A secure and lightwieght drones-access protocol for smart city surveillance

Muhammad Wahid Akram, Ali Kashif Bashir, Salman Shamshad, Muhammad Asad Saleem, Khalid Mahmood, Shehzad Ashraf Chaudhry, Ahmad Ali AlZubi, Yousaf Bin Zikria

***Abstract*—The rising popularity of ICT and the Internet has enabled Unmanned Aerial Vehicle (UAV) to offer advantageous assistance to Vehicular Ad-hoc Network (VANET), realizing a relay node's role among the disconnected segments in the road. In this scenario, the communication is done between Vehicles to UAVs (V2U), subsequently transforming into a UAV-assisted VANET. UAV-assisted VANET allows users to access real-time data, especially the monitoring data in smart cities using current mobile networks. Nevertheless, due to the open nature of communication infrastructure, the high mobility of vehicles along with the security and privacy constraints are the significant concerns of UAV-assisted VANET. In these scenarios, Deep Learning Algorithms (DLA) could play an effective role in the security, privacy, and routing issues of UAV-assisted VANET. Keeping this in mind, we have devised a DLA-based key-exchange protocol for UAV-assisted VANET. The proposed protocol extends the scalability and uses secure bitwise XOR operations, one-way hash functions, including user's biometric verification when users and drones are mutually authenticated. The proposed protocol can resist many well-known security attacks and provides formal and informal security under the Random Oracle Model (ROM).The security comparison shows that the proposed protocol outperforms the security performance in terms of running time cost and communication cost and has effective security features compared to other related protocols.**

***Index Terms*—Authentication Protocol, Internet of Drones, VANET, Intelligent Transportation Systems, Mutual Authentication, Information Security**

I. Introduction

THE emergence of Intelligent Transportation System (ITS) and Internet of Things (IoT) technology has empowered the Vehicular Ad-hoc Networks (VANETs) to offer a more pleasant and safer driving experience. However, due to the highly dynamic nature of network topologies, VANETs frequently realize the challenge of intermittent connection interruption. In order to encounter this limitation, Unmanned Aerial Vehicles (UAVs) can be utilized as the most advisable contender to enhance the connectedness of VANETs. For instance, UAV can provide assistance to the ground vehicle during the data transmission through Storage-Carry-Forward (SCF) technique which can adequately reduce the end-to-end delay and enhance the message delivery ratio.

UAVs are also termed drones who have proclaimed their encouraging capabilities in various real-time applications like disaster management, surveillance system, goods distribution, traffic surveillance systems in a smart city, data collection, object detection, tracking, and rescue system, health-care system, environmental monitoring, localization, and mapping. Drones are the recent advancement as the flying Internet of Things (IoT) objects that pretend to be sensing devices [1], [2]. Currently, drones are employing IoT technology to play their role in IoD. Each drone has a peculiar Fly Zone (FZ), from where it gathers the needed information from its surroundings. Later on, this information could be transferred to a specific user on his request. The control center remotely administers the communication of drones with the users. The embedded sensors in each drone send the captured information from its surrounding terrain to the control server via some source of wireless communication technology [3].

Drones are being endorsed worldwide, mainly because they acquired the expertise to visit isolated areas with little manpower, time, energy, and effort. No doubt, these features empower the end users, but it is expected critical as direct access privileges can cause worse security threats [4], [5]. IoD takes advantage of IoT technologies to attain its acute operations [6], [7]. IoD network's major requirements associated with the aspects of cost-effective operations are localization, trajectory planning, authorization, drone monitoring, security, and privacy. Regardless of the technological advancement and plenty of existing solutions, privacy and security are still primary issues in the IoD environment. IoD architectures are resource constrained because a drone has restricted power, computation, and storage sources [8]. Therefore, to reinforce an IoD paradigm's lifespan, it is necessary to construct an AKA protocol that consumes minor resources.

Muhammad Wahid Akram, Salman Shamshad, Muhammad Asad Saleem, Khalid Mahmood are with Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, Pakistan, e-mails(mwahid905@gmail.com, salmanshamshad01@gmail.com, masad@cuisahiwal.edu.pk, khalid.mahmood@cuisahiwal.edu.pk).

Ali Kashif Bashir (corresponding author) is with Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom, e-mail: (dr.alikashif.b@ieee.org).

Shehzad Ashraf Chaudhry is with Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey e-mail: (ashraf.shehzad.ch@gmail.com).

Ahmad Ali AlZubi is with Computer Science Department, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Saudi Arabia, (aalzubi@ksu.edu.sa).

Yousaf Bin Zikria (corresponding author) is with Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea, e-mail: (yousafbinzikria@ynu.ac.kr).

A. Motivation and Contributions

In IoD, the important data is interchanged between the user and drone over the public channel. Moreover, the drones are employed in an open are called as fly zone to gather the sensitive data and transfer it to the specific user. Therefore, drones and user may encounter various security threats by any third-party (say an adversary). However, to ensure shared data security, an efficient authentication infrastructure is required to effectively validate the participants' authenticity. No doubt, a number of researchers have provided their contribution in developing the protocols to secure the IoD environment, but it should be noted that these protocols are found vulnerable against numerous security threats. Additionally, some of them are not valuable for the resource-constrained environment like IoD because of their high computation and communication overhead. So, to provide resistance and efficiency against these challenges, we have presented a robust biometric-based and efficient IoD authentication protocol. Our proposed scheme can confront distinct security attacks and offers imperative security properties. The key contributions of the presented protocol are given below:

- Our article developed the biometric-based secure and lightweight authenticated key agreement protocol for the IoD environment. Additionally, we use lightweight cryptographic operations, including the primitives (e.g., XoR and Hash operations), making our protocol more effective.
- The proposed protocol also provides the location privacy of involved participants (e.g., $\mathcal{U}_m$ and $\mathcal{D}_n$) along with the authentication phase.
- The devised protocol utilized a generic one-way hash function instead of the map-to-point hash function. The generic hash function enables the electric service to validate diverse messages at the same time.
- The proposed protocol also allows the involved participants (e.g., $\mathcal{U}_m$ and $\mathcal{D}_n$) to affirm mutually to commute the common shared key between them for safe communication. Additionally, the procedure of shared key establishment include both long term (i.e. ID_m, ID_n and ephemeral (i.e. a_1, a_2, a_3). Moreover, these credentials are not interchanges in the insecure channel. Thereby, the proposed protocol ensures the security of the shared key.

B. Paper Roadmap

In the subsequent sections, we have described the related work in Section 2, and cryptographic preliminaries are discussed in Section 3. Our proposed protocol is presented in Section 4. The security analysis and performance evaluation of proposed protocol is described in Section 5 and 6. Lastly, we have given conclusive remarks in Section 7.

II. Related Work

In this section, several multi-factor authentications and key exchange protocols have been discussed. Lamport et al. [9] introduced a password based scheme in 1981. They designed their scheme using a one-way hash function, which makes their scheme more efficient and lightweight. Keeping the effectiveness of lightweightness, many researchers came up with several secure authentication schemes in numerous environments [10]–[13].

