Computers, Materials & Continua
DOI:10.32604/cmc.2022.020774
Article

A Secure Key Agreement Scheme for Unmanned Aerial Vehicles-Based Crowd Monitoring System

Bander Alzahrani[1], Ahmed Barnawi[1], Azeem Irshad[2], Areej Alhothali[1], Reem Alotaibi[1] and Muhammad Shafiq[3,*]

[1]Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[2]Department of Computer Science and Software Engineering, International Islamic University Islamabad, Pakistan
[3]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Korea
[*]Corresponding Author: Muhammad Shafiq. Email: shafiq@ynu.ac.kr
Received: 07 June 2021; Accepted: 16 August 2021

Abstract: Unmanned aerial vehicles (UAVs) have recently attracted widespread attention in civil and commercial applications. For example, UAVs (or drone) technology is increasingly used in crowd monitoring solutions due to its wider air footprint and the ability to capture data in real time. However, due to the open atmosphere, drones can easily be lost or captured by attackers when reporting information to the crowd management center. In addition, the attackers may initiate malicious detection to disrupt the crowd-sensing communication network. Therefore, security and privacy are one of the most significant challenges faced by drones or the Internet of Drones (IoD) that supports the Internet of Things (IoT). In the literature, we can find some authenticated key agreement (AKA) schemes to protect access control between entities involved in the IoD environment. However, the AKA scheme involves many vulnerabilities in terms of security and privacy. In this paper, we propose an enhanced AKA solution for crowd monitoring applications that require secure communication between drones and controlling entities. Our scheme supports key security features, including anti-forgery attacks, and confirms user privacy. The security characteristics of our scheme are analyzed by NS2 simulation and verified by a random oracle model. Our simulation results and proofs show that the proposed scheme sufficiently guarantees the security of crowd-aware communication.

Keywords: IoT; unmanned aerial vehicles; authentication; crowd monitoring

1 Introduction

Crowding usually occurs in major occasions, such as international games and sports competitions, cultural festivals, concerts, religious gatherings, etc. We cannot ignore the possibility of accidents in large gatherings, such as the Hajj 2006 or Love Parade 2010 in Germany, and the Kumbh Mela stampede reported in 2013 in the past few years [1,2]. The demand for crowd management solutions in urban metropolises is also becoming more and more common. Such gatherings always have risks, so precautions need to be taken in advance to ensure public safety. In addition, it is also important to use technology to identify anti-social and atypical behaviors in the population, and to distinguish these factors in order to take preventive measures to enhance public safety and security. Recently, the pandemic riot phenomenon needs to perceive crowd behavior without involving human factors, and further requires technological innovation to deal with it. In order to ensure public safety, the administrator or event manager must foresee and check the indicators of real-time data captured from the crowded terrain, and finally make timely decisions to curb unforeseen situations.

In the follow-up of major catastrophic situations such as floods, earthquakes, fire outbreaks, and rescue operations, Unmanned Aerial Vehicles (UAVs) are the first responders. According to observations, surveillance is one of the emerging fields, which has expanded the application range of UAVs (or drones). The sensors in drones help these devices effortlessly expand the scope of mission execution, so they are very suitable for surveillance-based rescue and monitoring operations [3,4]. The drone can focus on their target location and can easily provide the control team with key information about what is happening at that location. The economy of its use and the technological improvement of drones make these devices a strong competitor to improve the safety of surveillance and crowd monitoring operations.

UAVs can help police officers ensure the security and safety of large cities, because these devices can be introduced in real time to collect real-time updates on various actions on the spot. For example, police officers in the United Kingdom use drones to catch suspected robbers [5]. However, it becomes very challenging to manage the efficiency and effectiveness of such monitoring systems in cities. Other agencies such as the US Congress and the US Department of Justice have allowed the use of drones to manage large-scale events in large cities [6]. The combination of drones with multimedia streaming, safe wireless interaction, forensic applications, video detection technology for abnormal motion [7], and video recognition of human abnormal behavior [8] may help to achieve a peaceful living place.

Nevertheless, this development of drone network technology exposes new ways of cyber threats, such as eavesdropping, privacy, forgery, and data reconciliation issues, which makes crowd management very challenging. If any malicious adversary accesses surveillance-related data, it may disrupt the entire surveillance activity. If any legal mobile user wants to access the data collected by a specific drone introduced in the flight area, this must be possible in the follow-up process of the mutual authentication process, leading to an agreed session key. The gateway is a trusted entity that cannot be hacked by opponents, and the mobile user's equipment and drones may be physically compromised. Therefore, designing a secure and lightweight authentication key agreement is essential for the Internet of Drones (IoD) architecture to overcome the above shortcomings.

The salient features of the contribution are as follows:

- We propose a secure key agreement scheme for UAVs-based crowd sensing system. In the proposed scheme, police or intelligence personnel can safely obtain the real-time status of crowd dynamics with mobile devices by using crowd-sensing drones. These drones are used to report the perceived crowd information to the mobile user/police officer (CMD_i) through the reliable registration agency GRS_j after adopting an appropriate authentication process and using a mutually shared session key. However, this communication must be carried out between legitimate members after using a successful authentication procedure and establishing a mutually agreed session key

- We have verified the session key security of the proposed scheme using the ROR (Real-Or-Random) trusted model [9]. In addition, an informal security analysis was conducted to prove the security function of the proposed schemes against a capable adversary.
- We developed a simulation in NS2 to verify the efficiency of the proposed model in terms of throughput and latency benchmarks. The performance evaluation results show that the proposed scheme is sufficiently safe and efficient in computation and communication.

The rest of this article is organized as follows. Section 2 describes the related work. Section 3 explains the system model and adversary model. Section 4 demonstrates the proposed model. Section 5 analyzes the methods proposed on the formal and informal routes. Section 6 introduces the performance evaluation and comparative study of the proposed models. The conclusion is drawn in the last section.

2 Related Works

We can find some research articles on protecting drone-based surveillance [10]. In [11], for example, the authors proposed a UAV communication scheme for rescue operations. In [12], the authors demonstrated the advantages and disadvantages of using drones to monitor the US border. In [13], the authors proposed a security method based on multi-UAV architecture to manage catastrophic scenarios. In [14], the authors discussed equipment for monitoring crowds. In [15], the hierarchical intrusion detection is designed as a lightweight detection and response method to protect drone-based networks from known attacks. Since then, the Time Credential-based Anonymous Lightweight Authentication Scheme (TCALAS) has tried to solve the problems in key protocols related to drone networks. In [16], a certificate-less group key authentication protocol for untrusted drone architecture is proposed. In [17], the authors proposed another lightweight authentication protocol for drone Internet. However, this scheme does not support mutual authentication and so lacks a secure key agreement. In [18], the authors proposed a mobile user authentication protocol for wireless sensor networks related to the Internet-of-Things (IoTs) framework, which establishes an agreed session key with sensor nodes. However, this protocol is particularly suitable for sensor nodes with insufficient resources only and so it uses minimal hash-based operations and XOR operations to support mutual authentication among sensor nodes, mobile users, and gateway server nodes. In [19], authors proved that the scheme in [18] is vulnerable because it does not support anonymity and untraceability. In addition, this solution is susceptible to forgery attacks, stolen card attacks, and man-in-the-middle attacks. In [20], authors proposed a novel and efficient signature-based authentication protocol for IoT-based architecture, in which data is accessed from IoT sensors in real time after a mutual authentication process. However, no solution can meet the goals of real-world online application scenarios to make full use of a secure drone-based crowd sensing system.

