS}) \oplus H(ID_{MU_i} \parallel BK_{MU_i}^{ne} \parallel PW_{MU_i}^{ne})$ ,  $A_3^{ne} = (SID_{D_j} \parallel SP_{MU_i}) \oplus H(BK_{MU_i}^{ne} \parallel ID_{MU_i} \parallel PW_{MU_i}^{ne})$ ,  $A_{11}^{ne} = SID_{MS} \oplus SID_{MU_i} \oplus SP_{MU_i} \oplus SID_{D_j}$ , and  $AUTH_{ne} = H(ID_{MU_i} \parallel BK_{MU_i}^{ne} \parallel PW_{MU_i}^{ne} \parallel A_{11}^{ne})$ .  $MD_i$  deletes  $A_{11}^{ne}$  from its memory.
- 3) Step PUP-3: Finally,  $MD_i$  stores the parameters  $\{A_2^{ne}, A_3^{ne}, AUTH_{ne}, Gen(), Rep(), RP^{ne}, ET^{ne}\}$  in its memory.

Fig. 4 illustrates the summary of PUP.

### E. REVOCATION OR RE-ISSUE PHASE (RRP)

An authorized  $MU_i$  can get a new mobile device  $MD_i^{new}$  after losing the old  $MD_i^{old}$ . For this,  $MU_i$  requires to accomplish the following steps.

- 1) Step RRP-1:  $MU_i$  only needs to remember or maintain  $ID_{MU_i}$  and picks a new password  $PW_{MU_i}^n$ .
- 2) Step RRP-2:  $MD_i$  picks a new random number  $R_{IN}^n$ , computes  $RID_{MU_i}^n = H(ID_{MU_i} \parallel R_{IN}^n)$ , and sends  $RID_{MU_i}^n$  to MS through a secure channel.
- 3) Step RRP-3: Upon receiving  $RID_{MU_i}^n$  from  $MD_i$ , MS picks fresh/new timestamp  $T_{MS}^n$ , new master-key  $MSK_{MU_i}^n$  for  $MU_i$ , calculates  $S_{ms}^n = H(T_{MS}^n \parallel RID_{MU_i}^n \parallel$

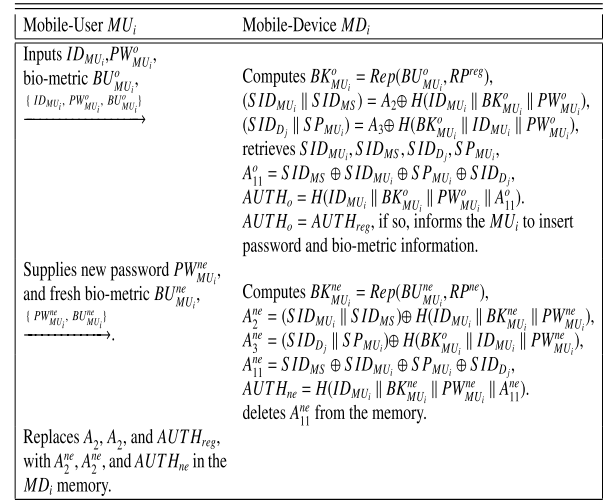


FIGURE 4. Password/bio-metric update phase.

- 4) Step RRP-4: Upon receiving the response from MS,  $MD_i$  calculates the following operations  $A_{11}^n = SID_{MS} \oplus SID_{MU_i}^n \oplus SP_{MU_i}^n \oplus SID_{D_j}$  and computes the new bio-metric key  $(BK_{MU_i}^n, RP^n) = Gen(BU_{MU_i}^n)$  by taking the fresh bio-information of the user (both the old and new bio-metric information are same) as input. Furthermore,  $MD_i$  also calculates  $A_2^n = SID_{MU_i} \parallel SID_{MS} \oplus H(ID_{MU_i} \parallel BK_{MU_i}^n \parallel PW_{MU_i}^n)$ ,  $A_3^n = SID_{D_j} \parallel SP_{MU_i}^n \oplus H(BK_{MU_i}^n \parallel ID_{MU_i} \parallel PW_{MU_i}^n)$ ,  $AUTH_{reg}^n = H(ID_{MU_i} \parallel BK_{MU_i}^n \parallel PW_{MU_i}^n \parallel A_{11}^n)$ , and deletes  $A_{11}^n$  from its memory.
- 5) Step RRP-5: Finally,  $MD_i$  stores the parameters  $\{A_2^n, A_3^n, AUTH_{reg}^n, Gen(), Rep(), RP^{reg}, ET^n\}$  in its memory.

Fig. 5 illustrates the summary of RRP.

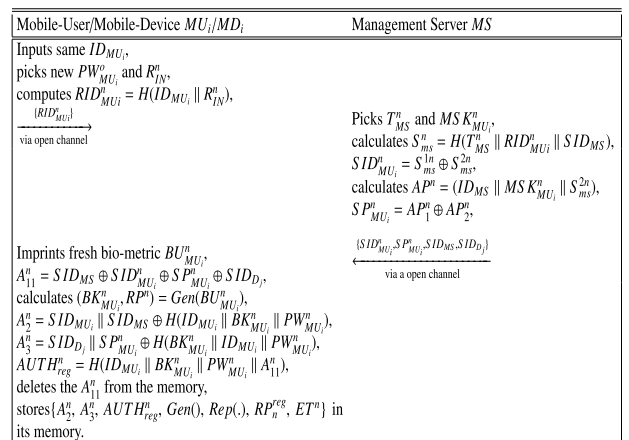


FIGURE 5. Revocation or re-issue phase.

## F. DYNAMIC DRONE DEPLOYMENT (DDD) PHASE

Following steps are required to execute for the deployment of new drone device, say  $D_j^{new}$  in some existing Fly Zone (FZ) with unique identity  $FID_k$ .

- 1) Step DDD-1: MS assigns a unique identity  $ID_{D_i}^{new}$  and Fly Zone Identity  $FID_k^{new}$  to drone  $D_j^{new}$ .
- 2) Step DDD-2: MS computes the temporary identity  $SID_{D_j}$  of drone by calculating  $Q^{new} = H(ID_{MS} \parallel SID_{MS} \parallel R_{D_i}^{new})$ , where  $R_{D_i}^{new}$  is a random number of 128 bits,  $SID_{D_i}^{new} = Q_1^{new} \oplus Q_2^{new}$ .
- 3) Step DDD-3: MS computes secret parameter  $SP_{D_j}^{new}$  for  $D_j^{new}$  by computing  $SP_{D_j}^{new} = R_{D_i}^{new} \oplus Q_1^{new}$ .
- 4) Step DDD-4: Finally, MS stores the parameters  $\{ID_{D_i}^{new}, SID_{D_i}^{new}, SP_{D_j}^{new}, FID_k^{new}\}$  in the memory of  $D_j^{new}$ . MS also stores these credentials in its memory.

## VI. SECURITY ANALYSIS

Both informal and formal security analyses have been conducted on LAKE-IoD to ascertain its immunity against various harmful attacks, such as device capture attack, Man-in-the-Middle (MITM) attack, and replay attack. BAN logic is applied to examine the logical completeness of LAKE-IoD. Scyther, a software tool, is utilized to examine the security characteristics of LAKE-IoD in an automatic way.

### A. INFORMAL SECURITY ANALYSIS

Following informal security analysis explicates that LAKE-IoD is immune to various attacks, and also guarantees user's un-traceability/anonymity.

