Computer Standards & Interfaces 80 (2022) 103566

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Designing secure and lightweight user access to drone for smart city surveillance

Sajid Hussain [a], Khalid Mahmood [b], Muhammad Khurram Khan [c], Chien-Ming Chen [d], Bander A. Alzahrani [e], Shehzad Ashraf Chaudhry [*,f]

[a] Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan
[b] Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus 57000, Pakistan
[c] College of Computer & Information Sciences, King Saud University, Riyadh, Saudi Arabia
[d] College of Computer Science and Engineering, Shandong University of Science and Technology, Shandong, China
[e] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[f] Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey

ARTICLE INFO

Keywords:
Three-factor authentication
Smart-card capture attack
Drone capture attack
Provable security
Smart City security
IoT security
Key-agreement
Internet of Drones
Authentications

ABSTRACT

The Internet of drones (IoD) is a very useful application of the Internet of things (IoT) and it can help the daily life comfort through various functions including the smart city surveillance. The IoD can enhance the comfort to reach inaccessible and hard to access sites and can save lot of effort, time and cost. However, in addition to traditional threats, the IoD may suffer from new threats and requires customized methods to combat the security weaknesses. Very recently, Wazid et al. proposed a security solution for securing IoD application scenario and claimed its security. However, in this paper we show that their scheme cannot resist stolen verifier and traceability attacks. Moreover, an attacker with access to the verifier, can impersonate any user, drone or server of the system. An enhanced scheme is then proposed to cope with these weaknesses. The security claims of proposed scheme are endorsed by formal and informal security analysis. Moreover, the performance and security comparisons show that proposed scheme completes a cycle of authentication with a slight increase in computation time, but it offers all the required security features as compared with the scheme of Wazid et al.

1. Introduction

Among many other applications of Internet of Things (IoT), the drones infrastructure also called as Internet of drones (IoD) can extend benefit in variety of ways including the smart city surveillance, and the remote cargo etc. For surveillance purposes, the IoD can enhance quality of life and can help in reducing crime rate as these can be deployed at inaccessible remote location like fire site of a tall building and can also reach many accessible remote locations like mountains peaks and depths, very fast in contrast with traditional way of transport [1–4]. The continuous and fast internet connectivity made the IoD dream a reality and as the population is increasing very rapidly, the use of traditional surveillance and traditional transportation to emergency sites may not be feasible for the safety of human lives, where the rapid response is a must [1]. A typical drone also called as unmanned aerial vehicle (UAV) is an automatic spacecraft without any pilot physically present in the drone. The network of coordinating and collaborating UAVs forms an IoD network [2], where the UAVs act within a layered and controlled network for the specified collective task like surveillance. With the same properties as of a typical IoT network, the UAVs are equipped with sensor, receiver and transmitted for communication with out-side entities including the control station and the drone user [1,2]. These drones within an IoD sense the required information and sends the data to the user connected through base station for decision making process. The collected data is real time and can help for making rapid and wise decisions [2]. However, like other internet based systems, the IoD or a single drone can be used by the deceitful adversaries for wicked intentions and can ultimately be harmful in many ways including the passive drone location tracing as well as disruption of the services [5]. Moreover, the attacker can try to stop the drone for performing its' designated tasks or can physically capture the drone. Figure 1 presents a typical IoD entities, where users are accessing the drone through public internet.

Although some schemes were proposed to secure similar structure in

* Corresponding author.
E-mail addresses: ashraf.shehzad.ch@gmail.com, sashraf@gelisim.edu.tr (S.A. Chaudhry).

https://doi.org/10.1016/j.csi.2021.103566
Received 27 September 2020; Received in revised form 28 April 2021; Accepted 5 August 2021
Available online 10 August 2021

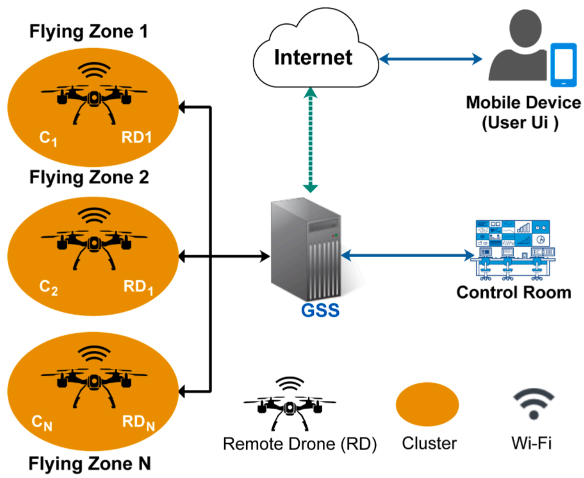

Fig. 1. IoD Architecture

User (U_m)	Server (S)
Select ID_m $\xrightarrow{\langle\{ID_m\}\rangle}$ **(via secure channel)**	
	Compute $RID_m = E_{K_{ms}}(ID_m \| r_m)$, $TC_m = h(ID_m \| r_m \| K_{ms})$. $\xleftarrow{\langle\{RID_m, ID_{DR_n}, TC_m, A\}\rangle}$ **(via secure channel)**
Choose PW_m and BIO_m Compute $Gen(BIO_m) = (\sigma_m, \tau_m)$ $RID_m^{'} = RID_m \oplus h(PW_m \| \sigma_m)$, $ID_{DR_n}^{'} = ID_{DR_n} \oplus h(ID_m \| PW_m \| \sigma_m)$, $TC_m^{'} = TC_m \oplus h(ID_m \| \sigma_m)$, $RPW_m = h(PW_m \| n)$, $A^{'} = A \oplus h(ID_m \| \sigma_m \| PW_m)$, $B = n \oplus h(PW_m \| ID_m \| \sigma_m)$, $C = h(A \| RID_{DR_n} \| RPW_m \| \sigma_m)$. Stores $\{RID_m^{'}, ID_{DR_n}^{'}, TC_u^{'}, A^{'},$ $B, C, \tau_m, Gen(), Rep(), h(), t\}$ into MD_m	Store $\{ID_m\}$

Fig. 2. User Registration

three party settings using three factors [6–16]. However, the tailored security schemes for IoD environments are very less [17–24]. In 2019, Tian et al. [17] designed modular exponentiation based frame work for authentication in edge based IoD architecture. Bera et al. [24] argued that the scheme of Tian et al. is computationally expensive. Bera at al. also proposed an elliptic curve cryptography (ECC) based access control scheme for IoD. Bera et al. utilized ECC based certificate for authentication and key exchange. Moreover, they employed blockchains for the transactions of the collected data. Due to usage of ECC and static certificate in each access control cycle between ground server station (GSS) and drone, the scheme is not only computationally expensive but also lacks anonymity. Similarly, Srinivas et al. [20] proposed symmetric key based lightweight authentication scheme specifically for securing the IoD environment. However, Ali et al. [23] proved some of the crucial weaknesses in the scheme of Srinivas et al.

1.1. Motivations and Contributions

Very recently, Wazid et al. [22] proposed a new authentication scheme in three party settings to secure *IoD* communication by establishing a secure channel between the legal user gathering the real time data and the drones. The protocol was designed by using only light-weight symmetric key computations, which makes it a very suitable candidate for resource limited *IoD* environments. However, a careful analysis in this paper proves that the scheme of Wazid et al. is vulnerable to traceability and stolen verifier attacks. Moreover, in their scheme upon receiving a request, the server after validation of the legality of the user, sends request message to all drones in a flying zone, such broadcasting may force all the drones in a specified zone to process the request causing the useless processing and ultimately draining the drone/s battery. The analysis in later part of this article also shows that any adversary by just getting verifier can impersonate any entity (user, drone or server) of the system. Moreover, an adversary with verifier can disclose any session key computed among a user and a drone by just listening to the communication channel. Hence, it is dispensable to design an authentication scheme for securing the drones. The contributions of this papers are as follows:

- Initially, the review and cryptanalysis of the scheme of Wazid et al. are presented.
- It is proved that the scheme of Wazid et al. is vulnerable to traceability and stolen verifier attacks.
- It is also argued in this article that due to a design flaw in the scheme of Wazid et al., the server broadcasts the connection request to all the drones, forcing them to process the request, which can ultimately drain the battery due to useless computation.
- An improved scheme is proposed to extend required security features and to resist known attacks.
- The security of the proposed scheme is verified through formal BAN logic along with a discussion of security features.
- Finally, proposed scheme is compared with related schemes using the performance and security features as comparison metrics.

1.2. Adversarial Model

In this paper, we have employed the common adversarial model (eCK) [25–29], where an adversary $\mathscr{A}$ is considered to be strong enough to control the public communication channel. Precisely, in the employed adversarial model, $\mathscr{A}$ can passively listen communication between user/GSS and drone. $\mathscr{A}$ can replay, and/or send a forged messages. $\mathscr{A}$ can also stop any message transmitted on the communication channel [30–32]. Using the power analysis, $\mathscr{A}$ can interpret the leaked data from a physically captured drone and from stolen smart card [25,32]. $\mathscr{A}$ can be a deceitful user as well as an outsider with the knowledge of all public parameters including identities of the registered entities. $\mathscr{A}$ cannot expose private key of any of the participant.

2. Review of the Scheme of Wazid et al.

In this section, we present a brief review of the scheme of Wazid et al. designed specifically for securing IoD. Their scheme consists of three participants, namely User ($\mathscr{U}_m$), Server ($\mathscr{S}$) and Drone ($\mathscr{DR}_n$). The Server $\mathscr{S}$ in their scheme provides the registration facility to the $\mathscr{U}_m$ and $\mathscr{DR}_n$. Following subsections provide brief overview of the phases presented in Wazid et al.'s scheme:

2.1. Pre-deployment Phase

The server $\mathscr{S}$ registers all the drones $\mathscr{DR}_n : \{n = 1, 2....N\}$ before deployment in the IoD environment. Initially, $\mathscr{S}$ picks a unique 160-bit secret number k and a unique identity ID_{DRn} then computes $RID_{DRn} = h(ID_{DRn} \| k)$, $TC_{DRn} = h(ID_{DRn} \| RTS_{DRn})$ relevant to $\mathscr{DR}_n$. $\mathscr{S}$ also selects a bi-variate polynomial $\mathscr{P}(x, y) = \sum_{i=0}^{n}\sum_{j=0}^{n} g_{i,j} x^i y^j \in GF(p)[x, y]$ for supporting to establish inter-drone secure connection. Then $\mathscr{S}$ generates TID_{DRn} and computes polynomial value $\mathscr{P}(TID_{DRn}, y)$. Then $\mathscr{S}$ engraves $\{TID_{DRn}, TC_{DRn}, \mathscr{P}(TID_{DRn}, y), RID_{DRn}\}$ in the memory of respective drone $\mathscr{DR}_n$ and stores $\{RID_{DRn}, TC_{DRn}, \mathscr{P}(x, y), k\}$ in its own database.

