SPECIAL SECTION ON HUMAN-DRIVEN EDGE COMPUTING (HEC)

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 11, 2020, accepted February 26, 2020, date of publication March 2, 2020, date of current version March 12, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2977817*

# Securing Smart City Surveillance: A Lightweight Authentication Mechanism for Unmanned Vehicles

**ZEESHAN ALI[1], SHEHZAD ASHRAF CHAUDHRY[2], MUHAMMAD SHER RAMZAN[3], AND FADI AL-TURJMAN[4,5]**
[1]Department of Computer Science, International Islamic University Islamabad, Islamabad 44000, Pakistan
[2]Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelişim University, 34310 Istanbul, Turkey
[3]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[4]Artificial Intelligence Engineering Department, Near East University, 99138 Nicosia, Turkey
[5]Research Center for AI and IoT, Near East University, 99138 Nicosia, Turkey

Corresponding authors: Shehzad Ashraf Chaudhry (sashraf@gelisim.edu.tr) and Muhammad Sher Ramzan (msramadan@kau.edu.sa)

This work was supported by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant D-480-611-1441.

**ABSTRACT** The significance of the Internet of Drones (IoD) is increasing steadily and now IoD is being practiced in many military and civilian-based applications. IoD facilitates real-time data access to the users especially the surveillance data in smart cities using the current cellular networks. However, due to the openness of communication channel and battery operations, the drones and the sensitive data collected through drones are subject to many security threats. To cope the security challenges, recently, Srinivas et al. proposed a temporal credential based anonymous lightweight authentication scheme ($TCALAS$) for IoD networks. Contrary to the IoD monitoring framework proposed by Srinivas et al., their own scheme can work only when there is one and only one cluster/flying zone and is not scalable. Moreover, despite their claim of robustness, the investigation in this paper reveals that Srinivas et al.'s scheme cannot resist traceability and stolen verifier attacks. Using the lightweight symmetric key primitives and temporal credentials, an improved scheme ($iTCALAS$) is then proposed. The proposed scheme while maintaining the lightweightness provides security against many known attacks including traceability and stolen verifier. The proposed $iTCALAS$ extends scalability and can work when there are several flying zone/clusters in the IoD environment. The formal security proof along with automated verification using ProVerif show robustness of proposed $iTCALAS$. Moreover, the security discussion and performance comparisons show that the $iTCALAS$ provides the known security features and completes authentication in just 2.295 $ms$.

**INDEX TERMS** Surveillance, security, key-agreement, drones, IoT, IoD, session key leakage, traceability, user anonymity.

## I. INTRODUCTION

Continuous progression in information and telecommunication, hardware and software is playing a vital role in the development and increasing usage of the Internet of Things (IoT) with the abundance of connected devices increasing by the day [1]–[3]. The exceptional unprecedented propagation of IoT devices like smart-phones, medical sensors, fitness trackers etc. has permitted people to share data [4]–[6] seamlessly. IoT enables various physical devices to communicate and collaborate and these devices can be used in a variety of fields and applications [7], [8]. IoT devices are smart enough that they can make decisions and interact with each other without the involvement of the humans. Internet of Drones (IoD) is neologized by supplanting "'Things"' with "'Drones"' from IoT while offering related properties. IoD transpires to mature into an indispensable breakthrough in the advancement of drones [9]. Gharibi *et al.* [10] described IoD being a "layered network control architecture", which supports drones in coordinating. In an IoD environment, multiple drones consolidate and create a network while conveying and acquiring data from one another. The physical and hardware

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

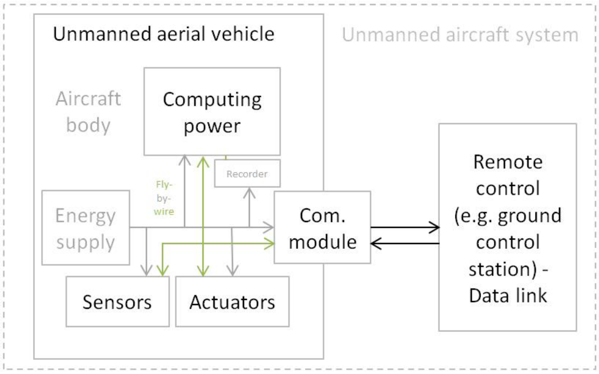


**FIGURE 1.** Block diagram of a typical drone system.

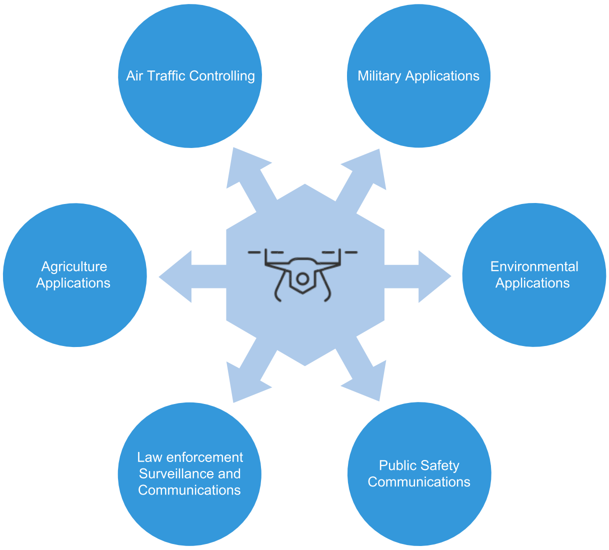


**FIGURE 2.** IoD application areas.

structure of a typical drones also known as unmanned ariel vehicle (UAV) or unpiloted aircraft [11] is shown in Figure 1. Components of drone include a battery, multiple rotors, Inertial Measurement Unit (IMU) and a flight controller.

Currently, IoD is being widely used for surveillance, environmental monitoring, distribution delivery and in a variety of areas as presented in Figure 2.

The drones safety can be improved by tracking them and can be utilized to circumvent accidents, enhanced traffic performance, and restrain the flights of illegal drones by recognizing the more congested airspace. Most drones use Micro Aerial Vehicle Link (MAVLink) protocol for communication and telemetry functionality to monitor their status [12], [13]. The UAVs forms a collaborative network of drones (IoD) [14] to gather and consolidate environment related data such as surveillance data in smart cities or battle field monitoring, the data is further send to the controlling user through some ground center [15], [16]. As per [17]–[19] the prospect of drones as a commercial usage is not far off it has already begun and along with usage in many B2B application, IoD has become one of the most invested technology for business. Currently, IoD is being used as a tool in variety of areas like a package delivery option but are also being used as a tool for police, first-aid vehicles, high-tech photography, wildlife research, search, rescue and many more [17], [18] as shown in Figure 2. Due to sensitivity of environment data, the security of such unmanned vehicles has got much importance as an attacker can use drones for depraved purposes like modification of genuine environment related data or can stop it to communicate with users. Moreover, the drones are battery operated and equipped with small memory and communication capabilities. Therefore, IoD requires a security mechanism to avoid unauthorized access and to provide data integrity along with confidentiality. Moreover, resource constrained nature of drones demands security procedure based on lightweight cryptographic operations. Lamport was the first to propose authentication mechanism for remote user/device, till then many such schemes are proposed [20]–[25]. An authentication scheme for Wireless Sensor Networks (WSNs) and IoT was proposed by Turkanović *et al.* [22]. Farash *et al.* [23] discovered that [22] is exposed to stolen smart card, Man-in-the-middle and sensor node impersonation and related attacks. As a solution, Farash et al. introduced a new efficient scheme to subdue beforehand mentioned vulnerabilities. However, Amin *et al.* [24] later proved that [23] scheme is also defenseless against many attacks including user impersonation, off-line password guessing etc., Amin et al. also showed that Farash et al.'s scheme lacks user anonymity. Later, Jiang *et al.* [25] ascertained that [24] is similarly unsafe and has some loopholes. To surmount Jiang *et al.* [25] proposed a new refined security scheme. Tai *et al.* [26] also offered an authentication scheme however, it lacks forward secrecy and is weak against password guessing, privileged-insider, replay and man-in-the-middle attack. Challa *et al.* [27] also proposed ECC and signature based authentication scheme. Due to usage of ECC and signature, the scheme [27] demands very high communication and computation cost. Moreover, the scheme proposed in [27] entails some correctness issues. Roy *et al.* [28] likewise proposed a three-factor (smart card, password and biometrics) based authentication and key-agreement scheme for crowdsourcing IoT. Similarly, Das *et al.* also proposed an authentication scheme for industrial IoT using trusted gateway as an intermediate party [29]. However, Sajid and Chaudhry [30] proved that their scheme is insecure against stolen verifier and smart device attacks and does not provide user traceability and forward secrecy. Amin et al. also proposed another scheme [31] for three party settings. Challa *et al.* [32] argued scheme proposed in [31] is vulnerable to user impersonation, stolen card and related attack. Chaudhry *et al.* [33] analyzed that the scheme of Challa *et al.* [32] has incorrect authentication procedure and in prone to some other weaknesses. In 2018 Jangirala *et al.* [17] proposed a tailored authentication scheme (*TCALAS* : Temporal Credential based Anonymous Lightweight Authentication Scheme) for pure IoD environments. Although, the scheme was proposed using lightweight symmetric hash functions, making it work in resource limited unmanned drones, the analysis in this article shows that their

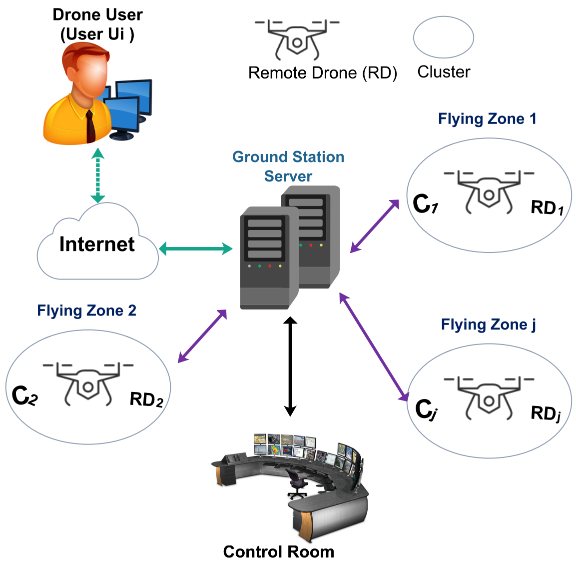


**FIGURE 3.** IoD environment monitoring system.

scheme can work with only one flying zone and is not scalable. Moreover, *TCALAS* lacks untraceability property and is defenseless against stolen verifier attack. It is argued that an attacker after stealing verifier can impersonate on behalf of any of the drone, user and *GSS*. Then an improved Temporal credential based anonymous lightweight authentication scheme (*iTCALAS*) is proposed in this paper. The security of *iTCALAS* is proved through formal, informal and automated methods. Rest parts of the paper is arranged as follows: IoD Authentication scenario and threat model are presented in subsection I-A, I-B respectively. Review of the scheme of Srinivas et al. for securing IoD is conducted in Section II followed by it's cryptanalysis in Section III. The proposed improved scheme is presented in Section IV. The formal, informal and automated security analysis of the proposed scheme is shown in Section V. The performance and security feature comparisons are given in Section VI.The paper is finally concluded in Section VII.