Turkanovic et al. [14], firstly introduced an unconventional AKA protocol for nodes and users without the intervention of gateway node. Turkanovic et al.'s protocol proves to be lightweight as they only use two cryptographic operations in the entire scheme. Where one operation is a one-way hash, and the other is a bitwise XOR operation. Farash et al. [15] indicate various vulnerabilities of Turkanovic et al.'s scheme. They identified that [14] does not offer user anonymity and also vulnerable to node impersonation and man-in-the-middle attack. After that, Farash et al. proposed an enhanced AKA protocol to mitigate the weaknesses present in [14]. Later on, Amin et al. [16] cryptanalyzed Farash et al.'s scheme and figured out that various attacks are possible on their scheme, including user impersonation, off-line password guessing attacks. Afterward, Amin et al. introduced a smart card based efficient and secure AKA scheme. After a while, Jiang et al. [17] found that, unfortunately, Amin et al.'s scheme cannot sustain security against off-line password guessing attacks and smart cart stolen attacks.

Inherently, key escrow and certificate management problems exist in conventional public key infrastructure (PKI) and identity base cryptography (IBC). Selvi et al. [18] and Li et al. [19] proposed key agreement and authentication schemes. However, both schemes cannot prove to be lightweight. After that, numerous Certificate-Less Public Key Cryptography (CL-PKC) key exchange protocols [20]–[23] are introduced. These protocols are proved lightweight authentication protocols because they are not using any pairing operation for computations. After that, a tag key-encapsulation mechanism using the certificate-less signcryption/authenticated encryption was presented by Seo et al. [24].

The former studies [9]–[25] provide evidence that the existing AKA solutions failed to maintain the trade off between security and overhead. For instance, when some protocols try to increase security, their overhead also rises and vice versa. Hence, the protocols examined are not compatible and do not offer privacy in the IoD environment. This paper has proposed a secure and lightweight AKA protocol for the IoD environment to deal with highlighted flaws discussed in the literature review. At the same time, the security analysis is presented to evaluate the performance of the proposed protocol.

III. Preliminaries

In this section, we have discussed some cryptographic primitives and their descriptions. Notations used throughout the article are also presented in Table I.

Table I: Notation Table

Notation	*Description*	*Notation*	*Description*
$\mathcal{CC}$	Control Center	s, MSK	Two secrets keys of $\mathcal{CC}$
SID_c	The pseudonym of $\mathcal{CC}$	z	Public parameter selected by $\mathcal{CC}$
$\mathcal{U}_m, \mathcal{D}_n$	The m^{th} user and n^{th} drone, respectively	ID_m, ID_n	The identities of $\mathcal{U}_m$ & $\mathcal{D}_n$
SID_m	The pseudonym of $\mathcal{U}_m$	SID_n	The pseudonym of $\mathcal{D}_n$
k_m	The master private key of $\mathcal{U}_m$	k_n	The master private key of $\mathcal{D}_n$
Bio_m	Biometric of $\mathcal{U}_m$	$Gen(.)$	Fuzzy biometric generator
$Rep(.)$	Fuzzy biometric reproduction	T_1	The current timestamp
ΔT	Threshold value	$h(l)$	One-way hash $h : \{0,1\}^* \rightarrow Z_n^*$
$\oplus$	Bitwise XOR operation	$\|$	Concatenation operation
$\mathcal{A}$	Adversary	a_1, a_2, a_3	Random numbers of $\mathcal{U}_m$ and $\mathcal{D}_n$ respectively

A. Threat Model

We have defined the capabilities of adversary $\mathcal{A}$ below as utilized in [26]–[30]:

1- $\mathcal{A}$ can control all messages communicated over the public channel. $\mathcal{A}$ has acquired the rights to modify, intercept, replay, and deleting the message(s).

2- $\mathcal{A}$ can presume the identity and password of the user and drone in polynomial time via a dictionary attack.

3- $\mathcal{A}$ can use malicious devices to intercept the password or excerpt the relevant parameters from any mobile device. Although, $\mathcal{A}$ cannot execute both actions simultaneously.

4- $\mathcal{A}$ can assault the forward secrecy, if the secret key of the control server or the password of user is revealed to $\mathcal{A}$.

B. The Network Model

To demonstrate the prototype of an IoD environment, we have presented the generic IoD environment below in Figure 1. Where an external entity needs to capture the data gathered by the drones in their particular FZ. This type of access is only possible if both the user and the drone authenticate with the control center's association. The control center is supposed to be a reliable and trusted entity in the IoD environment. However, the communication between involved entities is wireless, due to which numerous privacy and security concerns emanate in the IoD environment. Therefore, there is still a need to design a secure and lightweight authentication protocol. Additionally, recent research shows that the association of cryptographic and blockchain techniques provides superior protection against the security endangers in various environments, including IoD, where the communication among the involved entities is done over an insecure channel. This article has designed an improved protocol, which is more secure and reliable in terms of computation and communication overheads.

IV. Proposed Scheme

A. Initialization Phase

In this subsection, the initialization phase is briefly described below:

1) $\mathcal{CC}$: $\mathcal{CC}$ is deemed as trusted party and liable for registration of all users and drones. $\mathcal{CC}$ is able to generate secret keys for $\mathcal{U}_m$ and $\mathcal{D}_n$ against their identities.

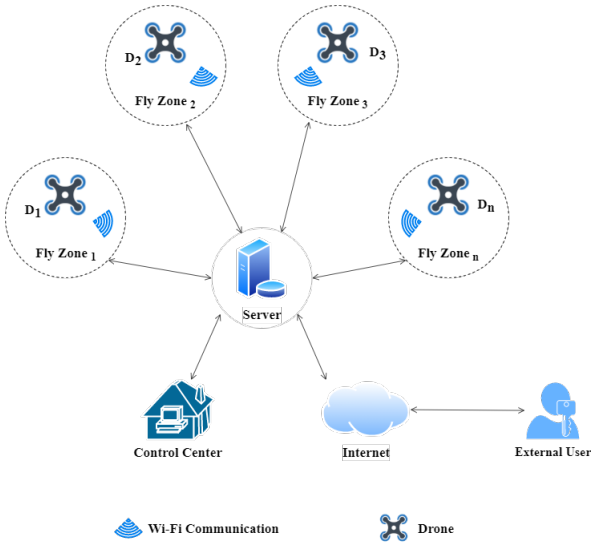

Figure 1: General IoD Architecture

2) $\mathcal{U}_m$: In an IoD environment, $\mathcal{U}_m$ receive his secret key during registration phase from $\mathcal{CC}$. So, $\mathcal{CC}$ should authenticate $\mathcal{U}_m$ before giving access to $\mathcal{U}_m$ to communicate with $\mathcal{D}_n$ in AKA phase.

3) $\mathcal{D}_n$: The drones also get registered by the $\mathcal{CC}$ before communication with the users. $\mathcal{D}_n$ and $\mathcal{U}_m$ can established a session key with each other after successful execution of authentication phase.

B. Setup Phase

In this phase, $\mathcal{CC}$ produces public parameters of the system along with its secret key. The setup phase is briefly described below:

1) Initially, $\mathcal{CC}$ randomly selects s and MSK as its secret keys and masked, respectively. At the same time, $\mathcal{CC}$ also selects z as public parameter of the system.
2) Next, $\mathcal{CC}$ selects a one-way hash function $h : \{0,1\}^* \rightarrow Z^*$ and his own identity ID_c. Thereafter, $\mathcal{CC}$ computes: $SID_c = h(ID_c \| s)$.
3) Finally, $\mathcal{CC}$ secretly keeps (MSK, s) and publishes (h, z, SID_c).

