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

Muhammad Tanveer1, Amjad Hussain Zahid2, Musheer Ahmad3, Abdullah Baz4, (Senior Member, IEEE), and Hosam Alhakami5, (Member, IEEE)

1 Telecommunications and Networking (TeleCoN) Research Laboratory, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan
2 Department of Informatics and Systems, University of Management and Technology, Lahore 54700, Pakistan
3 Department of Computer Engineering, Jamia Millia Islamia, New Delhi 110025, India
4 Department of Computer Engineering, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia
5 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). UAVs 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 summary of the main contributions are listed as follows:

- 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.

**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.

**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 offline-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 Daniel-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, stollen-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.

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 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 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 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() : It is Bio-metric Key (BK) generation algorithm. The Gen() is a probabilistic algorithm. It takes BU as an input and generates BK ∈ [0, 1]L and a Reproduction Parameter (RP), where L is the length of the BK, that is, Gen(BU) = (BK, RP).

2) Rep(): 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’) ≤ 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 constrained IoT devices. It requires less computational resources as compared to AES and AES-GCM.

3) The AEGIS is an encryption algorithm, which generates output (CT, AUTHtag), where CT is the ciphertext, AUTHtag is the authentication parameter, by taking the plaintext PT as input. Logical operation of AEGIS can be expressed as CT = EK([IV, AD, PT]) and AUTHtag, where K is the key, IV is the Initialization Vectors, and AD is the Associative Data. AUTHtag 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 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 Di |i = 1, 2..Nj is discussed, where Nj is the total number of Djs. It is assumed that the airspace is divided into k number of Fly Zones (FZ). Each FZ is assigned with a unique Zone Identity (FIDk |k = 1, 2..Nk). It is necessary to register Dj with the Management Server (MS) before its deployment in a specific FZ. It is assumed that MS has a unique identity IDMS and temporary identity SIDMS, which are known only to MS. The detailed process of Djs registration is given below:

1) Step DRP-1: MS assigns a unique identity IDDj and Fly Zone Identity FIDk to Djs before its deployment in a specific FZ.

2) Step DRP-2: MS computes the temporary identity SIDDj for Djs by computing Q = H(IDMS || SIDMS || R Dj), where R Dj is a random number of 128 bits, SIDDj = Q1 ⊕ Q2, where Q1 and Q2 are the two equal chunks (128-bits) of Q.

3) Step DRP-3: MS computes secret parameter SPDj for the drone Djs by computing SPDj = R Dj ⊕ Q1.

4) Step DRP-4: Finally, MS stores the parameters {IDDj, SIDDj, SPDj, FIDk} in the memory of Djs. MS also stores these credentials in its memory.