3 Preliminaries

There is always a communication security threat between entities in the IoD environment. This requires the development of effective and efficient authentication protocols. The network model of the proposed framework is shown in Fig. 1, including three participating roles, such as control room (CR), ground registration station (GRS_j), mobile user (MU_i), and crowd monitoring drone (CMD_i). The IoD network consists of multiple flight zones with specific identifiers (FZ_i), and a specific UAV is deployed to any specific FZ_i, and at the same time it can fly and communicate with other GRS_j and drones of the same FZ_i. GRS_j acts as a trusted entity and is connected to the CR endpoint. GRS_j registers all mobile users and remote drones by providing long-term keys based on their identity. Mobile user MU_i or police officer with smart device obtain is/her own long-term key through GRS_j. The drone CMD_i introduced in a specific FZ_i can report to GRS_j in real time after scanning and monitoring crowd-based information.

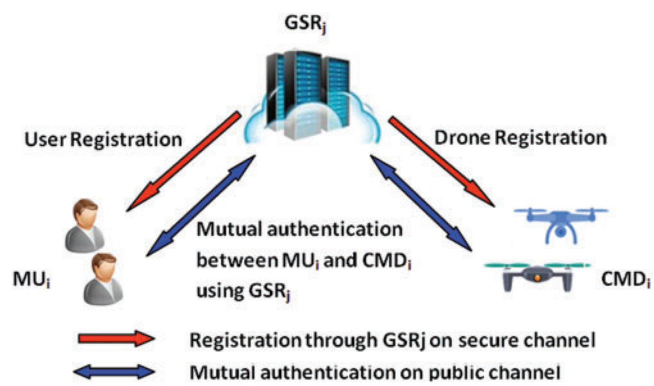

Figure 1: System model

We use the Dolev-Yao (DY) threat model to assume the capabilities of malicious adversaries. Under the DY threat model, adversary $\mathcal{A}$ can intercept, delete, modify, append or replay any eavesdropping messages exchanged on public channels. The adversary can physically capture the deployed drone in any FZ_i, steal the information stored in its memory and manipulate it to achieve its malicious objectives. It may also attempt to use this information to expose secret network communications by disrupting the data exchanged between the hijacked drone and other un-compromised drones. In addition, the $\mathcal{A}$ can also obtain smart card credentials such as identity, password, and biometric secrets by using power differential analysis attacks [21]. For the current solution, compared with the DY model, we assume another powerful threat model, namely the adversary model of Canetti and Krawczyk (also known as the CK-adversary model). Under the CK model, $\mathcal{A}$ can physically access the credentials of a single entity by recovering its content and calculating its corresponding session key and its session state.However, a sound agreement must retain the forward and backward secrecy under the CK model in the follow-up actions of the exposed credentials. In addition, assume that GRS_j is deployed in a physically protected lock system as a trusted entity in our IoD-based architecture, which is reliably protected from malicious attackers.

4 Proposed Scheme

Our proposed scheme consists of three sub-phases, namely the network establishment phase, the MU_i registration phase, the CMD_i registration phase and the mutual authentication procedure. Before we proceed, we have listed a summary of the symbols used in Tab. 1.

4.1 Network Setup

In the network setting, entities in the IoD network are initialized with key secret parameters before deployment on site. First, GRS_j constructs its master secret key and auxiliary parameters required in the protocol, as shown in the following.

- The GRS_j selects its 160-bit master secret key K_G as well as bit-mask key m_k along with a high entropy parameter n.

- The GRS_j selects its identity ID_{GR} and calculates $PID_G = h(ID_{GR} \| m_k)$.
- Next, GRS_j stores the parameters (K_G, m_k) secretly and publicizes the vector (h, n, PID_G).

Table 1: Summary of the notations

Notation	Description
MU_i, CMD_i	i-th mobile user, i-th crowd monitoring drone
GRS_j	j-th ground registration server, a trusted controlling authority
CR	Control room
ID_u, PW_u	Identity and password of MU_i
ID_d, ID_{GR}	Identities of CMD_i and GRS_j
K_G, m_k:	Master secret key and mask key of GRS_j
PID_u, PID_d, PID_G	Respective pseudonyms for MU_i, CMD_i and GRS_j
$\|$, $\oplus$:	Concatenation and exclusive-OR based functions

4.2 MU_i Registration Phase

In the MU_i registration phase, the user MU_i becomes part of the IoD system through the registration process. GRS_j uses confidential channels to perform MU_i registration by issuing secret parameters. This stage includes the following steps:

- The MU_i chooses its identity ID_u and password PW_u, and submits the identity ID_u as request message for registration towards GRS_j.
- Upon the receipt of registration message request from MU_i, the GRS_j calculates $PID_u = h(ID_u \| k)$, $B_i = h(ID_u \| K_G)$. Then, it stores the factors $\{ID_u, B_i, PID_u\}$ in its repository, and forwards the message $\{B_i, PID_u, PID_d\}$ to MU_i as shown in Fig. 2.
- The MU_i after receiving the message calculates $B_i' = h(ID_u \| PW_u) \oplus B_i$, $PID_u' = h(ID_u \| PW_u) \oplus PID_u$, and finally stores ($B_{i'}$, $PID_{u'}$, PID_d) in its memory.

4.3 CMD_i Registration Phase

The crowd monitoring drone CMD_i registers itself with GRS_j and becomes part of the IoD environment. In order to complete the registration, CMD_i performs the following steps:

- The CMD_i chooses its identity ID_d on random basis, and submits the same towards GRS_j to initiate the registration process.
- The GRS_j, then computes $PID_d = h(ID_d \| k)$, $B_j = h(ID_d \| K_G)$ and stores the parameters $\{ID_d, B_j, PID_d\}$ in its repository, and forwards the message $\{B_j, PID_d\}$ to CMD_i.
- The CMD_i, ultimately stores the same factors in its memory.