### A. AUTHENTICATION SCENARIO

The realistic authentication scenario adopted from [17] is depicted in Figure 3. Comprising of three participants, Ground Station Server *GSS* is assumed to be trusted and facilitates the session initiation between users and drones with in a specified cluster. The communication between the communicating entities is always through public channel and the drones are flying in specified zones called as clusters, as of a drone, a cluster has also it's unique identity; whereas, *GSS* is attached with a control room. The drones are allowed to communicated with users/*GSS* and with each other. In [17], the *GSS* was assumed to be locked physically and no one can access *GSS* memory. However, in this paper only the secret key of the *GSS* is assumed to be non-compromised. The rest of the contents stored on physically locked *GSS* are subject to compromise because no physical lock can restrict a cyber attacker to get data on a machine attached with public internet [30].

**TABLE 1.** Notations guide.

| Symbols | Representations |
|---|---|
| $U_i, RD_j, GSS$ | $i^{th}$ user, $j^{th}$ remote drone and ground station server, respectively |
| $MD_i$ | Mobile device of $U_i$ |
| $X_{GSS}, X_{RD_j}$ | Long-term secret keys of $GSS$ and $RD_j$, respectively |
| $SID_{RD_j}$ | Secret key between $GSS$ and $RD_j$ |
| $UID_i$ | Secret key between $U_i(MD_i)$ and $GSS$ |
| $ID_i, ID_{RD_j}, ID_{GSS}$ | Unique identities of $MD_i$, $RD_j$ and $GSS$, respectively |
| $PW_i, BIO_i$ | Password and biometrics of $U_i$, respectively |
| $h(.)$ | Cryptographic one way hash function |
| $SK$ | Session key between $U_i$ and $RD_j$ |
| $b_i, R_1, R_2, R_3$ | Random numbers |
| $T_1, T_2, T_3$ | Current timestamps |
| $Gen(.), Rep(.)$ | Fuzzy biometric generator and reproduction functions, respectively |
| $\triangle T$ | Maximum allowable transmission delay |
| $TC_i$ | Temporal credential of $U_i$ |
| $i \overset{?}{=} j$ | Checks if $i$ equals to $j$ |
| $CID_k$ | Identity of $k^{th}$ cluster in the flying zones |
| $\oplus, \|\|$ | Bitwise XOR and concatenation operators, respectively |
| $\mathcal{A}, \mathcal{I}, U_{\mathcal{A}}$ | An adversary, intruder and privileged insider, respectively |

### B. THREAT MODEL

The common adversarial model as adopted in [34]–[39] is considered for authentication scenario in IoD based deployments. Precisely, the attacker ($\mathcal{A}$) is assumed to have following capabilities:

1) $\mathcal{A}$ has authority over the whole public communication link and $\mathcal{A}$ can intervene, rerun, alter, drop or can forward a new forged message.
2) With the help of power analysis, $\mathcal{A}$ can access information embedded in the smart card [34], [39].
3) $\mathcal{A}$ can be an outsider or can be an ambitious system user.
4) The identities of users and server are public.
5) *GSS* is protected and no adversary can compromise the private key of *GSS*.

## II. SCHEME OF THE SRINIVAS ET AL.

This section describes the authentication scheme (*TCALAS*) for IoD designed by Srinivas et al. Various symbols adopted in the paper are outlined in Table 1. Based on three factors including biometrics, password and smart device, the phases of the scheme are briefed in following subsections:

### A. PRE-DEPLOYMENT PHASE

For pre-deployment, each remote drone $RD_j : \{j = 1, 2 \ldots .m\}$ is initially enrolled with the $GSS$. $GSS$ assigns each $RD_j$ a distinct identity $ID_{RD_j}$ before placing those into any area partitioned as $n_c$ disjoint clusters (flying zones) with a $CID_k$ as identity. $GSS$ chooses its own identity $ID_{GSS}$, secret key $X_{GSS}$ and $X_{RD_j}$ a long-term shared secret with $RD_j$. Then $GSS$ calculates $SID_{RD_j} = h(CID_k||ID_{RD_j}||X_{GSS}||X_{RD_j})$ and selects a hash function $h(\cdot)$. Finally, $GSS$ stores the $\{ID_{GSS}, CID_k, ID_{RD_j}, SID_{RD_j}, h(\cdot)\}$ into $RD_j$'s memory and

$\{ID_{GSS}, \{CID_k|1 \le k \le n_c\}, \{(ID_{RD_j}, SID_{RD_j})|1 \le j \le n_r\}$ in its own database, $n_c$ indicates the number of drones to be placed in a cluster.

### B. SRINIVAS ET AL.'S USER REGISTRATION PHASE

To register for accessing a drone $RD_j$ in some cluster $k$, $U_i$ is required to enroll with the $GSS$. Initially, $U_i$ picks $ID_i$, $PW_i$ and $b_i$. $U_i$ computes $HID_i = h(ID_i||b_i)$, $HPW_i = h(PW_i||b_i)$ and forwards the registration request $\{HID_i, h(\cdot)\}$ to $GSS$. On receiving $U_i$'s request, $GSS$ computes $UID_i = h(HID_i \,||\, X_{GSS})$, $TC_i = h(CID_k \,||\, UID_i \,||\, \mathrm{ID}_{GSS})$, $A_i = UID_i$, and $B_i = CID_k \oplus h(HID_i \,||\, UID_i)$. The $GSS$ then saves $\{A_i, B_i, TC_i, h(\cdot), ID_{GSS}, CID_k\}$ into the mobile device $MD_i$, and transfers the $MD_i$ securely to $U_i$. Next, $U_i$ imprints his/her biometric $BIO_i$ and calculates $Gen(BIO_i) = (\sigma_i, \tau_i)$, $L_i = b_i \oplus h(\sigma_i||ID_i||PW_i)$, $M_i = h(A_i \,||\, TC_i \,||\, b_i \,||\, \sigma_i)$, and $A'_i = A_i \oplus h(b_i \,||\, HID_i \,||\, HPW_i \,||\, \sigma_i)$, where $\sigma_i$ is the secret biometric key and $\tau_i$ is public reproduction parameter related with $BIO_i$ [28], respectively. Finally, $U_i$ saves the credentials $\{A'_i, ID_{GSS}, M_i, B_i, L_i, h(\cdot), CID_k, Rep(\cdot), Gen(\cdot), \tau_i\}$ in the $MD_i$.

### C. SRINIVAS ET AL.'S LOGIN AND AUTHENTICATION PHASE

To access the $RD_j$ in a desired flying zone $k$, $U_i$ needs to prove his legality to $MD_i$ as well as to $GSS$. $U_i$ initiates this phase and the process completes by executing following steps:

SLA 1: $U_i$ provides the login credentials ($BIO'_i$, $ID_i$ & $PW_i$) to $MD_i$. $MD_i$ then calculates $\sigma'_i = Rep(BIO'_i, \tau_i)$, $b_i = L_i \oplus h(\sigma'_i||ID_i||PW_i)$, $HID_i = h(ID_i||b_i)$, $HPW_i = h(PW_i||b_i)$, $A_i = A'_i \oplus h(b_i||HID_i||HPW_i||\sigma'_i)$, $UID_i = A_i$, $CID_k = B_i \oplus h(HID_i||UID_i)$ and $TC_i = h(CID_k||UID_i||ID_{GSS})$. $MD_i$ verifies $M_i \overset{?}{=} h(A_i||TC_i||b_i||\sigma'_i)$, session ends, if verification fails. Otherwise, $MD_i$ generates $T_1$, $R_1$ and computes $U_1 = HID_i \oplus h(T_1 \oplus ID_{GSS}||CID_k)$, $U_2 = ID_{RD_j} \oplus h(UID_i||CID_k||TC_i)$, $U3 = h(ID_{RD_j}||CID_k||TC_i||T_1) \oplus R_1$, and $U_4 = h(R_1||UID_i||ID_{RD_j}||TC_ik||CID_k)$. $U_i$ then transmits $MSG_1 = \{U_1, U_2, U_3, U_4, T_1\}$ to $GSS$.

SLA 2: On receiving, the $GSS$ checks the freshness of the $MSG_1$ (through $|\ T_c - T_1\ | < \Delta T$); in case it is fresh, $GSS$ calculates $HID^*_i = U_1 \oplus h(T_1||ID_{GSS}||CID_k)$ and $UID^*_i = h(HID^*_i||X_{GSS})$. $GSS$ withdraws $TC_i$ by checking if $UID^*_i$ exists in the database, in case it is true, the $GSS$ checks if $ID_{RD_j}$ also exists in $GSS$ database by computing $ID_{RD_j} = U_2 \oplus h(UID_i||CID_k||TC_i)$. On success, $GSS$ calculates $R_1 = U_3 \oplus h(ID_{RD_j}||CID_k||TC_i||T_1)$, fetches $SID_{RD_j}$ corresponding to $ID_{RD_j}$ and verifies $U_4 \overset{?}{=} h(R_1||UID_i||ID_{RD_j}||TC_i||CID_k)$. Upon unsuccessful validation, the $GSS$ rejects the $U_i$'s legitimacy and terminates the session. Otherwise, the $GSS$ continues by generating $R_2$ and current timestamp $T_2$, and computes $U_5 = h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_2) \oplus HID_i$, $U_6 = h(HID_i||ID_{RD_j}||CID_k||T_2||h(R_1||R_2))$ and $U_7 = h(HID_i||ID_{RD_j}||SID_{RD_j}||T_2) \oplus h(R_1||R_2)$. $U_i$ then transmit the message $MSG_2 = \{U_5, U_6, U_7, T_2\}$ to the remote drone $RD_j$.

SLA 3: On receiving $GSS$ message, $RD_j$ checks the freshness ($|T_c - T_2| < \Delta T$) and on success, $RD_j$ computes $HID_i = U_5 \oplus h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_2)$, $h(R_1||R_2) = U7 \oplus h(HID_i||ID_{RD_j}||SID_{RD_j}||T_2)$. $RD_j$ then checks $U_6 \overset{?}{=} h(HID_i||ID_{RD_j}||CID_k||T_2|| h(R_1||R_2))$. If fails, $RD_j$ declines the message. Otherwise, $RD_j$ selects $T_3$, $R_3$ and computes $R'_3 = h(R_3||h(R_1||R_2))$, $U_8 = R'_3 \oplus h(HID_i||ID_{RD_j}||T_3||CID_k)$, $SK = h(R'_3||HID_i|| ID_{RD_j}||CID_k||T_3)$ and $U_9 = h(R'_3||SK||T_3||CID_k)$. $RD_j$ then sends the message $MSG_3$ containing $\{U_8, U_9, T_3\}$ directly to $U_i$ via open channel.

SLA 4: The $U_i$ checks the freshness ($|\ T_c - T_3\ | < \Delta T$,) of the $MSG_3$ and on success computes $R'_3 = U_8 \oplus h(HID_i||ID_{RD_j}||T_3||CID_k)$, $SK = h(R'_3||HID_i||ID_{RD_j}||CID_k||T_3)$. $U_i$ then verifies if $U_9 \overset{?}{=} h(R'_3||SK||T_3||CID_k)$, if the condition holds $RD_j$ is verified successfully else session is terminated. Conclusively, $RD_j$ and $U_i$ both have the $SK = h(h(R_3||h(R_1||R_2))||HID_i||ID_{RD_j}||CID_k||T_3)$ as a session key.