#### 1) OFFLINE PASSWORD-GUESSING ATTACK

Presume that an adversary  $\mathcal{A}$  somehow gets or steals  $MD_i$  of  $MU_i$ .  $\mathcal{A}$  by applying the power-analysis attack [34] can procure the information stored in the memory of  $MD_i$ , such as  $\{A_2, A_3, AUTH_{reg}, Gen(), Rep(), RP^{reg}, ET\}$ . The extracted information does not provide any secret information to  $\mathcal{A}$  related to  $MU_i$ , such as  $ID_{MU_i}$ ,  $PW_{MU_i}$ , and  $BU_{MU_i}$ . Therefore, without knowing valid parameters, such as  $ID_{MU_i}$  and  $BU_{MU_i}$ , it is hard for  $\mathcal{A}$  to guess the correct  $PW_{MU_i}$  of  $MU_i$ . Hence, LAKE-IoD is resistant to the password-guessing attack.

#### 2) PASSWORD AND BIO-METRIC UPDATE ATTACK

Suppose that an adversary  $\mathcal{A}$  somehow has obtained the lost or stolen  $MU_i$ 's  $MD_i$  and extricates the stored information, such as  $\{A_2, A_3, AUTH_{reg}, Gen(), Rep(), RP^{reg}, ET\}$  by employing the power analysis attack [34]. Now,  $\mathcal{A}$  tries to update the password  $PW_{MU_i}$  and bio-metric information  $BU_{MU_i}$  of  $MU_i$ . For this purpose,  $\mathcal{A}$  picks bogus password  $PW_{MU_i}^{\mathcal{A}}$ , bio-metric information  $BU_{MU_i}^{\mathcal{A}}$ , identity  $ID_{MU_i}^{\mathcal{A}}$ , and calculates  $BK_{MU_i}^{\mathcal{A}} = Rep(BU_{MU_i}^{\mathcal{A}}, RP^{reg}), (SID_{MU_i}^{\mathcal{A}} \parallel SID_{CS}^{\mathcal{A}}) = A_2 \oplus H(ID_{MU_i}^{\mathcal{A}} \parallel BK_{MU_i}^{\mathcal{A}} \parallel PW_{MU_i}^{\mathcal{A}}), (SID_{D_j}^{\mathcal{A}} \parallel SP_{MU_i}^{\mathcal{A}}) = A_3 \oplus H(BK_{MU_i}^{\mathcal{A}} \parallel ID_{MU_i}^{\mathcal{A}} \parallel PW_{MU_i}^{\mathcal{A}}), A_{11}^{\mathcal{A}} = SID_{MS}^{\mathcal{A}} \oplus SID_{MU_i}^{\mathcal{A}} \oplus SP_{MU_i}^{\mathcal{A}} \oplus SID_{D_j}^{\mathcal{A}}$ , and  $AUTH_{\mathcal{A}} = H(ID_{MU_i}^{\mathcal{A}} \parallel BK_{MU_i}^{\mathcal{A}} \parallel PW_{MU_i}^{\mathcal{A}} \parallel A_{11}^{\mathcal{A}})$ , and checks the condition  $AUTH_{\mathcal{A}} =$

$AUTH_{reg}$ . To execute these computation,  $\mathcal{A}$  requires to know valid secret parameters, such as  $ID_{MU_i}$ ,  $BU_{MU_i}$ , and  $PW_{MU_i}$  of  $MU_i$ , which are secret and known only to  $MU_i$ . Without having the knowledge of these parameters, it is hard for  $\mathcal{A}$  to execute this attack. Therefore, LAKE-IoD is secure against the password/bio-metric update attack.

#### 3) IDENTITY-GUESSING ATTACK

During  $MU_i$  registration phase,  $MU_i$  sends  $RID_{MU_i}$  to MS, where  $RID_{MU_i} = H(ID_{MU_i} \parallel R_{IN})$ , through public communication channel. It is observed that the registration message does not provide any information about the identity  $ID_{MU_i}$  of  $MU_i$ . Now suppose that, insider attacker/adversary  $\mathcal{A}$  of MS has obtained lost/stolen device of  $MU_i$  and extricates the parameters  $\{A_2, A_3, AUTH_{reg}, Gen(), Rep(), RP^{reg}, ET\}$  stored on device. After getting these parameters,  $\mathcal{A}$  can not procure any significant information about the user identity  $ID_{MU_i}$ . Therefore, to guess the identity of  $MU_i$ ,  $\mathcal{A}$  needs to know both  $ID_{MU_i}$  and  $R_{IN}$  to compute  $HID_{MU_i} = H(ID_{MU_i} \parallel R_{IN})$ . Without knowing  $ID_{MU_i}$  and  $R_{IN}$ , it is hard for  $\mathcal{A}$  guess the correct identity of  $MU_i$ . Above discussion shows that LAKE-IoD is secure against the identity-guessing attack.

#### 4) USER ANONYMITY/UN-TRACEABILITY

According to the threat model described in Section III-B, an adversary  $\mathcal{A}$  can intercept the communicated messages  $M_1: \{T_1, A_4, C_1^{mu}, C_2^{mu}, AUTH_{tag1}\}$ ,  $M_2: \{T_2, A_{12}, C_1^{ms}, AUTH_{tag3}\}$ , and  $M_3: \{T_3, A_{19}, C_1^d, AUTH_{tag5}\}$ , where  $A_4 = SID_{MU_i} \oplus ZZ \oplus SID_{MS}$ , and  $A_{12} = SID_{MU_i} \oplus ZZ_2 \oplus SID_{D_j}$ , which are communicated during the AKE phase. However, without knowing the valid secret parameters and based on the discussion for the identity-guessing attack as in Section VI-A3, it is hard for  $\mathcal{A}$  to derive the real identity of  $MU_i$ . Thus, LAKE-IoD ensures the anonymity of  $MU_i$ . All the exchanged messages are dynamic in nature, which incorporates the latest timestamps, fresh random numbers, and random Initialization Vectors (IV). Therefore,  $\mathcal{A}$  can not correlate two messages of different AKE sessions. So, LAKE-IoD also ensures the user's un-traceability.

#### 5) DRONE CAPTURE ATTACK

From the threat model as discussed in Section III-B, it is possible for an adversary  $\mathcal{A}$  to capture the drone device  $D_j$  because they are deployed in the hostile environment. By utilizing the power analysis attack [34],  $\mathcal{A}$  can retrieve the secret information stored in memory of  $D_j$ , such as  $ID_{D_j}$ ,  $SID_{D_j}$ ,  $SP_{D_j}$ , and  $FID_k$  and can compromise the session key security of the captured  $D_j$ . However, by compromising the security of captured  $D_j$ ,  $\mathcal{A}$  can not breach the security of other non compromised  $D_j$  because of the uniqueness of the secret parameters  $ID_{D_j}$ ,  $SID_{D_j}$ ,  $SP_{D_j}$ , and  $FID_k$ . Therefore, LAKE-IoD is resilient against the drone captured attack.

#### 6) IMPERSONATION ATTACK

The succeeding impersonation attacks associated to LAKE-IoD are considered.