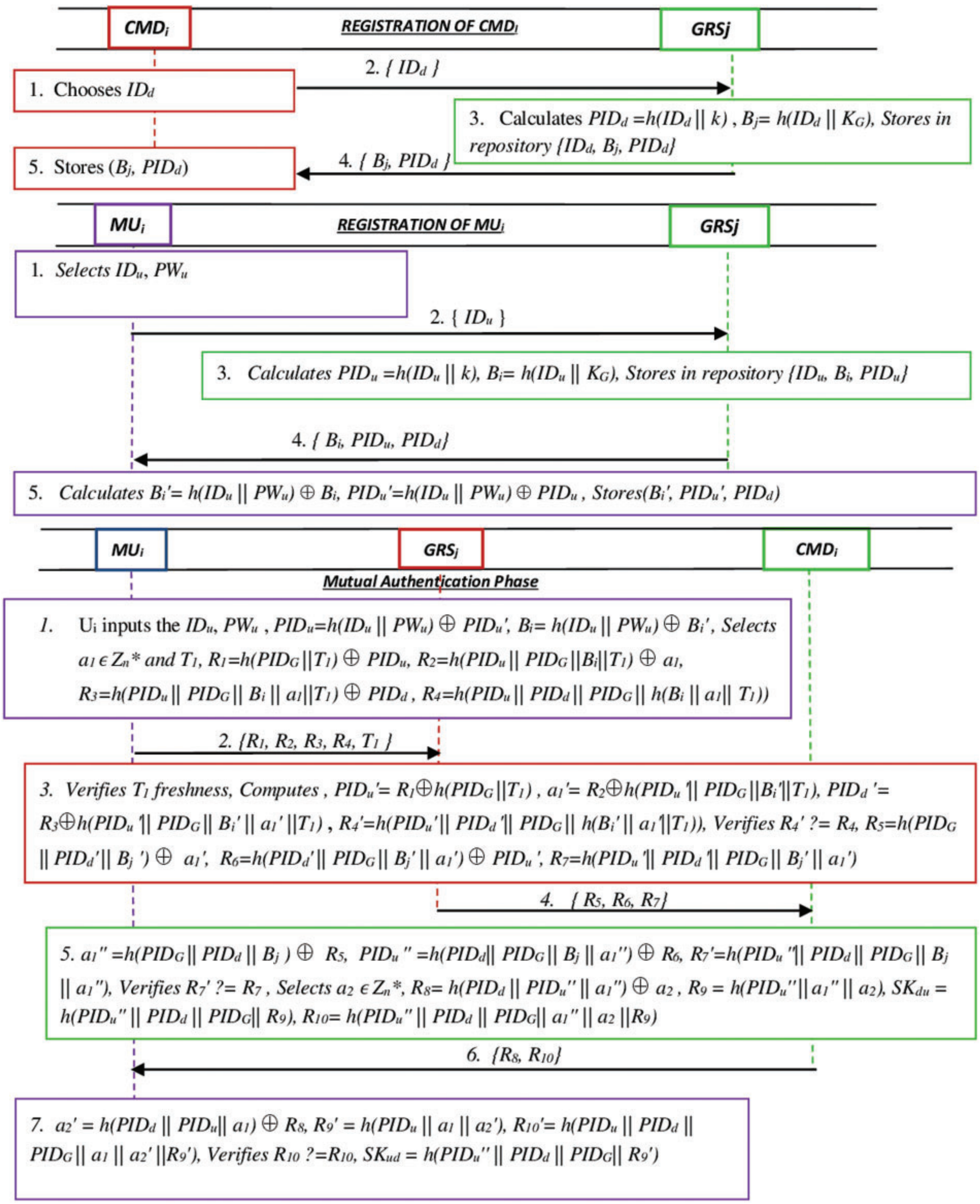

Figure 2: Proposed authentication model

4.4 Login and Authentication Phase

The MU_i and CMD_i participate in this stage to establish a mutual authentication session key at the end of the authentication session so that these entities can safely forward their data. The main steps at this stage can be described as follows:

- The MU_i inputs the identity ID_u and password PW_u into the mobile phone device. Then, the device calculates $PID_u = h(ID_u \,||\, PW_u) \oplus PID_{u'}$, $B_i = h(ID_u \,||\, PW_u) \oplus B_{i'}$. Next, it selects a random integer $a_1 \in Z_n^*$ and a fresh timestamp T_1. Next, it further computes $R_1 = h(PID_G \,||T_1) \oplus PID_u$, $R_2 = h(PID_u \,||\, PID_G \,||B_i \,||T_1) \oplus a_1$, $R_3 = h(PID_u \,||\, PID_G \,||\, B_i|| \,a_1 \,||T_1) \oplus PID_d$ and $R_4 = h(PID_u \,||\, PID_d \,||\, PID_G \,||\, h(B_i \,||\, a_1|| \,T_1))$. Next, it submits the message $\{R_1, R_2, R_3, R_4, T_1\}$ to the GRS_j.
- Upon the receipt of message from MU_i, the GRS_j verifies the freshness for T_1. If it is fresh, it calculates $PID_{u'} = R_1 \oplus h(PID_G \,||T_1)$ and retrieves $B_{i'}$ from repository LR, otherwise, rejects the session. Next, it calculates $a_{1'} = R_2 \oplus h(PID_u'||PID_G||B_i'||T_1)$, $PID_d' = R_3 \oplus h(PID_u'||PID_G||B_i'||a_1'||T_1)$, $R_4' = h(PID_u' \,||\, PID_d'||\, PID_G \,||\, B_i' \,||\, a_1'||T_1)$. Next, GRS_j verifies $R_4' \,?= R_4$, if it is false, it aborts the session. On the other hand, it calculates $R_5 = h(PID_G \,||\, PID_d' \,||\, B_j') \oplus a_1'$, $R_6 = h(PID_d' \,||\, PID_G \,||\, B_j' \,||\, a_1') \oplus PID_u'$, $R_7 = h(PID_u'||\, PID_d'|| \,PID_G \,||\, B_j' \,||\, a_1')$. In the last, it submits the message R_5, R_6, R_7 towards CMD_i.
- The CMD_i, after getting the message (R_5, R_6, R_7), calculates $a_1'' = h(PID_G \,||\, PID_d \,||\, B_j) \oplus R_5$, $PID_u'' = h(PID_d|| \,PID_G \,||\, B_j \,||\, a_1'') \oplus R_6$ and $R_7'' = h(PID_u''|| \,PID_d \,||\, PID_G \,||\, B_j \,||\, a_1'')$. Next, the CMD_i verifies the equality $R_7'' \,?= R_7$, it aborts the session if it is not true. On the other hand, it randomly selects a 160-bit integer $a_2 \in Z_n^*$ and calculates $R_8 = h(PID_d \,||\, PID_u'' \,||\, a_1'') \oplus a_2$, $R_9 = h(PID_u'' \,||\, a_1'' \,||\, a_2)$, $SK_{du} = h(PID_u'' \,||\, PID_d \,||\, PID_G|| \,R_9)$ and $R_{10} = h(PID_u'' \,||\, PID_d \,||\, PID_G|| \,a_1'' \,||\, a_2 \,||R_9)$. Finally, it submits the message R_8, R_{10} towards MU_i.
- The MU_i after getting the message (R_8, R_{10}) calculates $a_2' = h(PID_d \,||\, PID_u|| \,a_1) \oplus R_8$, $R_9' = h(PID_u \,||\, a_1 \,||\, a_2')$ and $R_{10}' = h(PID_u \,||\, PID_d \,||\, PID_G \,||\, a_1 \,||\, a2' \,||R_9')$. Next, it verifies the equation $R_{10} \,?= R_{10}$. If it does not hold valid, it terminates the session. On the other hand, it authenticates the CMD_i and calculates a mutual session key as $SK_{ud} = h(PID_u' \,||\, PID_d \,||\, PID_G|| \,R_9')$.

5 Security Evaluations and Analysis

We here formally prove that our scheme can resist the known attacks under the random oracle model. In addition, we informally stated that our plan is protected from contemporary threats. The following subsections consider both formal and informal security analysis.

5.1 Formal Security Analysis

We describe a model related to formal security analysis, which is described with the help of a game played between malicious $\mathcal{A}$ and challenger L. The adversary $\mathcal{A}$ is modeled as a Turing machine, which is simulated to operate in a possible polynomial amount of time (PPT) [22]. The challenger L models each oracle in the system. $\prod_g^x$ represents the x^{th} instance of the interactive participant $g = (MU_i, GRS_j, CMD_i)$. These oracles allow opponents to randomly issue a series of queries and trigger corresponding responses. The hash-based oracle keeps the hash list L_{Hs}. If $\mathcal{A}$ would execute hash-based query on message y, the challenger initially verifies the parameter using L_{Hs}. Upon the successful verification, the challenger returns the response $h(y)$ to the adversary and stores the vector (y, Y) in the list L_{Hs}. This query indicates the ability of an attacker to destroy a legitimate drone and obtain its private key. After the attacker executes the extraction query on the UAV ID_u's identity, the query returns the relevant key to the attacker. This oracle represents the capability of adversary for initiating an active attack. Upon submitting m to $\prod_g^x$, the attacker may receive the response from $\prod_g^x$ along with message *m*. In relation to the new oracle instance $\prod_g^x$, the attacker may launch submitting "Send ($\prod_g^x$, *Start*)" towards oracle.