B. MOBILE-USER REGISTRATION PHASE (MURP)

An MU 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

| Table 1. List of notations. |
|-----------------------------|
| Symbol | Description |
| IDMS, IDMS | Real-identity and temporary-identity of the Management Server (MS), respectively |
| IDMv, SIDMv, SPMv | Real-identity, temporary-identity, and authentication parameter for the user, respectively |
| MU, MDi | Mobile user, mobile device, respectively |
| IDsp, SIDsp, FIDk, SPSp | Real-identity, temporary-identity, Fly Zone (FZ) identity, and secret parameter for the drone, respectively |
| T1, T2, T3 | Timestamps utilized during the authentication phase, respectively |
| t1, t2, t3, and t4 | Maximum time delay and message received time, respectively |
| K1, K2, K3 | Shared-secret Key used during the Authentication phase |
| E(k)(x) | AES encryption and decryption of message “x” using the secret key “k”, respectively |
| Rmvi, RMv, SDj | Temporary random number used during authentication phase |
| Gen(), Rep() | Fuzzy-extractor key generation and reproduction function, respectively |
| BKMBV | Bio-metric key during registration and login phase, respectively |
| H(wx), w | Cryptographic hash-function, bit-wise XOR, and concatenation, respectively |
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 $R_{NU}$ and computes $RID_{MU_i} = H(ID_{MU_i} \parallel R_{NU})$ 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 $S_{ms} = H(T_{MS} \parallel RID_{MU_i} \parallel SID_{MS})$. $SID_{MS} = S_{ms} \oplus S_{ms}^{1}$, where $S_{ms}$ and $S_{ms}^{1}$ are the two equal 128-bits chunks of $S_{ms}$. MS calculates the Secret Parameter ($SP_{MU_i}$) for $MU_i$ by computing $AP = (ID_{MS} \parallel MSK_{MU_i} \parallel S_{ms}), SP_{MU_i} = AP \oplus AP_2$, 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 $A_{11} = SID_{MS} \oplus SID_{MU_i} \oplus SP_{MU_i} \oplus SID_{D_j}$, $(BK_{reg}, RP_{reg}) = Gen(BU_{MU_i}), A_2 = SID_{MS} \parallel SID_{D_j} \parallel H(ID_{MU_i} \parallel SP_{MU_i} \parallel BK_{reg} \parallel ID_{MU_i} \parallel PW_{MU_i}), AUTH_{reg} = H(ID_{MU_i} \parallel BK_{reg} \parallel PW_{MU_i} \parallel A_{11}),$ 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 $D_j$ calculates $BK_{LO_{MU_i}} = Rep(BU_{MU_i}, RP_{reg})$ provided the condition $HD(BU_{MU_i}, BU_{MU_i}^{reg}) \leq ET$ holds, where HD is the hamming distance between $BU_{MU_i}^{LO}$ and $BU_{MU_i}^{reg}$. Additionally, an $MU_i$ also computes $(SID_{MS}, SID_{MU_i}) = A_2 \oplus H(ID_{MU_i} \parallel BK_{LO_{MU_i}}, PW_{MU_i})$, $(SID_{D_j}, SID_{MU_i}) = A_2 \oplus H(ID_{MU_i} \parallel PW_{MU_i})$. Furthermore, an $MD_i$ calculates $A_{11} = SID_{MS} \oplus SID_{MU_i} \oplus SP_{MU_i} \oplus SID_{D_j}$. To verify the login request, an $MD_i$ computes $AUTH_{LO} = H(ID_{MU_i} \parallel BK_{LO_{MU_i}} \parallel PW_{MU_i} \parallel A_{11}^{LO})$ 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 $T_j$ 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 $l = 32, ZZ = Rot(Z_1, G), A_4 = SID_{MU_i} \parallel ZZ \parallel SID_{MS}$, 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 $(C_{mu}^{1}, C_{mu}^{2}) = E_{K_1}([IV_1, AD_1], P_1, P_2)$, and $AUTH_{tag_1}$ by using AEGIS encryption algorithm, where $AD_1$ is the associative data. Finally, $MD_i$ constructs the message $M_1 : (T_1, A_4, C_{mu}^{1}, C_{mu}^{2}, AUTH_{tag_1})$ 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 $T_{ad}^1 \geq |T_{ad} - T_1|$. If the condition holds, the received $M_1$ is considered to be a fresh message. Otherwise, MS rejects $M_1$. MS computes $G_1 = HW(SID_{MS}), Z_1 = (T_1 \parallel 0^l \parallel T_1 \parallel 1^l), ZZ_1 = Rot(Z_1, G_1), SID_{MU_i} = A_4 \parallel ZZ_1 \parallel SID_{MS}$, and verifies if $SID_{MS}$ exists in its database or not. If $SID_{MS}$ is found, MS retrieves $SP_{MU_i}$ related to $SID_{MS}$ from the database and continues the AKE process. Otherwise, MS aborts the AKE process promptly. Furthermore, MS computes $A_3 = H(SID_{MS} \parallel SP_{MU_i} \parallel SID_{MU_i})$ 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 $P_1, P_2 = D_{K_1}([IV_1, AD_2], C_{mu}^{1}, C_{mu}^{2}),$ and $AUTH_{tag_2}$ by using AEGIS decryption algorithm. To verify the authenticity of the received $M_1$, MS checks the condition $AUTH_{tag_1} = AUTH_{tag_2}$. 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 $P_3 = R_{MS} \parallel R_{MU_i}, G_2 = HW(SID_{D_j}), Z_2 = (T_2 \parallel 0^l \parallel T_2 \parallel 1^l), ZZ_2 = Rot(Z_2, G_2), A_9 = H(ID_{D_j} \parallel ID_{MU_i} \parallel SP_{MU_i}), K_2 = A_{10} \parallel A_1, A_{12} = SID_{D_j} \parallel ZZ_2 \parallel SID_{D_j}, AD_3 = A_{12},$ and $IV_3 = A_{10} \parallel AD_3$. Additionally, MS calculates $C_{ms}^{i} = E_{K_2}([IV_3, AD_3], P_3)$, and $AUTH_{tag_3}$ by employing the encryption algorithm. Finally, MS constructs the message $M_2 : (T_2, A_{12}, C_{ms}^{i}, AUTH_{tag_3})$ and dispatches $M_2$ to $D_j$ through a public channel.
Inputs $\text{ID}_{\text{MU}}$, $\text{PW}_{\text{MU}}$, and imprints $\text{BT}^{\text{MU}}$. computes $\text{BK}^{\text{MU}}(\text{ID}^{\text{MU}})$, $(\text{SID}_{\text{MU}} \lor \text{ID}_{\text{MU}}) = \text{K}_2 \oplus \text{H}(\text{ID}_{\text{MU}} \lor \text{BT}^{\text{MU}} \lor \text{PW}_{\text{MU}})$, $(\text{ID}_{\text{SU}} \lor \text{SP}_{\text{SU}}) = \text{A}_6 \oplus \text{H}(\text{ID}_{\text{SU}} \lor \text{SP}_{\text{SU}})$, $\text{AD}^\text{SU} = \text{SID}_{\text{SU}} \oplus \text{SP}_{\text{SU}} \lor \text{SP}_{\text{SU}} \lor \text{ID}_{\text{SU}}$. 