2.2. User Registration Phase

This subsection outlines the registration process for an arbitrary user

User ($\boldsymbol{U_m}$)	Server($\boldsymbol{S}$)	Drone ($\boldsymbol{DR_n}$)
Enter ID_m, password $PW_m^{'}$		
and biometrics $BIO_m^{'}$		
Compute $\sigma_m^{'} = Rep(BIO_m^{'}, \tau_m)$		
$RID_m = RID_m^{'} \oplus h(PW_m^{'} \vert \sigma_m^{'})$,		
$ID_{DR_n} = ID_{DR_n}^{'} \oplus h(ID_m \| PW_m^{'} \| \sigma_m^{'})$,		
$TC_m = TC_m^{'} \oplus h(ID_m \| \sigma_m^{'})$,		
$n = B \oplus h(PW'_m \| ID_m \| \sigma_m^{'})$,	Receive M_{sg1} from U_m	
$RPW_m^{'} = h(PW_m^{'} \| n)$,	$\vert T_1 - T_1^* \vert \leqslant \triangle T$? if so,	
$A = A^{'} \oplus h(ID_m \| \sigma_m^{'} \| PW^{'}_m)$,	$(ID_m \| r_m) = D_{K_{ms}}(M_1)$	
$C^{'} = h(A \| ID_{DR_n} \| RPW^{'}_m \| \sigma_m^{'})$.	If ID_m exists in server database. If so,	
Check if $C_m \stackrel{?}{=} C_m^{'}$ if so generate T_1, r_1.	Compute $TC_m = h(ID_m \| r_m \| K_{ms})$,	
Calculates: $M_1 = RID_m$	$ID_{DR_n} = M_2 \oplus h(TC_m \| ID_m \| T_1)$,	
$M_2 = ID_{DR_n} \oplus h(TC_m \| ID_m \| T_1)$,	$r_1^{'} = M_3 \oplus h(ID_s \| TC_m \| T_1)$,	
$M_3 = h(ID_s \| TC_m \| T_1) \oplus r_1$,	$M_4^{'} = h(ID_m \| ID_s \| RID_{DR_n} \| TC_m \| r_1^{'} \| T_1)$,	
$M_4 = h(ID_m \| ID_s \| ID_{DR_n} \| TC_m \| r_1 \| T_1)$.	verifies if $M_4^{'} \stackrel{?}{=} M_4$ if it is true, generate r_2, T_2, r_{mnew}	Receive M_{sg2} from S
(1) $\xrightarrow{\boldsymbol{M_1,\ M_2,\ M_3,\ M_4,\ T_1}}$ **(via open channel)**	Compute $TC_{DR_n} = h(ID_{DRn} \| K_{ms})$.	Check if $\vert T_2 - T_2^* \vert \leqslant \triangle T$?
	$M_5 = h(TC_{DR_n} \| ID_{DR_n}) \oplus h(ID_s \| r_1 \| r_2)$,	Compute $ID_m = M_6 \oplus h(TC_{DR_n} \vert T_2)$,
	$M_6 = h(TC_{DR_n} \| T_2) \oplus ID_m$,	$M_9 = M_5 \oplus h(TC_{DR_n} \| ID_{DR_n})$,
	$M_7 = h(TC_{DR_n} \| ID_{DR_n} \| h(ID_s \| r_1 \| r_2) \| T_2)$.	$M_{10} = h(TC_{DR_n} \| ID_{DR_n} \| M_9 \| T_2)$.
	$M_8 = E_{K_{ms}}(ID_m \| r_{mnew}) \oplus h(TC_m \| ID_m \| RID_m)$.	checks if $M_{10}^{'} \stackrel{?}{=} M_7$
	(2) $\xrightarrow{\boldsymbol{M_5,\ M_6,\ M_7,\ M_8,\ T_2}}$ **(via open channel)**	Generate r_3, T_3
		Compute $M_{11} = h(ID_{DR_n} \| ID_m \| T_3) \oplus r_3$
$\vert T_3 - T_3^* \vert \leqslant \triangle T$? if so,		$SK_{mn} = h(M_9 \| r_3 \| ID_m \| ID_{DR_n})$,
Computes $r_3^{'} = h(ID_{DR_n} \| ID_m \| T_3) \oplus M_{11}$		$M_{12} = h(ID_m \| ID_{DR_n} \| r_3) \oplus M_9$,
$M_9^{'} = h(ID_m \| ID_{DR_n} \| r_3^{'}) \oplus M_{12}$,		$M_{13} = h(SK_{mn} \| T_3)$.
$SK_{mn}^{'} = h(M_9^{'} \| r_3^{'} \| ID_m \| ID_{DR_n})$,		$\xleftarrow{\boldsymbol{\langle M_{11},\ M_{12},\ M_{13},\ T_3, M_8 \rangle}}$ (3) **(via open channel)**
$M_{14} = h(SK_{mn}^{'} \| T_3)$,		
Check if $M_{14} \stackrel{?}{=} M_{13}$		
$\overline{RID}_m = M_8 \oplus h(TC_m \| ID_m \| RID_m)$		
$RID_m = \overline{RID}_m$		
	U_n and DR_n saves the session key $SK_{mn}(= SK_{mn}^{'})$ for future secure communication	

Fig. 3. Proposed Login and Authentication

$\mathscr{U}_m$ for gaining the real-time information (Surveillance or otherwise) from desired drone $\mathscr{DR}_n$ in the IoD environment. $\mathscr{U}_m$ and $\mathscr{S}$ performs following steps to complete the registration process:

WR 1: Initially, $\mathscr{U}_m$ picks and sends ID_m to $\mathscr{S}$ secretly. On receiving the ID_m, $\mathscr{S}$ computes $RID_m = h(ID_m \parallel k)$, $RID_s = h(ID_s \parallel k)$ and $A = h(RID_s \parallel ID_m)$. Next, $\mathscr{S}$ selects $\mathscr{U}_m$'s master key MK_{Um}, registration time stamp RTS_{Um} and computes $TC_m = h(ID_m \parallel MK_{Um} \parallel RTS_{Um})$. $\mathscr{S}$ then securely sends reply $\{RID_m, RID_{DRn}, RID_s, TC_m, A\}$ to $\mathscr{U}_m$.
WR 2: On receiving the reply of $\mathscr{S}$, $\mathscr{U}_m$ picks PW_m, and inputs BIO_m using mobile device MD_m. MD_m creates the biometric secret key σ_m and its relevant τ_m as $Gen(BIOi) = (\sigma_m, \tau_m)$. MD_m then produces n (160 $-bit$ secret number) for $\mathscr{U}_m$ and computes $RID_m^{'} = RID_m \oplus h(PW_m \parallel \sigma_m)$, $RID_{DRn}^{'} = RID_{DRn} \oplus h(ID_m \parallel PW_m \parallel \sigma_m)$, $TC_{Um}^{'} = TC_m \oplus h(ID_m \parallel \sigma_m)$, and $RPW_m = h(PW_m \parallel n)$. MD_m further computes $RID_s^{'} = RID_s \oplus h(RID_m \parallel \sigma_m)$, $A^{'} = A \oplus h(RID_m \parallel \sigma_m \parallel PW_m)$, $B = n \oplus h(PW_m \parallel ID_m \parallel \sigma_m)$ and $C = h(A \parallel RID_{DRn} \parallel RPW_m \parallel \sigma_m)$. At the end, MD_m engraves $\{RID_m^{'}, RID_{DRn}^{'}, RID_s^{'}, TC_{Um}^{'}, A^{'}, B, C, \tau_m, Gen(.), Rep(.), h(.), t\}$ in its own memory. Moreover, $\mathscr{S}$ stores $\{ID_m \parallel RID_m \parallel TC_m \parallel RID_s\}$ in its verifier data-base.

2.3. Login and Authentication phase

Login and authentication phase of Wazid et al.'s scheme (WL) is invoked by $\mathscr{U}_m$ to get authenticated and establish a secure channel by sharing a secret key with $\mathscr{DR}_n$. Following steps accomplishes the login and authentication procedure:

WL1 1: $\mathscr{U}_m$ submits the pair $\{ID_m, PW_m\}$ to MD_m, and imprints BIO_m. MD_m computes $\sigma_m = Rep(BIO_m, \tau_m)$ and checks validity of biometrics. On successful validation of biometrics, MD_m computes: $RID_m = RID_m^{'} \oplus h(PW_m^{'} \parallel \sigma_m^{'})$, $RID_{DRn} = RID_{DRn}^{'} \oplus h(ID_m \parallel PW_m^{'} \parallel \sigma_m^{'})$, $TC_m = TC_{Um}^{'} \oplus h(ID_m \parallel \sigma_m^{'})$, $RID_s = RID_s^{'} \oplus h(RID_m \parallel \sigma_m^{'})$, $n = B \oplus h(PW_m^{'} \parallel ID_m \parallel \sigma_m^{'})$ and $RPW_m^{'} = h(PW_m \parallel n)$. MD_m then computes $A = A^{'} \oplus h(RID_m \parallel \sigma_m^{'} \parallel PW_m^{'})$ and $C^{'} = h(A \parallel RID_{DRn} \parallel RPW_m^{'} \parallel \sigma_m^{'})$. Afterward, MD_m verifies equality $C_m \stackrel{?}{=} C_m^{'}$. On failed equality, the process is aborted immediately. Next, MD_m picks the pair $\{T_1, r_1\}$ and computes $M_1 = RID_m \oplus h(RID_s \parallel T_1)$, $M_2 = RID_{DRn} \oplus h(TC_u \parallel ID_m \parallel T_1)$, $M_3 = h(RID_s \parallel TC_u \parallel T_1) \oplus r_1$, $M_4 = h(ID_m \parallel RID_s \parallel RID_{DRn} \parallel TC_u \parallel r_1 \parallel T_1)$. Finally, $M_{sg1} = (M_1, M_2, M_3, M_4, T_1)$ is sent to $\mathscr{S}$ on public channel.
WL 2: On receiving M_{sg1}, the $\mathscr{S}$ verifies the time-freshness ($\vert T_1 - T_1^* \vert \leqslant \triangle T$), on success, $\mathscr{S}$ computes $RID_m = M_1 \oplus h(RID_s \parallel T_1)$, $RID_{DRn} = M_2 \oplus h(TC_u \parallel ID_m \parallel T_1)$, $r_1 = M_3 \oplus h(RID_s \parallel TC_u \parallel T_1)$, $M_4^{'} = h(ID_m \parallel RID_s \parallel RID_{DRn} \parallel TC_u \parallel r_1 \parallel T_1)$ and verifies $M_4^{'} \stackrel{?}{=} M_4$, on success user is considered as authenticated else session is aborted immediately. $\mathscr{S}$ then picks the pair $\{r_2, T_2\}$ and computes $M_5 = h(TC_{DRn} \parallel RID_{DRn}) \oplus h(RID_s \parallel r_1 \parallel r_2)$, $M_6 = h(TC_{DRn} \parallel T_2) \oplus RID_m$ and $M_7 = h(TC_{DRn} \parallel RID_{DRn} \parallel h(RID_s \parallel r_1 \parallel r_2) \parallel T_2)$. $\mathscr{S}$ sends $M_{sg2} = \{M_5, M_6, M_7, T_2\}$ to Drone $\mathscr{DR}_n$.
WL 3: $\mathscr{DR}_n$ on reception of M_{sg2}, first checks time-freshness ($\vert T_2 - T_2^* \vert \leqslant \triangle T$) and on success, $\mathscr{DR}_n$ computes $RID_m = M_5 \oplus h(TC_{DRn} \parallel T_2)$, $M_8 = M_5 \oplus h(TC_{DRn} \parallel RID_{DRn})$, $M_9 = h(TC_{DRn} \parallel RID_{DRn} \parallel M_8 \parallel T_2)$, and checks $M_9^{'} \stackrel{?}{=} M_7$. On success, $\mathscr{S}$ is considered as authenticated by $\mathscr{DR}_n$; otherwise session is terminated immediately. $\mathscr{DR}_n$ then creates $\{r_3, T_3\}$ pair and computes $M_{10} = h(RID_{DRn} \parallel RID_m \parallel T_3) \oplus r_3$, $SK_{mn} = h(M_8 \parallel r_3 \parallel RID_m \parallel RID_{DRn})$, $M_{11} = h(RID_m \parallel RID_{DRn} \parallel r_3) \oplus M_8$ and $M_{12} = h(SK_{mn} \parallel T_3)$. Finally, $\mathscr{DR}_n$ transmits reply message $M_{sg2} = \{M_{10}, M_{11}, M_{12}, T_3\}$ directly to user $\mathscr{U}_m$ through public channel.
WL 4: $\mathscr{U}_m$ after receiving the authentication reply, first checks time-freshness ($\vert T_3 - T_3^* \vert \leqslant \triangle T$), upon success, $\mathscr{U}_m$ computes $r_3 = h(RID_{DRn} \parallel RID_m \parallel T_3) \oplus M_{10}$, $M_8 = h(RID_m \parallel RID_{DRn} \parallel r_3) \oplus M_{11}$,