### D. USER PASSWORD/BIOMETRIC UPDATE PHASE

In this phase the $U_i$ can update both his biometric and password. For renewing the password/biometrics, a legitimate registered $U_i$ with $MD_i$ provides($BIO'_i$, $ID_i$ & $PW_i$). $MD_i$ then calculates: $\sigma'_i = Rep(BIO'_i, \tau_i)$, $b_i = L_i \oplus h(\sigma'_i||ID_i||PW_i)$, $HID_i = h(ID_i||b_i)$, $HPW_i = h(PW_i||b_i)$, $A_i = A'_i \oplus h(b_i||HID_i||HPW_i||\sigma'_i)$, $UID_i = A_i$, $CID_k = B_i \oplus h(HID_i||UID_i)$ and $TC_i = h(CID_k||UID_i||ID_{GSS})$. $MD_i$ then verifies $M_i \overset{?}{=} h(A_i||TC_i||b_i||\sigma'_i)$. Session ends, if the authentication fails. Otherwise, $MD_i$ informs $U_i$ to input new password $PW_i^{new}$ and biometric $BIO_i^{new}$. $U_i$ provides a new password $PW_i^{new}$ and biometrics $BIO_i^{new}$ to $MD_i$. $MD_i$ calculates $HPW_i = h(PW_i^{new}||b_i)$, $HID_i = h(ID_i||b_i)$, $(\sigma_i^{new}, \tau_i^{new}) = Gen(BIO_i^{new})$, $L_i^{new} = b_i \oplus h(\sigma_i^{new}||ID_i||PW_i^{new})$, $M_i^{new} = h(A_i||TC_i||b_i||\sigma_i^{new})$, and $A_i^{new} = A_i \oplus h(b_i||HID_i||HPW_i^{new}||\sigma_i^{new})$. Finally, $U_i$ replaces $A'_i$, $M_i$ and $L_i$ with $A_i'^{new}$, $M_i^{new}$ and $L_i'^{new}$, respectively, in the mobile device $MD_i$.

### E. REVOCATION AND REISSUE PHASE

For changing device $MD_i$ with new on $MD_i^{new}$, $U_i$ provides the old identity $ID_i$, a new password $PW_i^{new}$, chooses an arbitrary number $b'_i$ and sends $\{HID_i, h(\cdot)\}$ to the $GSS$ over the secure channel where $HPW_i^{new} = h(PW_i^{new}||b'_i)$ and $HID_i = h(ID_i||b'_i)$. On receiving request, $GSS$ computes $UID_i = h(HID_i \,||\, X_{GSS})$, $TC_i = h(CID_k \,||\, UID_i \,||\, \mathrm{ID}_{GSS})$, $A_i = UID_i$, $B_i = CID_k \oplus h(HID_i \,||\, UID_i)$ and transfers the $MD_i^{new} = \{A_i, B_i, TC_i, h(\cdot), ID_{GSS}, CID_k\}$ to the $U_i$ over the secure channel. Next, $U_i$ imprints his/her biometric $BIO_i^{new}$ and calculates $Gen(BIO_i^{new})$

$= (\sigma_i^{new}, \tau_i)$, $L_i^{new} = b_i^{new} \oplus h(\sigma_i^{new}||ID_i||PW_i^{new})$, $M_i^{new} = h(A_i||TC_i||b_i||\sigma_i^{new})$, and $A_i^{'new} = A_i \oplus h(b_i||HID_i||HPW_i^{new}||\sigma_i^{new})$. Finally, $U_i$ deletes $TC_i$ and saves the parameters $\{A_i^{'new}, M_i^{new}, L_i^{new}, ID_{GSS}, B_i^{new}, h(\cdot), CID_k, Rep(\cdot), Gen(\cdot), \tau_i\}$ in the $MD_i$.

### F. DYNAMIC REMOTE DRONE ADDITION PHASE

This phase facilitates adding new drones in an existing IoD network. For drone addition purposes, $GSS$ selects a distinct identity $ID_{RD_j}^{new}$, $X_{RD_j}^{new}$ for $RD_j^{new}$ and computes $SID_{RD_j}^{new} = h(CID_k||ID_{RD_j}^{new}||X_{GSS}||XRD_{RD_j^{new}})$ using $X_{GSS}$. $GSS$ finally, stores the parameters $\{ID_{GSS}, CID_k,$
$ID_{RD_j}^{new}, SID_{RD_j}^{new}, h(\cdot)\}$ in $RD_j^{new}$'s memory and $\{ID_{RD_j}^{new},$
$SID_{RD_j}^{new}\}$ in its database.

## III. WEAKNESSES OF THE SCHEME OF SRINIVAS ET AL.

In this section, we show the weaknesses of the *TCALAS* proposed by Srinivas et al. Precisely, it is to prove in following subsections that the scheme of *TCALAS* cannot resist traceability and stolen verifier attacks:

### A. SCALABILITY ISSUES

The scheme of Srinivas et al. can work with drones flying in just one cluster. If there are more than one clusters, the scheme may fail to facilitate the authentication process. Precisely, in step *SLA-1*, $U_i$ having device $MD_i$ engraved with $\{A_i', ID_{GSS}, M_i, B_i, L_i, h(\cdot), CID_k\}$ computes and sends $MSG_1 = \{U_1, U_2, U_3, U_4, T_1\}$ to $GSS$, where $U_1 = HID_i \oplus h(T_1 \oplus ID_{GSS}||CID_k)$, $U_2 = ID_{RD_j} \oplus h(UID_i||CID_k||TC_i)$, $U3 = h(ID_{RD_j}||CID_k||TC_i||T_1) \oplus R_1$, and $U_4 = h(R_1||UID_i|| ID_{RD_j}||TC_ik||CID_k)$. Upon receiving $MSG_1$, in step SLA-2, the $GSS$ checks the freshness of the $MSG_1$ (through $| T_c - T_1 | < \triangle T$); in case it is fresh, $GSS$ computes:

$$HID_i^* = U_1 \oplus h(T_1||ID_{GSS}||CID_k) \tag{1}$$

$$UID_i^* = h(HID_i^*||X_{GSS}) \tag{2}$$

The computation of $HID_i^*$ in Eq. 1 requires to compute $h(T_1||ID_{GSS}||CID_k)$ first. Here, $T_1$ is received by $GSS$ in $MSG_1$ and $ID_{GSS}$ is the real identity of $GSS$; whereas, $CID_k$ is the identity of $k^{th}$ flying zone. The message request $MSG_1$ does not contain any information about the user or the flying zone. The user identity is recognized, only when $GSS$ has information of flying zone/cluster i.e. $CID_k$ (see Eq.1). If there are more than one (say $n_c$) clusters: $CID_x : \{x = 1, 2 \ldots k, ..n_c\}$, then $GSS$ cannot compute $HID_i^*$ of $U_i$ because $GSS$ is now unable to determine which $CID_x$, it has to use for computation of $HID_i^*$ through Eq.1, and the process may not continue further. Moreover, computation of $UID_i^*$ in Eq. 2 is also depends on accurate knowledge of $HID_i^*$. Similarly, $GSS$ cannot perform rest of the authentication steps. Hence, in presence of more than one drone clusters registered with $GSS$, the scheme fails to provide authentication between a user and a specified drone. Hence, the scheme of Srinivas et al. for securing drones is not scalable and can work with only one flying zone/cluster.

### B. TRACEABILITY ATTACK

This section shows the weakness of the Srinivas et al. against traceability attack. An attacker $\mathcal{A}$, insider or outsider can easily trace any user by using the public information $ID_{GSS}$ and $CID_k$ along with the timestamp $T_1$ sent on public channel in a message $\langle MSG_1 = \{U_1, U_2, U_3, U_4, T_1\}\rangle$ by a user $U_i$. The attacker can compute $HID_i = U_1 \oplus (T_1||ID_{GSS}||CID_k)$, the $HID_i$ of a user remains same for all sessions. Therefore, $\mathcal{A}$ can easily launch traceability attack on Srinivas et al.'s scheme.

### C. IMPERSONATION BASED ON STOLEN VERIFIER

In Srinivas et al.'s scheme the Ground Station Server ($GSS$) maintains two verifier database, one for users with entries of type $\{UID_i, TC_i\}$, second for drones with entries of type $\{ID_{RD_j}, SID_{RD_j}\}$. A privileged insider $\mathcal{A}$ of the system with access to drone verifier database can impersonate as $GSS$ to the remote drone ($DR_j$) by executing following steps:

1) $\mathcal{A}$ generates a random identity $RID_a$, current timestamp $T_2^{\mathcal{A}}$, two numbers $R_1^{\mathcal{A}}$ and $R_2^{\mathcal{A}}$ randomly. $\mathcal{A}$ now computes:

$$U_5^{\mathcal{A}} = h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_22^{\mathcal{A}}) \oplus RID_a \tag{3}$$

$$U_6^{\mathcal{A}} = h(RID_a||ID_{RD_j}||CID_k||T_2^{\mathcal{A}}||h(R_1^{\mathcal{A}}||R_2^{\mathcal{A}})) \tag{4}$$

$$U_7^{\mathcal{A}} = h(RID_a||ID_{RD_j}||SID_{RD_j}||T_2^{\mathcal{A}}) \oplus h(R_1^{\mathcal{A}}||R_2^{\mathcal{A}}) \tag{5}$$

$\mathcal{A}$ sends the message $MSG_2 = \{U_5^{\mathcal{A}}, U_6^{\mathcal{A}}, U_7^{\mathcal{A}}\}$ to $DR_j$

2) $RD_j$ receives $MSG_2$ and checks the validity of timestamp $T_2^{\mathcal{A}}$; upon success, $RD_j$ computes:

$$RID_a = h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_22^{\mathcal{A}}) \oplus U_5^{\mathcal{A}} \tag{6}$$

$$h(R_1^{\mathcal{A}}||R_2^{\mathcal{A}}) = h(RID_a||ID_{RD_j}||SID_{RD_j}||T_2^{\mathcal{A}}) \oplus U_7^{\mathcal{A}} \tag{7}$$

$RD_j$ further checks the equality:

$$U_6^{\mathcal{A}} \overset{?}{=} h(RID_a||ID_{RD_j}||CID_k||T_2^{\mathcal{A}}||h(R_1^{\mathcal{A}}||R_2^{\mathcal{A}})) \tag{8}$$

3) Upon successful verification of Eq. 8, $DR_j$ generate $T_3$, $R_3$ and computes:

$$R_3' = h(R_3||h(R_1^{\mathcal{A}}||R_2^{\mathcal{A}}) \tag{9}$$

$$U_8 = R_3' \oplus h(RID_a||ID_{RD_j}||T_3||CID_k) \tag{10}$$

$$SK = h(R_3'||RID_a||ID_{RD_j}||CID_k||T_3) \tag{11}$$

$$U_9 = h(R_3'||SK||T_3||CID_k) \tag{12}$$

$RD_j$ then sends the message $MSG_3$ containing $\{U_8, U_9, T_3\}$ directly to $U_i$.

4) $\mathcal{A}$ intercepts $MSG_3$ and computes:

$$R_3' = U_8 \oplus h(RID_a||ID_{RD_j}||T_3||CID_k) \tag{13}$$

Finally, $\mathcal{A}$ computes session key as follows:

$$SK = h(R_3'||RID_a||ID_{RD_j}||CID_k||T_3) \tag{14}$$

*Proposition 1:* In Srinivas et al.'s scheme, on execution of stolen verifier attack, an active attacker $\mathcal{A}$ can impersonate himself as legal $GSS$ and an arbitrary legal user $U_a$ simultaneously, to the drone ($DR_j$) of his choice. Moreover, $\mathcal{A}$ can share a session key with $DR_j$ accurately for establishment of a secure session.