Step ULP-6: After receiving $\text{LAKE-IoD}$ login and authentication phase.

**FIGURE 3.** $\text{LAKE-IoD}$ login and authentication phase.

Step ULP-7: $D_j$ picks $T_j$, $R_{D_j}$, and computes $P_4 = R_{D_j} \oplus \text{FID}_k \oplus P_3$, $A_{16} = \text{H}(\text{SID}_{\text{SU}} \lor \text{SID}_{\text{MU}} \lor \text{T}_3)$, and $K_3 = \text{A}_{17} \oplus \text{A}_{18}$. To secure the communication between $D_j$ and $M_D$, $D_j$ computes the session-key by computing $\text{SK}_X = \text{H}(\text{SID}_{\text{SU}} \lor \text{P}_4 \lor \text{SID}_{\text{MU}} \lor \text{T}_3)$. Moreover, $D_j$ calculates $A_{19} = \text{SK}_X^{1/2} \oplus \text{SK}_X^{1/2} \lor \text{SID}_{\text{MU}}$, $\text{AD}_{3} = \text{A}_{19}$, and $IV_{3} = \text{A}_{17} \lor \text{AD}_{3}$. Additionally, $D_j$ calculates $C_4 = \text{E}_{K_3}(\{\text{IV}_4, \text{AD}_{4}\}, P_4)$, and $\text{AUTH}_{\text{tag}5}$ by using AEGIS. Finally, $D_j$ constructs the message $M_3 : (T_3, A_{19}, C_4, \text{AUTH}_{\text{tag}5})$ and sends $M_3$ to $M_D$.

Step ULP-8: After receiving the message $M_3$ from $D_j$, $MD_j$ checks the freshness of $M_3$ by checking the condition $T_{\text{ad}}^3 \geq |T^R - T_2|$. If the condition holds, $M_3$ is considered to be outdated message. Otherwise, $D_j$ calculates $G_3 = \text{H}(<\text{SID}_{\text{SU}} \lor \text{SID}_{\text{MU}} \lor \text{T}_3 \oplus \text{ID}_{\text{SU}} \lor \text{ID}_{\text{MU}} \lor \text{T}_3 \oplus \text{ID}_{\text{SU}} \lor \text{SP}_{\text{SU}} \lor \text{SP}_{\text{SU}} \lor \text{ID}_{\text{SU}})$, $A_{16} = \text{H}(\text{ID}_{\text{SU}} \lor \text{SP}_{\text{SU}})$, and $\text{AUTH}_{\text{tag}4}$ by using AEGIS decryption algorithm. To establish the authenticity of the received message $M_2$, MS validates the condition $\text{AUTH}_{\text{tag}4} = \text{AUTH}_{\text{tag}3}$. If the condition does not hold, $D_j$ rejects the message and aborts the AKE process. Otherwise, $D_j$ retrieves $P_3 = R_{\text{MU}} \oplus R_{\text{MS}}$ from $C_1^{\text{MU}}$, which is received with $M_2$.