- $MU_i$  Impersonation Attack: According to the threat model described in Section III-B, an adversary  $\mathcal{A}$  can capture  $M_1: \{ T_1, A_4, C_1^{mu}, C_2^{mu}, AUTH_{tag1} \}$  transmitted by  $MU_i$  during the login and AKE phase. Further,  $\mathcal{A}$  can act as a legitimate  $MU_i$  by producing some bogus message  $M'_1$  to persuade MS that  $M'_1$  is from a valid  $MU_i$ . However,  $\mathcal{A}$  can generate the timestamp  $T'_1$  but without the knowledge of valid parameters, such as  $SID_{MU_i}$ ,  $SID_{MS}$ ,  $SP_{MU_i}$ , and  $K_1$ , it is hard for  $\mathcal{A}$  to generate a valid  $M_1$  because the authenticity of  $M_1$  is checked against the condition  $AUTH_{tag1} = AUTH_{tag2}$ . Without satisfying this condition,  $\mathcal{A}$  cannot impersonate as a legitimate user in IoD environment. Therefore, LAKE-IoD is resistant against  $MU_i$  impersonation attack.
- MS Impersonation Attack: An adversary  $\mathcal{A}$  can capture  $M_2: \{ T_2, A_{12}, C_1^{ms}, AUTH_{tag3} \}$  and also generate a fake message  $M'_2$  to make  $D_j$  believe that  $M'_2$  is from a legitimate MS. However,  $M_2$  received by  $D_j$  during the login and AKE phase will be checked against the condition  $AUTH_{tag3} = AUTH_{tag4}$ . If the condition holds,  $M_2$  will be accepted. Otherwise,  $D_j$  rejects  $M_2$ . Therefore, it is hard for  $\mathcal{A}$  to generate a valid message  $M_2$ , without the knowledge of the secret parameters, such as  $SID_{D_j}$ ,  $ID_{D_j}$ ,  $FID_k$ , and  $SP_{D_j}$ . Hence, LAKE-IoD is resistant against MS impersonation attack.
- $D_j$  Impersonation Attack: In this case, an adversary  $\mathcal{A}$  intercepts the message  $M_3: \{ T_3, A_{19}, C_1^d, AUTH_{tag5} \}$  transmitted by the  $D_j$  and generates a fake message  $M'_3$  on behalf of  $D_j$  to convince  $MU_i$  that  $M'_3$  is from a legitimate  $D_j$ . However, without the knowledge of secret parameters  $SID_{MU_i}$ , and  $SID_{D_j}$ , it is hard for  $\mathcal{A}$  to generate a fake message on behalf of  $D_j$ . Therefore, the proposed scheme is secure against  $D_j$  impersonation attack.

#### 7) MAN-IN-THE-MIDDLE ATTACK

During the login & authentication phase,  $\mathcal{A}$  tries to intercept the exchanged messages, such as  $M_1: \{ T_1, A_4, C_1^{mu}, C_2^{mu}, AUTH_{tag1} \}$ ,  $M_2: \{ T_2, A_{12}, C_1^{ms}, AUTH_{tag3} \}$ ,  $M_3: \{ T_3, A_{19}, C_1^d, AUTH_{tag5} \}$ , and attempts to modify the contents of  $M_1$ ,  $M_2$ , and  $M_3$ . By framing this attack, the objective of  $\mathcal{A}$  is to make the entities in IoD environment, such as  $MU_i$ ,  $MS$ , and  $D_j$ , which are involved in the AKE process believe that the messages are from a legitimate entity. However,  $\mathcal{A}$  can not frame this attack without computing valid secret credentials, such as  $K_1$ ,  $K_2$ , and  $K_3$  because these credentials are derived by using secret parameters  $SID_{MU_i}$ ,  $SID_{MS}$ ,  $SP_{MU_i}$ ,  $SID_{D_j}$ , and  $SP_{D_j}$ , which are unknown to  $\mathcal{A}$ . Therefore, without knowing these secret parameters, it is hard for  $\mathcal{A}$  to frame this attack. Hence, LAKE-IoD is secure against the Man-in-the-Middle attack.

#### 8) DANIEL-OF-SERVICE (DoS) ATTACK

In the proposed scheme LAKE-IoD,  $MU_i$  enters his/her secret credentials, such as password  $PW_{MU_i}$ , bio-metric information

$BU_{MU_i}$ , and identity  $ID_{MU_i}$  at the available interface of  $MD_i$ . These parameters are verified locally by checking the condition  $AUTH_{LO} = AUTH_{reg}$  before sending an authentication request to MS. If the condition holds,  $MD_i$  will then send authentication request to MS. If the condition does not hold,  $MD_i$  aborts AKE process promptly and prevent  $MU_i$  from sending too many authentication requests to MS. Above discussion shows that LAKE-IoD is resistant to the DoS attack.

#### 9) REPLAY ATTACK

In this attack, an adversary  $\mathcal{A}$  attempts to capture the communicated messages, such as  $M_1: \{ T_1, A_4, C_1^{mu}, C_2^{mu}, AUTH_{tag1} \}$ ,  $M_2: \{ T_2, A_{12}, C_1^{ms}, AUTH_{tag3} \}$ , and  $M_3: \{ T_3, A_{19}, C_1^d, AUTH_{tag5} \}$  during the AKE process in the proposed scheme to launch the replay attack by replying the forged instances of the messages to the receiver. However, all the exchanged messages incorporate the timestamps and fresh random numbers. At first, the receiver of the message checks the freshness of each message by checking the condition  $T_{ad}^1 \geq |T^R - T_1|$  for  $M_1$ ,  $T_{ad}^2 \geq |T^R - T_2|$  for  $M_2$ , and  $T_{ad}^3 \geq |T^R - T_3|$  for  $M_3$ . If all the received messages are within allowed delay time limit, the received messages are considered as latest/fresh messages. Otherwise, the receiver discards the delayed messages. Additionally, the receiver will validate the authenticity and integrity of each received message by checking the condition  $AUTH_{tag1} = AUTH_{tag2}$  for  $M_1$ ,  $AUTH_{tag3} = AUTH_{tag4}$  for  $M_2$ , and  $AUTH_{tag5} = AUTH_{tag6}$  for  $M_3$ . All the exchanged message during the AKE phase are considered to be authentic, if these satisfy these conditions. Without knowing the valid secret parameters, it is hard for  $\mathcal{A}$  to reproduce a valid message and cannot frame this attack. Therefore, LAKE-IoD is immune to the replay attack.

#### 10) EPHEMERAL SECRET LEAKAGE (ESL) ATTACK

It is possible that an adversary  $\mathcal{A}$  may compromise the long-term and short-term secret parameters of the communicating entities in IoD environment. By utilizing these compromised secret parameters,  $\mathcal{A}$  may reveal the secret session key between the two communicating entities. This type of attack is referred to as Ephemeral Secret Leakage (ESL) attack.

- Case-1: Suppose that the short-term (ephemeral) secret parameters, such as  $R_{MU_i}$ ,  $R_{MS}$ , and  $R_{D_j}$  are somehow revealed to the adversary  $\mathcal{A}$ . Now, the objective of  $\mathcal{A}$  is to generate the secret session key by computing  $SK_X (= SK_Y) = H(SID_{D_j} \parallel P_4 \parallel SID_{MU_i} \parallel T_3)$ . However, without knowing other long-term secret credentials  $SID_{D_j}$ ,  $SID_{MU_i}$ , and  $FID_k$ , it is hard for  $\mathcal{A}$  to generate the valid secret session key  $SK_X (= SK_Y)$ .
- Case-2: In this case, if the long-term secret credential  $SID_{D_j}$ ,  $SID_{MU_i}$ , and  $FID_k$  are somehow revealed to  $\mathcal{A}$ , still  $\mathcal{A}$  is required to know the short-term secret parameters, such as  $R_{MU_i}$ ,  $R_{MS}$ , and  $R_{D_j}$  to derive the valid session key  $SK_X (= SK_Y)$ .

It is clear from the above discussion that  $\mathcal{A}$  needs to know both the long-term and short-term secret parameters to breach

the security of the session key  $SK_X (= SK_Y)$ . Therefore, the proposed LAKE-IoD is secure against ESL attack.

### 11) MUTUAL AUTHENTICATION

LAKE-IoD renders the mutual authentication among the involved entities in the IoD environment. The details of the mutual authentication process are given below.