$SK'_{mn} = h(M'_8 \| r'_3 \| RID_m \| RID_{DRn})$ and $M'_{13} = h(SK'_{mn} \| T_3)$. At last $\mathscr{U}_m$ checks whether $M_{13} \stackrel{?}{=} M12$. If the condition is true, $\mathscr{U}_m$ considers $\mathscr{DR}_n$ as authenticated drone and the session key SK'_{mn} is considered as correct key for establishing future secure channels.

3. Weaknesses of the Scheme of Wazid et al.

This section explores some of the weaknesses of the scheme of Wazid et al. [22] in following subsections:

3.1. Traceability Attack

This subsection shows that a registered but unfair user $\mathscr{U}_A$ have privileges to launch successful traceability attack. $\mathscr{U}_A$ after registering with $\mathscr{S}$ gets his mobile device MD_A customized with $\{RID_A, RID'_{DRn}, RID'_s, TC'_A, A', B, C, \tau_m, Gen(.), Rep(.), h(.), t\}$. $\mathscr{U}_A$ using his device MD_A inputs $ID_{\mathscr{A}}$, $PW'_{\mathscr{A}}$, $BIO'_{\mathscr{A}}$ and computes:

$$\sigma_{\mathscr{A}} = Rep\left(BIO'_{\mathscr{U}_{\mathscr{A}}}, \tau_{\mathscr{A}}\right) \tag{1}$$

$\mathscr{U}_A$ extracts RID'_s from mobile device and further computes:

$$RID_s = RID'_s \oplus h\left(RID_{\mathscr{A}} \| \sigma'_{\mathscr{A}}\right) \tag{2}$$

$\mathscr{U}_A$ waits for any user (say $\mathscr{U}_m$) of the system to initiate a login request consisting $\{M_1, M_2, M_3, T_1\}$ where $M_1 = RID_m \oplus h(RID_s \| T_1)$. Once, a fair user $\mathscr{U}_m$ initiates login, $\mathscr{U}_A$ intercepts the request message and computes:

$$RID_m = M_1 \oplus h(RID_s \| T_1) \tag{3}$$

In Eq. 3, RID_m is pseudo identity of $\mathscr{U}_m$ and remains same for all sessions. Therefore $\mathscr{U}_A$ can successfully trace a legal user of system.

3.2. Stolen Verifier Attack

This subsection shows that the scheme of Wazid et al. is defenseless against stolen verifier (SV) attack. It is to show here that an active and privileged adversary $\mathscr{U}_A$ can impersonate a legal user, a drone or even the server, once he gets the verifier table stored in server memory containing $\{ID_m$, RID_m, TC_{Um}, RID_s, RID_{DRn}, $TC_{DRn}\}$. Moreover, any attacker with verifier can disclose a session key shared among two legal entities (user and drone). Once the verifier is stolen, the subsequent vulnerabilities of Wazid et al.'s scheme are simulated in below subsections:

3.2.1. User Impersonation Attack via Stolen Verifier

Let $\mathscr{U}_A$ be an adversary with access to the verifier/s $\{RID_{DR_n}, TC_{DR_n}, \mathscr{P}(x,y), k\}$ and $\{ID_m, RID_m, TC_{Um}, RID_s, RID_{DRn}, TC_{DRn}\}$ maintained by $\mathscr{S}$. $\mathscr{U}_A$ can forge request message $M_{sg1} = \{M_{\mathscr{A}1}, M_{\mathscr{A}2}, M_{\mathscr{A}3}, M_{\mathscr{A}4}, T_{\mathscr{A}1}\}$ by using RID_m related to some user $\mathscr{U}_m$ by simulating following steps:

1. $\mathscr{U}_A$ generates fresh $T_{\mathscr{A}1}$ and computes:

$$M_{\mathscr{A}1} = RID_m \oplus h(RID_s \| T_{\mathscr{A}1}) \tag{4}$$

$$M_{\mathscr{A}2} = RID_{DR_n} \oplus h(TC_{U_m} \| ID_m \| T_{\mathscr{A}1}) \tag{5}$$

2. Afterward, $\mathscr{U}_A$ generates $r_{\mathscr{A}1}$ randomly and computes:

$$M_{\mathscr{A}3} = h(RID_s \| TC_{U_m} \| T_{\mathscr{A}1}) \oplus r_{\mathscr{A}1} \tag{6}$$

$$M_{\mathscr{A}4} = h(ID_m \| RID_s \| RID_{DR_n} \| TC_m \| r_{\mathscr{A}1} \| T_{\mathscr{A}1}) \tag{7}$$

Now, $\mathscr{U}_A$ sends the forged message $M_{sg1} = \{M_{\mathscr{A}1}, M_{\mathscr{A}2}, M_{\mathscr{A}3}, M_{\mathscr{A}4}, T_{\mathscr{A}1}\}$ to $\mathscr{S}$.

3. $\mathscr{S}$ receives M_{sg1} and checks $|T_{\mathscr{A}1} - T_1^*| \leqslant \triangle T$?, and upon successful validation $\mathscr{S}$ computes:

$$RID_m = M_1 \oplus h(RID_s \| T_1) \tag{8}$$

4. $\mathscr{S}$ fetches ID_m, TC_m, relevant to the RID_m and computes:

$$RID_{DRn} = M_2 \oplus h(TC_m \| ID_m \| T_{\mathscr{A}1}) \tag{9}$$

$$r_{\mathscr{A}1} = M_3 \oplus h(RID_s \| TC_m \| T_{\mathscr{A}1}) \tag{10}$$

$$M'_4 = h(ID_m \| RID_s \| RID_{DRn} \| TC_m \| r_{\mathscr{A}1} \| T_{\mathscr{A}1}) \tag{11}$$

5. $\mathscr{S}$ verifies if $M'_4 = ?M_4$ and on successful verification, $\mathscr{S}$ authenticates the party on other side as $\mathscr{U}_m$. Afterward, $\mathscr{S}$ continues the process by computing and sending $M_{sg2} = \{M_5, M_6, M_7, T_2\}$ to Drone $\mathscr{DR}_n$.

Proposition 1. *In Wazid et al.'s scheme, an adversary $\mathscr{U}_A$ authenticates himself from $\mathscr{S}$ on behalf of a legal user $\mathscr{U}_m$ and shares a session key with a desired drone $\mathscr{DR}_n$.*

Proof. $\mathscr{U}_A$ initiates the login request by computing and sending $M_{sg1} = \{M_{\mathscr{A}1}, M_{\mathscr{A}2}, M_{\mathscr{A}3}, M_{\mathscr{A}4}, T_{\mathscr{A}1}\}$ to $\mathscr{S}$. The server $\mathscr{S}$ authenticates impersonated $\mathscr{U}_A$ on behalf of $\mathscr{U}_m$ by verifying the timestamp freshness and by checking the equality of $M_{\mathscr{A}4}$ computed by $\mathscr{U}_A$ in Eq. 7 with M'_4 computed by $\mathscr{S}$ in Eq. 11. $\mathscr{U}_A$ generated fresh timestamp $T_{\mathscr{A}1}$, so it will pass the freshness test. Moreover, $\mathscr{U}_A$ has access to all parameters ID_m, RID_s, RID_{DRn}, TC_m extracted from parameter and $r_{\mathscr{A}1}$ computed by $\mathscr{U}_A$ himself. Therefore, $M_{\mathscr{A}4}$ computed by $\mathscr{U}_A$ in Eq. 7 is same as $\mathscr{S}$ computed M'_4 in Eq. 11. Hence, $\mathscr{U}_A$ has successfully impersonated as another user $\mathscr{U}_m$ in the scheme of Wazid et al.'s scheme.□

3.2.2. Server Impersonation Attack via Stolen Verifier

The adversary $\mathscr{U}_A$ with stolen verifier can impersonate as the legal server $\mathscr{S}$, during the login and authentication phases, $\mathscr{U}_m$ transmits the login message $M_{sg1} = \{M_1, M_2, M_3, M_4, T_1\}$. $\mathscr{U}_A$ intercepts the message and simulates the attack as per following steps:

1. $\mathscr{U}_A$ using the intercepted message computes $RID_m = M_1 \oplus h(RID_s \| T_1)$ and extracts the corresponding $\{ID_m, TC_m\}$ from the stolen verifier. $\mathscr{U}_A$ computes:

$$RID_{DRn} = M_2 \oplus h(TC_m \| ID_m \| T_1) \tag{12}$$

$$r_1 = M_3 \oplus h(RID_s \| TC_m \| T_1) \tag{13}$$

$$M'_4 = h(ID_m \| RID_s \| RID_{DRn} \| TC_m \| r_1 \| T_1) \tag{14}$$

2. $\mathscr{U}_A$ verifies $M'_4 \stackrel{?}{=} M_4$, on success, it generates r_a and T_2. Next, $\mathscr{U}_A$ extracts TC_{DRn} corresponding to RID_{DRn} and computes:

$$M_5 = h(TC_{DRn} \| RID_{DRn}) \oplus h(RID_s \| r_1 \| r_2) \tag{15}$$

$$M_6 = h(TC_{DRn} \| T_2) \oplus RID_m \tag{16}$$

$$M_7 = h(TC_{DRn} \| RID_{DRn} \| h(RID_s \| r_1) \| T_2) \tag{17}$$

3. $\mathscr{U}_A$ now sends the forged message $M_{sg2} = \{M_5, M_6, M_7, T_2\}$ to $\mathscr{DR}_n$.
4. $\mathscr{DR}_n$ receives M_{sg2}, checks $|T_3 - T_3^*| \leqslant \triangle T$? and upon successful validation computes:

$$RID_m = M_5 \oplus h(TC_{DRn} \parallel T_2) \tag{18}$$

$$M_8 = M_5 \oplus h(TC_{DRn} \parallel RID_{DRn}) \tag{19}$$

$$M_9 = h(TC_{DRn} \| RID_{DRn} \| M_8 \| T_2) \tag{20}$$

5. $\mathscr{DR}_n$ checks $M_9' \stackrel{?}{=} M_7$ and upon successful validation authenticates the party on other side as legal $\mathscr{S}$. Afterward, $\mathscr{DR}_n$ continues the process by computing and sending $M_{sg3} = \{M_{10}, M_{11}, M_{12}, T_3\}$ to Drone $\mathscr{U}_m$.