7) Step ULP-7: $D_j$ picks $T_j$, $R_{D_j}$, and computes $P_4 = R_{D_j} \oplus \text{FID}_k \oplus P_3$, $A_{16} = \text{H}(\text{SID}_{\text{SU}} \lor \text{SID}_{\text{MU}} \lor \text{T}_3)$, and $K_3 = \text{A}_{17} \oplus \text{A}_{18}$. To secure the communication between $D_j$ and $M_D$, $D_j$ computes the session-key by computing $\text{SK}_Y = \text{H}(\text{SID}_{\text{SU}} \lor \text{P}_4 \lor \text{SID}_{\text{MU}} \lor \text{T}_3)$. Finally, $M_D$ computes $A_{23} = \text{SK}_Y^{1/2} \oplus \text{SK}_Y^{1/2} \lor \text{SID}_{\text{MU}}$ 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 $PW_{MU_i}$, and calculates the following operations, such as computes $BK_{MU_i} = Rep(BU_{MU_i}, R_{PUP})$ to reproduce the bio-metric key, $(SID_{MU_i} \parallel SID_{MS}) = A_2 \oplus H(ID_{MU_i} \parallel BK_{MU_i} \parallel PW_{MU_i})$, and retrieves $ID_{DM_i}, SID_{MD_i}, SID_{DS_i}$, and $SP_{MU_i}$. Furthermore, $MD_i$ computes $A_{11}^n = SID_{MS} \oplus SID_{MU} \oplus SP_{MU_i} \parallel SID_{DS_i}$, and $AUTH_n = H(ID_{MU_i} \parallel BK_{MU_i} \parallel PW_{MU_i} \parallel A_{11}^n)$. $MU_i$ checks the condition $AUTH_n = AUTH_{reg}$. 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 $PW_{MU_i}$, and calculates the following operations, such as computes $BK_{MU_i} = Rep(BU_{MU_i}, R_{PUP})$ to reproduce the bio-metric key, $A_{22}^n = (SID_{MU_i} \parallel SID_{MS}) \parallel H(ID_{MU_i} \parallel BK_{MU_i} \parallel PW_{MU_i} \parallel A_{11}^n), A_{23}^n = (SID_{MU_i} \parallel SP_{MU_i} \parallel H(BK_{MU_i} \parallel ID_{MU_i} \parallel PW_{MU_i} \parallel A_{11}^n), A_{24}^n = SID_{MS} \parallel SP_{MU_i} \parallel SID_{DS_i}$, and $AUTH_{ne} = H(ID_{MU_i} \parallel BK_{MU_i} \parallel PW_{MU_i} \parallel A_{23}^n). MD_i$ deletes $A_{11}^n$ from its memory.

3) Step PUP-3: Finally, $MD_i$ stores the parameters $A_{22}^n, A_{23}^n, AUTH_{reg}, Gen(\cdot), Rep(\cdot), R_{PUP}^n, ET_{ne}^n$ in its memory.

Fig. 4 illustrates the summary of PUP.

E. REVOCAUTION OR RE-ISSUE PHASE (RRP)

An authorized $MU_i$ can get a new mobile device $MD_i^{\text{new}}$ after losing the old $MD_i^{\text{old}}$. 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 $PW_{MU_i}^{\text{new}}$.

2) Step RRP-2: $MD_i$ picks a new random number $R_{\text{IN}}^{\text{new}}$, computes $RID_{MU_i}^{\text{new}} = H(ID_{MU_i} \parallel R_{\text{IN}}^{\text{new}})$, and sends $RID_{MU_i}^{\text{new}}$ to MS through a secure channel.

3) Step RRP-3: Upon receiving $RID_{MU_i}^{\text{new}}$ from $MD_i$, MS picks fresh/new timestamp $T_{MS}^n$, new master-key $MSK_{MU_i}^{\text{new}}$ for $MU_i$, calculates $S_{ms}^n = H(T_{MS}^n \parallel RID_{MU_i}^{\text{new}})$.

4) Step RRP-4: Upon receiving the response from MS, $MD_i$ calculates the following operations, such as computes $A_{22}^n = SID_{MU_i} \parallel SID_{MS} \parallel H(ID_{MU_i} \parallel BK_{MU_i} \parallel PW_{MU_i} \parallel A_{11}^n)$, and deletes $A_{11}^n$ from its memory.

5) Step RRP-5: Finally, $MD_i$ stores the parameters $A_{22}^n, A_{23}^n, AUTH_{reg}, Gen(\cdot), Rep(\cdot), R_{PUP}^n, ET_{ne}^n$ in its memory.

Fig. 5 illustrates the summary of RRP.
F. DYNAMIC DRONE DEPLOYMENT (DDD) PHASE

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

1) Step DDD-1: MS assigns a unique identity $ID_{D_i}^{\text{new}}$ and Fly Zone Identity $FID_k^{\text{new}}$ to drone $D_i^{\text{new}}$.