- $MU_i \rightarrow MS$ : MS after receiving the message  $M_1: \{T_1, A_4, C_1^{mu}, C_2^{mu}, AUTH_{tag1}\}$  authenticates  $MU_i$  by checking  $SID_{MU_i}$  in its database and ensures the authenticity of  $M_1$  by verifying the condition  $AUTH_{tag1} = AUTH_{tag2}$ .
- $MS \rightarrow D_j$ : Upon receiving the message  $M_2: \{T_2, A_{12}, C_1^{ms}, AUTH_{tag3}\}$  from MS,  $D_j$  computes  $SID_{MU_i} = A_{12} \oplus SID_{D_j}$ . Further,  $D_j$  verifies the authenticity of  $M_2$  by checking the condition  $AUTH_{tag3} = AUTH_{tag4}$  and extracts  $P_3 = R_{MS} \oplus R_{MU_i}$ .
- $D_j \rightarrow MU_i$ :  $M_3$ :  $MU_i$  receives the message  $\{T_3, A_{19}, C_1^d, AUTH_{tag5}\}$  from  $D_j$  and checks the condition to authenticate  $D_j$  by verifying the condition  $AUTH_{tag6} = AUTH_{tag5}$ . After the authentication of  $D_j$ ,  $MU_i$  retrieves the plaintext  $P_4 = R_{D_j} \oplus FID_k \oplus P_3$  from the ciphertext  $C_1^d$ .

Above discussion reveals that the proposed LAKE-IoD achieves the mutual authentication between  $MU_i$  and  $D_j$  with the help of MS. After achieving the mutual authentication, both entities  $MU_i$  and  $D_j$  establish a secret session-key  $SK_X (= SK_Y) = H(SID_{D_j} \parallel P_4 \parallel SID_{MU_i} \parallel T_3)$ .

### B. FORMAL SECURITY ANALYSIS

This section provides the formal analysis of the proposed scheme by employing the Burrows *et al.* [35] logic and software verification tool Scyther [36].

#### 1) MUTUAL AUTHENTICATION VERIFICATION BY USING BAN LOGIC

Burrows-Abadi-Needham (BAN) logic [35] is an epistemic logic devised for the analysis of communication security protocols. The BAN logic is a set of rules for describing and validating the completeness of an authentication protocol. Particularly, BAN logic assists its users to determine whether the exchanged information is reliable. The semantics of the BAN logic comprises of the expression presented in Table 2 and different inference derivation rules are specified in Table 3.

- 1) Assumptions: The subsequent assumptions are considered at the inception of the proposed scheme LAKE-IoD, to validate its mutual authentication.

- A-1:  $MD_j \models \#T_1, \#T_3, \#R_{MU_i}$
- A-2:  $MD_i \models (MD_i \xleftrightarrow{K_3} D_j)$
- A-3:  $MD_i \models D_j \implies (D_j \xleftrightarrow{SK} MD_i)$
- A-4:  $MD_i \models \implies D_j \sim P_4$
- A-4:  $MD_i \models (MD_i \xleftrightarrow{K_1} MS)$
- A-5:  $MS \models \#T_1, \#T_2, \#R_{MU_i}, \#R_{MS}$
- A-6:  $MS \models (MS \xleftrightarrow{K_1} MD_i)$

TABLE 2. BAN logic notations.

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

TABLE 3. BAN logic inference rules.

Notation	Description
Message-Meaning-Rule	$\frac{M \models M \xleftrightarrow{k} H, M \triangleleft \{X\}_k}{M \models H \mid \sim X}$
Jurisdiction-Rule	$\frac{M \models H \rightarrow X, M \models H \mid \sim X}{M \models X}$
Belief-Rule	$\frac{M \models (X, Y)}{M \models X}$
Nonce-Verification-Rule	$\frac{M \models \#(X), M \models H \mid \sim X}{M \models H \mid \sim X}$
Freshness-Rule	$\frac{M \models \#(X)}{M \models \#(X, Y)}$

- A-7:  $MS \models (MS \xleftrightarrow{K_2} D_j)$
- A-8:  $D_j \models MS \implies MS \mid \sim P_3$
- A-9:  $D_j \models \#T_2, \#T_3$
- A-10:  $D_j \models \#R_{MS}, \#R_{D_j}$
- A-11:  $D_j \models (D_j \xleftrightarrow{K_2} MS)$
- A-12:  $D_j \models (D_j \xleftrightarrow{K_3} MD_i)$

- 2) Idealized Form:

- IDF-1:  $\{T_1, A_4, R_{MU_i}, SID_{D_j}\}_{(MD_i \xleftrightarrow{K_1} MS)}$
- IDF-2:  $\{T_2, R_{MS}, \langle P_3 \rangle\}_{(MS \xleftrightarrow{K_2} D_j)}$
- IDF-3:  $\{T_3, P_4, (D_j \xleftrightarrow{SK} MD_i)\}_{(D_j \xleftrightarrow{K_3} MD_i)}$

- 3) Goals:

- G-1:  $D_j \models (D_j \xleftrightarrow{SK} MD_i)$
- G-2:  $MD_i \models (MD_i \xleftrightarrow{SK} D_j)$

#### 2) FORMAL VERIFICATION

We validate the mutual authentication property of LAKE-IoD formally by utilizing THE basic BAN Logic rules defined in Table 2, BAN Inference rules defined in Table 3, and by using the assumptions. Details of the steps are given below.

- FV-1: From IDF-1, by applying the A-6, A-7, and Message-Meaning Rule (MMR), we get
- FV-2: By using A-6 and Freshness Rule (FR), we obtain

$$\frac{MS \models \#T_1}{MS \models \#(T_1, A_4, R_{MU_i}, SID_{D_j})}$$

- FV-3: From FV-1, FV-2, and by using the Nonce-Verification Rule (NVR), we obtain
- FV-4: From IDF-2, by using A-9, A-10, A-11, A-12, and Message-Meaning Rule (MMR), we obtain

$$\frac{D_j \models (D_j \xleftrightarrow{K_2} MS), D_j \triangleleft \{T_2, A_{12}, \langle P_3 \rangle\}}{D_j \models MS \mid \sim \{T_2, A_{12}, \langle P_3 \rangle\}} \quad \begin{matrix} (D_j \xleftrightarrow{K_2} MS) \\ (MD_j \xleftrightarrow{K_2} MS) \end{matrix}$$

- FV-5: By employing A-10, A-11, and by using FR, we get

$$\frac{D_j \models \#T_2}{D_j \models \#(T_2, A_{12}, \langle P_3 \rangle)}$$

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

$$\frac{D_j \models \#(T_2, A_{12}, \langle P_3 \rangle), D_j \triangleleft (T_2, A_{12}, \langle P_3 \rangle)}{D_j \models MS \models (T_2, A_{12}, \langle P_3 \rangle)}$$

- FV-7: From FV-4, FV-5, FV-6, by applying A-19, and by using NVR, we get  $D_j \models R_{MU_i} \oplus R_{MS}$ .
- FV-8: Using FV-7, and by using A-9, A-10, A-11, and A-12, G-1 is achieved

$$D_j \models (D_j \xleftrightarrow{SK} MD_j).$$

- FV-9: From IDF-3, by using A-1, A-2, A-3, and A-4, and by applying MMR, we get
- FV-10: Using A-1 and by using FR, we obtain

$$\frac{MD_i \models \#T_3}{MD_i \models \#(T_3, P_4, (D_j \xleftrightarrow{SK} MD_i))}$$

- FV-11: From FV-9 and FV-10, and by applying NVR, we get
- FV-12: From FV-9, FV-10, FV-11, and by applying A-15, and NVR, we get  $MD_i \models P_4$ .
- FV-13: Using FV-12, by using A-2, and A-4, G-2 is achieved

$$MD_i \models (MD_i \xleftrightarrow{SK} D_j).$$

From FV-8 and FV-13, it is clear that  $M$  and  $D_j$  authenticate with each other through  $MS$ .