Proposition 2. *In Wazid et al.'s scheme, an adversary $\mathscr{U}_A$ authenticates himself from $\mathscr{DR}_n$ on behalf of the legal user $\mathscr{S}$ and mediates the sharing of session key between $\mathscr{U}_m$ and $\mathscr{DR}_n$.*

Proof. A legal $\mathscr{U}_m$ initiates the login request by computing and sending $M_{sg1} = \{M_1, M_2, M_3, M_4, T_1\}$ to $\mathscr{S}$. The attacker $\mathscr{U}_A$ intercepts the message and after verifying legality of $\mathscr{U}_m$, computes and sends $M_{sg2} = \{M_5, M_6, M_7, T_2\}$ to $\mathscr{DR}_n$. The $\mathscr{DR}_n$ authenticates impersonated $\mathscr{U}_A$ on behalf of $\mathscr{S}$ by verifying the timestamp freshness and by checking the equality of M_7 computed by $\mathscr{U}_A$ in Eq. 17 with M_9 computed by $\mathscr{DR}_n$ in Eq. 20. $\mathscr{U}_A$ generated fresh timestamp T_2, so it will pass the freshness test. Moreover, $\mathscr{U}_A$ has access to all parameters $TC_{DRn} \| RID_{DRn}$ extracted from verifier and M_8 computed by $\mathscr{U}_A$ himself, again by using verifier and the request of $\mathscr{U}_m$. Therefore, M_7 computed by $\mathscr{U}_A$ in Eq. 17 is same as $\mathscr{DR}_n$ computed M_9 in Eq. 20. Hence, $\mathscr{U}_A$ has successfully impersonated as the legal server $\mathscr{S}$ in the scheme of Wazid et al.'s scheme.□

3.2.3. Drone Impersonation Attack using Stolen Verifier

The adversary $\mathscr{U}_A$ with stolen verifier can also impersonate as a legal drone $\mathscr{DR}_n$, during the login and authentication phase, $\mathscr{U}_m$ transmits the login message $M_{sg1} = \{M_1, M_2, M_3, M_4, T_1\}$ to $\mathscr{S}$. The $\mathscr{S}$ receives the message and after authenticating $\mathscr{U}_m$ sends $M_{sg2} = \{M_5, M_6, M_7, T_2\}$ to $\mathscr{DR}_n$. $\mathscr{U}_A$ intercepts the message and simulates the attack as per following steps:

1. $\mathscr{U}_A$ using the intercepted message and the stolen verifier computes:

$$RID_m = M_5 \oplus h(TC_{DRn} \parallel T_2) \tag{21}$$

$$M_8 = M_5 \oplus h(TC_{DRn} \parallel RID_{DRn}) \tag{22}$$

$$M_9 = h(TC_{DRn} \| RID_{DRn} \| M_8 \| T_2) \tag{23}$$

2. $\mathscr{U}_A$ verifies $M_9 \stackrel{?}{=} M_7$ and on success it generates r_3, T_3 and computes:

$$M_{10} = h(RID_{DRn} \| RID_m \| T_3) \oplus r_3 \tag{24}$$

$$SK_{ij} = h(M_8 \| r_3 \| RID_m \parallel RID_{DRn}) \tag{25}$$

$$M_{11} = h(RID_m \| RID_{DRn} \| r_3) \oplus M_8 \tag{26}$$

$$M_{12} = h\left(SK_{ij} \parallel T_3\right) \tag{27}$$

3. $\mathscr{U}_A$ now sends the forged message $M_{sg3} = \{M_{10}, M_{11}, M_{12}, T_3\}$ to the $\mathscr{U}_m$.
4. $\mathscr{U}_m$ receives M_{sg3}, checks $|T_3 - T_3^*| \leqslant \triangle T$ and upon successful validation computes:

$$r_3 = h(RID_{DRn} \| RID_m \| T_3) \oplus M_{10} \tag{28}$$

$$M_8 = h(RID_m \| RID_{DRn} \| r_3) \oplus M_{10} \tag{29}$$

$$SK_{ij}' = h(M_8' \| r_3' \| RID_m \parallel RID_{DRn}) \tag{30}$$

$$M_{13} = h\left(SK_{ij}' \parallel T_3\right) \tag{31}$$

5. $\mathscr{U}_m$ checks $M_{13} \stackrel{?}{=} M_{12}$ and upon successful validation authenticates the party on other side as legal $\mathscr{DR}_n$ and use the key SK for secure communication with $\mathscr{U}_A$ on behalf of $\mathscr{DR}_n$.

Proposition 3. *In Wazid et al.'s scheme, an adversary $\mathscr{U}_A$ authenticates himself from $\mathscr{U}_m$ on behalf of a legal drone $\mathscr{DR}_n$ and shares a a session key with $\mathscr{U}_m$.*

Proof. A legal $\mathscr{U}_m$ initiates the login request by computing and sending $M_{sg1} = \{M_1, M_2, M_3, M_4, T_1\}$ to $\mathscr{S}$ and $\mathscr{S}$ upon checking legality of the message computes and sends $M_{sg2} = \{M_5, M_6, M_7, T_2\}$ to $\mathscr{DR}_n$. $\mathscr{U}_A$ intercepts the messages and computes and sends $M_{sg3} = \{M_{10}, M_{11}, M_{12}, T_3\}$ to $\mathscr{U}_m$. The $\mathscr{U}_m$ authenticates impersonated $\mathscr{U}_A$ on behalf of $\mathscr{DR}_n$ by verifying the timestamp freshness and by checking the equality of M_{13} computed by $\mathscr{U}_m$ in Eq. 31 with M_{12} computed by $\mathscr{U}_A$ in Eq. 27. $\mathscr{U}_A$ generated fresh timestamp T_3, so it will pass the freshness test. Moreover, $\mathscr{U}_A$ has access to all parameters $RID_m \| RID_{DRn}$ extracted from verifier and $r_3 = h(RID_{DRn} \| RID_m \| T_3) \oplus M_{10}$ computed by $\mathscr{U}_A$ himself M_{sg2}, again by using verifier and the received message of $\mathscr{S}$. Therefore, M_{12} computed by $\mathscr{U}_A$ in Eq. 27 is same as $\mathscr{U}_m$ computed M_{13} in Eq. 31. Hence, $\mathscr{U}_A$ has successfully impersonated as the legal drone $\mathscr{DR}_n$ in the scheme of Wazid et al.□

3.2.4. Session Key Disclosure

Once the Drone $\mathscr{DR}_n$ has verified that the $\mathscr{S}$ is legal, then $\mathscr{DR}_n$ will compute and send message $M_{sg3} = \{M_{10}, M_{11}, M_{12}, T_3\}$ directly to $\mathscr{U}_m$, where M_{12} is hiding the session key SK_{ij} hashed with timestamp T_3. Attacker $\mathscr{U}_A$ will intercept this message as it is transmitted over the public channel. $\mathscr{U}_A$ extracts the session key on the basis of stolen parameters $\{ID_m, RID_m, TC_{Um}, RID_s, RID_{DRn}, TC_{DRn}\}$ from server as illustrated below:

$$r_3 = h(RID_{DR_n} \| RID_m \| T_3) \oplus M_{10} \tag{32}$$

$$M_8 = h(RID_m \| RID_{DRn} \| r_3) \oplus M_1 0 \tag{33}$$

$$SK_{ij}' = h(M_8' \| r_3' \| RID_m \parallel RID_{DRn}) \tag{34}$$

$$M_{13}' = h\left(SK_{ij}' \parallel T_3\right) \tag{35}$$

In Eq. 34, $\mathscr{U}_A$ has successfully computed the session key shared among $\mathscr{U}_m$ and $\mathscr{DR}_n$. Moreover, $\mathscr{U}_A$ can verify the truthfulness of session key by checking the validity of $M_{13} \stackrel{?}{=} M_1 2$. Therefore, $\mathscr{U}_A$ has successful disclosed the session key shared among a user and a drone just by listening the communication link and using the verifier.

4. Proposed Scheme

In this section, we present a brief review of our devised scheme for securing IoD. The scheme consists of three participants, namely User ($\mathscr{U}_m$), Server ($\mathscr{S}$) and Drone ($\mathscr{DR}_n$). The Server $\mathscr{S}$ in the proposed scheme provides the registration facility to the $\mathscr{U}_m$ and $\mathscr{DR}_n$. Following subsections provide brief overview of the phases of our scheme:

4.1. Pre-deployment Phase

$\mathscr{S}$ registers all drone $\mathscr{DR}_n : \{j = 1, 2....n\}$ before deployment in the

Table 1
Notation Guide

Symbols	Representations
$\mathscr{U}_m$, MD_m, $\mathscr{DR}_n$, $\mathscr{S}$	User, Mobile device, drone, Server
ID_m, PWD_m, BIO_m	$\mathscr{U}_m$'s identity, password, biometrics
ID_s, ID_{DRn}	ID's of $\mathscr{S}$, $\mathscr{DR}_n$
RID_m, RID_s, RID_{DRn}	Pseudo IDs of $\mathscr{U}_m$, $\mathscr{S}$, $\mathscr{DR}_n$
RTS_{Um}, RTS_{DRn}	Reg. timestamps of $\mathscr{U}_m$, $\mathscr{DR}_n$, $\mathscr{S}$
MK_{Um}, MK_{DRn}	Master key of $\mathscr{U}_m$, $\mathscr{DR}_n$
ΔT	Maximum transmission delay
$GEN(.)$, $Rep(.)$	Fuzzy extractor generation and reproduction parameter
$E_p(a,b)$	A singular elliptic curve
$h(.)$, $\|$, $\oplus$	One way hash, Concatenation, Bitwise XoR Functions
σ_m, τ_m	$\mathscr{U}_m$'s biometric secret key and public reproduction parameter
SK_{mn}, $\mathscr{A}$	Session key among entities $\mathscr{U}_m$ and $\mathscr{DR}_n$, Adversary

Table 2
Comparison of functionality features

Scheme→ ↓Features	Zhang [35]	Tai [36]	Srinivas [20]	Wazid [22]	Farash [6]	Ever [37]	Our
S_{r1}	✓	✓	×	✓	×	✓	✓
S_{r2}	✓	×	✓	×	×	-	✓
S_{r3}	✓	×	✓	✓	✓	✓	✓
S_{r4}	✓	×	✓	✓	✓	-	✓
S_{r5}	✓	✓	✓	✓	✓	-	✓
S_{r6}	×	×	✓	×	✓	✓	✓
S_{r7}	×	✓	✓	✓	✓	✓	✓
S_{r8}	✓	✓	✓	✓	✓	-	✓
S_{r9}	×	✓	✓	×	×	✓	✓
S_{r10}	×	✓	✓	×	✓	✓	✓
S_{r11}	✓	×	×	✓	×	-	✓
S_{r12}	✓	×	✓	✓	✓	-	✓
S_{r13}	✓	×	✓	✓	✓	-	✓
S_{r14}	✓	✓	✓	✓	✓	✓	✓
S_{r15}	✓	✓	✓	✓	✓	-	✓
S_{r16}	✓	✓	✓	✓	✓	-	✓
S_{r17}	✓	✓	✓	✓	✓	-	✓
S_{r18}	✓	✓	✓	×	✓	✓	✓
S_{r19}	✓	✓	×	×	✓	-	✓

Note: Note: S_{r1}:User anonymity; S_{r2}:Privileged-insider attack; S_{r3}:Password guessing attack; S_{r4}:Stolen mobile device or smart card attack; S_{r5}:Denial of service attack; S_{r6}:User Impersonation attack; S_{r7}:Replay attack; S_{r8}:Man-in-the middle attack; S_{r9}:Mutual authentication; S_{r10}:Session key agreement; S_{r11}: Untraceability; S_{r12}:Drone capture attack; S_{r13}:Password update phase; S_{r14}: Drone/sensing device capture attack; S_{r15}:Biometric update phase; S_{r16}:Key management phase; S_{r17}:Formal security verification; S_{r18}:Server impersonation attack; S_{r19}:Session key Security. ✓: The scheme provides the security feature, × : The scheme Lacks the security feature.