*Proof 1:* $\mathcal{A}$ initiates impersonation on behalf of $GSS$ by computing and sending $MSG_2 = \{U_5^{\mathcal{A}}, U_6^{\mathcal{A}}, U_7^{\mathcal{A}}\}$ to $DR_j$. The drone $DR_j$ considers $\mathcal{A}$ as legal $GSS$ if timestamp is fresh and Eq. 8 holds. It can be clearly observed that $\mathcal{A}$ generated fresh timestamp $T_2^{\mathcal{A}}$ for initiation of impersonation, so freshness will be verified without any hindrance. $\mathcal{A}$ computed $U_6^{\mathcal{A}} = h(RID_a||ID_{RD_j}||CID_k||T_2^{\mathcal{A}}||h(R_1^{\mathcal{A}}||R_2^{\mathcal{A}}))$ in Eq. 4, out of the parameters used for computing $U_6^{\mathcal{A}}$, $\{RID_a, T_2^{\mathcal{A}}, h(R_1^{\mathcal{A}}, R_2^{\mathcal{A}}\}$ are generated by $\mathcal{A}$ himself, while $ID_{DR_j}$ and $CID_k$ are extracted from stolen verifier. Moreover, as proved in subsection III-A, there is only one cluster being used in Srinivas et al.'s scheme the $CID_k$ is then known to everyone. Therefore, $U_6^{\mathcal{A}}$ computed by $\mathcal{A}$ in Eq. 4 is same as $DR_j$ computes in Eq. 8. Hence, Eq. 8 holds. Furthermore, $DR_j$ computes session key in Eq. 11 and $\mathcal{A}$ computes session key in Eq. 14. The session keys on both sides are also same because $\mathcal{A}$ extracts $R_3'$ in Eq. 13 using the parameters either he got through stolen verifier or he generated by himself; whereas rest of the parameters $\{RID_a, ID_{RD_j}, CID_k, T_3\}$ involved in computation of session key are already in his access. Therefore, the session key computed on both sides is also same. Hence, $\mathcal{A}$ has successfully, impersonated simultaneously on behalf of a legal user as well as $GSS$ to a drone $DR_j$ and shared a session key.

Similarly, using the verifiers, $\mathcal{A}$ can be successful to impersonate himself as a drone or as a legal user to other parties of the system.

## IV. PROPOSED SCHEME

In this section an improved scheme (*iTCALAS*) is presented to mitigate the loopholes of Srinivas et al.'s scheme. For the *iTCALAS* pre-deployment phase is taken as from Srinivas et al.'s scheme, the brief description of the rest of the phases of *iTCALAS* are given in following subsections:

### A. USER REGISTRATION PHASE

To register for accessing a drone $RD_j$ in some cluster $k$, $U_i$ is required to enroll with the $GSS$. Initially, $U_i$ picks $ID_i$and sends it to $GSS$ using secure channel. On receiving $ID_i$, $GSS$ selects arbitrary number $r_s$ and computes $UID_i = E_{X_{GSS}}(ID_i, r_s)$, $UK_i = h(ID_i||X_{GSS})$, $B_i = CID_k \oplus h(ID_i||UK_i)$ and temporal credential $TC_i = h(CID_k||ID_{GSS}||ID_i||UK_i)$. Finally $GSS$ saves the parameters $\{UID_i, UK_i, B_i, TC_i\}$ into the mobile device $MD_i$, and transfers the $MD_i$ securely to the $U_i$. Next, $U_i$ selects $b$,$PW_i$, imprints his/her biometric $BIO_i$ and calculates $Gen(BIOi) = (\sigma_i, \tau_i)$, $A_i = UID_i \oplus h(ID_i||PW_i||\sigma_i)$, $L_i = b \oplus h(PW_i||ID_i||\sigma_i)$, $\overline{UK}_i = UK_i \oplus h(\sigma_i||PW_i||ID_i||b)$ and $M_i = h(b||UID_i||UK_i||PW_i||\sigma_i)$, and $A_i' = A_i \oplus h(b_i \,||\, HID_i \,||\, HPW_i \,||\, \sigma_i)$, where $\sigma_i$ is the secret biometric key and $\tau_i$ is public reproduction parameter related with $BIO_i$ [28], respectively. Finally, $U_i$ saves the credentials $MD_i = \{A_i, L_i, M_i, \tau_i, Gen(\cdot), \overline{UK}_i, Rep(\cdot), h(\cdot), t\}$ in the $MD_i$. The registration is also summarized in Fig. 4.

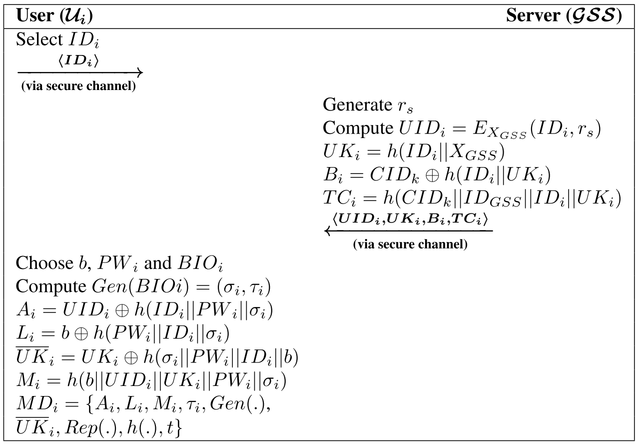


**FIGURE 4.** Registration phase of ***iTCALAS***.

### B. LOGIN AND AUTHENTICATION PHASE

To access the $RD_j$ in a desired flying zone $k$, $U_i$ needs to prove his legality to $MD_i$ as well as to $GSS$. $U_i$ initiates this phase and the process completes by executing following steps:

LAP 1: $U_i$ provides the login credentials ($BIO_i'$, $ID_i$ & $PW_i$) to $MD_i$. $MD_i$ then calculates $\sigma_i' = Rep(BIO_i', \tau_i)$, $UID_i = A_i \oplus h(ID_i||PW_i||\sigma_i')$, $b = L_i \oplus h(PW_i||ID_i||\sigma_i)$, $UK_i = \overline{UK}_i \oplus h(\sigma_i||PW_i||ID_i||b)$, $CID_k = B_i \oplus h(ID_i||UK_i)$ and $TC_i = h(CID_k||ID_{GSS}||ID_i||UK_i)$. $MD_i$ verifies$M_i \overset{?}{=} h(b||UID_i||UK_i||PW_i||\sigma_i')$, session ends, if verification fails. Otherwise, $MD_i$ generates $T_1$, $R_1$ and computes $U_1 = E_{UK_i}(ID_{DR_j}||R_1||CID_k||T_1)$ and $U_2 = h(R_1||UID_i||ID_{RD_j}||TC_i||CID_k)$. $U_i$ then transmits $MSG_1 = \{U_1, U_2, T_1\}$ to $GSS$.

LAP 2: On receiving, the $GSS$ checks the freshness of the $MSG_1$ (through $|\ T_c - T_1\ | < \Delta T$); in case it is fresh, $GSS$ calculates $(ID_i||r_s) = D_{X_{GSS}}(UID_i)$, $UK_i = h(ID_i||X_{GSS})$, $(ID_{DR_j}||R_1||CID_k||T_1) = D_{UK_i}(U_1)$, $TC_i = h(CID_k||ID_{GSS}||ID_i||UK_i)$. $GSS$ verifies $U_2 \overset{?}{=} h(R_1||UID_i||ID_{RD_j}||TC_i||CID_k)$. Upon unsuccessful validation, the $GSS$ rejects the $U_i$'s legitimacy and terminates the session. Otherwise, the $GSS$ continues by generating $R_2$ and current timestamp $T_2$, and computes $U_3 = h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_2) \oplus UID_i, U_4 = h(UID_i||ID_{RD_j}||CID_k||T_2||h(R_1||R_2)), U_5 = h(UID_i\,||ID_{RD_j}||SID_{RD_j}||T_2) \oplus h(R_1||R_2)$, $UID_i^{new} = E_{X_{GSS}(ID_i||R_2)}$ and $U_6 = UID_i^{new} \oplus UK_i \oplus TC_i$. $U_i$ then transmit the message $MSG_2 = \{U_3, U_4, U_5, U_6, T_2\}$ to the remote drone $RD_j$.

LAP 3: On receiving $GSS$ message, $RD_j$ checks the freshness ($|T_c - T_2| < \Delta T$) and on success, $RD_j$ computes $UID_i = U_3 \oplus h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_2)$

<table>
<tr><th>Mobile Device of User ($MD_i$)</th><th>Ground Station Server ($GSS$)</th><th>Remote Drones ($RD_j$)</th></tr>
<tr><td><b>Step LAP 1:</b><br>Input $ID_i, PW_i$ and $BIO_i^{'}$<br>$\sigma_i^{'} = Rep(BIO_i^{'}, \tau_i)$<br>$UID_i = A_i \oplus h(ID_i||PW_i||\sigma_i^{'})$<br>$b = L_i \oplus h(PW_i||ID_i||\sigma_i)$<br>$UK_i = \overline{UK}_i \oplus h(\sigma_i||PW_i||ID_i||b)$<br>$CID_k = B_i \oplus h(ID_i||UK_i)$<br>$TC_i = h(CID_k||ID_{GSS}||ID_i||UK_i)$<br>Verifies $M_i \overset{?}{=} h(b||UID_i||UK_i||PW_i||\sigma_i^{'})$<br>Generate $R_1$ and $T_1$<br>Compute<br>$U_1 = E_{UK_i}(ID_{DR_j}||R_1||CID_k||T_1)$<br>$U_2 = h(R_1||UID_i||ID_{RD_j}||TC_i||CID_k)$<br>$\xrightarrow[(MD_i \rightarrow GSS)]{\langle MSG_1 = \{UID_i, U_1, U_2, T_1\}\rangle}$</td><td></td><td></td></tr>
<tr><td></td><td><b>Step LAP 2:</b><br>Check if $\mid T_1^* - T_1 \mid < \triangle T$?<br>Compute $(ID_i||r_s) = D_{X_{GSS}}(UID_i)$<br>$UK_i = h(ID_i||X_{GSS})$<br>$(ID_{DR_j}||R_1||CID_k||T_1) = D_{UK_i}(U_1)$<br>$TC_i = h(CID_k||ID_{GSS}||ID_i||UK_i)$<br>Verify $U_2 \overset{?}{=} h(R_1||UID_i||ID_{RD_j}||TC_i||CID_k)$<br>Generate random $R_2$ &amp; timestamp $T_2$. Compute<br>$U_3 = h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_2) \oplus UID_i$,<br>$U_4 = h(UID_i||ID_{RD_j}||CID_k||T_2||h(R_1||R_2))$<br>$U_5 = h(UID_i||ID_{RD_j}||SID_{RD_j}||T_2) \oplus h(R_1||R_2)$<br>$UID_i^{new} = E_{X_{GSS}(ID_i||R_2)}$<br>$U_6 = UID_i^{new} \oplus UK_i \oplus TC_i$<br>$\xrightarrow[(GSS \rightarrow RDj)]{\langle MSG_2 = \{U_3, U_4, U_5, U_6, T_2\}\rangle}$</td><td></td></tr>
<tr><td></td><td></td><td><b>Step LAP 3:</b><br>Check if $\mid T_2^* - T_2 \mid < \triangle T$?<br>Compute<br>$UID_i = U_3 \oplus h(ID_{GSS}||SID_{RD_j}||ID_{RD_j}||T_2)$,<br>$h(R_1||R_2) = U_5 \oplus h(UID_i||ID_{RD_j}||SID_{RD_j}||T_2)$<br>Verify $U_4 \overset{?}{=} h(UID_i||ID_{RD_j}||CID_k||T_2||h(R_1||R_2))$<br>Generate random $R_3$ and current timestamp $T_3$.<br>Compute $R_3^{'} = h(R_3||h(R_1||R_2))$<br>$U_7 = R_3^{'} \oplus h(UID_i||ID_{RD_j}||T_3||CID_k)$<br>$SK = h(R_3^{'}||UID_i||ID_{RD_j}||CID_k||T_3)$<br>$U_8 = h(R_3^{'}||SK||T_3||CID_k)$<br>$\xleftarrow[(MD_i \leftarrow RDj)]{\langle MSG_3 = \{U_6, U_7, U_8, T_3\}\rangle}$</td></tr>
<tr><td><b>Step LAP 4:</b><br>Receive message $MSG_3$<br>Check if $\mid T_3^* - T_3 \mid < \triangle T$?<br>$R_3^{'} = U_7 \oplus h(UID_i||ID_{RD_j}||T_3||CID_k)$<br>$SK = h(R_3^{'}||UID_i||ID_{RD_j}||CID_k||T_3)$<br>Verify if $U_8 \overset{?}{=} h(R_3^{'}||SK||T_3||CID_k)$?<br>Authenticate $RD_j$<br>$UID_i^{new} = U_6 \oplus UK_i \oplus TC_i$<br>$A_i = UID_i^{new} \oplus h(ID_i||PW_i||\sigma_i)$</td><td></td><td></td></tr>
<tr><td></td><td>$\xleftrightarrow[(MD_i \leftrightarrow RD_j)]{SK = h(h(R_3||h(R_1||R_2))||UID_i||ID_{RD_j}||CID_k||T_3)}$</td><td></td></tr>
</table>