### 3) SECURITY ANALYSIS USING SCYTHYR TOOL

We employ Scyther tool [36] to analyze security properties and potential weaknesses of the proposed LAKE-IoD formally. The details of the Scyther tool are given below.

- Scyther tool is used for automatic validation of the security schemes. It is better and effective tool for falsification, verification, and analysis of proposed security protocols as compared to other verification tools, such as ProVerif and AVISPA.
- Scyther is based on the perfect cryptographic assumptions. It means that an adversary can not decrypt the encrypted information without knowing the secret key.
- Scyther utilizes the Security-Protocol Description-Language (SPDL) for modeling the user defined security scheme. In SPDL specification, each communicating entity is described by *Role* that can perform various functions such as *Send*, *Recv*, *event*, and security *claim*.
- Scyther tool follows the Dolev-Yao (DY) model and 9 other adversarial models such as eCK model and CK model, etc.
- Scyther renders a set of tests and claims to validate the security properties such as secrecy, authentication, synchronization, aliveness, weak agreement, and agreement.

There are three basic roles involved during the login and authentication phase of the LAKE-IoD, which are the Mobile-User  $MU_i$ , the Management Server  $MS$ , and drone  $D_j$ . The proposed scheme is implemented in SPDL. Scyther takes the SPDL file as input and performs various analyses on the LAKE-IoD scheme. Fig. 6 shows the results generated by Scyther after the analysis of the LAKE-IoD, which demonstrates that the proposed security scheme is secure under the claims as specified.

## VII. PERFORMANCE EVALUATION

In this section a detailed comparison between the proposed scheme LAKE-IoD and other relevant AKE schemes, such as Wazid *et al.* [6], Das *et al.* [24], Challa *et al.* [26], Srinivas *et al.* [7], and Challa *et al.* [22] is presented. LAKE-IoD is compared in term of security features, storage overhead, communication overhead, and computational cost during the AKE phase.

### A. SECURITY FEATURE COMPARISON (SFC)

This section presents the comparison of LAKE-IoD security features and other related AKE schemes. It is obvious from

$$\frac{MS \models (MS \xleftrightarrow{K_1} MD_i), MS \triangleleft \{T_1, A_4, R_{MU_i}, SID_{D_j}\}}{MS \models MD_i \mid \sim \{T_1, A_4, R_{MU_i}, SID_{D_j}\}} \quad \begin{matrix} (MD_i \xleftrightarrow{K_1} MS) \\ (MD_i \xleftrightarrow{K_1} MS) \end{matrix}$$

$$\frac{MS \models \#(T_1, A_4, R_{MU_i}, SID_{D_j}), MS \triangleleft (T_1, A_4, R_{MU_i}, SID_{D_j})}{MS \models MD_i \models (T_1, A_4, R_{MU_i}, SID_{D_j})}$$

Scyther results : autoverify					
Claim			Status	Comments	
LAKE_IoD	MU	LAKE_IoD,MU2	Secret_Hidden_2	OK	No attacks within bounds.
		LAKE_IoD,MU3	Secret_Hidden_1	OK	No attacks within bounds.
		LAKE_IoD,MU4	Secret T1	OK	No attacks within bounds.
		LAKE_IoD,MU5	Secret authtag5	OK	No attacks within bounds.
		LAKE_IoD,MU6	Secret C4	OK	No attacks within bounds.
		LAKE_IoD,MU7	Secret T3	OK	No attacks within bounds.
		LAKE_IoD,MU8	Alive	OK	No attacks within bounds.
		LAKE_IoD,MU9	Weakagree	OK	No attacks within bounds.
		LAKE_IoD,MU10	Niagree	OK	No attacks within bounds.
		LAKE_IoD,MU11	Nisynch	OK	No attacks within bounds.
MS	LAKE_IoD,MS1	Secret_Hidden_5	OK	No attacks within bounds.	
	LAKE_IoD,MS2	Secret_Hidden_4	OK	No attacks within bounds.	
	LAKE_IoD,MS3	Secret_Hidden_3	OK	No attacks within bounds.	
	LAKE_IoD,MS4	Secret T2	OK	No attacks within bounds.	
	LAKE_IoD,MS5	Secret T1	OK	No attacks within bounds.	
	LAKE_IoD,MS6	Alive	OK	No attacks within bounds.	
	LAKE_IoD,MS7	Weakagree	OK	No attacks within bounds.	
	LAKE_IoD,MS8	Niagree	OK	No attacks within bounds.	
	LAKE_IoD,MS9	Nisynch	OK	No attacks within bounds.	
DI	LAKE_IoD,DI2	Secret_Hidden_6	OK	No attacks within bounds.	
	LAKE_IoD,DI3	Secret T3	OK	No attacks within bounds.	
	LAKE_IoD,DI4	Secret T2	OK	No attacks within bounds.	
	LAKE_IoD,DI5	Alive	OK	No attacks within bounds.	
	LAKE_IoD,DI6	Weakagree	OK	No attacks within bounds.	
	LAKE_IoD,DI7	Niagree	OK	No attacks within bounds.	
	LAKE_IoD,DI8	Nisynch	OK	No attacks within bounds.	

FIGURE 6. Scyther analysis results of LAKE-IoD.

Table 4 that the scheme of Wazid et al. [6] does not render *SFC1*, *SFC3*, and *SFC6*, Das et al. [24] is insecure against *SFC1*, *SFC3*, *SFC4*, and *SFC16*, Challa et al. [26] is vulnerable to *SFC1*, *SFC2*, and *SFC3*, *SFC4*, and *SFC6*, Srinivas et al. [7] is not protected against *SFC1*, *SFC3*, and *SFC6*, and Challa et al. [22] is unprotected against *SFC5*, *SFC6*, *SFC7*, *SFC8*, and *SFC9*. Security is one of the most important parameters of concern of an AKE scheme. The proposed LAKE-IoD provides more security features as compared to other related AKE schemes. Table 4 illustrates the security feature comparison between LAKE-IoD and other related schemes.

TABLE 4. Security features comparison.

S-Feature	Wazid [6]	Das [24]	Challa [26]	Srinivas [7]	Challa [22]	LAKE-IoD
<i>SFC1</i>	×	×	×	×	✓	✓
<i>SFC2</i>	✓	✓	×	✓	✓	✓
<i>SFC3</i>	×	×	×	×	✓	✓
<i>SFC4</i>	✓	×	×	✓	✓	✓
<i>SFC5</i>	✓	✓	✓	✓	×	✓
<i>SFC6</i>	×	✓	×	×	×	✓
<i>SFC7</i>	✓	✓	✓	✓	×	✓
<i>SFC8</i>	✓	✓	✓	✓	×	✓
<i>SFC9</i>	✓	✓	✓	✓	×	✓
<i>SFC10</i>	✓	✓	✓	✓	✓	✓
<i>SFC11</i>	✓	✓	✓	✓	✓	✓
<i>SFC12</i>	✓	✓	✓	✓	✓	✓
<i>SFC13</i>	✓	✓	✓	✓	✓	✓
<i>SFC14</i>	✓	✓	✓	✓	✓	✓
<i>SFC15</i>	✓	✓	✓	✓	✓	✓
<i>SFC16</i>	✓	×	✓	✓	✓	✓