Table 3
Comparison of Communication Costs

Scheme	Number of messages	Number of bits
Zhang et al. [35]	3	1472
Tai et al. [36]	4	2560
Srinivas et al. [20]	3	1536
Wazid et al. [22]	3	1696
Farash et al. [6]	4	2752
Ever [37]	6	1920
Our	3	2061

IoD environment. Initially, $\mathscr{S}$ picks a unique identity ID_{DRn} then computes $TC_{DRn} = h(ID_{DRn} \| K_{ms})$ relevant to $\mathscr{DR}_n$. Then, $\mathscr{S}$ engraves $\{TC_{DRn}, ID_{DRn}\}$ in the memory of respective drone $\mathscr{DR}_n$ and stores the identity ID_{DRn} in its own database.

4.2. User Registration Phase

This subsection outlines the registration process for an arbitrary user $\mathscr{U}_m$ for gaining the real-time information (Surveillance or otherwise) from desired drone $\mathscr{DR}_n$ in the IoD environment. $\mathscr{U}_m$ and $\mathscr{S}$ performs following steps to complete registration.

PR 1: Initially, $\mathscr{U}_m$ picks and sends ID_m to $\mathscr{S}$ secretly. On receiving the ID_m, server $\mathscr{S}$ computes $RID_m = E_{K_{ms}}(ID_m \| r_m)$ and $TC_m = h(ID_m \| r_m \| K_{ms})$. Next $\mathscr{U}_m$ generates A randomly and sends $\{RID_m, ID_{DR_n}, TC_m, A\}$ to $\mathscr{U}_m$ and stores ID_m in its' identity table.
PR 2: On receiving the reply of $\mathscr{S}$, $\mathscr{U}_m$ picks PW_m, and inputs BIO_m using mobile device MD_m. MD_m creates the biometric secret key σ_m and its relevant τ_m as $Gen(BIOi) = (\sigma_m, \tau_m)$. Next, MD_m produces n ($160-bit$ secret number) for $\mathscr{U}_m$ and computes $RID_m^{'} = RID_m \oplus h(PW_m \| \sigma_m)$, $ID_{DR_n}^{'} = ID_{DR_n} \oplus h(ID_m \| PW_m \| \sigma_m)$, $TC_m^{'} = TC_m \oplus h(ID_m \| \sigma_m)$, $RPW_m = h(PW_m \| n)$, $A^{'} = A \oplus h(ID_m \| \sigma_m \| PW_m)$, $B = n \oplus h(PW_m \| ID_m \| \sigma_m)$ and $C = h(A \| RID_{DR_n} \| RPW_m \| \sigma_m)$. Now, MD_m engraves $\{RID_m^{'}, ID_{DRn}^{'}, TC_{Um}^{'}, A^{'}, B, C, \tau_m, Gen(.), Rep(.), h(.), t\}$ in its own memory.

4.3. Login and Authentication phase

Login and authentication phase of the devised scheme is invoked by $\mathscr{U}_m$ to get authenticated and establish a secure channel by sharing a secret key with $\mathscr{DR}_n$. Following steps accomplishes the login and authentication procedure:

PL 1: $\mathscr{U}_m$ submits $\{ID_m, PW_m\}$ pair to MD_m, and imprints BIO_m. MD_m computes $\sigma_m = Rep(BIO_m, \tau_m)$ and checks validity of biometrics. On successful validation of biometrics, MD_m computes: $RID_m = RID_m^{'} \oplus h(PW_m^{'} \| \sigma_m^{'})$, $ID_{DRn} = ID_{DRn}^{'} \oplus h(ID_m \| PW_m^{'} \| \sigma_m^{'})$, $TC_m = TC_m = TC_m^{'} \oplus h(ID_m \| \sigma_m^{'})$,$n = B \oplus h(PW_m^{'} \| ID_m \| \sigma_m^{'})$ and $RPW_m^{'} = h(PW_m \| n)$. MD_m then computes $A = A^{'} \oplus h(ID_m \| \sigma_m^{'} \| PW_m^{'})$ and $C^{'} = h(A \| ID_{DRn} \| RPW^{'}{}_m \| \sigma_m^{'})$. Afterward, MD_m verifies equality $C_m \stackrel{?}{=} C_m^{'}$. On failed equality, the process is aborted immediately. MD_m then picks $\{T_1, r_1\}$ pair and computes $M_1 = RID_m$, $M_2 = ID_{DRn} \oplus h(TC_u \| ID_m \| T_1)$, $M_3 = h(ID_s \| TC_m \| T_1) \oplus r_1$, $M_4 = h(ID_m \| ID_s \| ID_{DR_n} \| TC_m \| r_1 \| T_1)$. Finally, $M_{sg1} = (M_1, M_2, M_3, M_4, T_1)$ is sent to $\mathscr{S}$ on public channel.
PL 2: On receiving M_{sg1}, the $\mathscr{S}$ verifies the time-freshness ($|T_1 - T_1^*| \leqslant \Delta T$), on success, $\mathscr{S}$ computes $(ID_m \| r_m) = D_{K_{ms}}(M_1)$, and if ID_m exists, then further computes: $TC_m = h(ID_m \| r_m \| K_{ms})$, $ID_{DR_n} = M_2 \oplus h(TC_m \| ID_m \| T_1)$, $r_1^{'} = M_3 \oplus h(ID_s \| TC_m \| T_1)$ and $M_4^{'} = h(ID_m \| ID_s \| RID_{DR_n} \| TC_m \| r_1^{'} \| T_1)$. Now $\mathscr{S}$ verifies $M_4^{'} \stackrel{?}{=} M_4$, on success user is considered as authenticated else session is aborted immediately. $\mathscr{S}$ then picks the pair $\{r_2, T_2\}$ and computes: $TC_{DR_n} = h(ID_{DRn} \| K_{ms})$, $M_5 = h(TC_{DR_n} \| ID_{DR_n}) \oplus h(ID_s \| r_1 \| r_2)$, $M_6 = h(TC_{DRn} \| T_2) \oplus ID_m$, $M_7 = h(TC_{DRn} \| ID_{DRn} \| h(ID_s \| r_1 \| r_2) \| T_2)$ and $M_8 = E_{K_{ms}}(ID_m \| r_{mnew}) \oplus h(TC_m \| ID_m \| RID_m)$. $\mathscr{S}$ sends $M_{sg2} = \{M_5, M_6, M_7, M_8, T_2\}$ to $\mathscr{DR}_n$.
PL 3: $\mathscr{DR}_n$ on reception, first checks time-freshness ($|T_2 - T_2^*| \leqslant \Delta T$) and on success, $\mathscr{DR}_n$ computes $ID_m = M_6 \oplus h(TC_{DR_n} | T_2)$, $M_9 = M_5 \oplus h(TC_{DR_n} \| ID_{DR_n})$, $M_8 = M_5 \oplus h(TC_{DRn} \| RID_{DRn})$, $M_{10} = h(TC_{DR_n} \| ID_{DR_n} \| M_9 \| T_2)$, and checks $M_{10}^{'} \stackrel{?}{=} M_7$. On success, $\mathscr{S}$ is considered as authenticated by $\mathscr{DR}_n$; otherwise session is terminated immediately. $\mathscr{DR}_n$ then creates $\{r_3, T_3\}$ pair and computes

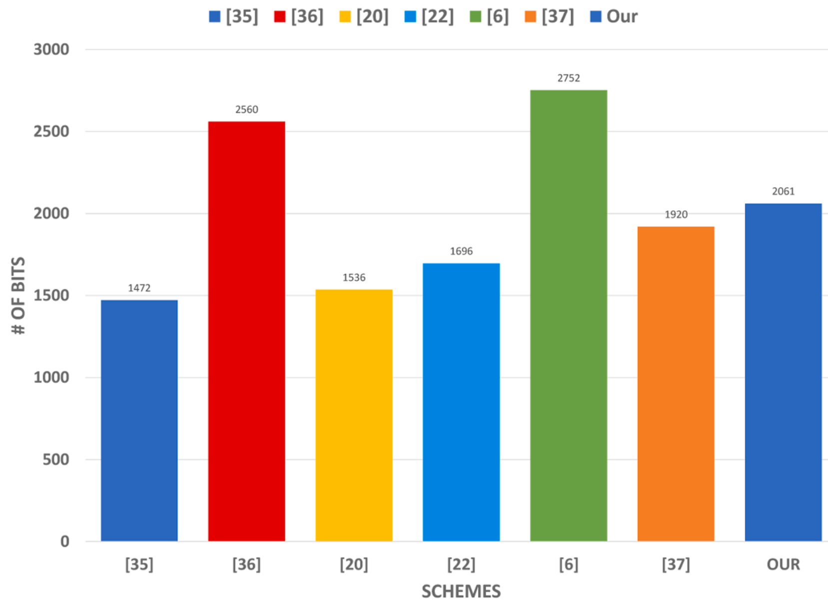

Fig. 4. Communication cost comparison graph

Table 4
Experimental Results

↓Operation/ Device→	Mobile	Server	Drone
T_b: Bilinear-Pairing	17.36	4.038	12.52
T_e: ECC Point Multiplication	5.116	0.926	4.107
T_a: ECC Point Addition	0.013	0.006	0.018
T_h: One way Hash	0.009	0.004	0.006
T_r: Random number Generation	2.011	0.118	1.185
T_{se}: Symmetric key Operations	0.017	0.08	0.013

Table 5
Comparison of Computation Costs

Protocol	User	Server	Drone	Total
Zhang et al. [35]	$10T_h$	$7T_h$	$7T_h$	$\approx 0.16ms$
Tai et al. [36]	$7T_h$	$6T_h$	$10T_h$	$\approx 0.147ms$
Srinivas et al. [20]	$14T_h + 1T_{fe}$	$9T_h$	$30T_h + 1T_{fe}$	$\approx 18.699ms$
Wazid et al. [22]	$16T_h + 1T_{fe}$	$8T_h$	$7T_h$	$\approx 5.334ms$
Farash et al. [6]	$11T_h$	$7T_h$	$14T_h$	$\approx 0.211ms$
Ever [37]	$5T_h + 2T_b$	$3T_h + 2T_b$	$9T_h + 2T_b + 4_e$	$\approx 84.375ms$
Our	$1T_{fe} + 15T_h$	$1T_{se} + 1T_{se} + 9T_h$	$7T_h$	$\approx 5.489ms$

$M_{11} = h(ID_{DR_n} \| ID_m \| T_3) \oplus r_3$, $SK_{mn} = h(M_9 \| r_3 \| ID_m \| ID_{DR_n})$, $M_{12} = h(ID_m \| ID_{DR_n} \| r_3) \oplus M_9$ and $M_{13} = h(SK_{mn} \| T_3)$. The Drone $\mathscr{DR}_n$ transmits reply message $M_{sg3} = \{M_{11}, M_{12}, M_{13}, T_3, M_8\}$ directly to user $\mathscr{U}_m$ through public channel.