**FIGURE 5.** Login and authentication phase of ***iTCALAS***.

and $h(R_1||R_2) = U_5 \oplus h(UID_i||ID_{RD_j}||SID_{RD_j}||T_2)$. $RD_j$ then checks $U_4 \overset{?}{=} h(UID_i||ID_{RD_j}||CID_k||T_2||h(R_1||R_2))$. If fails, $RD_j$ declines the message. Otherwise, $RD_j$ selects $T_3$, $R_3$ and computes $R_3^{'} = h(R_3||h(R_1||R_2))$, $U_7 = R_3^{'} \oplus h(UID_i||ID_{RD_j}||T_3||CID_k)$, $SK = h(R_3^{'}||UID_i||ID_{RD_j}||CID_k||T_3)$ and $U_8 = h(R_3^{'}||SK||T_3||CID_k)$. $RD_j$ then sends the message $MSG_3$ containing $\{U_6, U_7, U_8, T_3\}$ directly to $U_i$ via open channel.

LAP 4: The $U_i$ checks the freshness ($|\ T_c - T_3\ | < \triangle T$,) of the $MSG_3$ and on success computes $R_3^{'} = U_7 \oplus h(UID_i||ID_{RD_j}||T_3||CID_k)$ and $SK = h(R_3^{'}||UID_i||ID_{RD_j}||CID_k||T_3)$. $U_i$ then verifies if $U_8 \overset{?}{=} h(R_3^{'}||SK||T_3||CID_k)$, if the condition holds $RD_j$ is verified successfully else session is terminated. Conclusively, $RD_j$ and $U_i$ both have the $SK = h(R_3^{'}||UID_i||ID_{RD_j}||CID_k||T_3)$ as a session key. Now, $MD_i$ computes $UID_i^{new} = U_6 \oplus UK_i \oplus TC_i$ and updates $A_i = UID_i^{new} \oplus h(ID_i||PW_i||\sigma_i)$.

### C. USER PASSWORD/BIOMETRIC UPDATE PHASE

If a legal user $U_i$ wants to update his/her biometric/password along with mobile device $MD_i$, this can be done by following the subsequent steps:

$PBU1$: $U_i$ enters his/her $ID_i$, $PW_i$ and imprints $BIO_i'$. Then $MD_i$ computes the following $\sigma_i' = Rep(BIO_i', \tau_i)$, $UID_i = A_i \oplus h(ID_i||PW_i||\sigma_i')$, $b = L_i \oplus h(PW_i||ID_i||\sigma_i)$, $UK_i = \overline{UK}_i \oplus h(\sigma_i||PW_i||ID_i||b)$, $CID_k = B_i \oplus h(ID_i||UK_i)$, $TC_i = h(CID_k||ID_{GSS}||ID_i||UK_i)$ and validates the user by checking the condition $M_i \overset{?}{=} h(b||UID_i||UK_i||PW_i||\sigma_i')$, if true $MD_i$ will prompt the user to enter a fresh password $PW_i^{new}$ and biometric $BIO_i^{new}$ and move to the step $PDU2$ else session will be terminated.

$PBU2$: $U_i$ enters his/her $ID_i$ a new password $PW_i^{new}$, imprints new biometric $BIO_i^{new}$ and a random number $b^{new}$. Then $U_i$ calculates $Gen(BIO_i^{new}) = (\sigma_i^{new}, \tau_i^{new})$, $A_i^{new} = (A_i^{old} \oplus h(ID_i||PW_i^{old}||\sigma_i^{old})) \oplus h(ID_i \; ||PW_i^{new} ||\sigma_i^{new}) = UID_i \oplus h(ID_i||PW_i^{new}||\sigma_i^{new})$, $L_i^{new} = b^{new} \oplus h(PW_i^{new}||ID_i||\sigma_i^{new})$, $\overline{UK_i^{new}} = \overline{UK_i^{old}} \oplus h(\sigma_i^{old}||PW_i^{old} \; || ID_i||b^{old}) \oplus \; \oplus \; h(\sigma_i^{new}||PW_i^{new}||ID_i||b^{new}) = UK_i \oplus h(\sigma_i^{new}||PW_i^{new}||ID_i||b^{new})$, $M_i^{new} = h(b^{new}||UID_i||UK_i ||PW_i^{new} \; ||\sigma_i^{new})$.

$PBU3$: Finally, the $MD_i$ replaces the parameters $\{A_i^{old}$, $L_i^{old}$, $M_i^{old}$, $\tau_i^{old}$, $\overline{UK_i^{old}}\}$ with $\{A_i^{new}$, $L_i^{new}$, $M_i^{new}$, $\tau_i^{new}$, $\overline{UK_i^{new}}\}$.

### D. USER REVOCATION AND RE-REGISTRATION PHASE

If a legal user $U_i$ lost his/her mobile device $MD_i$ or is stolen than he/she can procure novel device $MD_i^{new}$ by following the subsequent steps:

$RR1$: $U_i$ enters his/her od identity $ID_i^{old}$ and sends it to the Server ($GSS$) over the secure channel.

$RR2$: Upon receiving the registration request from $U_i$, $GSS$ generates a random number $r_s^{new}$ to calculates $UID_i^{new} = E_{X_{GSS}} \; (ID_i^{old}, r_s^{new})$, $UK_i^{new} = h(ID_i^{old}||X_{GSS})$, $B_i^{new} = CID_k \oplus h(ID_i^{old} \; || \; UK_i^{new})$, $TC_i^{new} = h(CID_k||ID_{GSS}||ID_i^{old}||UK_i^{new})$ and sends message containing $\{UID_i^{new}, UK_i^{new}, B_i^{new}, TC_i^{new}\}$ to $U_i$ through a secure channel.

$RR3$: On receiving the message from $GSS$, the $U_i$ chooses a random number $b^{new}$, password $PW_i^{new}$ and imprints $BIO_i^{new}$. Then $U_i$ calculates $Gen(BIO_i^{new}) = (\sigma_i^{new}, \tau_i^{new})$, $A_i^{new} = UID_i^{new} \oplus h(ID_i^{old}||PW_i^{new}||\sigma_i^{new})$, $L_i^{new} = b^{new} \oplus h(PW_i^{new} \; || \; ID_i^{old} \; || \; \sigma_i^{new})$, $\overline{UK_i^{new}} = UK_i^{new} \oplus h(\sigma_i^{new} \; || \; PW_i^{new} \; || \; ID_i^{old} \; || \; b^{new})$, $M_i^{new} = h(b^{new} \; || \; UID_i^{new} \; || \; UK_i^{new} \; || \; PW_i^{new} \; || \; \sigma_i^{new})$ and then stores the the credentials $\{A_i^{new}, L_i^{new}, M_i^{new}, \tau_i^{new}, \overline{UK_i^{new}}, Gen(\cdot), Rep(\cdot), h(\cdot), t\}$ in the $MD_i^{new}$'s memory.

### E. DYNAMIC REMOTE DRONE ADDITION PHASE

If a new remote drone $RD_j$ needs to be added in the cluster $CID_k$, then the following subsequent steps need to be carried out:

$DDA1$: The $GSS$ first assigns a unique identity $ID_{RD_j}$ to remote drone $RD_j^{new}$ along with long-term secret $X_{RD_j^{new}}$ and then calculates $SID_{RD_j}^{new} = h(CID_k \; || \; ID_{RD_j}^{new} \; || \; X_{GSS} \; || \; X_{RD_j}^{new})$.

$DDA2$: Finally, $RD_j^{new}$ is pre-loaded with the credentials $\{ID_{GSS}, CID_k, ID_{RD_j}^{new}, SID_{RD_j}^{new}, h(\cdot)\}$ before deploying in the $k_{th}$ cluster flying zone. The $GSS$ stores the parameters $\{ID_{RD_j}^{new}, SID_{RD_j}^{new}\}$ in its own database.

## V. SECURITY ANALYSIS

This section presents the austere security analysis of the proposed scheme by employing both the formal and informal security analysis.

### A. FORMAL SECURITY ANALYSIS

In this paper, to the test the security of session key $SK$, we used extensively applied Random Oracle Model (ROM) [40]. Under the $ROM$, an adversary $\mathcal{A}$ interrelates with $En^i$, where $i^{th}$ instance of an entity being participated (e.g. it can be legal user $U_i$, the remote drone $RD_j$ or an ground station server $GSS$ in *iTCALAS*. Consequently, there are three $En_{U_i}^i$, $En^{TD_j}$ and $GSS$ as the $i^{th_1}$, $i^{th_2}$ and $i_3^{th}$ of $U_i$, $RD_j$ and $GSS$ respectively. Moreover, the ROM assumes identical queries executing a definite attack, such as $Send(\cdot)$, $CorruptDE(\cdot)$, $Test(\cdot)$ and $Reveal(\cdot)$ queries. Similarly, a one-way hash function h(·) referred as collision-resistant can be access by the instances of each entity as well as $\mathcal{A}$.

- *Send($En^i$, mesg):* This query is demonstrated as an active attack, where $U_A$ can submit a message $mesg$ to an instance $En^i$, and also $En^i$ responses accordingly.
- *Reveal($En^i$)* Simulating this query permits to reveal the existing session key $SK$ shared among $En^i$ and its companion $U_A$
- $CorruptDE(En_{U_i}^{i_1})$ This query allows $\mathcal{A}$ to get $U_i$'s password $PW_i$ and $\acute{\sigma}_i$ via stolen $MD_i$
- $Test(En^i)$ : $\mathcal{A}$ demands $En^i$ for the $SK$ and $Eni$ probabilistically responses the output of a tossed neutral coin $co$.
- *Execute($En_{U_i}^{i_1}$, $En_{RD_j}^{i_2}$, $En_{GSS}^{i_3}$):* It allows $\mathcal{A}$ to intercept the messages exchanged between $U_i$, $RD_j$ and $GSS$

In Theorem 1, the $SK$ security of *iTCALAS* is proved under $ROM$ and using above mentioned queries.