2) Step DDD-2: MS computes the temporary identity $SID_{D_i}$ of drone by calculating $Q_{ Sid_{D_i}} = H(\text{ID}_{MS} \parallel \text{SID}_{MS} \parallel R_{Sid_{D_i}}^{\text{new}})$, where $R_{Sid_{D_i}}^{\text{new}}$ is a random number of 128 bits, $SID_{D_i}^{new} = Q_{Sid_{D_i}}^{\text{new}} \oplus Q_{Sid_{D_i}}^{\text{new}}$.

3) Step DDD-3: MS computes secret parameter $SP_{D_j}$ for $D_j^{\text{new}}$ by computing $SP_{D_j}^{\text{new}} = R_{Sid_{D_j}}^{\text{new}} \oplus Q_{Sid_{D_j}}^{\text{new}}$.

4) Step DDD-4: Finally, MS stores the parameters $(ID_{D_i}^{\text{new}}, \text{SID}_{D_i}^{\text{new}}, SP_{D_j}^{\text{new}}, FID_k^{\text{new}})$ in the memory of $D_i^{\text{new}}$. 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 immutability 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. Scythe, 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 $A$ somehow gets or steals $MD_i$ of $MU_i$. $A$ by applying the power-analysis attack [34] can procure the information stored in the memory of $MD_i$, such as $\{A_2, A_3, AUTH_{reg}, \text{Gen}(), \text{Rep}(), R_{\text{preg}}, ET\}$. The extracted information does not provide any secret information to $A$ 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 $A$ 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 $A$ somehow has obtained the lost or stolen $MU_i$’s $MD_i$ and extricates the stored information, such as $\{A_2, A_3, AUTH_{reg}, \text{Gen}(), \text{Rep}(), R_{\text{preg}}, ET\}$ by employing the power analysis attack [34]. Now, $A$ tries to update the password $PW_{MU_i}$ and bio-metric information $BU_{MU_i}$ of $MU_i$. For this purpose, $A$ picks bogus password $PW_{MU_i}^3$, bio-metric information $BU_{MU_i}^3$, identity $ID_{MU_i}^3$, and calculates $BK_{MU_i}^3 = \text{Rep}(BU_{MU_i}^3, R_{\text{preg}}^3), (SID_{MU_i}^3, \text{SID}_{SP}^3)$:

$$A_2 \oplus H(ID_{MU_i}^3 \ || \ BK_{MU_i}^3 \ || PW_{MU_i}^3, (SID_{D_i}^3 \ || SP_{MU_i}^3, A_{11} = SID_{MS} \ \ (\text{SID}_{MU_i}^3 \ \ (\text{SID}_{SP}^3 \ || A_{11})),$$

and checks the condition $AUTH_A = AUTH_{reg}$. To execute these computations, $A$ 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 $A$ 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 $RID_{MU_i} = H(\text{ID}_{MU_i} \parallel R_{IN})$ 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 $A$ of MS has obtained lost/stolen device of $MU_i$ and extricates the parameters $\{A_2, A_3, AUTH_{reg}, \text{Gen}(), \text{Rep}(), R_{\text{preg}}, ET\}$ stored on device. After getting these parameters, $A$ can not procure any significant information about the user identity $ID_{MU_i}$. Therefore, to guess the identity of $MU_i$, $A$ 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 $A$ to 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 $A$ can intercept the communicated messages $M_1: \{T_1, A_4, C_{11m}, C_{11r}, AUTH_{tag1}, \}$, $M_2: \{T_2, A_{12}, C_{21}, AUTH_{tag2}, \}$, and $M_3: \{T_3, A_{19}, C_{1d}, AUTH_{tag5}\}$, where $A_4 = SID_{MU_i} \ || ZZ \ || SID_{MS}$, and $A_{12} = SID_{MU_i} \ || ZZ \ || SID_{D_j}$, 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 on Section VI-A3, it is hard for $A$ 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, $A$ cannot 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 $A$ to capture the drone device $D_j$ because they are deployed in the hostile environment. By utilizing the power analysis attack [34], $A$ can retrieve the secret information stored in memory of $D_j$, such as $ID_{D_j}$, $SID_{D_j}$, $SP_{D_j}$, and $FID_{D_j}$ and can compromise the session key security of the captured $D_j$. However, by compromising the security of captured $D_j$, $A$ 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 $FID_{D_j}$. Therefore, LAKE-IoD is resilient against the drone captured attack.