C. User Registration Phase

In this phase, $\mathcal{U}_m$ gets himself registered at the control center $\mathcal{CC}$ and receives his private key via a private channel. The user registration phase is described subsequently:

1) $\mathcal{U}_m$ selects his ID_m and PW_m randomly and imprints his/her biometric Bio_m into the mobile device. Thereafter, $\mathcal{U}_m$ calculates $Gen(Bio_m) = (\alpha_m, \beta_m)$ and sends ID_m as a registration request to $\mathcal{CC}$.
2) On receiving the registration request, $\mathcal{CC}$ computes $SID_m = h(ID_m \| s)$, $k_m = h(SID_m \| MSK)$ and generate a random number $a_m \leftarrow Z_n^*$. Next, $\mathcal{CC}$ computes $MID_m = Enc_{MSK}(SID_m \| \alpha_m)$. Then $\mathcal{CC}$ sends (k_m, SID_m, SID_n) to $\mathcal{U}_m$ via a private channel.
3) Whenever, $\mathcal{U}_m$ receives (k_m, SID_m, SID_n). Afer that, $\mathcal{U}_m$ will compute $\gamma_m = h(ID_m \| PW_m \| \alpha) \oplus k_m$, $SID_m^u = h(ID_m \| PW_m) \oplus SID_m$. In the end, $\mathcal{U}_m$ stores $(\gamma_m, SID_m^u, SID_n)$ securely.

D. Drone Registration Phase

In this phase, $\mathcal{D}_n$ registered itself with the control center $\mathcal{CC}$ and receives its private key via secure channel. The drone registration is described subsequently:

1) At first, $\mathcal{D}_n$ chooses its identity ID_n and forwards it to $\mathcal{CC}$ as registration request.
2) When $\mathcal{CC}$ receives registration request, $\mathcal{CC}$ computes $SID_n = h(ID_n \| s), k_n = h(SID_n \| MSK)$ and saves (ID_n, k_n, SID_n) in database list DB_b securely, encrypted with MSK. In the end, $\mathcal{CC}$ sends (k_n, SID_n) to $\mathcal{D}_n$ via private channel.
3) After receiving response from $\mathcal{CC}$, $\mathcal{D}_n$ gets (k_n, SID_n) and saves them securely.

E. Authentication and Key Agreement Phase

Whenever, $\mathcal{U}_m$ needs to communicate with $\mathcal{D}_n$, he has to perform authentication and key agreement phase. The authentication and key agreement phase is presented in Figure 2 and described subsequently and the flow diagram is presented in Figure 3:

1) Firstly, $\mathcal{U}_m$ enters his identity ID_m and password PW_m and also imprints his/her biometrics at the sensor of device. Then, $\mathcal{U}_m$ calculates $\alpha_m = Rep(Bio_m, \beta_m)$, $SID_m = SID_m^u \oplus h(ID_m \| PW_m), k_m = \gamma_m \oplus h(ID_m \| PW_m \| \alpha_m)$. After that, $\mathcal{U}_m$ randomly selects a number $a_1 \epsilon Z_n^*$ and computes: $A_1 = h(SID_m \| SID_c \| k_m) \oplus a_1$, $A_2 = h(SID_m \| SID_c \| k_m \| a_1) \oplus SID_n$ and $A_3 = h(SID_m \| SID_n \| SID_c \| k_m \| a_1)$. Finally, it sends authentication request message to $\mathcal{CC}$ through public channel MID_m, A_1, A_2, A_3.

2) After receiving the authentication message (MID_m, A_1, A_2, A_3) from $\mathcal{U}_m$. Firstly, $\mathcal{CC}$ computes $(SID_m \| \alpha_m) = Dec_{MSK}(MID_m)$, $k_m = h(SID_m \| MSK)$ and $a_1' = A_1 \oplus h(SID_m' \| SID_c \| k_m')$. Moreover, $\mathcal{CC}$ computes $SID_n' = A_2 \oplus h(SID_m' \| SID_c \| k_m' \| \alpha_1')$ and verifies k_n against SID_n' from the list DB_b. Finally, $\mathcal{CC}$ computes: $A_3' = h(SID_m' \| SID_n' \| SID_c \| k_m' \| a_1')$. Additionally, $\mathcal{CC}$ confirms the validation $A_3' \stackrel{?}{=} A_3$. If it is not true, $\mathcal{CC}$ will not entertain the authentication request. Otherwise, $\mathcal{CC}$ will authenticate the $\mathcal{U}_m$ and generate two random numbers a_2 and a_m^{new}. Then $\mathcal{CC}$ computes: $MID_m^{new} = Enc_{MSK}(SID_m \| a_m^{new}), A_4 = h(SID_n' \| k_n) \oplus (a_1' \| a_2 \| MID_m^{new})$, $A_5 = h(SID_n' \| SID_c \| k_n \| a_1') \oplus SID_m'$ and $A_6 = h(SID_m' \| SID_n' \| SID_c \| k_n \| a_1' \| a_2)$. Finally, $\mathcal{CC}$ forwards message (A_4, A_5, A_6) to $\mathcal{D}_n$ via a public channel.

3) Whenever, $\mathcal{D}_n$ receives request message (A_4, A_5, A_6) from $\mathcal{CC}$, $\mathcal{D}_n$ will compute: $(a_1'' \| a_2' \| MID_m^{new}) = A_4 \oplus h(SID_n \| k_n)$, $SID_m'' = A_5 \oplus h(SID_n \| SID_c \| k_n \| a_1'')$ and $A_6' = h(SID_m'' \| SID_n \| SID_c \| k_n \| a_1'' \| a_2')$. After that, $\mathcal{D}_n$ checks $A_6' \stackrel{?}{=} A_6$. $\mathcal{D}_n$ will terminate the session, if validation goes wrong and if it holds $\mathcal{D}_n$ can authenticate the control center $\mathcal{CC}$ and randomly selects a number $a_3 \epsilon Z_n^*$. Afterwards, $\mathcal{D}_n$ will perform: $A_7 = h(SID_n \| SID_m'' \| a_1'') \oplus (a_2 \| a_3' \| MID_m^{new})$, $A_8 = h(a_1'' \| a_2 \| a_3')$, $SK_{nm} = h(SID_m'' \| SID_n \| SID_c \| A_8)$ and $A_9 = h(SID_m'' \| SID_n \| SID_c \| a_2 \| a_3' \| A_8)$. In the end, $\mathcal{D}_n$ sends challenge message (A_7, A_9) to $\mathcal{U}_m$ using a public channel.

4) On receiving message (A_7, A_9) from $\mathcal{D}_n$, $\mathcal{U}_m$ will compute: $(a_2' \| a_3'' \| MID_m^{new}) = A_7 \oplus h(SID_n \| SID_m \| a_1)$, $A_8' = h(a_1 \| a_2' \| a_3'')$, $A_9' = h(SID_m \| SID_n \| SID_c \| a_2' \| a_3'' \| A_8')$. After that, $\mathcal{U}_m$ validates $A_9' \stackrel{?}{=} A_9$. $\mathcal{U}_m$ will terminate the session, if validation does not holds. Otherwise, $\mathcal{U}_m$ can authenticate $\mathcal{D}_n$ and share the session key $SK_{mn} = h(SID_m'' \| SID_n \| SID_c \| A_8') = SK_{nm}$ with $\mathcal{D}_n$.

V. Security Analysis

In this section, the security analysis of the proposed protocol is presented in detail. Initially, we show that our proposed protocol is more secure under Random Oracle Model (ROM). Then, we describe how our proposed protocol satisfies the security requirements.