The "Reveal" query models the erroneous use of the session key in the session. Upon the execution of Reveal query, in case the instance is effectively created, the challenger would return the session key SK for the instance $\prod_g^x$. On the other hand, it will return $\perp$. Using the Execute query (Execute (MU_i, CMD_i)), the adversary may eavesdrop all communication messages exchanged previously on insecure channel.

After the use of Test query (Test ($\prod_g^x$)), the attacker may distinguish among original session key and the randomly selected key. The adversary may execute this query just one time. The challenger selects a bit $b \epsilon (0, 1)$ at random and would return valid session key to adversary in case $b = 1$. On the other hand, it would return randomly selected secret key of the same size (i.e., $b = 0$). Alternatively, in case the queried oracle does not about the session key, challenger would return $\perp$ to adversary.

The adversary may employ the above mentioned queries, i.e., *Send, Reveal, Extract* after initiating the *Test* query [23]. Here, one disadvantage to $\mathcal{A}$ is that it may not launch the Reveal query either for oracle or the pattern oracle which employed the Test query for its execution. Finally, the adversary returns the output Φ' after making its guess Φ. Here we can remark that the adversary could auspiciously win this game as a result of breaking the authenticated key agreement (AKE) of contributed protocol Σ in case Φ' becomes equal to Φ. The benefit of $\mathcal{A}$ may be described as $adv_{\Sigma}^{AKE}(\mathcal{A}) = |2\Pr[\Phi' = \Phi] - 1|$.

Definition 1 (AKE-secure): When there is a negligible polynomial probability, the adversary may auspiciously win that game with a non-negligible benefit $adv_{\Sigma}^{AKE}(\mathcal{A})$, and we may infer that the contributed protocol Σ is AKE-secure.

The adversary may positively compromise the mutual authenticity of the contributed protocol Σ, in case the adversary could forge the legitimate authentication message, i.e., either authentication request or corresponding response. Suppose $E_{\text{MU-GRS}}$ represents the event that the adversary forges the MU_i and constructs the login request acknowledged by GRS_j. Also $E_{\text{MU-CMD}}$ characterizes the event that $\mathcal{A}$ masquerades the CMD_i and produces the response which is acknowledged by MU_i. The benefit of the adversary for being successful in this game can be described as $adv_{\Sigma}^{ME}(\mathcal{A}) = \Pr[E_{\text{MU-GRS}}] + \Pr[E_{\text{MU-CMD}}]$.

Definition 2 (ME-secure): In case there exists no probability for any polynomial time attacker such that one may auspiciously win the game with considerable benefit $adv_{\Sigma}^{ME}(\mathcal{A})$, we term the proposed protocol Σ as ME-Secure.

5.2 Proof

We acknowledge that there lies no adversary $\mathcal{A}$ that may impersonate as a legitimate authentication and response message with non-negligible chance. This certifies that the contributed protocol is AKE-secure and ME-secure regarding the provable security strength.

Lemma1: We assume that a polynomial time attacker $\mathcal{A}$ may compute a legitimate authentication request and response message with non-negligible chance. Thus, there lies a challenger C who may estimate a 160-bit randomly defined integer with success having non-negligible probability.

Proof: The challenger chooses a 160-bit randomly generated integer q, and submits the factors $\{h, n\}$ towards the adversary. The challenger produces a new hash-list L_{Hs}, which is blank on initial basis, and is meant for recording the query inputs as well as outputs for hash-based oracles. Then, it chooses two random drone identities, such as ID_U and ID_D to proceed. We assume that the rest of the oracles may be queried once the hash-based oracles perform their function. The queries' responses are illustrated as under:

$h(y_i)$: The challenger initially verifies the occurrence of y_i in the L_{Hs} list. If it exists in the list, the challenger would return Y_i to attacker. Otherwise, it selects a random integer Y_i, inserts $(y_i,\ Y_i)$ in the L_{Hs} list and returns the same Y_i to attacker.

Extract (ID_U): In case $u \neq U$, D, the challenger searches for the tuple $(ID_u \,||\, K_G,\ B_i)$ in L_{Hs} list, and would return B_i to the $\mathcal{A}$. On the other hand, the challenger aborts the oracle query and terminates the game. Send $(\prod_g^x, m)$: The attacker may use the Send query for modeling this active threat in four ways:

Send ($\prod_g^x Start$): The challenger searches for the hashing list L_{Hs} to find the secret key B_i for MU_i by checking the inequality for $u \neq U$ in the list. Using the secret key B_i, the challenger selects a randomly defined integer $n_1 \epsilon Z_n{}^*$, the fresh timestamp T_1, and calculates $(R_1,\ R_2,\ R_3,\ R_4,\ T_1)$. However if the equality does not hold, the challenger chooses three random integers $V_1,\ V_2,\ V_3 \ \epsilon\ Z_n{}^*$ and would set $R_1 \leftarrow V_1,\ R_2 \leftarrow V_2,\ R_3 \leftarrow R_3$. It then calculates $V_1 = h(PID_G \,||T_1) \oplus PID_U$ and returns the $(R_1,\ R_2,\ R_3,\ R_4)$ to the attacker.

Send ($\prod_{\mathrm{CMD_i}}^k$, $(R_5,\ R_6,\ R_7)$): The challenger upon the receipt of message, verifies the inequality for $d \neq D$. If the equality holds, the challenger discards the message and chooses randomly two integers $V_2,\ V_3\ \epsilon Z_n{}^*$ and will set $R_8 \leftarrow V_4,\ R_{10} \leftarrow V_5,\ R_3 \leftarrow V_3$. On the other hand, the challenger searches for the hash list L_{Hs} to find the secret key B_j for CMD_i and proceeds with the normal execution of the protocol.

Send ($\prod_{\mathrm{MU_i}}^t$, $(R_8,\ R_{10})$): The challenger now confirms the equality for $d \neq D$. If it is valid, then searches for CMD_i's secret B_j in the hash-list L_{Hs}. It generates randomly an integer $n_2 \ \epsilon\ Z_n{}^*$ and computes $(R_8,\ R_{10})$ using B_j. If the inequality does not hold, it chooses three integers on random basis as $V_4,\ V_5,\ V_6 \ \epsilon\ Z_n{}^*$, and would set $n_2 \ \leftarrow V_4,\ R_8 \leftarrow V_5,\ R_{10} \leftarrow V_6$ and returns the tuple $(R_8,\ R_{10})$ to MU_i.

Reveal ($\prod_g^t$)): In case the instance $\prod_g^t$ is accepted, the challenger would return the valid session key as SK_{ud}, otherwise it will return $\perp$. We assume that an attacker may compute valid login message request or response with success, or alternatively it may compute the responses $(R_1,\ R_2,\ R_3,\ R_4)$ to Send ($\prod_{\mathrm{MU_i}}^t$, *Start*) oracle query having $u = U$ and $(R_8,\ R_{10})$ to Send ($\prod_{\mathrm{CMD_i}}^k$, $(R_5,\ R_6,\ R_7)$) oracle query with d=D are verified by the GRS_j and MU_i entities. For computing the advantage for the challenger, we define the under-mentioned events as: E_{v1}: The modeling is not terminated. E_{v2}: The attacker sends the computed login request $(R_1,\ R_2,\ R_3,\ R_4)$ by employing Send ($\prod_{\mathrm{MU_i}}^t$, *Start*) or some valid response message $(R_8,\ R_{10})$ using Send ($\prod_{\mathrm{CMD_i}}^k$, $(R_5,\ R_6,\ R_7)$), however the queries Extract(ID_U) and Extract(ID_D) were never employed. E_{v3}: MU_i=MU_U or CMD_i=CMD_D. E_{v4}: The challenger may select any of the valid records from hash-list L_{Hs}.