6) IMPERSONATION ATTACK

The succeeding impersonation attacks associated to LAKE-IoD are considered.
· **MU** Impersonation Attack: According to the threat model described in Section III-B, an adversary \( \mathcal{A} \) can capture \( M_1: \{T_1, A_4, C_{1}^{\text{mu}}, C_{2}^{\text{mu}}, \text{AUTH}_{\text{tag1}}\} \) transmitted by \( MU \) during the login and AKE phase. Further, \( \mathcal{A} \) can act as a legitimate \( MU \) by producing some bogus message \( M'_1 \) to persuade MS that \( M'_1 \) is from a valid \( MU \). However, \( \mathcal{A} \) can generate the timestamp \( T'_1 \) but without the knowledge of valid parameters, such as \( SID_{MU}, SID_{MS}, SP_{MU}, \) 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 \( \text{AUTH}_{\text{tag1}} = \text{AUTH}_{\text{tag2}} \). Without satisfying this condition, \( \mathcal{A} \) cannot impersonate as a legitimate user in IoD environment. Therefore, LAKE-IoD is secure against \( MU \) impersonation attack.

· **MS** Impersonation Attack: An adversary \( \mathcal{A} \) can capture \( M_2: \{T_2, A_{12}, C_{1}^{\text{ms}}, \text{AUTH}_{\text{tag3}}\} \) 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 \( \text{AUTH}_{\text{tag3}} = \text{AUTH}_{\text{tag4}} \). 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 \( SID_{D_j}, ID_{D_j}, FID_{D_j}, \) and \( SP_{D_j} \). Hence, LAKE-IoD is resistant against MS impersonation attack.

· **Dj** Impersonation Attack: In this case, an adversary \( \mathcal{A} \) intercepts the message \( M_3: \{T_3, A_{19}, C_{1}^{\text{d}}, \text{AUTH}_{\text{tag5}}\} \) transmitted by the \( D_j \) and generates a fake message \( M'_3 \) on behalf of \( D_j \) to convince \( MU \) that \( M'_3 \) is from a legitimate \( D_j \). However, without the knowledge of secret parameters \( SID_{MU} \), 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 \( M_1: \{T_1, A_4, C_{1}^{\text{mu}}, C_{2}^{\text{mu}}, \text{AUTH}_{\text{tag1}}\}, M_2: \{T_2, A_{12}, C_{1}^{\text{ms}}, \text{AUTH}_{\text{tag3}}\}, M_3: \{T_3, A_{19}, C_{1}^{\text{d}}, \text{AUTH}_{\text{tag5}}\} \), 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 \( MU, MS, \) 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 \( SID_{MU}, SID_{MS}, SP_{MU}, \) and \( SP_{D_j} \). 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, \( MU \) enters his/her secret credentials, such as password \( PW_{MU} \), bio-metric information \( BU_{MU} \), and identity \( ID_{MU} \), at the available interface of \( MD_i \). These parameters are verified locally by checking the condition \( \text{AUTH}_{\text{LO}} = \text{AUTH}_{\text{req}} \) 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 \( MU \) 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 \( M_1: \{T_1, A_4, C_{1}^{\text{mu}}, C_{2}^{\text{mu}}, \text{AUTH}_{\text{tag1}}\}, M_2: \{T_2, A_{12}, C_{1}^{\text{ms}}, \text{AUTH}_{\text{tag3}}\}, \) and \( M_3: \{T_3, A_{19}, C_{1}^{\text{d}}, \text{AUTH}_{\text{tag5}}\} \) during the AKE process in the proposed scheme to launch the replay attack by replaying 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 \( T_{ad} \geq |T^R - T| \) for \( M_1, T_{ad} \geq |T^R - T_2| \) for \( M_2, \) and \( T_{ad} \geq |T^R - T_3| \) for \( M_3 \). If all the received messages are with in 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 \( \text{AUTH}_{\text{tag1}} = \text{AUTH}_{\text{tag2}} \) for \( M_1, \text{AUTH}_{\text{tag3}} = \text{AUTH}_{\text{tag4}} \) for \( M_2, \) and \( \text{AUTH}_{\text{tag5}} = \text{AUTH}_{\text{tag6}} \) for \( M_3 \). All the exchanged messages 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 Epheermal Secret Leakage (ESL) attack.

- **Case-1**: Suppose that the short-term (ephemeral) secret parameters, such as \( R_{MU}, R_{MS}, \) 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 \( SK_{x} = H(SID_{D_j} || P_4 || SID_{MU} || T_3) \). However, without knowing other long-term secret credentials \( SID_{D_j}, SID_{MU}, \) and \( FID_{D_k} \), it is hard for \( \mathcal{A} \) to generate the valid secret session key \( SK_{x} \).