A. Formal Security Analysis

In this subsection, we present the security model of proposed protocol as discussed in Choi et al [31]. The attacker $\mathcal{A}$ simulates as a polynomial-time restricted Turing-Machine. We denote $\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}$ as $\{t_1^{th}, t_2^{th}, t_3^{th}\}$ instances of $\{\mathcal{U}_m, \mathcal{D}_n, \mathcal{CC}\}$ respectively. Whereas, these oracles are can easily simulate under the supervision of Challenger C. Moreover, these oracles grant $\mathcal{A}$ to execute multiple queries and provide the respective feedback. Several queries used in random oracle model are described below:

- $\mathbf{h}(x)$: is a function of x oracle and it preserves a hash-list L_h. Whenever, $\mathcal{A}$ runs their hash query of specific oracle with their respective message $\{x, C\}$, firstly $\mathcal{A}$

Figure 2: Authentication And Key Agreement Phase

checks for x. Whether x exists in L_h or not. If exists then, C provides the output to $\mathcal{A}$ i.e $h(x)$. Contrary, C selects $X \in Z_n^*$ randomly, and $\mathcal{A}$ gets x from C. In the end, C keeps their perspective (x, X) in L_h.

- **Extract**(ID_m) : By utilizing this query, $\mathcal{A}$ can fraudulent the legitimate drone and gets their secret key. Whenever, $\mathcal{A}$ runs his malicious extract query on drones identity ID_n, then C returns and $\mathcal{A}$ gains access to the secret key of particular $\mathcal{D}_n$.
- **Send**$(\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}, M)$: An active attack can be surfaced by $\mathcal{A}$ by using this query, whenever, $\mathcal{A}$ forwards the specific M (message) to any instance $\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}$. Whereas, $\mathcal{A}$ will receive the respective reply from $\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}$ with specific message M. $\mathcal{A}$ can inaugurate by executing the $Send(\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}, Start)$ query to the oracle against any specific instance $\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}$.
- **Reveal**(Π^t) : This query reproduces the erroneous adoption of session key. When $\mathcal{A}$ runs the *Reveal* query and any of the instance $\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}$ has strongly developed, then C will send back the session key of instance $\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}$. Contrary, C will return $\perp$.
- **Execute**$(\mathcal{U}_m, \mathcal{D}_n)$: By using the $Execute$ query $\mathcal{A}$ will have the ability to snoop specific information from the public medium. Whenever, $\mathcal{A}$ runs $Execute$ query in the public channel, it can get all information during the executions.
- **Test**$(\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\})$: The $Test$ query helps $\mathcal{A}$ to differentiate among random secret and real session keys. Additionally, $\mathcal{A}$ will have the ability to run $Test$ query only one time. At this point, C checks that if $b = 1$. Afterwards, C arbitrarily selects the value of b as $b \in 0, 1$. In that case, C will return the real session key to $\mathcal{A}$. Other than that, C will return the arbitrary secret key to $\mathcal{A}$, when $(b = 0)$. Another case is that, if there is no session key exists related to any instance $\{\Pi_{\mathcal{U}_m}^{t_1}, \Pi_{\mathcal{D}_n}^{t_2}, \Pi_{\mathcal{CC}}^{t_3}\}$, then C will returns $\perp$ to $\mathcal{A}$.

$\mathcal{A}$ can initiate other queries like $Extract$, $Send$, $Reveal$ and $Execute$, after execution on the $Test$. At this point, the restriction of $\mathcal{A}$ is that it cannot furnish *Reveal* and its arrangement related to the $Test$. Lastly, $\mathcal{A}$ produced b' in the replacement of b. Here Σ if $b' = b$, then we can conclude that $\mathcal{A}$ successfully damage the AKA of devised protocol. The influence of $\mathcal{A}$ is prescribed as: $adv_{\Sigma}^{AKA}(A) = |2Pr[b^{'} = \text{b}] - 1|$.

- **Define 1** $(AKA - Secure)$: $\mathcal{A}$ can strongly conquer the game with insignificant improvement to $adv_{\Sigma}^{AKA}(A)$ in polynomial time. Here, we can conclude that our proposed AKA-Secure protocol Σ is more secure and reliable. Though $\mathcal{A}$ can easily damage the authentication and key exchange phase of devised protocol Σ. Where, if $\mathcal{A}$ generates the valid login request message or response message. Let E_{U-CC} shows the action that if $\mathcal{A}$ can produce a login message by impersonating the user $\mathcal{U}_m$. Additionally, the generated login message accepted got accepted by the $\mathcal{CC}$ successfully. Suppose that E_{U-D} express the

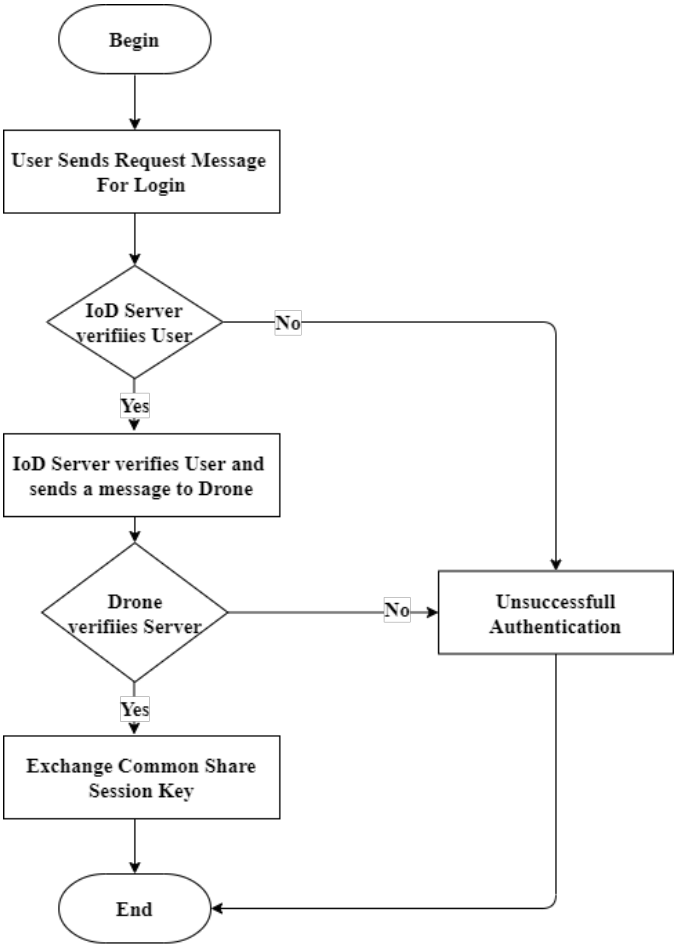

Figure 3: Flow Diagram of Login and Authentication Phase

action that $\mathcal{A}$ easily impersonate $\mathcal{D}_n$ and produces the response message which is accepted by legal $\mathcal{U}_m$. $adv^{\Sigma}_{MA}(A) = P_r[E_{U-CC} + P_r[E_{U-V}]]$ shows that how adversary $\mathcal{A}$ break the mutual authentication and impersonate $\mathcal{U}_m$ and $\mathcal{D}_n$.

- **Define 2**$(AKA-Secure)$: In this scenario, $\mathcal{A}$ can easily break the mutual authentication with insignificant improvement $adv^{MA}_{\Sigma}(A)$ in polynomial time. Here, we can conclude that our devised AKA-Secure protocol Σ is more reliable and secure.