We assume q_{sd}, q_{LR} and q_{LHs} represent the number of *Send*, L_R and L_{Hs} queries executed by the adversary.

$$\Pr[E_{v1}] \geq \frac{1}{q_{sd}} \tag{1}$$

$$\Pr[E_{v2}|E_v 1] \geq \epsilon \tag{2}$$

$$\Pr[\mathrm{E_{v4}}|\mathrm{E_{v3}} \wedge \mathrm{E_{v2}} \wedge \mathrm{Ev1}] \geq \frac{1}{q_{LR}}\frac{1}{q_{LR-1}} + \frac{a}{q_{LHs}}\frac{b}{q_{LHs}-a}$$

where a represents the valid record index in Send ($\prod_{\mathrm{MU_i}}^{t}$,*Start*) oracle, while b characterizes the frequency of Send ($\prod_{\mathrm{MU_i}}^{t}$, ($R_8$, R_{10})) queries. Thus, the challenger would guess 160-bit random integer auspiciously with non-negligible prospect as shown in Eqs. (1) and (2).

$$\Pr[\mathrm{E_{v1}} \wedge \mathrm{E_{v2}} \wedge \mathrm{E_{v3}} \wedge \mathrm{E_{v4}}] = \Pr[\mathrm{E_{v4}}|\mathrm{E_{v3}} \wedge \mathrm{E_{v2}} \wedge \mathrm{E_{v1}}]\Pr[\mathrm{E_{v3}}|\mathrm{E_{v2}} \wedge \mathrm{E_{v1}}]\Pr[\mathrm{E_{v2}}|\mathrm{E_{v1}}]$$

$$= \frac{1}{q_{sd}}\frac{1}{q_{LR}}\left(\frac{1}{q_{LR}}\frac{1}{q_{LR}} - 1 + \frac{a}{q_{LHs}}\frac{b}{q_{LHs}-a}\right)\epsilon \tag{3}$$

Nonetheless, this shows the contradiction regarding the hardness for guessing 160-bit random integer as shown in Eq. (3). Alternatively, the attacker may not construct a legitimate login request or response message, so the drones in the protocol may verify the authenticity of one another.

Theorem 1. The proposed protocol is ME-Secure for rigid guessing of 160-bit random integer.

According to Lemma1, no adversary may construct a legitimate login request or response message for guessing the high entropy 160-bit random integer. Thus, the contributed protocol is ME-Secure.

Theorem 2. The proposed protocol is AKE-Secure for rigid guessing of 160-bit random integer.

Proof. We assume that the probabilistic polynomial-time attacker produces the valid *b'=b* with non-negligible chance ϵ upon the execution of Test oracle query. Consequently the challenger may deduce 160-bit randomly defined integer with success having non-negligible prospect. For calculating the advantage of challenger, the understated events are described here:

- $\mathrm{E_{SKi}}$: The adversary may get the legitimate session key upon the execution of Test query.
- $\mathrm{E_{MU}}$: The adversary runs the Test query for the instance $\prod_{\mathrm{CMD_i}}$ auspiciously.
- $\mathrm{E_{CMD}}$: The adversary runs the Test query with success for the instance $\prod_{\mathrm{CMD_D}}$.

$\mathrm{E_{MUi\text{-}GRSj\text{-}CMDi}}$: The adversary may disrupt the authentication session between $\mathrm{MU_i}$ and $\mathrm{GRS_j}$, as well as between $\mathrm{MU_i}$ and $\mathrm{CMD_i}$. It is known that the attacker may guess the valid b with the missing information of b as $\frac{1}{2}$. Hence we have the equation $\Pr[\mathrm{E_{SKi}}] \geq \epsilon/2$

$$\begin{aligned}\Pr[\mathrm{E_{SKi}}] =& \Pr[\mathrm{E_{SKi}} \wedge \mathrm{E_{CMDi}}] + \Pr[\mathrm{E_{SKi}} \wedge \mathrm{E_{CMDi}} \wedge \mathrm{E_{MUi-GRSj-CMDi}}] + \Pr[\mathrm{E_{SKi}} \wedge \mathrm{E_{CMDi}} \wedge \\ & \neg\mathrm{E_{MUi-GRSj-CMDi}}] \leq \Pr[\mathrm{E_{SKi}} \wedge \mathrm{E_{MUi}}] + \Pr[\mathrm{E_{MUi-GRSj-CMDi}}] \\ & + \Pr[\mathrm{E_{SKi}} \wedge \mathrm{E_{CMDi}} \wedge \mathrm{E_{MUi-GRSj-CMDi}}]\end{aligned} \tag{4}$$

Hence

$$\begin{aligned}&\Pr[\mathrm{E_{SKi}} \wedge \mathrm{E_{MUi}}] + \Pr[\mathrm{E_{SKi}} \wedge \mathrm{E_{CMDi}} \wedge \neg\mathrm{E_{MUi-GRSj-CMDi}}] \geq \Pr[\mathrm{E_{Ski}}] \\ &- \Pr[\mathrm{E_{MUi-GRSj-CMDi}}] - \Pr[\mathrm{E_{MUi-GSRi-CMDi}}] \geq \varepsilon/2 - \Pr[\mathrm{E_{MUi-GSRi-CMDi}}]\end{aligned} \tag{5}$$

In relation to $\Pr[E_{CMDi}E_{MUi\text{-}GRSj\text{-}CMDi}] = \Pr[E_{CMDi}]$

We have $\Pr[E_{SKi} \wedge E_{CMDi}] \geq \frac{\epsilon}{4} - \frac{\Pr[E_{MUi-GRSj-CMDi}]}{2}$

The events $E_{SKi} \wedge E_{CMDi}$ depicts that the adversary forges MU_i and receives the valid session key with success. In accordance with the Lemma 1, $E_{MUi\text{-}GRSj\text{-}CMDi}$ has quite insignificant probability, hence the probability $\frac{\epsilon}{4} - \frac{\Pr[E_{MUi-GRSj-CMDi}]}{2}$ is not negligible as shown in Eqs. (4) and (5). This suggests that the probability the adversary may compromise the legitimate session key is not negligible which contradicts the hardness assumption for guessing 160-bit random integer.

5.3 Informal Security Analysis

5.3.1 Mutual Authentication

The proposed scheme provides mutual authenticity to both participants by devising a unique and mutual agreed session key between them. We know that the benefit that adversary may take by launching the login as well as an authentication request and response message is quite negligible due to illustrated lemma1 in above section [24]. Hence, the MU_i and CMD_i could mutually authenticate one another with the assistance of GRS_j. Hence, the proposed approach supports mutual authentication.

5.3.2 Anonymity

In the contributed protocol the MU_i does not send its identity plainly on pubic channel, rather it is masked in the form of $PID_u = h(ID_u \,||\, k)$. Furthermore, PID_u is integrated in the message $R_1 = h(PID_G \,||T_1) \oplus PID_u$ during mutual authentication process. It is hard problem in polynomial terms to recover the 160-bit random integer on account of guessing the values [25], so it is not feasible to calculate the legitimate identity of mobile drone CMD_i without compromising the high entropy factor k. Thus our scheme affirms anonymity to the participants in protocol.