PL 4: $\mathscr{U}_m$ after receiving the authentication reply, first checks time-freshness ($|T_3 - T_3^*| \leqslant \Delta T$), upon success, $\mathscr{U}_m$ computes $r_3' = h(ID_{DR_n} \| ID_m \| T_3) \oplus M_{11}$, $M_9' = h(ID_m \| ID_{DR_n} \| r_3') \oplus M_{12}$, $SK_{mn}' = h(M_9' \| r_3' \| ID_m \| ID_{DR_n})$ and $M_{14} = h(SK_{mn}' \| T_3)$. At last $\mathscr{U}_m$ checks whether $M_{14} \stackrel{?}{=} M13$. If the condition is true then Drone $\mathscr{DR}_n$ is considered as authenticated by User $\mathscr{U}_m$, and the session key SK_{mn}' is considered as correct for establishing secure communication in future. Now, $\mathscr{U}_m$ no computes $\overline{RID}_m = M_8 \oplus h(TC_m \| ID_m \| RID_m)$ and assigns $RID_m = \overline{RID}_m$.

5. Formal Security Analysis

In this section the formal analysis of the proposed scheme is conducted using the popular Burrows-Abadi-Needham (BAN) logic [33]. To perform the security analysis, we have defined the goals, idealized formation of the message and assumptions. At the end, we have demonstrated that the protocol achieves mutual authentication among the U_m, S and DR_n successfully. The following are the notations, followed for BAN logic analysis.

- $P| \equiv W$: P accepts statement W.
- $\#W$: The message W is fresh.
- $P \triangleleft W$: P sees W.
- $P| \sim W$: P once said W.
- $P| \Rightarrow W$: P has got jurisdiction over W
- $< W >_X$: The formulae W is hashed with X.
- $\{W\}_K$: W is encrypted by K.
- $P \stackrel{K}{\longleftrightarrow} Q$: P and Q can used shared key to communicate with each other.

BAN logic rules are as follows:

Rule 1: Message meaning rule

$\frac{P|\equiv P \stackrel{K}{\longleftrightarrow} Q, P \triangleleft <X>_K}{P|\equiv Q|\sim X}$

Rule 2: Nonce verification rule

$\frac{P|\equiv \#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$

Rule 3: Jurisdiction rule

$\frac{P|\equiv Q \Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$

Rule 4: Freshness rule

$\frac{P|\equiv \#(X)}{P|\equiv \#(X,Y)}$

Rule 5: Acceptance Conjunction

$\frac{P|\equiv X, P|\equiv Y}{P|\equiv (X,Y)}$

Rule 6: Session Key

$\frac{P|\equiv \#(X), P|\equiv Q \equiv X}{P|\equiv P \stackrel{K}{\longleftrightarrow} Q}$

To verify the mutual authentication following goals are set.

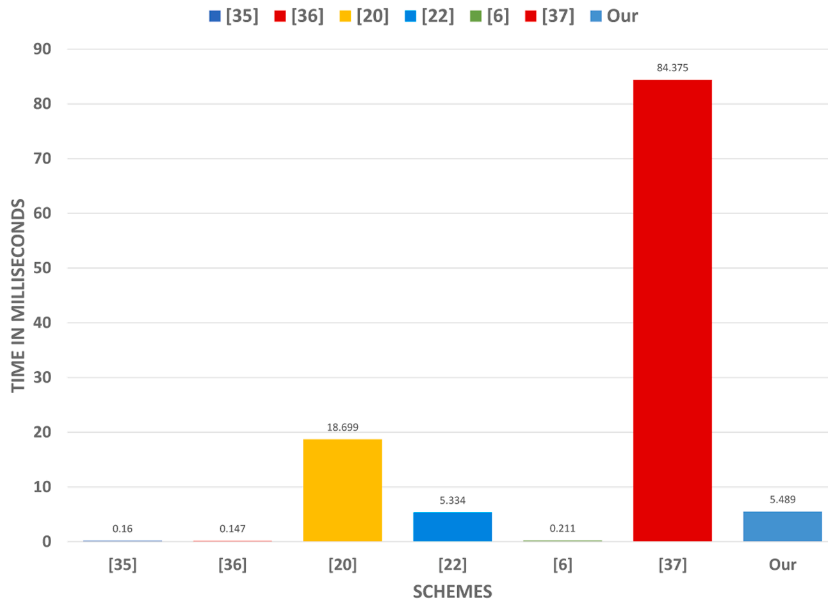

Fig. 5. Computation cost comparison graph

- Goal1 : $S|\equiv(r_1)$
- Goal2 : $S|\equiv U_m|\equiv(r_1)$
- Goal3 : $DR_n|\equiv(r_1)$
- Goal4 : $DR_n|\equiv U_m|\equiv(r_1)$
- Goal5 : $DR_n|\equiv(r_2)$
- Goal6 : $DR_n|\equiv S|\equiv(r_2)$
- Goal7 : $U_m|\equiv(r_3)$
- Goal8 : $U_m|\equiv DR_n|\equiv(r_3)$

Generic form of the protocol is shown as under:

- M1: $U_m\rightarrow S : M_1, M_2, M_3, M_4, T_1$
- M2: $S\rightarrow DR_n : M_5, M_6, M_7, M_8, T_2$
- M3: $DR_n\rightarrow U_m : M_{11}, M_{12}, M_{13}, T_3, M_8$

Protocol idealized form is as follows:

- M1: $U_m\rightarrow S : \{(U_m \overset{ID_{DR_n}}{\longleftrightarrow} S)_{(TC_m, ID_m)}, \langle r_1\rangle_{(U_m \overset{ID_{DR_s}}{\longleftrightarrow} S, TC_m)}, \langle r_1,\ TC_m\rangle_{(ID_m,\ U_m \overset{ID_I}{\longleftrightarrow}}$ $D_s,\ IDR_nS)\}$
- M2: $S\rightarrow DR_n : \{S \overset{TC_{DR_n},\ ID_{DR_n},\ ID_s}{\longleftrightarrow} \langle DR_n\rangle_{(r_1, r_2)}, \langle S\longleftrightarrow TC_{DR_n}DR_n\rangle_{ID_m}\}$
- M3: $DR_n\rightarrow U_m : \{\langle ID_{DR_n}, ID_m\rangle_{r_3}, \langle ID_m, ID_{DR_n}\rangle_{(r_3,\ M_9)}, U_m \overset{SK_{mn}}{\longleftrightarrow} DR_n\}$

For the BAN logic analysis following assumption are made.

- $A1 : S|\equiv\#(r_1)$
- $A2 : U_m|\equiv\#(r_3)$
- $A3 : DR_n|\equiv\#(r_2)$
- $A4 : DR_n|\equiv\#(r_1)$
- $A5 : DR_n|\equiv(Dr_n \overset{SK}{\longleftrightarrow} U_m)$
- $A6 : U_m|\equiv(DR_n \overset{SK}{\longleftrightarrow} U_m)$
- $A7 : S|\equiv U_m\Rightarrow(r_1)$
- $A8 : DR_n|\equiv S\Rightarrow(r_2)$
- $A9 : U_m|\equiv DR_n\Rightarrow(r_3)$

The proofs proceeds as follows:
According to Message 1:

- $S1 : S\triangleleft\{(U_m \overset{ID_{DR_n}}{\longleftrightarrow} S)_{(TC_m, ID_m)},\ <r_1>_{(U_m \overset{ID_{DR_s}}{\longleftrightarrow} S, TC_m)},\ <r_1,\ TC_m>_{(ID_m, U_m \overset{ID_{ID_s}, ID}{\longleftrightarrow}}$ $R_nS)\}$

From the message meaning rule according to S1 and A1:

- $S2 : S\Big|\equiv\{(U_m \overset{ID_{DR_n}}{\longleftrightarrow} S)_{(TC_m, ID_m)}, <r_1>_{(U_m \overset{ID_{DR_s}}{\longleftrightarrow} S, TC_m)}, <r_1, TC_m>_{(ID_m, U_m \overset{ID_{ID_s}, ID}{\longleftrightarrow}}$ $R_nS)\}$

In the view of A1, S2 nonce verification and freshness conjucatenation rules, we attain:

- $S3 : S|\equiv U_m|\equiv\#\{(U_m \overset{ID_{DR_n}}{\longleftrightarrow} S)_{(TC_m, ID_m)}, <r_1>_{(U_m \overset{ID_{DR_s}}{\longleftrightarrow} S, TC_m)}, <r_1, TC_m>_{(I}$ $D_m, U_m \overset{ID_{ID_s, IDR_n}}{\longleftrightarrow} S)\}$

According to A7, S3 and Jurisdiction rule:

- $S4 : S\Big|\equiv\{(U_m \overset{ID_{DR_n}}{\longleftrightarrow} S)_{(TC_m, ID_m)},\ <r_1>_{(U_m \overset{ID_{DR_s}}{\longleftrightarrow} S, TC_m)},\ <r_1,\ TC_m>_{(ID_m, U_m \overset{I}{\longleftrightarrow}}$ $D_{ID_s, IDR_n}S)\}$

According to S4:

- $S6 : S|\equiv U_m|\equiv U_m|\equiv r_1$ **(Goal 2)**

According to the jurisdiction rule with S6,and A1, we get:

- $S7 : S|\equiv r_1$ **(Goal 1)**

Assuming the second idealized of Message 2:

- $M2 : S\rightarrow DR_n : \{\langle S \overset{TC_{DR_n}, ID_{DR_n}, ID_s}{\longleftrightarrow} DR_n>_{(r_1, r_2)}, < S \overset{TC_{DR_n}}{\longleftrightarrow} DR_n>_{ID_m}\}$

By putting on seeing rule, we get:

- $S8 : DR_n\triangleleft : \{\langle S \overset{TC_{DR_n}, ID_{DR_n}, ID_s}{\longleftrightarrow} DR_n>_{(r_1, r_2)}, < S \overset{TC_{DR_n}}{\longleftrightarrow} DR_n>_{ID_m}\}$