B. Provable Security Block

This subsection proved that $\mathcal{A}$ could alter the valid login message. Additionally, $\mathcal{A}$ can respond against the login request in a non-negligible polynomial time and it states that the devised protocol is MA and AKA secure.

- **Lemma-1:** Lets assume, $\mathcal{A}$ could portend a valid login request or respond to a login request in a non-negligible polynomial time. After that, C could easily guess an arbitrary number in a non-negligible polynomial time.
- **Proof.** C selects an arbitrary number msk. Moreover, C forwards the parameter h, n to $\mathcal{A}$. Afterwards, C produces L_h. Initially, L_h does not contains any record. First, records are inserted in oracles and select two simultaneously identities of drones ID_i and ID_j. Let say all various oracles suppose to be inquired after the hash oracle models of drones. The different answers to queries are as below:
- $\mathbf{h}(x_i) : C$ initially audits whether x_i remains in the archive L_h. If it stays, then C hands-over X_i to $\mathcal{A}$. Otherwise, C select a digit X_i, adds (x_i, X_i) in the archive L_h and returns X_i to $\mathcal{A}$.
- **Extract**(ID_i) : If $i \neq I, J$, then C will explore a new tuple $(ID_i||msk, \alpha_i)$ in the archive L_h and returns α_i to $\mathcal{A}$. Beside that, C renounces the query and nullify the game.
- **Send**$(\Pi^t_{\wedge}, M)$: $\mathcal{A}$ can launch the $Send$ query to reproduce the working intrusion in four different ways:

  1- **Send**$(\Pi^t_{\mathcal{U}_m}, Start)$ C initially finds out whether, $i \neq I$ and subsequently explores L_s (the hash-list) for $\mathcal{U}'_m s$ undisclosed/private key α_i if they are equivalent. By taking the help of secret key α_i, C selects an arbitrary number $r1 \in Z^*_n$, the modern time-stamp ST_1 and computes (A_1, A_2, A_3), and if the computed (A_1, A_2, A_3) are not commensurate, C chooses $R1, R2, R3 \in Z^*_n$ as the three random numbers and now C sets $M2 \leftarrow R1, M3 \leftarrow R2, M4 \leftarrow R3$. Compute $M_1 = h(SID_c||T_1) \oplus SID_m$ and return (A_1, A_2, A_3) to $\mathcal{A}$.

  2- **Send**$(\Pi^k_{\mathcal{D}_n}, (A_4, A_5, A_6))$: As C receives the request message, first C verifies that j and J are equal. If validation holds, C will proceed and select two random numbers $R_4, R_5 \epsilon Z^*_n$ and set $A_7 \leftarrow R_4, A_9 \leftarrow R_5$. Otherwise, C will look in hash list L_h for the secret key s of D_m and remaining computations will be done as in sequence.

  3- **Send**$(\Pi^t_{\mathcal{U}_m}, (A_7, A_9))$: First C verifies $j \neq J$. If the condition not holds, then C look out hash list L_s for D_m's secret key s. By using the secret key s, C will select a random number $a_2 \epsilon Z^*_n$ and computes (A_7, A_9). If condition holds, C will randomly choose three numbers $R_4, R_5, R_6 \epsilon Z^*_n$, and set $r_{2\leftarrow} R_4, A_7 \leftarrow R_5, A_9 \leftarrow R_6$ and send A_7, A_9 to $\mathcal{U}_m$.

  4- **Reveal**(Π^t_{Λ}): If instance Π^t_{Λ} of entities $(\mathcal{U}_m, \mathcal{CC}, \mathcal{D}_n)$ has been acquired, C will returns session key SK_{Λ}, otherwise C will returns $\perp$.

It is one of the scenario that adversary $\mathcal{A}$ can succeed in computation of a validate-able login request or a response message, respectively. Then the result parameters (MID_m, A_1, A_2, A_3) to $Send(\Pi^t_{\mathcal{U}_m}, Start)$ query with $i = I$ and $(A_7, A_9$ to $Send(\Pi^t_{\mathcal{D}_n}, (A_4, A_5, A_6))$ query with $j = J$ will pass the validation check done by both the $\mathcal{CC}$ & $\mathcal{U}_m$. Some events are specified to figure out how C could be beneficial. These events are illustrated below:

- E_1: The simulation is not terminated.
- E_2: $\mathcal{A}$ can submit a legal login request message MID_m, A_1, A_2, A_3 using $Send(\Pi^t_{\mathcal{U}_m}, Start)$ query or a valid response message (A_7, A_9) using $Send(\Pi^t_{\mathcal{D}_n}, (A_4, A_5, A_6))$ query, in the meantime, $Extract(ID_M)$ and $Extract(ID_N)$ have never been questioned.

- E_3: $\mathcal{U}_m = U_M$ or $\mathcal{D}_n = D_N$.
- E_4: C can select the accurate pair from the hash list L_h.

Suppose q_s, q_{L_s} and q_{L_h} express the number $Send$-query, L_s-query and L_h-query performed by $\mathcal{A}$.

$$Pr[E_1] \geq \frac{1}{q_s} \tag{1}$$

It is clear that

$$\begin{aligned} Pr[E_2|E_1] &\geq \epsilon Pr[E_3|(E_2 \Lambda E_1)] \geq \frac{1}{q_{L_s}} \\ Pr[E_4|(E_3 \Lambda E_2 \Lambda E_1)] &\geq \frac{1}{q_{L_s}} \frac{1}{q_{L_s}-1} + \frac{a}{q_{L_h}} \frac{b}{q_{L_h}-a} \end{aligned} \tag{2}$$

In above computations, $\mathcal{A}$ is an accurate pair number in query-$Send(\Pi^t_{\mathcal{U}_m}, Start)$ and $'b'$ is the accurate number of $Send(\Pi^t_{\mathcal{U}_m}, (A_7, A_9))$-query. Hence, the C infers successfully, the random-number with length $160 - bits$, where the win-probability is non-negligible and can be presented as:

$$\begin{aligned} Pr[E_1 \Lambda E_2 \Lambda E_3 \Lambda E_4] &= Pr[E_4|E_3 \Lambda E_2 \Lambda E_1] Pr[E_3|E_2 \Lambda E_1] \\ &\quad Pr[E_2|E_1] Pr[E_1] \end{aligned} \tag{3}$$

$$= \frac{1}{q_s} \frac{1}{q_{L_s}} \left(\frac{1}{q_{L_s}} \frac{1}{q_{L_s}-1} + \frac{a}{q_{L_h}} \frac{b}{q_{L_h}-a}\right)\epsilon \tag{4}$$

Nevertheless, its difficult to guess an arbitrary number. Thus, $\mathcal{A}$ can never be able to produce a legitimate login request or legitimate response message. Therefore, users and drones can mutually authenticate in the devised protocol securely.

Theorem 1: Our proposed protocol is mutually authenticated secure against guessing of the 160-bits random number (RN).

Proof. According to Lemma-1, $\mathcal{A}$ cannot generate a validate-able login or response message due to hardness of guessing 160-bits RN. Hence, our proposed protocol is mutually authenticated securely.

Theorem 2: Our proposed protocol is AKA-Secure against guessing of a 160-bits RN.