- **Case-2**: In this case, if the log-term secret credential \( SID_{D_j}, SID_{MU}, \) and \( FID_{D_k} \) are somehow revealed to \( \mathcal{A} \), still \( \mathcal{A} \) is required to know the short-term secret parameters, such as \( R_{MU}, R_{MS}, \) and \( R_{D_j} \) to derive the valid session key \( SK_{x} \).

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 $M_1$: \{\text{T}_1, A_4, C_{\text{mu}}^1, C_{\text{mu}}^2, \text{AUTH}_{\text{tag1}}\} authorizes $MU_i$ by checking $SID_{MU_i}$ in its database and ensures the authenticity of $M_1$ by verifying the condition $\text{AUTH}_{\text{tag1}} = \text{AUTH}_{\text{tag2}}$.

- **MS $\rightarrow D_j$:** Upon receiving the message $M_2$: \{\text{T}_2, A_{12}, C_{\text{ms}}^1, \text{AUTH}_{\text{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 $\text{AUTH}_{\text{tag3}} = \text{AUTH}_{\text{tag4}}$ and extracts $P_y = R_{MS} \oplus R_{MU_i}$.

- **$D_j \rightarrow MU_1$:** $MU_1$ receives the message $\{\text{T}_3, A_{19}, C_{\text{mu}}^1, \text{AUTH}_{\text{tag5}}\}$ from $D_j$ and checks the condition to authenticate $D_j$ by verifying the condition $\text{AUTH}_{\text{tag6}} = \text{AUTH}_{\text{tag5}}$. After the authentication of $D_j$, $MU_1$ retrieves the plaintext $P_4 = R_{D_j} \oplus FID_k \oplus P_3$ from the ciphertext $C_{\text{mu}}^1$.

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:** $MD_j \equiv \#T_1, \#T_3, \#R_{MU_i}$
- **A-2:** $MD_j \equiv (MD_j \xleftarrow{K_2} D_j)$
- **A-3:** $MD_j \equiv D_j \implies (D_j \xleftarrow{SK} MD_j)$
- **A-4:** $MD_j \equiv (MD_j \xleftarrow{K_1} MS)$
- **A-5:** $MS \equiv \#T_1, \#T_2, \#R_{MU_i}, \#R_{MS}$
- **A-6:** $MS \equiv (MS \xleftarrow{K_1} MD_j)$

| Feature | Description |
|---------|-------------|
| $M_1 \Rightarrow H$ | If $M$ is true then $H$ is also true |
| $M \Rightarrow X$ | $M$ believes if $X$ is true |
| $M < X$ | $M$ once said $X$ |
| $M \equiv X$ | $M$ sees $X$ |
| $X \equiv k$ | $k$ is a shared-secret between $M$ and $H$ |
| $\{X\}_k$ | Message $X$ is encrypted with the secret key $k$ |
| $(X)_Y$ | $X$ is combine with $Y$ |
| $M \Rightarrow X$ | $M$ has jurisdiction over $X$ |

2) Idealized Form:

- **IDF-1:** \{\text{T}_1, A_4, R_{MU_i}, SID_{D_j}\} ($MD_j \xleftarrow{K_1} MS$)
- **IDF-2:** \{\text{T}_2, R_{MS}, \langle P_3 \rangle\} ($MS \xleftarrow{K_2} D_j$)
- **IDF-3:** \{\text{T}_3, P_4, (D_j \xleftarrow{SK} MD_j)\} ($D_j \xleftarrow{K_3} MD_j$)

3) Goals:

- **G-1:** $D_j \equiv (D_j \xleftarrow{SK} MD_j)$
- **G-2:** $MD_j \equiv (MD_j \xleftarrow{SK} D_j)$

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 $MS \equiv \#(T_1, A_{4}, R_{MU_i}, SID_{D_j})$.
• 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
  \[ D_j \models (D_j \overset{K_2}{\rightarrow} MS), D_j \overset{\sim}{\models} \{T_2, A_{12}, \langle P_3 \rangle\} \]
  \[ (D_j \overset{K_2}{\rightarrow} MS) \]

  \[ D_j \models MS \lhd \{T_2, A_{12}, \langle P_3 \rangle\} \]

• FV-5: By employing A-10, A-11, and by using NVR, we get
  \[ D_j \models \#(T_2, A_{12}, \langle P_3 \rangle) \]

• FV-6: From FV-4, FV-5, and by applying NVR, we achieve
  \[ D_j \models \#(T_2, A_{12}, \langle P_3 \rangle) \]

• 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
  \[ D_j \models (D_j \overset{SK}{\leftrightarrow} MD_j). \]

• 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
  \[ MD_j \models \#(T_3, P_4, (D_j \overset{SK}{\leftrightarrow} MD_j)). \]

• 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 \( MD_j \models P_4 \).