According to S8, A3 and message meaning rule,

- $S9: DR_n| \equiv S \sim \{\langle S \overset{TC_{DR_n}, ID_{DR_n}, ID_s}{\longleftrightarrow} DR_n >_{(r_1, r_2)}, < S \overset{TC_{DR_n}}{\longleftrightarrow} DR_n >_{ID_m}\}$

According to A3, S9, nonce verification and freshness conjucatenation rules we achieve:

- $S10: DR_n| \equiv S| \equiv \#\{S \overset{TC_{DR_n}, ID_{DR_n}, ID_s}{\longleftrightarrow} DR_n >_{(r_1, r_2)}, < S \overset{TC_{DR_n}}{\longleftrightarrow} DR_n >_{ID_m}\}$

According to the nonce verification rule and S10, we get:

- $S11: DR_n| \equiv S| \equiv \{\langle S \overset{TC_{DR_n}, ID_{DR_n}, ID_s}{\longleftrightarrow} DR_n >_{(r_1, r_2)}, < S \overset{TC_{DR_n}}{\longleftrightarrow} DR_n >_{ID_m}\}$

According to the belief rule with S10, we get:

- $S12: DR_n| \equiv S| \equiv r_2$ **(Goal 6)**

According to A8, S12, and Jurisdiction rule :

- $S15: DR_n| \equiv r_2$ **(Goal 5)**

Considering the third idealized of Message 3:

- $M3: DR_n \rightarrow U_m: \{\langle ID_{DR_n}, ID_m >_{r_3}, < ID_m, ID_{DR_n} >_{(r_3, M_9)}, U_m \overset{SK_{mn}}{\longleftrightarrow} DR_n\}$

By applying seeing rule, we get:

- $S16: U_m \triangleleft : \{\langle ID_{DR_n}, ID_m >_{r_3}, < ID_m, ID_{DR_n} >_{(r_3, M_9)}, U_m \overset{SK_{mn}}{\longleftrightarrow} DR_n\}$

According to S16, A9 and message meaning rule:

- $S17: U_m| \equiv DR_n \sim \{\langle ID_{DR_n}, ID_m >_{r_3}, < ID_m, ID_{DR_n} >_{(r_3, M_9)}, U_m \overset{SK_{mn}}{\longleftrightarrow} DR_n\}$

According to A9, S17, nonce verification and freshness conjucatenation rules we achieve:

- $S18: U_m| \equiv DR_n| \equiv \#\{\langle ID_{DR_n}, ID_m >_{r_3}, < ID_m, ID_{DR_n} >_{(r_3, M_9)}, U_m \overset{SK_{mn}}{\longleftrightarrow} DR_n\}$

From the nonce verification rule and according to S18, we get:

- $S19: U_m| \equiv DR_n| \equiv \{\langle ID_{DR_n}, ID_m >_{r_3}, < ID_m, ID_{DR_n} >_{(r_3, M_9)}, U_m \overset{SK_{mn}}{\longleftrightarrow} DR_n\}$

According to the belief rule with S19, we get:

- $S20: U_m| \equiv DR_n| \equiv r_3$ **(Goal 4)**

According to A2, S20, and Jurisdiction rule:

- $S21: U_m| \equiv r_3$ **(Goal 7)**

Referring the BAN logic analysis, our proposed scheme successfully get mutual authentication between DR_n, S and U_m.

6. Further security discussion

This section informally verify that the proposed scheme is secure against different well known attacks. The detailed analysis is given in the following subsections:

6.1. Replay attack

Assume that $\mathscr{U}_A$ detains all messages $M_{sg1} = (M_1, M_2, M_3, M_4, T_1)$, $M_{sg2} = \{M_5, M_6, M_7, M_8, T_2\}$ and $M_{sg3} = \{M_{11}, M_{12}, M_{13}, T_3, M_8\}$ the exchanged among the participants in the course of the login and authentication phase over the insecure channel. Lets assume, $\mathscr{U}_A$ may attempt to replay the messages to find some useful information from the exchanged data. After the verification any delay or modification will detected as each message includes current timestamp and random numbers which will be limit the $\mathscr{U}_A$ to launch replay attack.

6.2. Offline password guessing attack

Let $\mathscr{U}_m$ be a registered valid user of the system and his/her smart device is accidentally stolen by an attacker which can be insider or outsider $\mathscr{U}_A$. The adversary $\mathscr{U}_A$ can retrieve the sensitive information $\{RID_m^{'}, ID_{DR_n}^{'}, TC_u^{'}, A^{'}, B, C, \tau_m, Gen(\mathring{u}), Rep(\mathring{u}), h(\mathring{u}), t\}$ from the mobile device through power analysis [30,34]. However, $\mathscr{U}_A$ cannot extract the unique parameters $A_i = h(ID_m \| PWD_m \| \sigma_m)$ because of the biometric key. Also, due to the hash function's one-way property $\mathscr{U}_A$ cannot retrieve the password and identity concurrently. Hence password guessing is not possible for the adversary $\mathscr{U}_A$.

6.3. User impersonation attack

Let $\mathscr{U}_A$ impersonate as a $\mathscr{U}_m$ to $\mathscr{S}$. To produce a correct login request message M_{sg1}, $\mathscr{U}_A$ is required to produce these credentials $\{RID_m^{\mathscr{U}_A}, ID*_{DR_n}^{'\mathscr{U}_A}, r_1^{\mathscr{U}_A}, M_i^{\mathscr{U}_A}, T_1^{\mathscr{U}_A}\}$ in order to pretend as legal user $\mathscr{U}_m$. Hence $\mathscr{U}_A$ is required to calculate all the above parameters in order to send the M_{sg1}. But $\mathscr{U}_A$ can create its own timestamp $T_u^{\mathscr{U}_A}$, chooses his own random number $r_1^{\mathscr{U}_A}$ and tries to computes $M_{sg1}^{\mathscr{U}_A}$, but without knowing unique parameters $\{RID_m, ID_m, PWD_m, r_1, \sigma_i\}$, $\mathscr{U}_A$ cannot initiate the login request message M_{sg1}. Hence, $\mathscr{U}_A$ will be unable to impersonate as a valid user $\mathscr{U}_m$.

6.4. Server impersonation attack

If $\mathscr{U}_A$ tries to impersonate as a server $\mathscr{S}$ towards the the drone $\mathscr{DR}_n$, in order to perform this $\mathscr{U}_A$ builds a message $M_{sg2}^{\mathscr{U}_A} = \{M_5^{\mathscr{U}_A}, M_{DR_6}^{\mathscr{U}_A}, M_6^{\mathscr{U}_A}, M_7^{\mathscr{U}_A}, M_8^{\mathscr{U}_A}, T_2^{\mathscr{U}_A}\}$. But, without knowing the $\{TC_{DR_n}, RID_{DR_n}, K_{ms}, ID_{DR_n}\}$, $\mathscr{U}_A$ can't impersonate as a server.

6.5. Drone impersonation attack

If $\mathscr{U}_A$ may impersonate as $\mathscr{DR}_n$ and forms a message $M_{sg3}^{\mathscr{U}_A} = \{M_{11}^{\mathscr{U}_A}, M_{12}^{\mathscr{U}_A}, M_{13}^{\mathscr{U}_A}, T_3^{\mathscr{U}_A}, M_8^{\mathscr{U}_A}\}$ by generating its own current timestamp $T_3^{\mathscr{U}_A}$ and initiating message to $\mathscr{U}_m$. However, without knowing $\{ID_{DR_n}, TC_{DR_n}, ID_m, r_3\}$, $\mathscr{U}_A$ cannot impersonate as a valid drone towards $\mathscr{U}_m$.

6.6. Anonymity and un-traceability

User $\mathscr{U}_m$ is not traceable to $\mathscr{U}_A$, because for each new session new random numbers and current timestamps are created, which guarantees distinct messages $M_{sg1}, M_{sg2}, M_{sg3}$ for each new session. Also, the pseudo or real identities of user and drone are never shared publically. These identities are used by both user and drone to communicate with each other and are discarded after each session. Hence, the scheme is anonymous and un-traceable.

6.7. Denial of service (DoS) attack

In proposed scheme, the user verification is performed locally by the smart device. The user $\mathscr{U}_m$ submits his credentials including password, identity and biometrics and based on the user input the device computes $C^{'} = h(A \| RID_{DR_n} \| RPW_m^{'} \| \sigma_m^{'})$ and checks it's equality with the stored C_m. The request is sent to server only if the $C_m \overset{?}{=} C_m^{'}$ holds. Therefore, proposed scheme cannot become a prey of DoS, on wrong inputs by the user.

6.8. Stolen mobile device attack

As shown in subsection 6.2 that if $\mathscr{U}_A$ steals the mobile device, still unable to retrieve the secret credentials. Hence, the stolen mobile device attack is not possible in the proposed scheme.

6.9. Man-in-the-Middle Attack

If $\mathscr{U}_{\mathscr{A}}$ intercepts the messages exchanged through public channel, and tries to modify M_{sg1} to another valid message $M_{sg1}^{\mathscr{U}_A}$, then create random nonce r_1 and current timestamp $T_1^{\mathscr{U}_A}$ and want to compute $M_1 = RID_m,\ M_2 = ID_{DRn} \oplus h(TC_u \| ID_m \| T_1),\ M_3 = h(ID_s \| TC_m \| T_1) \oplus r_1,\ M_4 = h(ID_m \| ID_s \| ID_{DR_n} \| TC_m \| r_1 \| T_1)$. Without the knowledge of secret parameters $RID_m,\ ID_m, TC_m,\ ID_{DR_n},\ ID_s,\ \mathscr{U}_{\mathscr{A}}$ will be unable to compute M_{sg1} and other two messages. Hence, our scheme is secure against man-in-the-middle attack.

6.10. Mutual authentication

In the proposed scheme, when MD_m receives the login request, it verifies the authenticity of the user $\mathscr{U}_m$ by the condition $C_m \stackrel{?}{=} C_m^{'}$ and if the condition is true MD_m authenticate the $\mathscr{U}_m$. On receiving M_{sg1}, the On receiving $\mathscr{S}$ verifies the condition $M_4^{'} \stackrel{?}{=} M_4$ to check the authenticity of the $\mathscr{U}_m$ and on the successful verification passes the message to the $\mathscr{DR}_n$. $\mathscr{DR}_n$ checks the authenticity of the $\mathscr{S}$ by verifying the condition $M_{10}^{'} \stackrel{?}{=} M_7$ and if condition is true, $\mathscr{DR}_n$ authenticates $\mathscr{U}_A$ directly and $\mathscr{S}$ indirectly. Moreover, on receiving the response message M_{sg3}, $\mathscr{U}_m$ also verifies the authenticity of the $\mathscr{DR}_n$ by checking $M_{14} \stackrel{?}{=} M13$. On successful validation $\mathscr{U}_m$ authenticates $\mathscr{S}$ indirectly and authenticates $\mathscr{DR}_n$ directly. Furthermore, the session key validation is done at $\mathscr{U}_m$ to confirm, both $\mathscr{U}_m$ and $\mathscr{DR}_n$ share the identical session key. Hence, it is evident that the participants successfully attain mutual authentication.

7. Performance Analysis

In this section, comparison of the proposed scheme with some recently related schemes [6,20,22,35–37] with respect to security features availability, computation cost and communication cost is shown.