Proof. Let $\mathcal{A}$ be the guesser and is bounded by polynomial time and with non-negligible probability (N-NP) ϵ, it produced a $b' = b$, then there is a C with abilities to reveal $160 - bits$ RN with N-NP. To calculate the edge of C for comfort, there are some events which are described below:

- E_{SK}: $\mathcal{A}$ has the ability to fetch the valid session key after $Test$-query.
- E_U: $\mathcal{A}$ could execute a $Test$-query to instance Π_{D_I} with ease.
- E_D: $\mathcal{A}$ could also execute $Test$-query to instance Π_{D_J} with ease.
- E_{U-CC-D}: $\mathcal{A}$ could violate the mutual authentication among $\mathcal{U}_m$, $\mathcal{D}_n$ and $\mathcal{CC}$.

There is a probability of $\mathcal{A}$ against trying to guess the valid b without needed any other information. Therefore, we can get $Pr[]E_{SK} \geq \epsilon/2$ by applying the following computations:

$$\begin{aligned} Pr[E_{SK}] &= Pr[E_{SK} \Lambda E_D] + Pr[E_{SK} \Lambda E_D \Lambda E_{U-CC-D}] \\ &\quad + Pr[E_{SK} \Lambda E_D \Lambda \neg E_{U-CC-D}] \\ &\leq PR[E_{SK} \Lambda E_U] + Pr[E_{U-CC-D}] \\ &\quad + Pr[E_{SK} \Lambda E_D \neg E_{U-CC-D}] \end{aligned} \tag{5}$$

Then we have,

$$\begin{aligned} &Pr[ESK\ \Lambda\ E_U] + Pr[E_{SK} \Lambda E_D \Lambda \neg E_{U-CC-D}] \\ &\geq Pr[E_{SK}] - Pr[E_{U-CC-D}] \\ &\geq \epsilon/2 - Pr[E_{U-CC-D}] \end{aligned} \tag{6}$$

Owing to $Pr[E_D \Lambda \neg E_{U-CC-D}] = Pr[E_D]$, therefore,

$$Pr[E_{SK} \Lambda E_D] \geq \frac{\epsilon}{4} - \frac{Pr[E_{U-CC-D}]}{2} \tag{7}$$

The events $E_{SK} \Lambda E_{\mathcal{D}_n}$ show that $\mathcal{A}$ impersonate user $\mathcal{U}_m$ and captures the valid session key. According to Lemma, $Pr[E_{U-CC-D}]$ is negligible, so $\frac{\epsilon}{4} - \frac{Pr[E_{U-CC-D}]}{2}$ is non-negligible. In other words, the probability for $\mathcal{A}$ to capture the valid session key value is non-negligible, that is a contradiction because of the difficulty of guessing a RN.

C. Informal Security Analysis

In this subsection, the informal security of our proposed protocol is analyzed. The informal security analysis illustrates that the proposed protocol is resilient against various security attacks, which is described below:

1) Anonymity and Untraceability: In our proposed protocol, the identity ID_m of $\mathcal{U}_m$ is not shared in plaintext on any public channel. While, during the authentication phase, $\mathcal{U}_m$ sends login request message $\{MID_m, A_1, A_2, A_3\}$ to $\mathcal{CC}$ through a public channel. Whereas, MID_m, A_1, A_2 and A_3 needs SID_m to perform the computation and SID_m is calculated with $\mathcal{U}_m$'s identity ID_m and s secret key of $\mathcal{CC}$. As $\mathcal{A}$ cannot be able to know the secret key of $\mathcal{CC}$. Therefore, $\mathcal{A}$ also cannot access the ID_m of $\mathcal{U}_m$. Furthermore, each messages being transmitted over public channel involve arbitrary nonce (i.e., a_1, a_2, and a_3). So, our proposed protocol ensures the privacy of entities and also offers the user anonymity and untraceability.

2) User Impersonation Attack : If $\mathcal{A}$ forwards a login request to $\mathcal{CC}$ on behalf of legitimate $\mathcal{U}_m$ and $\mathcal{A}$ got authenticated by $\mathcal{CC}$. After authentication, $\mathcal{CC}$ successfully forwards a message to $\mathcal{D}_n$ then this scenario is called as user impersonation attack. Despite that, in the proposed protocol, if $\mathcal{A}$ attempts to send request message $\{MID_m, A_1, A_2, A_3\}$ on behalf of legal $\mathcal{U}_m$ then he/she has to calculate all the values for $\{MID_m, A_1, A_2, A_3\}$ correctly. Since, to calculate A_3, $\mathcal{A}$ needs SID_m, k_m and identity ID_m of legal $\mathcal{U}_m$, which is only in the access of $\mathcal{U}_m$. Therefore, our proposed protocol can resist user impersonation attack.

3) Control Center Impersonation Attack : In control center impersonation attack, $\mathcal{A}$ attempts to manipulate legitimate user $\mathcal{U}_m$ and entertains every login request of $\mathcal{U}_m$ on behalf of legal control center. In the proposed protocol,

whenever $\mathcal{A}$ wants to impersonate legal $\mathcal{CC}$, then $\mathcal{A}$ needs to relay login message with additional values. Whereas, the calculations of message $\{A_4, A_5, A_6\}$ requires $\mathcal{CC}$'s mask key MSK. Whereas, the MSK is only known to $\mathcal{CC}$. Therefore, $\mathcal{A}$ cannot have any access to MSK. Hence, our proposed protocol can resist control center impersonation attack.

4) Drone Impersonation Attack : In drone impersonation attack, suppose $\mathcal{A}$ sends a challenge message as a legal $\mathcal{D}_n$ to $\mathcal{U}_m$. If $\mathcal{D}_n$ has been authenticated by $\mathcal{U}_m$ and $\mathcal{U}_m$ is able to share session key with drone against his challenge message. Then it is referred as drone impersonation attack. In our proposed protocol, if $\mathcal{A}$ tries to forward challenge message $\{A_7, A_9\}$ on behalf of legal $\mathcal{D}_n$, then he has to calculate the value of A_7 correctly. So, for the computation of A_7, $\mathcal{A}$ should have valid values of SID_n, SID''_m, a_1, a_2 and a_3. As SID_n needs the identity ID_m of $\mathcal{U}_m$ and secret key s of $\mathcal{CC}$, which is not available to $\mathcal{A}$. Moreover, a_1, a_2, a_3 are specific session and their values updates in every session. Therefore, our proposed protocol can provide resistance against drone impersonation attack.

5) Session Key Agreement : In our protocol, $\mathcal{U}_m$ and $\mathcal{D}_n$ shares the session key $SK_{mn} = h(SID''_m \| SID_n \| SID_c \| A'_8) = SK_{nm}$ to keep the communication safe between $\mathcal{U}_m$ and $\mathcal{D}_n$. It is to be noted that the security of session key relies on the privacy of the involved random numbers a_1, a_2, a_3. That means their values are updating and shared among the entities in each session. Therefore, our proposed protocol can provide the session key agreement.

VI. Performance Analysis

In this section, we have compared our proposed protocol's performance with recently presented related protocols including [29], [32]–[34]. The comparison is presented in terms of time complexity cost, communication cost and security features.

A. Implementation Scenario

The proposed and related protocols comprise of three entities, including user $\mathcal{U}_m$, control center $\mathcal{CC}$ and drone $\mathcal{D}_n$, respectively. As the registration phase performed only once in proposed and contemporary related protocols; therefore, we have discarded both user and drone registration phases in performance evaluation. Moreover, we have neglected cryptographic operations, like string concatenation and XOR, since they have negligible computational costs. Therefore, to determine the experimental results, the cryptographic operations that are used at $\mathcal{U}_m$'s end are implemented on a mobile device. At the same time, Arduino is used to performing the cryptographic operations at $\mathcal{D}_n$'s end. Similarly, a desktop system has been used to implement the cryptographic operations at $\mathcal{CC}$'s end. Furthermore, the notations of cryptographic operations and their description and execution time for the specific environment is illustrated below in Table II. Table III presents the summary of system specifications for devices on which the cryptographic operations are implemented.

