Computers, Materials & Continua
DOI: 10.32604/cmc.2022.026179
Article

Tech Science Press

A Mutual Authentication and Cross Verification Protocol for Securing Internet-of-Drones (IoD)

Saeed Ullah Jan[1], Irshad Ahmed Abbasi[2,*] and Fahad Algarni[3]

[1]Department of Computer Science & IT, University of Malakand, Chakadara, 18800, Pakistan
[2]Faculty of Science & Arts Belqarn, Department of Computer Science, University of Bisha, Sabtul Alaya 61985, Saudi Arabia
[3]Faculty of Computing and Information Technology, University of Bisha, Bisha 67714, Saudi Arabia
*Corresponding Author: Irshad Ahmed Abbasi. Email: aabasy@ub.edu.sa
Received: 17 December 2021; Accepted: 22 February 2022

Abstract: With the rapid miniaturization in sensor technology, Internet-of-Drones (IoD) has delighted researchers towards information transmission security among drones with the control station server (CSS). In IoD, the drone is different in shapes, sizes, characteristics, and configurations. It can be classified on the purpose of its deployment, either in the civilian or military domain. Drone's manufacturing, equipment installation, power supply, multi-rotor system, and embedded sensors are not issues for researchers. The main thing is to utilize a drone for a complex and sensitive task using an infrastructure-less/self-organization/resource-less network type called Flying Ad Hoc Network (FANET). Monitoring data transmission traffic, emergency and rescue operations, border surveillance, search and physical phenomenon sensing, and so on can be achieved by developing a robust mutual authentication and cross-verification scheme for IoD deployment civilian drones. Although several protocols are available in the literature, they are either design issues or suffering from other vulnerabilities; still, no one claims with conviction about foolproof security mechanisms. Therefore, in this paper, the researchers highlighted the major deficits in prior protocols of the domain, i.e., these protocols are either vulnerable to forgery, side channel, stolen-verifier attacks, or raised the outdated data transmission flaw. In order to overcome these loopholes and provide a solution to the existing vulnerabilities, this paper proposed an improved and robust public key infrastructure (PKI) based authentication scheme for the IoD environment. The proposed protocol's security analysis section has been conducted formally using BAN (Burrows-Abadi-Needham) logic, ProVerif2.03 simulation, and informally using discussion/pragmatic illustration. While the performance analysis section of the paper has been assessed by considering storage, computation, and communication cost. Upon comparing the proposed protocol with prior works, it has been demonstrated that it is efficient and effective and recommended for practical implementation in the IoD environment.

Keywords: Cryptography; authentication; confidentiality; reachability; ZSP

1 Introduction

Earlier, drones were mainly used for military mission delivery. However, with the invention of small unmanned aerial vehicles (UAVs or drones) becoming opened new possibilities to be applied in incident monitoring, search and rescue operations, disaster relief, and packages' delivery. A popular Mobile Ad hoc Network (MANET) paradigm is Flying Ad hoc Networks (FANETs) are used for data transmission in the IoD environment. In contrast to other ad hoc network types, FANETs are distinguished by many unique features because it changes their topology dynamically [1]. Due to which it presents the research community with security challenges. The only solution to these security challenges is to design a robust authentication protocol for FANET to establish an efficient data transmission with the control station server. Besides security, energy consumption is also a big issue in UAVs (drones). As some tasks assigned to a drone need maximum flight time, while the electric power is not too to accomplish it, if the internal processing capabilities become modified without affecting its external functionalities, it can guarantee a complex operation with minimum power consumption. In this regard, the computation process to generate shared session key also needs to be adequate to transmit information among all the participants of IoD efficiently. This challenge is also possible by designing a protocol with fast and secure computation and communication features for data broadcasting [2].

In the Information society, cryptographic algorithms play a crucial role, and they secure us when we use debit cards or credit cards, call someone on a cell phone, get access to health care services, or buy something on the internet. These algorithms ensure that our transactions and bank accounts are secure, our telephone, voice-over-internet protocol (VoIP), or instant messaging cannot be listened to by anyone, and that confidential health information is protected from unauthorized access. Cryptographic protocols support digital signatures, user and data authentication, and more advanced functionalities such as electronic money or electronic voting, e-government, and e-commerce in the near term [3]. Moreover, a cryptographic hash function is a technique for verifying data validity, can run on data for checksum purposes, and cryptographically encompasses algorithms for cyclic redundancy checks. It translates data of arbitrary size into a fixed valued numerical string called a hash [4]. In the same way, the researchers in this paper have used cryptographic algorithms to design a security mechanism for working in the IoD environment.