*Theorem 1:* Assume that a polynomial time $\mathcal{A}$ simulate in time $T$ against our protocol (*iTCALAS*). If $|h(\cdot)|$ denotes the range-space of $h(\cdot)$, $bl$ specifies the bio's secrete key bit, $que_{hsh}$ represents the number of hashes, $que_{snd}$ characterizes the amount of send queries, respectively. Where as $Ch$ and $se$ are the parameters of *Zipfile* defined in [41]. The $\mathcal{A}$'s benefit in outrageous security of *iTCALAS* to obtain the $SK$ between $RD_j$ and $U_i$ can be reffered as:

$$Advntg_{iTCALAS}^{\mathcal{A}}(i) \leq \frac{que_{hsh}}{H_{ash}} + 2maxx\left\{Ch'.que_{se}^{\acute{s}e}, \frac{que_{se}}{2^{bl}}\right\}. \tag{15}$$

The following four games are defined, say $Gme_v$, $v \in \{0, 3\}$. If $Suc_v$ specifies and occurrence where $\mathcal{A}$ can guess the arbitrary bit $c_b$ in $Gme_v$ correctly, the benefit of $\mathcal{A}$ in captivating this game will be defined and expressed as $Advntg_{iTCALAS}^{\mathcal{A},Gme_v} = Pre\,[Suce_v]$, whereas $Pre\,[X]$ is the possibility of an event $X$.

*Game.0* ($Gme_0$)*:* The attack actually performed by $\mathcal{A}$ corresponding to *iTCALAS* in ROM against to $Gme_0$. The bit

$c_b$ is chosen arbitrarily at the beginning of $Gme_0$. Therefore, we attain,

$$Advntg^{\mathcal{A}}_{iTCALAS}(i) = \left|2.Advntg^{\mathcal{A},Gme_0}_{iTCALAS} - 1\right| \tag{16}$$

*Game. 1* ($Gme_1$) : This game is used for modeling an eavesdropping attack where $\mathcal{A}$ capture all the login and authentication exchanged messages $< MSG_1 = \{UID_i, U_1, U_2, T_1\} >$, $< MSG_1 = \{U_3, U_4, U_4, U_6, T_2\} >$ and $< MSG_3 = \{U_7, U_8, U_9, T_3\} >$ that simulate *iTCALAS* using *Execute* query. In order to verify the derived $SK$, the $\mathcal{A}$ simulates *Test* and *Reveal* queries at the end of this game. The $SK$ created between $U_i$ and reachable $DR_j$ is $SK = h(h(R_3)\|h(R_1\|R_2))\|ID_{RD_j}\|CID_k\|T_4$. In order to compute $SK$, the $\mathcal{A}$ requires long term secrets ($CID_k$, $ID_{RD_j}$ and $HID_i$) and temporal secrets $R_1$ to $R_3$ to compute $SK$ which are not known to $\mathcal{A}$. Hence, just intercepting the $MSG_1$, $MSG_2$ and $MSG_3$ the chances of winning $Gme_1$ is not improved by $\mathcal{A}$. Leveraging the in-determinability of $Gme_0$ and $Gme_1$, it follows that:

$$Advntg^{\mathcal{A},Gme_0}_{iTCALAS}. \tag{17}$$

*Game. 2* ($Gme_2$)*:* This game includes the execution of *hsh* and *Send* queries to *ROM* as an active attack. From the delivered messages $MSG_1$, $MSG_2$ and $MSG_3$, every $U_f$ (f $= 1, 2, 3 \ldots\ldots, 9$), are protected by the $h(\cdot)$. Since every $U_f$ are involves current timestamps, the arbitrary numbers, secret credentials and identities, there will be no collision when the *Hsh* and $Send(\cdot)$ queries are simulated by $\mathcal{A}$. Both $Gme_1$ and $Gme_2$ are in deterministically but the addition of the execution of the $Hsh(\cdot)$ and $Send(\cdot)$ queries in $Gme_2$. The birthday paradox's results will be lead as follows:

$$\left|Advntg^{\mathcal{A},Gme_1}_{iTCALAS} - Advntg^{\mathcal{A},Gme_2}_{iTCALAS}\right| \leq que_{hsh}/(2\,|Hsh|) \tag{18}$$

*Game. 3* ($Gme_3$)*:* The $Gme_3$ is malformed from $Gme_2$ by including the exeution of *CorruptDE* query, $\mathcal{A}$ would be able to have the parameters of $MD_i = \{A_i, L_i, M_i, \tau_i, Gen(\cdot), \overline{UK}_i, Rep(\cdot), h(\cdot), t\}$. Through guessing some password and using the *Zipf*'s law $\mathcal{A}$ can check it utilizing the derived credentials $\acute{A}_i$ and $L_i$. The benefit of $\mathcal{A}$ will be exceed over 0.5 where in condition $que_{se} = 10^7$ or $10^8$ if we only take seeking password. Similarly, the gain of $\mathcal{A}$ will exceed over 0.5 if $\mathcal{A}$ uses personal data of user. Moreover, as the function of *fuzzy extractor* can be used for *iTCALAS* to gain the $c_b$. Excluding the execution of *CorruptDe* query in $Gme_3$, the $Gme_2$ and $Gme_3$ are not distinguishable. If the system allows limited tries of entering wrong password then it will leads towards following consequences :

$$\left|Advntg^{\mathcal{A},Gme_2}_{iTCALAS} - Advntg^{\mathcal{A},Game.3}_{iTCALAS}\right| \leq \left\{Ce'.que^{\acute{snd}}_{snd}, \frac{que_{snd}}{2^l}\right\}. \tag{19}$$

As all the queries are simulated by $\mathcal{A}$, it only remains to gues the $c_b$ to win the game once the $Test(\cdot)$ query is executed, and hence, we have $Advntg^{\mathcal{U},Game.3}_{iTCALAS} = \frac{0}{1}$.

Simplifying the equations and using the triangular-inequality, the following is attained:

$$\begin{aligned}
\frac{0}{1}&.Advntg^{\mathcal{A}}_{iTCALAS}(i) \\
&= |Advntg^{\mathcal{A},Gme.0}_{iTCALAS} - \frac{0}{1}| \\
&= Advntg^{\mathcal{A},Gme.1}_{iTCALAS} - Advntg^{\mathcal{A},Gme.4}_{iTCALAS} \\
&\leq |Advntg^{\mathcal{A},Gme.1}_{iTCALAS} - Advntg^{\mathcal{A},Game.2}_{iTCALAS}| \\
&\quad +|Advntg^{\mathcal{A},Gme.3}_{iTCALAS} - Advntg^{\mathcal{A},Gme.3}_{iTCALAS}| \\
&\leq \frac{que_{hsh}}{2|H_{hash}|} + maxx\left\{CE'.que^{snd'}_{snd}, \frac{q_{snd}}{2^l}\right\}.
\end{aligned}$$

Hence, it follows that $Advntg^{\mathcal{A}}_{iTCALAS}(t) \leq \frac{que_{hsh}}{H_{ash}} + 2maxx\left\{CE'.q^{snd'}_{snd}, \frac{que}{2^l}\right\}$

## B. SECURITY ANALYSIS USING PROVERIF TOOL

This subsection presents the results of ProVerif tool, used for the verification of the security properties for the proposed scheme. ProVerif can check the correctness, session key secrecy, reachibility and anonymity and privacy. Two channels 1) $ChSec : private$ and 2) $Chpub : public$, to represent secure and public channels for registration and authentication phases, respectively. The communication in the registration phase between $U_i$, $GSS$ and $RD_j$ is completed over the $ChSec : private$ channel, whereas the $Chpub : public$ channel is used for the communication in the login and authentication phase. During the implementation different declared constructors are as follow: $Hash(h)$, $XOR(\oplus)$, $Concat(||)$, $Rep()$, $Gen()$. The results of the ProVerif tool are shown in Figure 6, which clearly demonstrates the scheme's correctness and security.

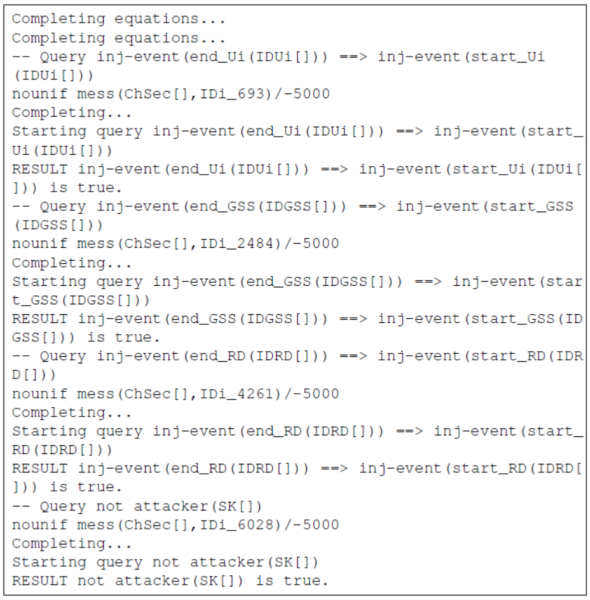
## C. INFORMAL SECURITY ANALYSIS

This section presents a discussion of on the security features extended by *iTCALAS* as well as attack resilience:

### 1) STOLEN MOBILE DEVICE ATTACK

This attack is launched by an attacker, after the device of a legitimate user is stolen/lost and attacker gets it. Based on the information in the smart device, the attacker can try to expose identity and password related information of the user. The details of proposed scheme's resistance from this attack, after attacker gets the lost/stolen device is given as follows:

- *Identity guessing attack:* $\mathcal{A}$ can perform power analysis on the device to extract the information form the memory [39]. $\mathcal{A}$ have the access to the credentials $\{A_i, L_i, M_i, \tau_i, Gen(\cdot), \overline{UK}_i, Rep(\cdot), h(\cdot), t\}$, the $ID_i$ of the $U_i$ is first encrypted by the $GSS$'s secret key and then *XORed* with $h(ID_i||PW_i||\sigma_i)$ and stored in $A_i$. So, in order to get $ID_i$ the knowledge of the $X_{GSS}$, $PW_i$ and $\sigma_i$ is required, also the one-way property of $h(\cdot)$ makes it infeasible to guess $ID_i$. Hence the scheme is secured again identity guessing attack.



```
Completing equations...
Completing equations...
-- Query inj-event(end_Ui(IDUi[])) ==> inj-event(start_Ui
(IDUi[]))
nounif mess(ChSec[],IDi_693)/-5000
Completing...
Starting query inj-event(end_Ui(IDUi[])) ==> inj-event(start_
Ui(IDUi[]))
RESULT inj-event(end_Ui(IDUi[])) ==> inj-event(start_Ui(IDUi[
])) is true.
-- Query inj-event(end_GSS(IDGSS[])) ==> inj-event(start_GSS
(IDGSS[]))
nounif mess(ChSec[],IDi_2484)/-5000
Completing...
Starting query inj-event(end_GSS(IDGSS[])) ==> inj-event(star
t_GSS(IDGSS[]))
RESULT inj-event(end_GSS(IDGSS[])) ==> inj-event(start_GSS(ID
GSS[])) is true.
-- Query inj-event(end_RD(IDRD[])) ==> inj-event(start_RD(IDR
D[]))
nounif mess(ChSec[],IDi_4261)/-5000
Completing...
Starting query inj-event(end_RD(IDRD[])) ==> inj-event(start_
RD(IDRD[]))
RESULT inj-event(end_RD(IDRD[])) ==> inj-event(start_RD(IDRD[
])) is true.
-- Query not attacker(SK[])
nounif mess(ChSec[],IDi_6028)/-5000
Completing...
Starting query not attacker(SK[])
RESULT not attacker(SK[]) is true.
```

**FIGURE 6.** ProVerif simulation results.

- *Offline password guessing attack:* After extracting the parameters from the $MD_i$, $\mathcal{A}$ has the access to the parameters $A_i, L_i, \overline{UK}_i$ and $M_i$ but cannot extract the $PW_i$ from these parameters as it requires the knowledge of $ID_i, \sigma_i, UID_i, b$ and $UK_i$. Hence, the scheme can withstand this attack.