5.3.3 Un-traceability

We employ random integers a_1 and a_2 along with fresh timestamps in different sessions which enable the constructed messages (R_1, R_2, R_3, R_4) in a session to be unique each time these are generated [26,27]. The attacker may not be able to distinguish the exchanged messages among for MU_i and CMD_i across various sessions. Furthermore, the legal identifies such as ID_u or PID_u are used in collision-resistant one hash function which enables the protocol in affording the untraceability feature.

5.3.4 Protected Session Key

In the proposed scheme, the MU_i confirms the authenticity of CMD_i through validating R_{10}, which ensures that both of these entities are having the legitimate randomly generated factors, a_1 and a_2. In this manner both entities construct a secure session key $SK = SK_{ud} = SK_{du} = h(PID_u \,||\, PID_d \,||\, PID_G || \, R_9)$ to interact in the future. Hence our scheme supports secure key agreement between the members.

5.3.5 Impersonation Threat

In case the adversary is able to capture the legal drone physically, it may access all of the stored information in its memory including pseudonym identities for CMD_i [28,29]. Then if the adversary attempts to forge the legal MU_i, it would construct the legal messages (R_1, R_4) and submit towards GRS_j. Now the adversary may compute the correct $R_1 = h(PID_G \,||T_1) \oplus PID_u$ and $R_4 = h(PID_u \,||\, PID_d \,||\, PID_G \,||\, h(B_i \,||\, a_1|| \, T_1))$, while a_1 and B_i depict the random integers as chosen by the adversary for random number and the protected key, respectively. After the receipt of (R_1, R_4), the GRS_j initially would parse from R_1 and recover the related secret as B_i in the list LHs. Thereafter, the GRS_j calculates the parameter R_4' along with another factor B_i and verifies the equation validity as $R_1' = R_1$. Therefore, the attacker does not expose the valid parameter B_i, and make the GRS_j distinguish the MU_u from legal user.

5.3.6 Server Masquerading Attack

The attacker may impersonate himself as GRS_j and submits the message R_7 towards the CMD_i. Then the attacker calculates $R_7 = h(PID'_u || PID'_d || PID_G || B'_j || a'_1)$, where B_j acts as a random integer chosen as CMD_i's private key by the adversary. After the receipt of R_7, the CMD_i constructs R_4' along with B_j and also checks the equality for $R_7' ?= R_7$. Nonetheless, the adversary may not access the B_j parameter or the CMD_i accesses the malicious server. Thus, our scheme is resistant to the spoofing attack.

5.3.7 CMD_i Capture Threat

The drones are vulnerable in the hands of adversaries, and could be physically compromised at any time. We assume that the adversary captures e number of drones and access the stored contents including $B_j = h(ID_d || K_G)$, $PID_d = h(ID_d || k)$, and $SK_{ud} = h(PID_u || PID_d || PID_G || R'_9)$ where $j = (1 \leq j \leq e)$ [30]. The master secret K_G and other masking key k are also used to mask the crucial factors in collision resistant hash function. Despite the access of information in the compromised several drones e, the adversary might not be able to access the K_G and k. At the same time, the session key $SK_{ud} = h(PID_u || PID_d || PID_G || R'_9)$ is composed of random integers and pseudonyms, the attacker may not calculate the subsequent session keys if it is not able to access the random integers. Consequently, our proposed model is immune to all physical drone capture threats.

5.3.8 Stolen MU_i's Smart Device Threat

In case the adversary is able to approach the MU_i's smart device and recover its contents (B'_i, PID'_u, PID_d) using differential analysis, where $B'_i = h(ID_u || PW_u) \oplus B_i$ and $PID'_u = h(ID_u || PW_u) \oplus PID_u$. The attacker may guess the password from B'_i only if it can test its accuracy, however without the MU_i's identity it cannot verify it. Thus, our scheme is resistant to the stolen device threat.

5.3.9 Replay Attack

The participants MU_i and GRS_j select random numbers and compute the login request message and response message as R_4 and R_{10}, respectively. Since the random nonces are fresh, the participants GRS_j, CMD_i and MU_i might discern the legitimate requests from the replayed messages through verification checks. Hence, our scheme is immune to this replay attack threat.

5.3.10 Known Session Key Attack

If an attacker becomes familiar about the current session key of any session in our scheme, it may not compute the previous session keys employing the current session key [31]. This is because the attacker needs to approach crucial pseudonym parameters besides the random nonces to construct the legal session key, however these parameters are protected under collision resistant one way hash function and cannot be compromised in polynomial amount of time.

6 Performance Evaluations

This section evaluates the performance of contributed protocol against the comparative studies including Wazid et al., Singh et al., Challa et al., and Turkanovic et al. on the basis of computational and communicational costs. The execution latency for the crypto-primitives employed by the comparative schemes [17,18,20,22] is depicted as T_{fe} to execute fuzzy extractor operation, T_h to execute one-way hash operation, T_{ex} to execute modular exponentiation operation, T_m to execute modular multiplication operation, T_{ecm} to execute (Elliptic Curve Cryptography) ECC-based point multiplication [31]. These crypto-primitive operations have been implemented for mobile user device as client and desktop computer as server. The mobile drones or user devices are equipped with biochemical detectors, infrared, microphone and camera-based sensors. We calculate the cost of computations with the help of MIRACL library [23] and Android-enabled MU_i/CMD_i client (Lenovo Zuk Z1 having 2.5Ghz Quad-core microprocessor, Android V5.1.2 OS, and 4GB RAM). To simulate the GRS_j environment we used desktop computer (HP E8300 Core i5 2.96Ghz, Ubuntu 16.12 OS and 8GB RAM). The experiments were conducted on the discussed client and server hardware platform that provides varying execution costs for various primitives. We select a multiplicative cyclic group G with order n having 160-bit prime integer.

This group G helps to achieve the 1024-bit RSA level of security. Using the above simulation, the execution timing of various crypto-primitives such as $T_{fe} \approx T_{ecm}$, T_h, T_{ex}, T_m and T_{ecm} is computed as 16.403, 0.078, 3.943, 0.012 and 0.012 ms for MU_i/CMD_i, and 6.276, 0.013, 0.438, 0.003 and 0.003 ms, respectively. In [17], the mobile user takes $1T_{fe} + 16T_h$ computational cost with 17.6 ms of execution latency. The CMD_i takes seven T_h operations and GRS_j incurs eight T_h operations with computational cost 0.54 ms and 0.104 ms respectively. In [22], the GRS_j does not participate in the mutual authentication process. Therefore, in this phase the MU_i and CMD_i require $2T_{ex} + 5T_m$ and $2T_{ex} + 7T_m$, i.e., 7.946 ms and 7.97 ms of computational cost, respectively. In [20], the MU_i and CMD_i entities bear 98.8 ms and 65.8 ms computational cost with given primitives $1T_{fe} + 5T_{ecm} + 5T_h$ and $3T_h + 4T_{ecm}$ respectively. On the server's end, it bears 31.43 ms of computational latency with $4T_h + 5T_{ecm}$ computations [18] bears 0.54 ms latency for both MU_i and CMD_i with 7 hash operations ($7T_h$) each, while on the GRS_j's end it incurs 19 hash operations with 1.482 ms computational latency. The proposed scheme employs $10T_h$, $7T_h$, $7T_h$ operations with 0.78 ms, 0.54 ms, and 0.54 ms of computational costs for MU_i, CMD_i and GRS_j, respectively. Tab. 2 describes the computational costs of [17,18,20,22] that are compared with the proposed schemes. For being lightweight symmetric crypto-operation, the hash function $h(.)$ with T_h is suitable for crowd sensing drone-based ecosystem to save the energy of mobile devices and ultimately improve their uptime.