Note: *SFC1*: Stolen-device attack; *SFC2*: Password guessing attack; *SFC3*: Privileged-insider attack; *SFC4*: User anonymity/traceability; *SFC5*: Mutual authentication *SFC6*: Impersonation attack; *SFC7*: DoS attack; *SFC8*: Replay-attack; *SFC9*: Man-in-the-Middle/Forgery attack; *SFC10*: ESL attack; *SFC11*: Sensor/drone capture attack; *SFC12*: Password update phase; *SFC13*: Bio-metric update; *SFC14*: Revocable phase; *SFC15*: Un-authorized login detection; *SFC16*: Session key security; ✓: Shows feature is supported; ×: Shows not supported feature;

### B. STORAGE COST COMPARISON

This section presents the storage cost comparison between LAKE-IoD and other related AKE schemes, such as the scheme of Wazid et al. [6], Das et al. [24], Challa et al. [26], Srinivas et al. [7], and Challa et al. [22]. LAKE-IoD requires to store  $\{A_2, A_3, AUTH_{reg}, Gen(\cdot), Rep(\cdot), RP^{reg}, ET\}$ ,  $\{SID_{D_j}, ID_{D_j}, SP_{D_j}, FID_k\}$ , and  $\{SID_{MU_i}, SP_{MU_i}, (ID_{D_j}, SID_{D_j}, SP_{D_j}, FID_k)\}$  of sizes  $\{256 + 256 + 128 + 160 + 8\} = 808$  bits,  $\{128 + 128 + 128 + 128\} = 512$  bits, and  $\{128 + 128 + 128 + 128 + 128 + 128\} = 768$  bits on  $MU_i, D_j$ , and MS respectively. The total storage required for LAKE-IoD is  $\{808 + 512 + 768\} = 2088$  bits. The comparison exhibits that LAKE-IoD requires less storage cost as compared to Wazid et al. [6], Das et al. [24], Challa et al. [26], Srinivas et al. [7], and slightly high storage cost as compared to Challa et al. [22]. However, LAKE-IoD renders more security than Challa et al. [22], which is the most important parameter of concern for security scheme. Table 5 illustrates the storage

$$\frac{MD_i \mid \equiv (MD_i \xleftrightarrow{K_3} D_j), MD_i \triangleleft \{T_3, P_3, (D_j \xleftrightarrow{SK} MD_i)\}}{(MD_i \xleftrightarrow{K_3} D_j)} \quad MD_i \mid \equiv D_j \mid \sim \{T_3, P_4, (D_j \xleftrightarrow{SK} MD_i)\}}{(MD_i \xleftrightarrow{K_3} D_j)}$$

$$\frac{MD_i \mid \equiv \#(T_3, P_4, (D_j \xleftrightarrow{SK} MD_i)), MD_i \triangleleft (T_3, P_4, (D_j \xleftrightarrow{SK} MD_i))}{MD_i \mid \equiv D_j \mid \equiv \#(T_3, P_4, (D_j \xleftrightarrow{SK} MD_i))}$$

cost comparison of the proposed LAKE-IoD and other related AKE schemes.

**TABLE 5. Storage overhead.**

Scheme	User Side	Server Side	Drone Side	Total
Wazid [6]	1288 bits	$1120+(m+1)\log_2(p)$ bits	$480+(m+1)\log_2(p)$ bits	$2888 + 2 * (m+1)\log_2(p)$ bits
Das [24]	768 bits	$992+(m+1)\log_2(p)$ bits	$640+(m+1)\log_2(p)$ bits	$2400 + 2 * (m+1)\log_2(p)$ bits
Challa [26]	488 bits	1928 bits	1600 bits	4016 bits
Srinivas [7]	1120 bits	1120 bits	640 bits	2888 bits
Challa [22]	648 bits	808 bits	320 bits	1776 bits
LAKE-IoD	808 bits	768 bits	512 bits	2088 bits

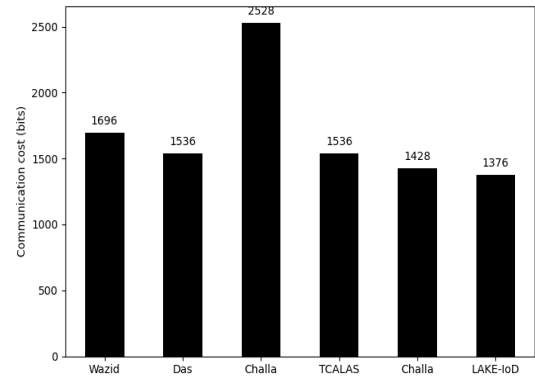
Note:  $m$  is the degree of symmetric bi-variate polynomial and  $p$  is the length of symmetric key

**C. COMMUNICATION OVERHEAD COMPARISON**

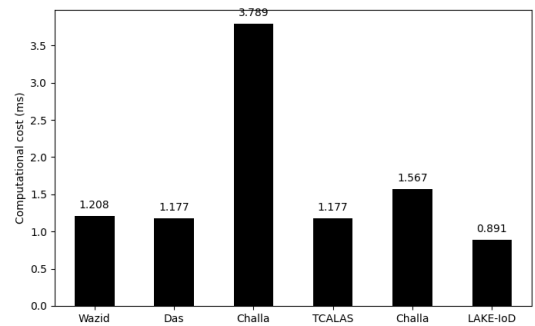
In this section, LAKE-IoD is compared with the existing schemes regarding the communication overhead of different involved entities during the AKE phase. The sizes of various credentials, we considered, such as timestamps, identities, random numbers, and EC points are 32 bits, 128 bits, 128 bits, and 160 bits, respectively. Moreover, the output hash function is 256 bits. Furthermore, the key size for the AEGIS is 128 bits and the size of parameter  $AUTH_{tagx} = 128$ , where  $x = 1, 2, 3$ . Table 6 illustrates the comparison of communication overhead during the AKE phase between LAKE-IoD and related schemes. LAKE-IoD exchanges three messages during the AKE process, such as  $M_1: \{T_1, A_4, C_1^{mu}, C_2^{mu}, AUTH_{tag1}\}$ ,  $M_2: \{T_2, A_{12}, C_1^{ms}, AUTH_{tag3}\}$ , and  $M_3: \{T_3, A_{19}, C_1^d, AUTH_{tag5}\}$  with length  $M_{au1} = 32 + 128 + 128 + 128 + 128 = 544$  bits,  $M_{au2} = 32 + 128 + 128 + 128 = 416$  bits, and  $M_{au3} = 32 + 128 + 128 + 128 = 416$  bits, respectively. Total communication overhead during the authentication process of the LAKE-IoD is  $\sum_{a=1}^3 |M_a| = (544 + 416 + 416) = 1376$  bits. Contrarily, the existing authentication scheme proposed by Wazid *et al.* [6], Das *et al.* [24], Challa *et al.* [26], Srinivas *et al.* [7], and Challa *et al.* [22] require 1696 bits, 1536 bits, 2528 bits, 1536 bits, and 1428 bits, respectively. Table 6 and Fig.7 manifest that LAKE-IoD requires less communication overhead as compared to the recent related schemes.