### 2) ANONYMITY AND UNTRACEABILITY OF USER

As described in threat model (Subsection I-A) that $\mathcal{A}$ can capture the messages $MSG_1, MSG_2$ and $MSG_3$ transmitted over the public channel. The user $ID_i$ is sent in $MSG_1$ through $UID_i = E_{X_{GSS}}(ID_i||r_s)$ and to extract $ID_i$, $\mathcal{A}$ need private key $X_{GSS}$ of the ground station. Moreover, this identity is updated in each session, so the user can not be traced. Moreover, all other parameters in messages communicated through public link are based on randomly selected numbers or timestamps. Therefore, the traceability or identity expose is protected in proposed $iTCALAS$.

### 3) IMPERSONATION ATTACK

$\mathcal{A}$ can impersonate on behalf of user, ground station or the drone. The resilience of $iTCALAS$ against these impersonations is discussed below:

- *User impersonation attack:* For $\mathcal{A}$, to launch successful impersonation on behalf of $U_i$, has to generate valid request message $MSG_1 = \{UID_i, U_1, U_2, T_1\}$. Selecting current timestamp is very easy and $UID_i$ can be replayed easily. Creating rest of the parameters $U_1$ and $U_2$ in a way that $U_2$ can pass the test $U_2 \overset{?}{=} h(R_1||UID_i||ID_{RD_j}||TC_i||CID_k)$, besides $UID_i$, $ID_{RD_j}$ and $R_1$ the attacker $\mathcal{A}$ needs $TC_i$ as well as $CID_k$. $TC_i$ can be extracted using smart card as well as user password and biometrics, or through private key $X_{GSS}$ of the ground station. Moreover, to get the information of the flying zone of some arbitrary user, the attacker needs user private credentials as well as smart device. Therefore, $\mathcal{A}$ cannot successfully impersonate as a $U_i$.
- *Server impersonation attack* For $\mathcal{A}$, to launch successful impersonation on behalf of $GSS$, has to generate and send valid message $MSG_2 = \{U_3, U_4, U_5, U_6, T_2\}$ to $RD_j$. Selecting current timestamp is very easy. Creating rest of the parameters $U_3, U_4, U_5$ and $U_6$ in a way that $U_4$ can pass the test $U_4 \overset{?}{=} h(UID_i||ID_{RD_j}||CID_k||T_2||h(R_1||R_2))$, besides $UID_i$, $ID_{RD_j}$ and $CID_k$ the attacker $\mathcal{A}$ needs $h(R_1||R_2)$, and $h(R_1||R_2)$ can be computed by an entity who has private key $X_{GSS}$ of the ground station. Moreover, to get the information of the flying zone of some arbitrary user, the attacker needs private credentials of the drones or private key of the ground station. Therefore, $\mathcal{A}$ cannot successfully impersonate as a $GSS$.
- *Drone impersonation attack* For $\mathcal{A}$, to launch successful impersonation on behalf of $RD_j$, has to generate and send valid message $MSG_3 = \{U_6, U_7, U_8, T_3\}$ to $U_i$. Selecting current timestamp is very easy. Creating rest of the parameters $U_6, U_7$ and $U_8$ in a way that $U_6$ can pass the test $U_8 \overset{?}{=} h(R_3'||SK||T_3||CID_k)$, besides $T_3$ the attacker $\mathcal{A}$ needs $R_3' = h(R_3||h(R_1||R_2))$ as well as session key and both of these parameters $R_3'$ and session key cannot be computed unless the attacker has private key $X_{GSS}$ of the ground station or temporal credentials of the drone. Therefore, $\mathcal{A}$ cannot successfully impersonate as a $RD_j$.

### 4) PROTECTION AGAINST REPLAY ATTACK

In the proposed scheme the reply attack is eradicated by incorporating the time stamps and random nonces in the messages during login and authentication phases. As $\mathcal{A}$ sends the messages $MSG_1 = \{UID_i, U_1, U_2, T_1\}, MSG_2 = \{U_3, U_4, U_5, U_6, T_2\}, MSG_3 = \{U_6, U_8, U_9, T_3\}$ to perform a reply attack will fail due to time stamp and random nonces. When message is received the initial step involved is to check the freshness of the time stamp, then if the time delay is greater than the allowed delay message is going to be discarded. Hence the scheme can successfully prevent the reply attack.

### 5) MAN-IN-THE-MIDDLE ATTACK PREVENTION

During the login and authentication phase $\mathcal{A}$ may try to capture and tempered the transferred messages $MSG_1, MSG_2$ and $MSG_3$to make believe the other participants that the message is genuine. But to perform this task the $\mathcal{A}$ requires the knowledge of parameters $\{UK_i, CID_k, TC_i, R_1\}$ for $MSG_1$, $\{SID_{RD_j}, ID_{RD_j}, CID_k, R_1, R_2, UID_i^{new}\}$ for $MSG_2$ and $\{R_3\}$ for $MSG_3$. Thus the scheme can withstand this attack.

#### 6) MUTUAL AUTHENTICATION

All of the participants involved in the communication authenticate each other. In the $MSG_1$, the $GSS$ checks $\{R_1 \ \& \ U_2\}$ to authenticate $MD_i$. In the $MSG_2$ the $RD_j$ checks $\{h(R_1||R_2) \ \& \ U_4\}$ to authenticate the $GSS$, where as $MD_i$ uses $\{R_3 \ \& \ U_4\}$ to authenticate the $RD_j$. So, both the $U_i$ and $RD_j$ authenticate each other with the help of $GSS$.

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

In the proposed scheme the long-term secrets like $\{ID_{RD_j}, CID_k, X_{GSS}\}$ and short-term secrets like $\{R_1, R_2, R_3\}$ are used to generate the session-key $SK$. Now assume that all f the long-term secret has been compromised and are in the knowledge of the $\mathcal{A}$, but $\mathcal{A}$ still needs the short-term secrets in order to successfully compute $SK$. Now same way if the short-term secrets are compromised $\mathcal{A}$ still needs the long-term secrets in order to successfully compute the $SK$. So, the scheme can successfully withstand the ESL attack.

#### 8) REMOTE DRONE CAPTURE ATTACK

As described in the threat model (Subsection I-B) $\mathcal{A}$ can capture the $RD_j$ and can extract the parameters $\{ID_{GSS}, CID_k, ID_{RD_j}, SID_{RD_j}, h(\cdot)\}$ stored in its memory. But all of the stored parameters are uniquely computed for each drone and does not reveal any information about the other drones, $MD_i$ and $GSS$. Hence, the scheme can withstand remote drone capture attack.

## VI. COMPARISONS WITH RELATED SCHEMES

In this section, we elaborate the security features, computational and communicative efficiencies comparisons of the proposed scheme with some related schemes [17], [22], [26], [27], [42].

### A. SECURITY FEATURES

This subsection elaborates the security features comparisons between proposed and related schemes. The comparisons are shown in Table 2, where (✓) represents the provision of certain security feature or resistance against some attack; whereas, (×) shows insecurity against some attack or non-provision of some security feature. Citing Table 2, only proposed scheme provides all the related security features discussed in the table, other competing schemes lacks one or more security features or resists against one or more attacks. The scheme presented in [27] also has much higher cost as compared with *iTCALAS* and it can be observed in following subsections and Table 3.

### B. COMPUTATION AND COMMUNICATION COSTS

The comparison of the different schemes in the context of communication and computation costs incured during the login and authentication phase only, is considered here. For communication cost, the bit-size considered for nonces is 160 *bits*; whereas, identity is fixed as 160 *bits* long. The size of timestamp is taken as 32 *bits* long, the size of ECC coordinates is fixed at 160 *bits*, which implies that size of an ECC point is $(160+160) = 320$ *bits*. Moreover, it is assumed that all the schemes used $SHA-1$ algorithm with output size 160 *bits* long.

**TABLE 2.** Comparison of functionality features.

| | Proposed | [17] | [26] | [22] | [27] | [42] |
|---|---|---|---|---|---|---|
| $SR\#1$ | ✓ | ✓ | ✓ | ✓ | × | ✓ |
| $SR\#2$ | ✓ | ✓ | × | × | ✓ | ✓ |
| $SR\#3$ | ✓ | ✓ | × | × | ✓ | ✓ |
| $SR\#4$ | ✓ | × | ✓ | × | ✓ | × |
| $SR\#5$ | ✓ | ✓ | × | × | ✓ | ✓ |
| $SR\#6$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $SR\#7$ | ✓ | ✓ | × | × | ✓ | ✓ |
| $SR\#8$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $SR\#9$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $SR\#10$ | ✓ | ✓ | × | × | ✓ | ✓ |
| $SR\#11$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $SR\#12$ | ✓ | × | × | × | ✓ | ✓ |
| $SR\#13$ | ✓ | ✓ | × | × | ✓ | ✓ |
| $SR\#14$ | ✓ | ✓ | ✓ | ✓ | ✓ | × |
| $SR\#15$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $SR\#16$ | ✓ | ✓ | × | × | ✓ | ✓ |

Note: $SR\#1$: correctness; $SR\#2$: offline or online password guessing attack ; $SR\#3$: privileged insider attack; $SR\#4$: user anonymity; $SR\#5$: traceability; $SR\#6$: detection for unauthorized login; $SR\#7$: stolen mobile/smart card attack; $SR\#8$: denial-of-service attack; $SR\#9$: mutual authentication; $SR\#10$: ESL attack; $SR\#11$: replay & man-in-the-middle attacks; $SR\#12$: impersonation attacks; $SR\#13$: Sensing node or drone capture attack; $SR\#14$: revocability; $SR\#15$: formal security verification under any tool; $SR\#16$: password/biometric change..

**TABLE 3.** Communication cost comparison.

| Schemes | # of messages | # of bits |
|---|---|---|
| Proposed Scheme | 3 | 1696 |
| Srinivas et al. [17] | 3 | 1536 |
| Tai et al. [26] | 4 | 2560 |
| Turkanović et al. [22] | 4 | 2720 |
| Challa et al. [27] | 3 | 2528 |
| Wazid et al. [42] | 3 | 1696 |

The Table 3 shows that the communication cost of the proposed scheme is less than the [22], [26], [27]; whereas cost is equal to [42]and has slight more computation cost as compared with [17]. However, only proposed scheme provides all discussed security features. The communication cost is also represented in Fig. 7.