In order to compare communication costs, we assume that |G| characterize 1024-bit element size, while $|Z_n|$ represents the 160-bit of each element in Z_n. Similarly, the |ID| depict the 32-bit size of timestamp as well as MUi's identity. We make the functionality comparison of our scheme against Wazid, Singh, Challa and Turkanovic et al. schemes in Tab. 4. The incurred communication cost of protocols [17,18,20,22] is compared against the proposed scheme as shown in Tab. 3. The Wazid et al. [17] bears the communication cost of 1696-bits which is calculated as $10|Z_n| + 3|ID|$ having 10 Z_n operations and 3 ID operations. Similarly, the [18,20,22] bear 4256-bits, 2528-bits, 2720-bits against $4|G| + 4|ID|$, $10|Z_n| + 3|ID|$ and $10|Z_n| + 3|ID|$ crypto-operations, respectively. In comparison with other schemes, the proposed scheme has remarkably less communication cost of 1472-bits against $9|Z_n| + |ID|$ operations.

Table 2: Computational cost

	User's end	Mobile drone	Server's end	Total
[17]	$1T_{fe} + 16T_h \approx$ *17.651* ms	$7T_h \approx$ *0.54* ms	$8T_h \approx$ *0.104* ms	$31T_h + 1T_{fe} \approx$ *18.295* ms
[22]	$2T_{ex} + 5T_m \approx$ *7.946* ms	$2T_{ex} + 7T_m \approx$ *7.97* ms	-	$12T_m + 4T_{ex} \approx$ *15.916* ms
[20]	$1T_{fe} + 5T_{ecm}+5T_h \approx$ *98.8* ms	$3T_h + 4T_{ecm} \approx$ *65.8* ms	$4T_h+5T_{ecm} \approx$ *31.43* ms	$12T_h + 14T_{ecm} + 1T_{fe} \approx$ *196.03* ms
[18]	$7T_h \approx$ *0.54* ms	$7T_h \approx$ *0.54* ms	$5T_h \approx$ *0.065* ms	$19T_h \approx$ *1.482* ms
Ours	$10T_h \approx$ *0.78* ms	$7T_h \approx$ *0.54* ms	$7T_h \approx$ *0.091* ms	$24T_h \approx$ *1.872* ms

Table 3: Communication cost

	Communication cost	Length (bits)
[17]	$10\|Z_n\| + 3\|ID\|$	1696
[22]	$4\|G\| + 4\|ID\|$	4256
[20]	$10\|Z_n\| + 3\|ID\|$	2528
[18]	$10\|Z_n\| + 3\|ID\|$	2720
Ours	$9\|Z_n\| + \|ID\|$	1472

Table 4: Functionality comparison

	[17]	[22]	[20]	[18]	[Ours]
Supports mutual authentication	×	×	×	✓	✓
Supports anonymity	✓	×	✓	×	✓
Unlinkability	✓	×	✓	×	✓
Supports session key agreement	×	×	✓	×	✓
Resists forgery attack	✓	×	✓	×	✓
Resists server impersonation attack	✓	✓	✓	×	✓
Immune to CMD_i physical capture threat	✓	×	✓	✓	✓
Immune to stolen device threat	✓	✓	×	✓	✓
Resists replay attack	✓	✓	✓	✓	✓
Resists man in the middle threat	✓	✓	✓	✓	✓
Resists offline password guessing attack	✓	×	✓	×	✓
Resists denial of service threat	✓	✓	✓	✓	✓
Supports formal analysis using ROM	✓	✓	✓	✓	✓

We now discuss the simulation details of the proposed model based on NS2 and the simulation details of the comparison schemes in [17,18,20,22]. We performed the simulation by using Ubuntu 14.04 long-term support (LTS platform) on the NS2 2.35 simulator [27]. We discussed the simulation parameters in Tab. 5. The total time taken by simulation is set as 2400 s (40 min). The entities CMDi, MUi, and Sj symbolize for ith drone, ith mobile user device, and jth IoT sensor in the compared schemes. We consider the various mobility parameters as 20, 30 and 40 mps for CMDi, MUi and Sj. We also assume a fixed server gateway across all of these schemes. The communication messages as exchanged among these participants are shown in Tab. 3. In the simulated experiment, three network performance-based benchmarks are evaluated, i.e., packet loss rate (number of packets), EED (sec) and throughput (bps). We now discuss the impact on these factors in the experiment in the following.

Table 5: Simulation parameters

Parameter	Semantics
Operating system	Ubuntu 14.04
Simulation tool	NS2 2.35
Domain(m)	450 m × 150 m × 20 m
Number of servers	1
Number of mobile users (MU_i)	3
Number of CMD_i/sensors	50
Mobility for MU_i/CMD_i	3 mps–20 mps/25 mps–35 mps
Communication range (CMD_i)	250 m
Total time for simulation	2400 sec

6.1 Throughput

We calculate the throughput based on the number of bits transmitted per unit of time i.e., $(r_p \times |p_s|)/T_s$, where T_s represents the total amount of time in seconds, $|p_s|$ shows the size of the packet, and r_p represents the received The total number of packets. The total simulation time is 2400 s. Fig. 3 shows that the throughput of contribution models [17,18,20,22] are 297.21, 225.34, 216.53, 284.76 and 267.12 bps, respectively. Obviously, the throughput of our model is higher than other protocols. This ensures that the proposed solution generates less communication cost for the small-sized communication messages exchanged during the protocol.

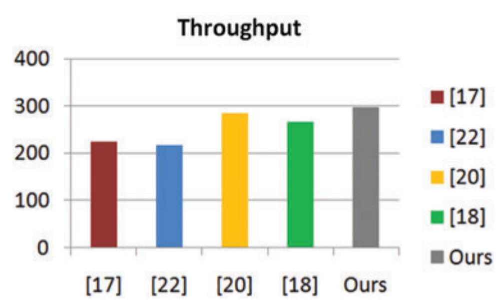

Figure 3: Throughput

6.2 End-to-End Delay (EED)

The EED shows the average time of packets to get to the sink or destination. This factor may be represented in numeric terms as $\sum_{j=1}^{n_{pkt}} (T_r - T_s)/n_{pkt}$, where T_r and T_s show the receiving and forwarding time of the exchanged packet, and n_{pkt} shows the number of packets to the destination. According to the Fig. 4, the EED values for [17,18,20,22] and proposed scheme are 0.041, 0.105, 0.29152, 0.04621, 0.033 sec, respectively. It is obvious that the EED factor of the contributed model is considerably less than the compared schemes and this attributes to the small size of the authentication messages.

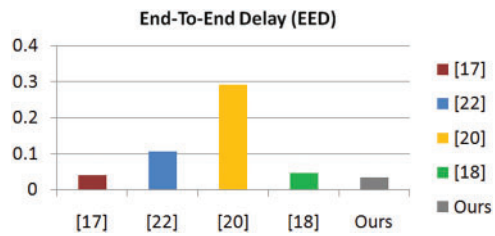

Figure 4: End-to-end delay

6.3 Packet Loss Rate (PLR)

The PLR factor describes the number of lost data packets per unit time and can be expressed as (n_{ip}/T_d), where T_d represents the total time in seconds, and n_{ip} represents the number of lost data packets. This factor must be as small as possible to make network-based communication more reliable. Fig. 5 shows the packet loss rate of different scenarios considering the comparison scheme and the contribution model. Obviously, the contribution model has a lower PLR compared with other schemes.