7.1. Security Requirements

The security features comparison is shown in Table 2. The proposed scheme has provision of all the mentioned security features; whereas, other scheme [6,20,22,35–37] lacks one or more security features.

7.2. Communication Cost Comparison

Table 3 exhibits the communication cost comparison of some recently proposed scheme and the scheme proposed in this paper. For communication cost comparison, user identities are assumed as 160 bits, random numbers are assumed as 128 bits and timestamps are considered to be of 32 bits. Hash digest (if we apply the Secure Hash Standard (SHA-1) hash algorithm) takes 160 bits. Our scheme endures little bit higher communication cost than Zhang et al.'s, Srinivas et al.'s, Waizd et al's. and Ever schemes and less than the Tai et al.'s and Farash et al.'s schemes; in contrast, it is evident from earlier proofs that our scheme is more secure than the rest of the schemes. The communication cost comparison is also illustrated graphically in Figure 4.

7.3. Computation Cost Comparison

For counting the computation time and cost, we establish a real-time setup, where we perform an demonstration using MIRACL Library, over Smartphone: Xiaomi Redmi Note 8, with 4GB RAM and Octa-core Max 2.01GHz processor, the android version is 9 and MIUI version is 11.0.7, the smartphone exhibits a user/mobile device. For Server, we used HP EliteBook 8460P with Intel(R) Core(TM) i7-2620M 2.7 GHz Processor and 4GB RAM over Ubuntu 16.0 LTS operating system. Similarly, we have utilized Pi3 B+ with Cortex-A53(ARMv8) 64-bit SoC @ 1.4GHz processor, 1GB LPDDR2 SDRAM RAM to clone a drone. The simulation outcomes on each device are shown is Table 4; also, similarly as of [22], we consider $T_{fe} \approx T_e$, where T_{fe} is the running time of executing a fuzzy extractor. The table 5 expresses the relative computation cost analysis of our scheme and corresponding schemes [6,20,22,35–37]. It is evident from table 5 and figure 5 that the computation cost of our scheme is less than the [20,37], comparable with [22] and higher than the [6,35,36]. But, our scheme provides enhanced security and functionality features as compared to the [6,20,22,35–37].

8. Conclusion

In this article, we examined Wazid et al. scheme for the *IoD* environment. We proved that Wazid et al.'s scheme does not provide untracebility property, additionally, it is insecure against stolen verifier based user, server and drone impersonation attacks, as well as Wazid et al.'s scheme cannot resist the session key leakage attack against an adversary with knowledge of verifier. We have also shown that the server in Wazid et al.'s scheme broadcasts an authentication message towards all drones which badly effects computation power and battery life of drones. An improved scheme is then proposed to overcome the weaknesses of existing schemes including the scheme of Wazid et al. The performance analysis, formal BAN logic based security analysis and the discussion provided in this paper prove that proposed scheme resists known attacks with slight more computation and communication costs as compared with some of the existing schemes including the scheme of Wazid et al.

CRediT authorship contribution statement

Sajid Hussain: Writing – original draft, Writing – review & editing, Visualization. **Khalid Mahmood:** Validation, Formal analysis, Supervision, Writing – review & editing. **Muhammad Khurram Khan:** Writing – review & editing, Formal analysis. **Chien-Ming Chen:** Validation. **Bander A. Alzahrani:** Visualization, Investigation, Validation. **Shehzad Ashraf Chaudhry:** Conceptualization, Methodology, Software, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This paper is supported by Researchers Supporting Project number (RSP-2021/12), King Saud University, Riyadh, Saudi Arabia.

References

[1] G. Choudhary, V. Sharma, T. Gupta, J. Kim, I. You, Internet of drones (iod): Threats, vulnerability, and security perspectives. arXiv preprint arXiv:1808.00203 (2018).
[2] M. Gharibi, R. Boutaba, S.L. Waslander, Internet of drones, IEEE Access 4 (2016) 1148–1162.
[3] S.A. Chaudhry, K. Yahya, F. Al-Turjman, M.H. Yang, A secure and reliable device access control scheme for iot based sensor cloud systems, IEEE Access 8 (2020) 139244–139254, https://doi.org/10.1109/ACCESS.2020.3012121.
[4] S.A. Chaudhry, M.S. Farash, N. Kumar, M.H. Alsharif, Pflua-diot: A pairing free lightweight and unlinkable user access control scheme for distributed iot

environments, IEEE Systems Journal (2020) 1–8, https://doi.org/10.1109/JSYST.2020.3036425.
[5] C. Lin, D. He, N. Kumar, K.R. Choo, A. Vinel, X. Huang, Security and privacy for the internet of drones: Challenges and solutions, IEEE Communications Magazine 56 (1) (2018) 64–69.
[6] M.S. Farash, M. Turkanović, S. Kumari, M. Hölbl, An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment, Ad Hoc Networks 36 (2016) 152–176, https://doi.org/10.1016/j.adhoc.2015.05.014.
[7] J. Won, S. Seo, E. Bertino, Certificateless cryptographic protocols for efficient drone-based smart city applications, IEEE Access 5 (2017) 3721–3749, https://doi.org/10.1109/ACCESS.2017.2684128.
[8] A. Irshad, M. Usman, S.A. Chaudhry, H. Naqvi, M. Shafiq, A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework, IEEE Transactions on Industry Applications 56 (4) (2020) 4425–4435, https://doi.org/10.1109/TIA.2020.2966160.
[9] Z. Ali, S.A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, Y.B. Zikria, A clogging resistant secure authentication scheme for fog computing services, Computer Networks 185 (2021) 107731, https://doi.org/10.1016/j.comnet.2020.107731.
[10] F. Li, Y. Han, C. Jin, Practical access control for sensor networks in the context of the internet of things, Computer Communications 89-90 (2016) 154–164, https://doi.org/10.1016/j.comcom.2016.03.007.Internet of Things Research challenges and Solutions
[11] C.-I. Fan, Y.-H. Lin, Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics, IEEE Transactions on Information Forensics and Security 4 (4) (2009) 933–945.
[12] X. Huang, Y. Xiang, A. Chonka, J. Zhou, R.H. Deng, A generic framework for three-factor authentication: Preserving security and privacy in distributed systems, IEEE Transactions on Parallel and Distributed Systems 22 (8) (2010) 1390–1397.
[13] S. Kumari, M. Karuppiah, A.K. Das, X. Li, F. Wu, N. Kumar, A secure authentication scheme based on elliptic curve cryptography for iot and cloud servers, The Journal of Supercomputing 74 (12) (2018) 6428–6453.
[14] D. Kumar, H.K. Singh, C. Ahlawat, A secure three-factor authentication scheme for wireless sensor networks using ecc, Journal of Discrete Mathematical Sciences and Cryptography (2019) 1–22.
[15] S. Challa, A.K. Das, V. Odelu, N. Kumar, S. Kumari, M.K. Khan, A.V. Vasilakos, An efficient ecc-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks, Computers & Electrical Engineering 69 (2018) 534–554, https://doi.org/10.1016/j.compeleceng.2017.08.003.
[16] D. He, N. Kumar, J.-H. Lee, Privacy-preserving data aggregation scheme against internal attackers in smart grids, Wireless Networks 22 (2) (2016) 491–502.
[17] Y. Tian, J. Yuan, H. Song, Efficient privacy-preserving authentication framework for edge-assisted internet of drones, Journal of Information Security and Applications 48 (2019) 102354, https://doi.org/10.1016/j.jisa.2019.06.010.
[18] M. Tanveer, A.H. Zahid, M. Ahmad, A. Baz, H. Alhakami, Lake-iod: Lightweight authenticated key exchange protocol for the internet of drone environment, IEEE Access 8 (2020) 155645–155659, https://doi.org/10.1109/ACCESS.2020.3019367.
[19] S. Hussain, S.A. Chaudhry, O.A. Alomari, M.H. Alsharif, M.K. Khan, N. Kumar, Amassing the security: An ecc-based authentication scheme for internet of drones, IEEE Systems Journal (2021) 1–8, https://doi.org/10.1109/JSYST.2021.3057047.
[20] J. Srinivas, A.K. Das, N. Kumar, J.J.P.C. Rodrigues , Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment, IEEE Transactions on Vehicular Technology 68 (7) (2019) 6903–6916.
[21] S.A. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A.K. Bashir, Y.B. Zikria, Gcacs-iod: A certificate based generic access control scheme for internet of drones, Computer Networks 191 (2021) 107999, https://doi.org/10.1016/j.comnet.2021.107999.
[22] M. Wazid, A.K. Das, N. Kumar, A.V. Vasilakos, 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 of Things Journal 6 (2) (2019) 3572–3584.
[23] Z. Ali, S.A. Chaudhry, M.S. Ramzan, F. Al-Turjman, Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles, IEEE Access 8 (2020) 43711–43724.
[24] B. Bera, D. Chattaraj, A.K. Das, Designing secure blockchain-based access control scheme in iot-enabled internet of drones deployment, computer communications, Volume 153 (2020) 229–249.
[25] D. Dolev, A. Yao, On the security of public key protocols, IEEE Transactions on information theory 29 (2) (1983) 198–208.
[26] A. Irshad, S.A. Chaudhry, O.A. Alomari, K. Yahya, N. Kumar, A novel pairing-free lightweight authentication protocol for mobile cloud computing framework, IEEE Systems Journal (2020) 1–9, https://doi.org/10.1109/JSYST.2020.2998721.
[27] S.A. Chaudhry, I.L. Kim, S. Rho, M.S. Farash, T. Shon, An improved anonymous authentication scheme for distributed mobile cloud computing services, Cluster Computing 22 (1) (2019) 1595–1609.
[28] D. He, N. Kumar, S. Zeadally, A. Vinel, L.T. Yang, Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries, IEEE Transactions on Smart Grid 8 (5) (2017) 2411–2419.
[29] S.A. Chaudhry, Correcting æpalk: Password-based anonymous lightweight key agreement framework for smart gridg, International Journal of Electrical Power & Energy Systems 125 (2021) 106529, https://doi.org/10.1016/j.ijepes.2020.106529.
[30] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, M.T.M. Shalmani, On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme. Annual International Cryptology Conference, Springer, 2008, pp. 203–220.
[31] R. Amin, S.H. Islam, G. Biswas, M.K. Khan, N. Kumar, An efficient and practical smart card based anonymity preserving user authentication scheme for tmis using elliptic curve cryptography, Journal of medical systems 39 (11) (2015) 180.
[32] P. Kocher, J. Jaffe, B. Jun, Differential power analysis. Annual International Cryptology Conference, Springer, 1999, pp. 388–397.
[33] M. Burrows, M. Abadi, R.M. Needham, A logic of authentication, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 426 (1871) (1989) 233–271.
[34] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE transactions on computers 51 (5) (2002) 541–552.
[35] Y. Zhang, D. He, L. Li, B. Chen, A lightweight authentication and key agreement scheme for internet of drones, Computer Communications (2020).
[36] W.-L. Tai, Y.-F. Chang, W.-H. Li, An iot notion–based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks, Journal of Information Security and Applications 34 (2017) 133–141.
[37] Y.K. Ever, A secure authentication scheme framework for mobile-sinks used in the internet of drones applications, Computer Communications (2020).