Table II: Cryptographic Operations and their Execution Time

Op	Description	Execution Time		
		MD	DS	Arduino
T_h	Hash function	1.003 ms	0.0022 ms	2.103 ms
T_{pm}	Point multiplication	0.234 ms	0.0026 ms	0.524 ms
T_s	Symmetric enc-decryption	0.430 ms	0.0032 ms	0.941 ms

Note: Op=Operation, MD=Mobile Device, DS=Desktop System.

Table III: Summary of System Specifications Diverse Devices

Items → Specifications ↓	Arduino	Mobile	System
Model	Mircocontroller:ATmega328	Vivo S1	Intel Corei5
RAM	SRAM: 2 KB (ATmega328)	6 GB	16 GB
Generation	-	-	7th
OS	Windows	Android	Windows
Processor	16 MHz	1.7 GHz	2.9 GHz
Library/IDE	Arduino IDE	PyCharm	PyCharm

B. Time Complexity Cost Comparison

In this subsection, we compute the time complexity cost of proposed and related protocols using the time complexity of cryptographic operations defined in Table II. Each protocol has a registration phase, which is a one time process. Therefore we only considered the hash, point multiplication, symmetric encryption/decryption, and cryptographic operations of authentication and key agreement phase to compute the time complexity cost of proposed and related protocols [29], [32]–[34]. In our proposed protocol, $\mathcal{U}_m$ logged in to the system using his ID_m, PW_m and Bio_m. After that, the system will perform the reproduction function to produce biometric of the specific user and executes two hash operations to validate the legitimacy of inputs. To validate the legitimacy of the provided credentials, the system needs to perform nine hash functions to commence the login request message. Therefore, the accumulative time complexity cost for $\mathcal{U}_m$'s end would be $9 \times T_h \approx 9.027ms$. On receiving the login request, $\mathcal{CC}$ forwards a message to $\mathcal{D}_n$, where it renders 7 hash and 2 symmetric encryption/decryption operations. Thus, the accumulative time complexity cost at $\mathcal{CC}$'s side is $7 \times T_h + 2 \times T_s \approx 0.0218ms$. After that, when $\mathcal{D}_n$ receives the message from $\mathcal{CC}$, it executes 9 hash operations. So the accumulative time complexity cost at $\mathcal{D}_n$'s side will be $7 \times T_h \approx 14.721ms$. Accordingly, the entire time complexity cost of our proposed protocol's authentication and key agreement phase is $9.027 + 0.0218 + 14.721 \approx 23.7698ms$. The time complexity cost of related protocols [29], [32]–[34] is computed in the same way, which is shown in Table IV. Figure 4 presents the graphical view of time complexity

Table IV: Analysis of Time Complexity and Communication Costs

Pt	$\mathcal{U}_m$'s Side	$\mathcal{CC}$'s Side	$\mathcal{D}_n$'s Side	Comp.	Comm.
Our	$9T_h \approx 9.027ms$	$7T_h + 2T_s \approx 0.0218ms$	$7T_h \approx 14.721ms$	23.7698ms	2304 bits
[32]	$10T_h \approx 10.03ms$	$7T_h \approx 0.0154ms$	$7T_h \approx 14.721ms$	24.7664ms	2464 bits
[33]	$11T_h + 1T_s \approx 11.463ms$	$8T_h + 2T_s \approx 0.024ms$	$9T_h \approx 18.927ms$	30.414ms	2944 bits
[29]	$16T_h \approx 16.048ms$	$7T_h \approx 0.0154ms$	$7T_h \approx 14.721ms$	30.7844ms	3040 bits
[34]	$14T_h \approx 14.042ms$	$10T_h \approx 0.022ms$	$8T_h \approx 16.824ms$	30.888ms	2784 bits

Note: Pt=Protocols, Comp=Aggregated Time Complexity Cost, Comm=Aggregated Communication Cost.

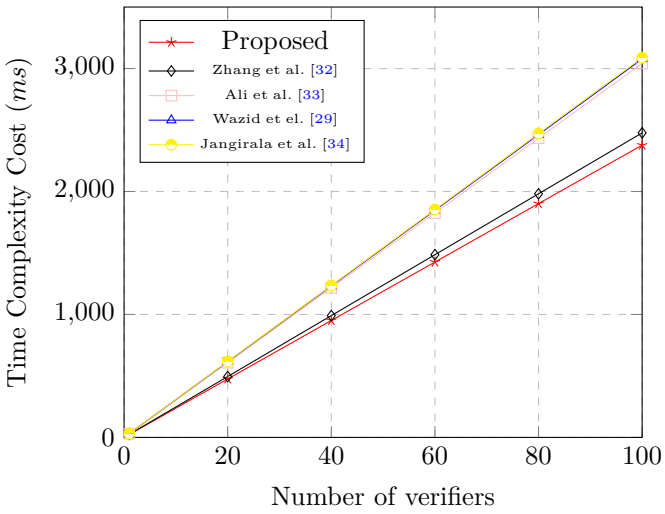

Figure 4: Total Time Complexity Complexity

cost comparison between proposed and related protocols. The number of verifiers is listed on the x-axis, and time complexity is shown on the y-axis. It is clear from Figure 4 that the time complexity cost of our proposed protocol is less than the related protocols.

C. Communication Cost Comparison

This section presents the precise comparison of the communication cost of devised and related protocols [29], [32]–[34]. Moreover, it is significant to mention that, during the calculation of communication cost of proposed and related protocols, we have only examined the messages that are communicated while the authentication and key agreement phase between the $\mathcal{U}_m, \mathcal{CC}$ and $\mathcal{D}_n$. Following notations and assumptions are solicited for the sizes of different communicating parameters: 'e/d' denotes Symmetric encryption/ decryption with 128 bits length, Hash Function is represented by 'h' with 256 bits length; whereas, the identities, timestamps, and ECC points are considered as 160 bits long and are represented by 'id', 't' and 'pm', respectively.

In authentication and key agreement phase of our proposed protocol, the entities $\mathcal{U}_m$, $\mathcal{CC}$ and $\mathcal{D}_n$ exchange three messages $\{MID_m, A_1, A_2, A_3\}, \{A_4, A_5, A_6\}$ and $\{A_7, A_9\}$ with each other. The communication cost of these messages is: $\{128{+}256{+}256{+}256\}, \{256{+}256{+}256\}$ and $\{256{+}256\}$. Therefore, the accumulative communication cost of our proposed protocol is $896 + 768 + 512 = 2176$ bits. The communication cost of related protocols is computed in the same way where Zhang et al. [32] transmits 2464 bits, Ali et al. [33] transmits 2912 bits, Wazid et el. [29] transmits 3040 bits and Jangirala et al. [34] transmits 2784 bits and the detailed communication cost comparison is shown in Table IV.Moreover, in Figure 5, we have presented the comparative analysis of the communicational cost needs to implement while authentication and key agreement phase of protocols against various times. The x-axis in Figure 5, represents our proposed and related protocols, whereas the y-axis shows the total data to be transmitted in bits. We can quickly figure out by viewing the overall analysis that our proposed protocol has better performance in terms of communication cost other than the rest of the protocols. Moreover, it is also visible in Table V that our proposed protocol resists all the major security attacks, whereas the related protocols [29], [32]–[34] are failed to resist various security attacks.