Furthermore, the already available cryptographic algorithms can also be used to secure the transmission path of drones with the control station server or external user to perform a tactical task. However, due to the existence of a strong adversary, only cryptographic-based protocols cannot achieve the goal of sensitive transmission security in IoD. It must need to be appropriately formalized; so that one must determine what the opponent/adversary is permitted to do and when the attack is successful. Under any complexity assumption, a cryptosystem would be "secure" if it demonstrates that the security principle is fulfilling, and the attacker could not crack the protocol [5]. However, a cryptographic system's security is most often proximate: its security is based on an assumption of complexity which is commonly believed in confidentiality. In the cryptographic research community, these methodologies are now the standard [6]. We, too, will first identify all possible threats to the system, design a cryptographic-based security mechanism, then evaluate its security as stated above and pragmatically illustrate them in the informal security analysis section of the paper to make it trustworthy in drone information transmission security for IoD.

Although, the increasing use of drones is raising security issues. Without incorporating the issue of security in the IoD, we cannot mitigate all other associated issues and challenges like power and navigation, product and traffic, privacy and obstacle detection, etc. Therefore, this research focuses

on designing cryptographic hash functions, XOR operations, and public key infrastructure (PKI) based authentication protocol for IoD using FANET. Because the security of exchanged information among all the IoD's participants is a challenging issue, it needs a robust, lightweight authentication protocol. The authentication protocol presented in this paper can extract dynamic identities and random numbers to ensure the dynamism feature in the protocol. The cryptographic hash-based function assimilates different security features like untraceability and anonymity and caters to the flaw of outdated data broadcasting. All these cryptographic algorithms (PKI, hash, XOR, SHA-1, MD5, AES, etc.) collaboratively used for the protocol design can guarantee to mitigate forgery, side channel, privileged insider, and stolen verifier attacks often seen in prior authentication protocols. Furthermore, it can show resistance to known attacks such as denial of service (DoS), man-in-the-middle, replay, drone capture attacks, and spoofing with other drones.

2 System Model

According to this model, a valid user must first register with the control station registry, and then a drone must also register with CSS. It is worth noting that the control station server has been designated as a wholly trusted individual. Their confidence must be consistent as a lack of trust could jeopardize the system's reliability. The proposed scheme means that the user and the remote server will fully trust the registration center, while any other entity alone cannot be fully trusted. Gharibi et al. [1] defined the flying zone strategy for a large geographical region in detail. We also consider their zone strategy for achieving impartiality, modularity, and standardization so that a drone can securely communicate with the ground station and external users. According to Gharibi et al. [1], for each drone, the Zone Service Provider (ZSP) is responsible to facilitate a drone for navigation services and designate zone on the request of a drone. Also, ZSP has the authority to put orders for landing drone, hold the drone in the current flying zone, or switch drones from one flying zone to another. ZSP planned collision-free navigation services to a drone, route maintenance between two drones, along many performance characteristics.

Furthermore, to cover a larger area, such as an entire country, the ground stations must communicate logically with one another. This technique would track drones in a cluster at various flying zones, traffic, and drone switching from one flying zone to another and provide mandatory statistics. [1] also clarified handover strategies when a drone moves from one flying zone to another, as shown in Fig. 1.

In Fig. 1, the certification for all drones is considered from a specialized framework installed within CSS, providing networking, information management support, and real-time problem-solving capabilities. The CSS is in charge of controlling, monitoring, and supervising drone navigation services. Network services and a wireless communication interface are also needed on all drones and are closely supervised by the CSS. The flight zones are another challenge for the CSS, and the drone must be operationalized in pre-determined flight zones/clusters. An external user can access a designated drone from a specific zone is also monitored by the CSS. CSS controls its flight and verifies its existence when a drone enters the IoD environment. The confirmation authenticity of a legitimate drone or the identification of an unauthorized drone in the flying zone can also easily be detected by the CSS due to its services agent capabilities.

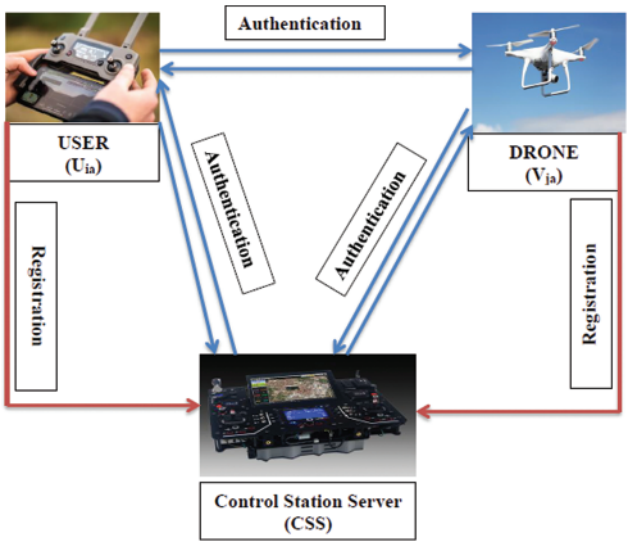


Figure 1: System model

2.1 Adversary Model

Any public networked-based correspondence may be altered, eavesdropped on, or snooped on by an intruder. An adversary can pose as an authentic node at a specific location and initiate contact with the legal peer. However, an adversary cannot reach the server to access the internal secret without authorization. However, he or she may compromise some tags to obtain the shared session key. Furthermore, an adversary has complete authority to begin negotiations with a legitimate client, to insert false tags with the standard message in a public network channel during contact, to remove the entire or part of the message, to copy the message and replay it at a later time [6].