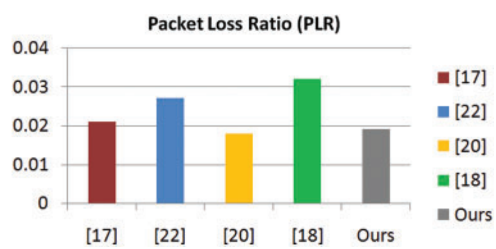

Figure 5: Packet loss ratio

7 Conclusions

The security and privacy requirements for reliable distribution of aerial monitoring and surveillance-based services have received increasing attention due to the vulnerability of the drone terrain. If the underlying authentication key agreement between the participating entities is not secure, the attacker may launch various attacks to disrupt the communication. In order to solve the security and privacy issues in such networks, we demonstrated a new identity verification protocol based on crowd monitoring drones, which enables participants to establish an agreed session key between them, and secure communication afterwards. Formal analysis under the Random Oracle Model (ROM) proved the proposed scheme. In addition, we used NS2 simulation to compare the proposed scheme with the existing scheme. Our analysis proves that the proposed scheme outperforms other schemes in terms of throughput, end-to-end delay and packet loss rate. Performance evaluation and benchmark factors show that the proposed scheme is secure compared with other contemporary studies in the same field. In the future, we can explore the prospect of using distributed systems based on blockchain to protect air surveillance.

Acknowledgement: The authors express their gratitude to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the Project Number (227).

Funding Statement: This work was supported by the Deputyship for Research & Innovation, Ministry of Education (in Saudi Arabia) through the Project Number (227).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] W. Xiao, M. Li, B. Alzahrani, R. Alotaibi, A. Barnawi *et al.*, "A blockchain-based secure crowd monitoring system using UAV swarm," *IEEE Network*, vol. 35, no. 1, pp. 108–115, 2021.

[2] A. Trotta, U. Muncuk, M. F. Di and K. R. Chowdhury, "Persistent crowd tracking using unmanned aerial vehicle swarms: A novel framework for energy and mobility management," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 96–103, 2020.

[3] T. Alladi, V. Chamola and N. Kumar, "PARTH: A two-stage lightweight mutual authentication protocol for UAV surveillance networks," *Computer Communications*, vol. 160, pp. 81–90, 2020.

[4] L. Nkenyereye, S. R. Islam, M. Bilal, M. Abdullah-Al-Wadud, A. Alamri *et al.*, "Secure crowd-sensing protocol for fog-based vehicular cloud," *Future Generation Computer Systems*, vol. 120, pp. 61–75, 2021.

[5] G. Cardone, A. Cirri, A. Corradi, L. Foschini, R. Ianniello *et al.*, "Crowdsensing in urban areas for city-scale mass gathering management: Geofencing and activity recognition," *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4185–4195, 2014.

[6] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman and D. Alghazzawi, "UAV assistance paradigm: State-of-the-art in applications and challenges," *Journal of Networks and Computer Applications*, vol. 166, pp. 102706, 2020.

[7] Y. Jiang, Y. Miao, B. Alzahrani, A. Barnawi, R. Alotaibi *et al.*, "Ultra large-scale crowd monitoring system architecture and design issues," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10356–10366, 2021.

[8] O. P. Popoola and K. Wang, "Video-based abnormal human behavior recognition—a review," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 42, no. 6, pp. 865–878, 2012.

[9] M. Abdalla, P. Fouque and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. IWPKC*, Les Diablerets, Switzerland, vol. 3386, pp. 65–84, 2005.

[10] C. Lin, D. He, N. Kumar, K. K. R. Choo, A. Vinel *et al.*, "Security and privacy for the internet of drones: Challenges and solutions," *IEEE Magazine*, vol. 56, no. 1, pp. 64–69, 2018.

[11] M. Silvagni, A. Tonoli, E. Zenerino and M. Chiaberge, "Multipurpose UAV for search and rescue operations in mountain avalanche events," *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, 2017.

[12] J. Blazakis, "Border security and unmanned aerial vehicles," *Connections*, vol. 5, no. 2, pp. 154–159, 2006.

[13] I. Maza, F. Caballero, J. Capitán, J. R. Martínez-de Dios and A. Ollero, "Experimental results in multi-uAV coordination for disaster management and civil security applications," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 563–585, 2011.

[14] N. H. Motlagh, M. Bagaa and T. Taleb, "UAV-Based IoT platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[15] H. Sedjelmaci, S. M. Senouci and N. Ansari, "A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 48, no. 9, pp. 1594–1606, 2017.

[16] B. Semal, K. Markantonakis and R. N. Akram, "A certificateless group authenticated key agreement protocol for secure communication in untrusted UAV networks," in *Proc. DASC*, London, UK, pp. 1–8, 2018.

[17] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos and J. J. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3572–3584, 2018.

[18] M. Turkanovic, 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 Networks*, vol. 20, pp. 96–112, 2014.

[19] M. S. Farash, M. Turkanovic, 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 Networks*, vol. 36, pp. 152–176, 2016.

[20] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy *et al.*, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

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

[22] J. Singh, A. Gimekar and S. Venkatesan, "An efficient lightweight authentication scheme for human-centered industrial internet of things," *International Journal of Communication Systems*, pp. e4189, 2019.

[23] Y. Zhang, D. He, S. Zeadally, D. Wang and K. K. R. Choo, "Efficient and provably secure distributed signing protocol for mobile devices in wireless networks," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5271–5280, 2018.

[24] M. Tao, X. Li, H. Yuan and W. Wei, "UAV-Aided trustworthy data collection in federated-WSN-enabled IoT applications," *Information Sciences*, vol. 532, pp. 155–169, 2020.

[25] A. S. Abdalla, K. Powell, V. Marojevic and G. Geraci, "UAV-Assisted attack prevention, detection, and recovery of 5G networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 40–47, 2020.

[26] K. Gai, Y. Wu, L. Zhu, K. K. R. Choo and B. Xiao, "Blockchain-enabled trustworthy group communications in UAV networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4118–4130, 2020.

[27] J. W. Jung, S. W. Chang and S. S. Lee, "Appropriate module configuration for vehicular networking using NS2 simulator," in *Proc. ICTC*, Busan, Korea, pp. 611–612, 2014.

[28] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. EUROCRYPT '21'*, Innsbruck, Austria, Springer, pp. 453–74, 2001.

[29] I. U. Khan, I. M. Qureshi, M. A. Aziz, T. A. Cheema and S. B. H. Shah, "Smart IoT control-based nature inspired energy efficient routing protocol for flying ad hoc network (FANET)," *IEEE Access*, vol. 8, pp. 56371–56378, 2020.

[30] I. U. Khan, R. Alturki, H. J. Alyamani, M. A. Ikram and M. A. Aziz, "RSSI-Controlled long-range communication in secured IoT-enabled unmanned aerial vehicles," *Mobile Information Systems*, vol. 2021, 2020.

[31] I. U. Khan, S. Z. Zukhraf, A. Abdollahi, S. A. Imran, I. M. Qureshi *et al.* "Reinforce based optimization in wireless communication technologies and routing techniques using internet of flying vehicles," in *Proc. ICFNDS*, New York, NY, USA, pp. 1–6, 2020.