**D. COMPUTATIONAL OVERHEAD COMPARISON**

This paper considers the experimental results presented in the Table 7 to compute the computational overhead of the LAKE-IoD and other proposed schemes. The execution time of various operations employed in LAKE-IoD is computed using the system Intel(R) Pentium(R) CPU @ 2.5GHz, with Ubuntu (64 bits) operating system, and RAM 2 GB. Total computational overhead of the LAKE-IoD and the schemes of Wazid *et al.* [6], Das *et al.* [24], Challa *et al.* [26], Srinivas *et al.* [7], and Challa *et al.* [22] require  $13T_{SH} + 6T_{AG} + 1T_{BU} \approx 0.8943$  ms,  $31T_{SH} + 1T_{BU} \approx 1.2114$  ms,  $30T_{SH} + T_{BU} \approx 1.1803$  ms,  $12T_{SH} + 14T_{ec} + T_{BU} \approx 3.8354$  ms,  $30T_{SH} + 1T_{BU} \approx 1.1803$  ms, and  $19T_{SH} + 3T_{ec} +$



**FIGURE 7. Communication overhead.**



**FIGURE 8. Computational overhead.**

$T_{BU} \approx 1.5800$  ms, respectively. Table 8 and Fig. 8 shows the computational cost comparison of LAKE-IoD and the related AKE schemes. Moreover, the proposed LAKE-IoD requires computational cost at the drone side  $3T_{SH} + 2T_{AG} \approx 0.2052$  ms, which is comparable with the existing recent related schemes, such as Wazid [6], Das [24], Challa [26], Srinivas [7], and Challa [22], require  $7T_{SH} \approx 0.2177$  ms,  $7T_{SH} \approx 0.2177$  ms,  $3T_{SH} + 4T_{ec} \approx 1.0825$  ms,  $7T_{SH} \approx 0.2177$  ms, and  $5T_{SH} \approx 0.1555$  ms computational cost at the drone/sensor side, respectively. LAKE-IoD requires slightly high computation cost at drone side as compared to

**TABLE 6. Communication overhead.**

Scheme	Messages exchanged among the entities during AKE	Total (bits)
Wazid [6]	$MU_i/U_i \xrightarrow{672} MS/GW \xrightarrow{512} D_j/SN_j \xrightarrow{512} MU_i/U_i$	1696
Das [24]	$MU_i/U_i \xrightarrow{672} GW/CS \xrightarrow{512} D_j/SN_j \xrightarrow{352} MU_i/U_i$	1536
Challa [26]	$MU_i/U_i \xrightarrow{992} GW/CS \xrightarrow{1024} D_j/SN_j \xrightarrow{512} MU_i/U_i$	2528
Srinivas [7]	$MU_i/U_i \xrightarrow{672} MS/CS \xrightarrow{512} D_j/SN_j \xrightarrow{352} MU_i/U_i$	1536
Challa [22]	$MU_i/U_i \xrightarrow{832} GW/CS \xrightarrow{244} D_j/SN_j \xrightarrow{352} MU_i/U_i$	1428
LAKE-IoD	$MU_i/U_i \xrightarrow{544} MS/GW \xrightarrow{416} D_j/SN_j \xrightarrow{416} MU_i/U_i$	1376

Note:  $MU_i/U_i$  is the mobile user,  $MS$  is a management server,  $GW$  is the gateway,  $CS$  is the control server,  $D_j$  is the drone, and  $SN_j$  is the sensor node.

**TABLE 7. Execution time for various operations.**

Notation	Cryptographic Operation	Approximate Time (ms)
$T_{SH}$	Hash function (SHA-256)	0.0311
$T_{AG}$	AEGIS	0.056
$T_{EC}$	ECC point addition	0.2473
$T_{BU} \approx T_{EC}$	Fuzzy extractors	0.2473

**TABLE 8. Computational overhead.**

Scheme	User Side	Server Side	Drone Side	Total Time
Wazid [6]	$16T_{SH} + T_{BU}$	$8T_{SH}$	$7T_{SH}$	$31T_{SH} + T_{BU} \approx 1.2114$ ms
Das [24]	$14T_{SH} + T_{BU}$	$9T_{SH}$	$7T_{SH}$	$30T_{SH} + T_{BU} \approx 1.1803$ ms
Challa [26]	$5T_{SH} + 5T_{ec} + T_{BU}$	$4T_{SH} + 5T_{ec}$	$3T_{SH} + 4T_{ec}$	$12T_{SH} + 14T_{ec} + T_{BU} \approx 3.8354$ ms
Srinivas [7]	$14T_{SH} + T_{BU}$	$9T_{SH}$	$7T_{SH}$	$30T_{SH} + T_{BU} \approx 1.1803$ ms
Challa [22]	$10T_{SH} + 2T_{ec} + T_{BU}$	$4T_{SH} + T_{ec}$	$5T_{SH}$	$19T_{SH} + 3T_{ec} + T_{BU} \approx 1.5800$ ms
LAKE-IoD	$6T_{SH} + 2T_{AG} + T_{BU}$	$2T_{SH} + 2T_{AG}$	$3T_{SH} + 2T_{AG}$	$11T_{SH} + 6T_{AG} + T_{BU} \approx 0.8943$ ms

Challa [22] and less computational cost as compared to other related AKE schemes. However, the proposed LAKE-IoD is secure and renders more security functionality as compared to Challa [22], which is a critical feature of an AKE scheme.

## VIII. CONCLUSION

IoD is a providential technology that will predominate in the anticipated future, and there is an inevitable requirement to guarantee secure communication in IoD environment. The drones collect critical data and outsource it to the cloud and the users can collect buffered data from the cloud or (real-time data) directly from the drone. User authentication is inevitable and one of the principal security requirements to ensure secure communication between a specific drone and authorized user. In this paper, we devised a novel Lightweight Authenticated Key Exchange Protocol for the Internet of Drone Environment (LAKE-IoD) which is a three-factor security scheme employing user's password, mobile device, and bio-metric information. LAKE-IoD is examined meticulously for its security characteristics by employing formal security analysis using BAN logic and Scyther tool and also using informal security analysis. A comprehensive comparison of LAKE-IoD and other relevant security strategies illustrates that LAKE-IoD renders better security functionalities and incurs less computational and communication overhead for IoD resource constricted environment.

## ACKNOWLEDGMENT

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by grant code 19-COM-1-01-0015.

## REFERENCES

- [1] G. S. Ilgi and Y. K. Ever, "Critical analysis of security and privacy challenges for the Internet of drones: A survey," in *Drones in Smart-Cities*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 207–214.
- [2] H. Lei, D. Wang, K.-H. Park, I. S. Ansari, J. Jiang, G. Pan, and M.-S. Alouini, "Safeguarding UAV IoT communication systems against randomly located eavesdroppers," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1230–1244, Feb. 2020.
- [3] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–3742, Apr. 2019.
- [4] S. Ullah, K.-I. Kim, K. H. Kim, M. Imran, P. Khan, E. Tovar, and F. Ali, "UAV-enabled healthcare architecture: Issues and challenges," *Future Gener. Comput. Syst.*, vol. 97, pp. 425–432, Aug. 2019.
- [5] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of drones," *IEEE Access*, vol. 4, pp. 1148–1162, 2016.
- [6] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.
- [7] 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.
- [8] S. H. Alsamhi, O. Ma, M. S. Ansari, and F. A. Almalki, "Survey on collaborative smart drones and Internet of Things for improving smartness of smart cities," *IEEE Access*, vol. 7, pp. 128125–128152, 2019.
- [9] Y.-J. Chen and L.-C. Wang, "Privacy protection for Internet of drones: A network coding approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1719–1730, Apr. 2019.
- [10] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of UAV: A survey," *Mobile Netw. Appl.*, vol. 25, no. 1, pp. 95–101, Feb. 2020.
- [11] C. Lin, D. He, N. Kumar, K.-K.-R. Choo, A. Vinel, and X. Huang, "Security and privacy for the Internet of drones: Challenges and solutions," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 64–69, Jan. 2018.
- [12] M. Wazid, A. K. Das, and J.-H. Lee, "Authentication protocols for the Internet of drones: Taxonomy, analysis and future directions," *J. Ambient Intell. Humanized Comput.*, pp. 1–10, Aug. 2018, doi: 10.1007/s12652-018-1006-x.
- [13] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–43724, 2020.
- [14] M. S. Farash and M. A. Attari, "An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards," *Int. J. Commun. Syst.*, vol. 29, no. 13, pp. 1956–1967, Sep. 2016.
- [15] Y. Lu, L. Li, H. Peng, and Y. Yang, "An anonymous two-factor authenticated key agreement scheme for session initiation protocol using elliptic curve cryptography," *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 1801–1815, Jan. 2017.
- [16] Z. Zhang, Q. Qi, N. Kumar, N. Chilamkurti, and H.-Y. Jeong, "A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography," *Multimedia Tools Appl.*, vol. 74, no. 10, pp. 3477–3488, May 2015.
- [17] A. Irshad, S. A. Chaudhry, S. Kumari, M. Usman, K. Mahmood, and M. S. Faisal, "An improved lightweight multiserver authentication scheme," *Int. J. Commun. Syst.*, vol. 30, no. 17, p. e3351, Nov. 2017.
- [18] S. Kumari, X. Li, F. Wu, A. K. Das, K.-K.-R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Gener. Comput. Syst.*, vol. 68, pp. 320–330, Mar. 2017.
- [19] Q. Feng, D. He, S. Zeadally, and H. Wang, "Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment," *Future Gener. Comput. Syst.*, vol. 84, pp. 239–251, Jul. 2018.
- [20] R. Ali and A. K. Pal, "An efficient three factor-based authentication scheme in multiserver environment using ECC," *Int. J. Commun. Syst.*, vol. 31, no. 4, p. e3484, Mar. 2018.
- [21] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multiserver environment," *Secur. Commun. Netw.*, vol. 2019, pp. 1–15, Jun. 2019.
- [22] S. Challa, A. K. Das, V. Odelu, N. Kumar, S. Kumari, M. K. Khan, and A. V. Vasilakos, "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Comput. Electr. Eng.*, vol. 69, pp. 534–554, Jul. 2018.
- [23] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, Jan. 2018.
- [24] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.
- [25] S. Hussain and S. A. Chaudhry, "Comments on 'biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment,'" *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10936–10940, Dec. 2019.