• FV-13: Using FV-12, by using A-2, and A-4, G-2 is achieved
  \[ MD_j \models (MD_j \overset{SK}{\leftrightarrow} D_j). \]

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

3) SECURITY ANALYSIS USING SCYTHER 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
TABLE 4. Security features comparison.

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

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/Portery attack; SFC10: BSL attack; SFC11: Sensor/drone capture attack; SFC12: Password update phase; SFC13: Biometric update; SFC14: Revocable phase; SFC15: Unauthorized login detection; ✓: Shows feature is supported; x: 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(., Rep(., R_{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\} = 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 cost comparison.
cost comparison of the proposed LAKE-IoD and other related AKE schemes.

### TABLE 5. Storage overhead.

| Scheme | User Side | Server Side | Denote Side | Total |
|--------|-----------|-------------|-------------|-------|
| Wazid [6] | 128 bits | 1120 bits | 148 bits | 1356 bits |
| Das [24] | 768 bits | 992 bits | 128 bits | 1988 bits |
| Challa [26] | 648 bits | 816 bits | 100 bits | 1864 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 \( AUTH_{tag} \approx 128 \), 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 \( M_1 \): \( T_1, A_1, C_1^{au}, C_2^{au}, AUTH_{tag1} \), \( M_2 \): \( T_2,A_{12}, C_1^{agt}, AUTH_{tag3} \), and \( M_3 \): \( T_3, \hat{A}_{19}, C_1^{agt}, AUTH_{tag5} \) with length \( M_{au1} = 32 + 128 + 128 + 128 + 544 \) 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, 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 \( 13T_{BU} + 6T_{AG} + 1T_{BU} \approx 0.8943 \) ms, \( 31T_{SH} + 17T_{BU} \approx 1.2114 \) ms, \( 30T_{SH} + 1T_{BU} \approx 1.1803 \) ms, \( 12T_{SH} + 14T_{ec} + T_{BU} \approx 3.8354 \) ms, \( 30T_{SH} + 1T_{BU} \approx 1.1803 \) ms, and \( 19T_{BU} + 3T_{ec} + T_{BU} \approx 0.1555 \) ms computational cost at the drone/sensor side, respectively. LAKE-IoD requires slightly high computation cost at drone side as compared to

| Scheme | Total (bits) |
|--------|--------------|
| Wazid [6] | 1696 |
| Das [24] | 1536 |
| Challa [26] | 2528 |
| Srinivas [7] | 1536 |
| Challa [22] | 1428 |
| LAKE-IoD | 1376 |

Note: \( MU/U \) is the mobile user, \( MS \) is a management server, \( GW \) is the gateway, \( CS \) is the control server, \( D_i \) is the drone, and \( S_{nj} \) 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                |
| $T_{AE}$ | AEGIS                  | 0.056                 |
| $T_{EC}$ | ECC point addition     | 0.2475                |
| $T_{AC} + T_{EC}$ | Puzzy extractors | 0.2475                |

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 constrained 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. 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. Srivinas, A. K. Das, N. Kumar, and J. P. C. Rodrigues, “TACLAS: 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. Intell. 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, “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. Chilamkurthy, 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. Kumar, M. Usman, K. Mahmood, and M. S. Faisal, “An improved lightweight multi-server 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, “Anonymious 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 multisever environment,” Secure 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. 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.
MUHAMMAD TANVEER received the B.S. degree in electronics from GCU Lahore, Pakistan, and the M.S. degree in computer science from the Institute of Management of Sciences (IMS), Lahore, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Computer Sciences and Engineering. He is also a member of the Telecommunications and Networking (TeleCoN) Research Laboratory, GIK Institute of Engineering Sciences and Technology. His current research 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 (NCRC), 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 ETI 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, M.L., AI, VLSI design, EDAC/AD tools, embedded systems, 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 TRANSACTIONS ON NETWORKS AND LEARNING SYSTEMS, JOURNAL OF INFORMATION SECURITY AND APPLI CATIONS, and the IEEE JOURNAL OF INFORMATION SECURITY AND APPLICATIONS. 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.

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.