Table V: Comparison of Security Features

Protocols → **Sec. Feat. ↓**	**Ours**	**[32]**	**[33]**	**[29]**	**[34]**
UIA	●	○	●	●	●
CCIA	●	●	●	○	●
DIA	●	○	●	●	●
MA	●	●	●	●	●
UA	●	○	●	●	●
UT	●	●	○	●	○
SKA	●	●	●	○	●

Note: Sec.Feat.=Security Features, UIA=User Impersonation Attack, CCIA=Control Center Impersonation Attack, DIA=Drone Impersonation Attack, MA=Mutual Authentication, UA=User anonymity, UT=Un-traceability, SKA=Session Key Agreement, ● Provides; ○ Does not Provide.

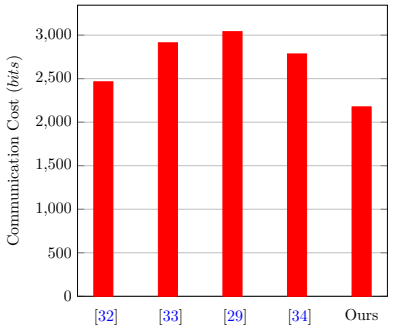

Figure 5: Communication cost comparison of different protocols

VII. Conclusion

This article proposes a multi-factor lightweight authentication protocol for a UAV-assisted VANET environment. The security analysis shows that the proposed protocol ensures the integrity and completeness properties of the UAV-assisted VANET environment. Furthermore, the proposed protocol's performance analysis compared with related authentication protocols shows that our protocol requires the least computational overhead and communication overhead.

Acknowledgement

Ali Kashif Bahsir and Yousaf Bin Zikria are the corresponding authors. This work was supported by the Researchers Supporting Project (No. RSP-2021/395), King Saud University, Riyadh, Saudi Arabia.

References

[1] A. Al-Qerem, M. Alauthman, A. Almomani, and B. Gupta, "Iot transaction processing through cooperative concurrency control on fog–cloud computing environment," *Soft Computing*, vol. 24, no. 8, pp. 5695–5711, 2020.

[2] B. B. Gupta and M. Quamara, "An overview of internet of things (iot): Architectural aspects, challenges, and protocols," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, p. e4946, 2020.

[3] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 976–985, 2020.

[4] B. Sejdiu, F. Ismaili, and L. Ahmedi, "Integration of semantics into sensor data for the iot: A systematic literature review," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 16, no. 4, pp. 1–25, 2020.

[5] R. Arul, G. Raja, A. K. Bashir, J. Chaudry, and A. Ali, "A console grid leveraged authentication and key agreement mechanism for lte/sae," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2677–2689, 2018.

[6] S. P. Ahuja and N. Wheeler, "Architecture of fog-enabled and cloud-enhanced internet of things applications," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 10, no. 1, pp. 1–10, 2020.

[7] N. M. F. Qureshi, I. F. Siddiqui, M. A. Unar, M. A. Uqaili, C. S. Nam, D. R. Shin, J. Kim, A. K. Bashir, and A. Abbas, "An aggregate mapreduce data block placement strategy for wireless iot edge nodes in smart grid," *Wireless personal communications*, vol. 106, no. 4, pp. 2225–2236, 2019.

[8] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, "Compound rank-k projections for bilinear analysis," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 7, pp. 1502–1513, 2015.

[9] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[10] J. H. Cheon, K. Han, S.-M. Hong, H. J. Kim, J. Kim, S. Kim, H. Seo, H. Shim, and Y. Song, "Toward a secure drone system: Flying with real-time homomorphic authenticated encryption," *IEEE access*, vol. 6, pp. 24325–24339, 2018.

[11] S. A. Chaudhry, I. L. Kim, S. Rho, M. S. Farash, and T. Shon, "An improved anonymous authentication scheme for distributed mobile cloud computing services," *Cluster Computing*, vol. 22, no. 1, pp. 1595–1609, 2019.

[12] Z. Ali, A. Ghani, I. Khan, S. A. Chaudhry, S. H. Islam, and D. Giri, "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *Journal of Information Security and Applications*, vol. 52, p. 102502, 2020.

[13] C. Wang, Y. Zhu, W. Shi, V. Chang, P. Vijayakumar, B. Liu, Y. Mao, J. Wang, and Y. Fan, "A dependable time series analytic framework for cyber-physical systems of iot-based smart grid," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 1, pp. 1–18, 2018.

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

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

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

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

[18] S. S. D. Selvi, S. S. Vivek, and C. P. Rangan, "Certificateless kem and hybrid signcryption schemes revisited," in *International Conference on Information Security Practice and Experience*, pp. 294–307, Springer, 2010.

[19] F. Li, M. Shirase, and T. Takagi, "Certificateless hybrid signcryption," in *International Conference on Information Security Practice and Experience*, pp. 112–123, Springer, 2009.

[20] D. He, J. Chen, and J. Hu, "A pairing-free certificateless authenticated key agreement protocol," *International Journal of Communication Systems*, vol. 25, no. 2, pp. 221–230, 2012.

[21] M. Geng and F. Zhang, "Provably secure certificateless two-party authenticated key agreement protocol without pairing," in *2009 International Conference on Computational Intelligence and Security*, vol. 2, pp. 208–212, IEEE, 2009.

[22] G. Yang and C.-H. Tan, "Strongly secure certificateless key exchange without pairing," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 71–79, 2011.

[23] H. Sun, Q. Wen, H. Zhang, and Z. Jin, "A novel pairing-free certificateless authenticated key agreement protocol with provable security," *Frontiers of Computer Science*, vol. 7, no. 4, pp. 544–557, 2013.

[24] S.-H. Seo, J. Won, and E. Bertino, "pclsc-tkem: a pairing-free certificateless signcryption-tag key encapsulation mechanism for a privacy-preserving iot.," *Trans. Data Priv.*, vol. 9, no. 2, pp. 101–130, 2016.

[25] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E.-J. Yoon, and K.-Y. Yoo, "Secure signature-based authenticated key establishment scheme for future iot applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[26] S. A. Chaudhry, "Correcting "palk: Password-based anonymous lightweight key agreement framework for smart grid"," *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106529, 2021.

[27] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and 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*, vol. 56, no. 4, pp. 4425–4435, 2020.

[28] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "Pflua-diot: A pairing free lightweight and unlinkable user access control scheme for distributed iot environments," *IEEE Systems Journal*, pp. 1–8, 2020.

[29] 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.

[30] S. A. Chaudhry, K. Yahya, F. Al-Turjman, and M. H. Yang, "A secure and reliable device access control scheme for iot based sensor cloud systems," *IEEE Access*, vol. 8, pp. 139244–139254, 2020.

[31] K. Y. Choi, J. Y. Hwang, D. H. Lee, and I. S. Seo, "Id-based authenticated key agreement for low-power mobile devices," in *Australasian Conference on Information Security and Privacy*, pp. 494–505, Springer, 2005.

[32] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for internet of drones," *Computer Communications*, 2020.

[33] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–43724, 2020.

[34] J. Srinivas, A. K. Das, N. Kumar, and J. J. Rodrigues, "Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6903–6916, 2019.