- [26] 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.
- [27] X. Jia, D. He, L. Li, and K.-K. R. Choo, "Signature-based three-factor authenticated key exchange for Internet of Things applications," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 18355–18382, 2018.
- [28] G. Sharma and S. Kalra, "A lightweight user authentication scheme for cloud-IoT based healthcare services," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 43, no. S1, pp. 619–636, Jul. 2019.
- [29] 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, May 2020.
- [30] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
- [31] 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 Things J.*, vol. 5, no. 4, pp. 2884–2895, Aug. 2018.
- [32] H. Wu and B. Preneel, "AEGIS: A fast authenticated encryption algorithm," in *Selected Areas in Cryptography—SAC 2013*, T. Lange, K. Lauter, and P. Lisoněk, Eds. Berlin, Germany: Springer, 2014, pp. 185–201.
- [33] S. Sharaf and H. Mostafa, "A study of authentication encryption algorithms (POET, Deoxys, AEZ, MORUS, ACORN, AEGIS, AES-GCM) for automotive security," in *Proc. 30th Int. Conf. Microelectron. (ICM)*, Dec. 2018, pp. 303–306.
- [34] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.
- [35] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 426, no. 1871, pp. 233–271, 1989.
- [36] (Mar. 2020). *The Scyther Tool*. [Online]. Available: <https://people.cispa.io/cas.cremers/scyther/>



interests include remote user authentication, cyber security, security and privacy, cryptography, the Internet of Things, 6LoWPAN, and the Internet of Drone.



**AMJAD HUSSAIN ZAHID** received the Ph.D. degree in computer science (information security) from the University of Engineering and Technology, Lahore, Pakistan. He is currently working as an Assistant Professor with the University of Management and Technology (UMT), Lahore. He is also the Program Advisor for BS(IT) program and member of many academic bodies. He has been an Active Member of Higher Education Commission (HEC) National Curriculum Revision Committee (NCR), Pakistan. He has been an Active Member of Faculty Board of Studies with the Punjab University College of Information Technology (PUCIT) and the Virtual University of Pakistan. He possesses quality monitoring and maintaining capabilities along with the strong interpersonal, leadership, and team management skills. He has more than 23 years of qualitative experience in teaching. He is vigorous in academic research. His research interests include information security, programming languages, algorithm design, enterprise architecture, technology management, IT infrastructure, block chain, and so on. He is serving as an Efficient and an effective Reviewer in several reputed international research journals of high impact factor in the domain of information security.



**MUSHEER AHMAD** received the B.Tech. and M.Tech. degrees from the Department of Computer Engineering, Aligarh Muslim University, India, in 2004 and 2008, respectively, and the Ph.D. degree in chaos-based cryptography from the Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India. From 2007 to 2010, he has worked with the Department of Computer Engineering, Aligarh Muslim University. Since 2011, he has been working as an Assistant Professor with the Department of Computer Engineering, Jamia Millia Islamia. He has published over 80 research papers in international reputed refereed journals and conference proceedings of the IEEE/Springer/Elsevier. He has more than 1000 citations of his research works with an H-index of 18. His research interests include multimedia security, chaos-based cryptography, cryptanalysis, machine learning for security, image processing, and optimization techniques. He has served as a reviewer and a technical program committee member of many international conferences. He has also served as referee of some renowned journals, such as *Signal Processing*, *Information Sciences*, the *Journal of Information Security and Applications*, *IEEE ACCESS*, the *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, the *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, *Wireless Personal Communications*, *Neural Computing and Applications*, the *International Journal of Bifurcation and Chaos*, *Optik, Optics and Laser Technology*, *Neurocomputing*, *IET Information Security*, *Security and Communication Networks*, *Complexity*, *Computers in Biology and Medicine*, *Chaos Solitons & Fractals*, *Physica A: Statistical Mechanics and its Applications*, *Signal Processing: Image Communication*, the *Journal of the Chinese Institute of Engineers*, *Computational and Applied Mathematics*, and *ETRI Journal*.



**ABDULLAH BAZ** (Senior Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Umm Al-Qura University (UQU), in 2002, the M.Sc. degree in electrical and computer engineering from KAU, in 2007, and the M.Sc. degree in communication and signal processing and the Ph.D. degree in computer system design from Newcastle University, in 2009 and 2014, respectively. From 2014 to 2020, he was a Vice-Dean and the Dean of the Deanship of Scientific Research with UQU. He is currently an Assistant Professor with the Computer Engineering Department, a Vice-Dean of DFMEA, the General Director of the Decision Support Center, and the Consultant of the University Vice Chancellor with UQU. His research interests include data science, ML, AI, VLSI design, EDA/CAD tools, intelligent transportation, computer system and architecture, smart systems, and smart health. Since 2015, he has been served as a Review Committee Member of the IEEE International Symposium on Circuits and Systems (ISCAS) and a member of the Technical Committee of the IEEE VLSI Systems and Applications. He served as a Reviewer in a number of journals, including the *IEEE INTERNET OF THINGS*, *IET Computer Vision*, *Artificial Intelligence Review*, and *IET Circuits, Devices and Systems*.



**HOSAM ALHAKAMI** (Member, IEEE) received the B.Sc. degree in computer science from King Abdulaziz University, Saudi Arabia, in 2004, the M.Sc. degree in internet software systems from Birmingham University, Birmingham, U.K., in 2009, and the Ph.D. degree in software engineering from De Montfort University, in 2015. From 2004 to 2007, he worked with Software Development Industry, where he implemented several systems and solutions for a national academic institution. From 2015 to 2020, he was the Vice-Dean of the Deanship of Admission and Registration for Academic affairs with Umm Al-Qura University (UQU). He is currently an Associate Professor with the Computer Science Department, UQU. He focuses on enhancing real-world matching systems using machine learning and data analytics in a context of supporting decision-making. His research interests include algorithms, semantic web, and optimization techniques.

...