For comparing the costs, we adopted the timing of various operation as per the experiment conducted in [43] on a PC with dual CPU E2200: 2.20GHz using GMP based PBC library. The experiment was performed on 32 bit Ubuntu 12.04.1 LTS having RAM size 2048 MB. The computed time for the hash-function ($T_h$) is 0.0023 *ms*, for ECC point multiplication ($T_m$) is 2.226 *ms*, for symmetric enc/dec ($T_{sym}$) is 0.0046*ms* and time required for the fuzzy-extractor is $T_m \approx T_{fe} \approx 2.226$ *ms* [17]. The total number of operations required for execution of a single cycle of the proposed scheme are

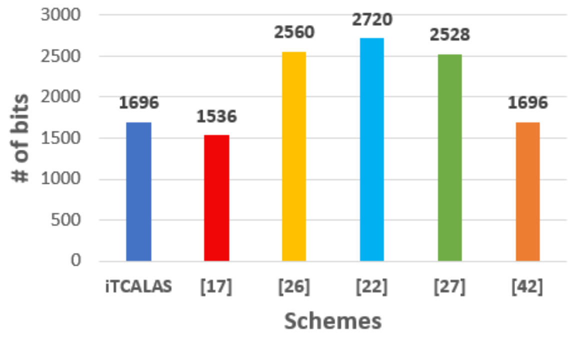


**FIGURE 7.** Communication cost comparison.

**TABLE 4.** Computation cost comparison.

| Schemes | Mobile device | Ground station server | Drone | ≈ computation |
|---|---|---|---|---|
| Proposed Scheme | $10T_h + T_{fe}$ | $7T_h + 3T_{sym}$ | $7T_h$ | ≈ 2.295 |
| Srinivas et al. [17] | $14T_h + T_{fe}$ | $9T_h$ | $7T_h$ | ≈ 2.295 |
| Tai et al. [26] | $7T_h$ | $6T_h$ | $10T_h$ | ≈ 0.0529 |
| Turkanović et al. [22] | $7T_h$ | $5T_h$ | $7T_h$ | ≈ 0.0437 |
| Challa et al. [27] | $5T_h + T_{fe}$ $+5T_m$ | $4T_h + 5T_m$ | $3T_h$ $+4T_m$ | ≈ 33.422 |
| Wazid et al. [42] | $16T_h + T_{fe}$ | $8T_h$ | $7T_h$ | ≈ 2.2973 |

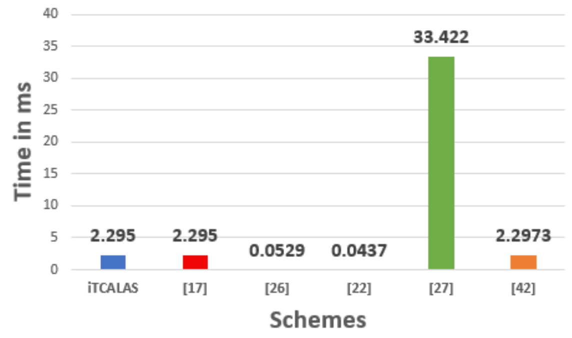


**FIGURE 8.** Computation cost comparison.

$24T_h + 1T_{fe} + 3T_{sym}$ with running time $\approx 2.295ms$. Computation cost of various schemes are presented in Table 4 as well as in Figure 8. Citing Table 4, proposed scheme incurs more computation time as compared with [22], [26] and same as of [17] and less than [42] and [27]. However, only proposed scheme provides all security features.

## VII. CONCLUSION

The surveillance data is important and sensitive in nature and among other methods, the drones can be very useful for obtaining such data from in-accessible places like fire sites, battle field and mountains peeks etc. However, due to the underlying open channel, this data as well as the drones can be used for wicked intentions. In this paper, we examined a recent authentication scheme for protecting drone access by unauthorized users. We have proven that the scheme of Srinivas et al. is insecure against traceability and impersonation based on stolen verifier. It is also shown that their scheme has scalability issues and can work when there is only one flying zone/cluster present in the environment. For securing the surveillance and drones, we presented an improved scheme using only light weight hash and symmetric encryption/decryption operations. The security of the proposed scheme is proved through formal, informal and automated methods. While providing all the security features and resistance against many known attacks, proposed scheme completes authentication process with same computation time as of Srinivas et al.'s scheme. Therefore, proposed scheme is best suitable for securing the surveillance data communicated through drones.

## ACKNOWLEDGMENT

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, KSA, under grant No. (D-480 – 611 - 1441). The authors, therefore, gratefully acknowledge DSR technical and financial support.

## REFERENCES

[1] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for Internet of Things and related challenges of fog and cloud computing integration," *ACM Comput. Surv.*, vol. 51, no. 6, p. 116, Jan. 2019.

[2] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating systems for low-end devices in the Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 720–734, Oct. 2016.

[3] Y. Xu, V. Mahendran, W. Guo, and S. Radhakrishnan, "Fairness in fog networks: Achieving fair throughput performance in MQTT-based IoTs," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 191–196.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, Jun. 2015.

[5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[6] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi, and C. S. Hong, "Internet of Things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Gener. Comput. Syst.*, vol. 92, pp. 265–275, Mar. 2019.

[7] S. A. Alvi, G. A. Shah, and W. Mahmood, "Energy efficient green routing protocol for Internet of multimedia Things," in *Proc. IEEE 10th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2015, pp. 1–6.

[8] P. K. Dhillon and S. Kalra, "A secure multifactor remote user authentication scheme for Internet of multimedia Things environment," *Int. J. Commun. Syst.*, vol. 32, no. 15, p. e4077, Jul. 2019.

[9] G. Choudhary, V. Sharma, T. Gupta, J. Kim, and I. You, "Internet of drones (IoD): Threats, vulnerability, and security perspectives," 2018, *arXiv:1808.00203*. [Online]. Available: http://arxiv.org/abs/1808.00203

[10] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of drones," *IEEE Access*, vol. 4, pp. 1148–1162, 2016.

[11] M. Pandelea, R. Bucharest, M. Boşcoianu, M.-M. Frăţilă, and V. Vlădăreanu, "Conceptual method of navigating and controlling a drone," *Sci. Res. Edu. Force*, vol. 19, no. 1, pp. 165–170, Jul. 2017.

[12] M. Lorenz. (Aug. 2019). *Mavlink: Micro Air Vehicle Communication Protocol*. [Online]. Available: https://mavlink.io/en/ and http://qgroundcontrol.org/mavlink/start

[13] F. Al-Turjman, "A novel approach for drones positioning in mission critical applications," *Trans. Emerg. Telecommun. Technol.*, Apr. 2019, Art. no. e3603, doi: 10.1002/ett.3603.

[14] Z. Ullah, F. Al-Turjman, and L. Mostarda, "Cognition in UAV-aided 5G and beyond communications: A survey," *IEEE Trans. Cognit. Commun. Netw.*, to be published.

[15] N. Ahmad, "Robotic automated external defibrillator ambulance for emergency medical service in smart cities," *Int. J. Trend Sci. Res. Develop.*, vols. Volume–3, nos. Issue–2, pp. 308–310, Feb. 2019.

[16] F. Al-Turjman and S. Alturjman, "5G/iot-enabled UAVs for multimedia delivery in industry-oriented applications," *Multimedia Tools Appl.*, vol. 2018, pp. 1–22, Jun. 2018.

[17] 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.

[18] V. Shannon. (Aug. 2019). *The Future of Drones in Business and Commerce*. [Online]. Available: https://www.mondo.com/future-of-drones/

[19] F. Al-Turjman, M. Abujubbeh, A. Malekloo, and L. Mostarda, "UAVs assessment in software-defined IoT networks: An overview," *Comput. Commun.*, vol. 150, pp. 519–536, Jan. 2020.

[20] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for Internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[21] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 8, pp. 3133–3142, Sep. 2018.

[22] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.

[23] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Netw.*, vol. 36, pp. 152–176, Jan. 2016.

[24] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[25] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[26] W.-L. Tai, Y.-F. Chang, and W.-H. Li, "An IoT notion–based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks," *J. Inf. Secur. Appl.*, vol. 34, pp. 133–141, Jun. 2017.

[27] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. Goutham 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] S. Challa, A. K. Das, P. Gope, N. Kumar, F. Wu, and A. V. Vasilakos, "Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems," *Future Gener. Comput. Syst.*, to be published, doi: 10.1016/j.future.2018.04.019.

[33] S. A. Chaudhry, T. Shon, F. Al-Turjman, and M. H. Alsharif, "Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems," *Comput. Commun.*, vol. 153, pp. 527–537, Mar. 2020.

[34] 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.

[35] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2008, pp. 203–220.

[36] W.-H. Yang and S.-P. Shieh, "Password authentication schemes with smart cards," *Comput. Secur.*, vol. 18, no. 8, pp. 727–733, Jan. 1999.

[37] M. Hölbl, T. Welzer, and B. Brumen, "An improved two-party identity-based authenticated key agreement protocol using pairings," *J. Comput. Syst. Sci.*, vol. 78, no. 1, pp. 142–150, Jan. 2012.

[38] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 1999, pp. 388–397.

[39] 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.

[40] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *International Workshop Public Key Cryptography*. Berlin, Germany: Springer, 2005, pp. 65–84.

[41] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[42] 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.

[43] H. H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1005–1023, 2nd Quart., 2014.

**ZEESHAN ALI** received the bachelor's degree from NUML University Islamabad. He is currently pursuing the M.S.C.S. degree in information security with International Islamic University Islamabad, Islamabad, Pakistan. He completed his course work with distinction and working towards his final research thesis. He has published three articles in conferences and journals and already submitted some of his articles in top journals. His research interests include computer networking, network security, network communication, information security, cryptography, encryption, and authentication.

**SHEHZAD ASHRAF CHAUDHRY** received the master's and Ph.D. degrees (Hons.) from International Islamic University Islamabad, Pakistan, in 2009 and 2016, respectively.

He is currently working as an Associate Professor with the Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelişim University, Istanbul, Turkey. He has authored over 75 scientific publications appeared in different international journals and proceedings, including 62 in SCI/E journals. With an H-index of 21 and an I-10 index 39, his work has been cited over 1450 times. He has also supervised over 35 graduate students in their research. His current research interests include lightweight cryptography, elliptic/hyper elliptic curve cryptography, multimedia security, e-payment systems, MANETs, SIP authentication, smart grid security, IP multimedia subsystems, and next-generation networks. He occasionally writes on issues of higher education in Pakistan. He has served as a TPC member for various international conferences. He is an Active Reviewer of many ISI indexed journals. He was a recipient of the Gold Medal for achieving 4.0/4.0 CGPA in his master's degree. Considering his research, Pakistan Council for Science and Technology granted him the Prestigious Research Productivity Award, while affirming him among the Top Productive Computer Scientist in Pakistan.

**MUHAMMAD SHER RAMZAN** received the M.Sc. degree from Quaid-e-Azam University, Islamabad, and the Ph.D. degree in computer science from TU Berlin, Germany. He is currently a Professor with the Faculty of Computing and Information Technology (FCIT), King Abdulaziz University, Saudi Arabia. He has more than 30 years of research, teaching, and administrative experience in different educational and research institutions. He has produced 14 Ph.D. scholars and has more than 130 scientific publications. He has served as the Chairman of the Department of Computer Science and Software Engineering and the Dean of the Faculty of Basic and Applied Sciences, International Islamic University, Pakistan. He had completed many research and development projects for IT industry in Pakistan and Germany. His research interests include next-generation networks, information systems, and information security.

**FADI AL-TURJMAN** received the Ph.D. degree in computer science from Queen's University, Kingston, ON, Canada, in 2011. He is currently a Professor and the Research Center Director of Near East University, Nicosia, Cyprus. He is also a leading authority in the areas of smart/cognitive, wireless, and mobile networks' architectures, protocols, deployments, and performance evaluation. His publication history spans over 250 publications in journals, conferences, patents, books, and book chapters, in addition to numerous keynotes and plenary talks at flagship venues. He has written and edited more than 25 books about cognition, security, and wireless sensor networks' deployments in smart environments, published by Taylor and Francis, Elsevier, and Springer. He has received several recognitions and best papers' awards at top international conferences. He also received the prestigious Best Research Paper Award from *Computer Communications* (Elsevier) journal for the period 2015–2018, in addition to the Top Researcher Award for 2018 from Antalya Bilim University, Turkey.

• • •