2.2 Threat Model

Malicious users (attackers) have become more powerful nowadays. Therefore, all possible attacks are easy to launch against a legitimate user. Further, malicious users have many capabilities, such as editing, deleting, modifying, and blocking messages over IoD wireless networks. The possible threats against real users are: routing and session key threats, unauthorized access untraceability threats, perfect forward secrecy and data leakage threats, signal jamming and privacy threat, flight control and collation threats, signal spoofing and forgery threats, insider and deauthentication threats, stolen verifier and desynchronization threats, masquerade and impersonation threats, and clogging and ephemeral secret leakage threats.

2.3 Public Key Infrastructure

During peer authentication, efficient and secure management of keys (random numbers or public/private) pair is difficult to keep secret from a strong adversary. However, cryptographers [7,8] developed a scenario in which first the key pair is generated, secondly professionally deployed (public key is for encryption and private for decryption), and finally, the process of overturning it is performed. The overturning or invalidation step is initiated when the whole session is accomplished then the key pair becomes null or compromised. Therefore, to achieve secure communication over the public, insecure networks, protocols for the mutual authentication of two parties, and the generation of a

cryptographically generated shared key among the participants are fundamental. In contrast, the cryptographically-hash-based message authentication code depends on cross-verified session shared keys that need dynamic updates, as shown in Fig. 2.

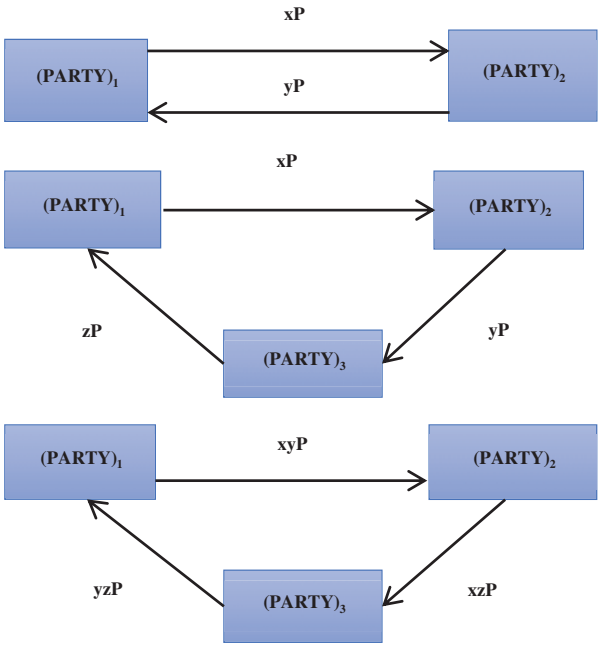


Figure 2: Single, double, and triple secret exchange scenarios

2.4 XOR Operation

For encrypting/decrypting the message using a single key or to secretly transmit a message without changing its size, a bit-wise XOR technique is used. It is a well-known technique in information security; a message having XOR cannot be cracked, which is also known as a one-time-pad [6].

2.5 Research Contribution

This article presents a PKI-based authentication protocol for IoD using FANET. The protocol offered in this research paper authenticates each participant (U_{ia}, V_{ja}, CSS) before procuring data from the drone using FANET. This lightweight and resource-efficient authentication protocol use SHA-1, PKI, XOR operations, and AES (Advanced Encryption Standard) for a secure key generation before broadcasting data with each other. The main contributions of the research work are as under:

1. The protocol concentrates on generating secure keys among users, drone, and CSS, consisting of user's password change, dynamic drone addition, and drone revocation/reissue phases. Besides, the hash function, which is used for cyclic checksum, has fewer storage overheads and high security. It also allows joint public network channels between User → Drone, Drone → CSS, CSS → Drones, and Drone → CSS without performance loss.
2. The protocol offered in the article is validated using BAN logic and ProVerif2.03. A comprehensive, pragmatic illustration for prominent attacks shows that the scheme is verifiably protected against each.

3. A comparative analysis section has been offered by considering three aspects, computation, storage, and communication costs which shows that the scheme is better than the state-of-the-art protocols.

3 Related Works

The drone's computing resources are severely limited, making it vulnerable to various security threats such as replay attacks, forgery attacks, and man-in-the-middle attacks. Seriously, a drone's surveillance work in smart cities could cause serious harm at any moment. He et al. [5] suggested an elliptic curve cryptography-based lightweight identity authentication scheme. However, they neglect to mention drone addition, revocation, and password update phases. According to [9], malicious drone in contact between ground stations and drone causes data transmission and instruction data leakage. The identity authentication, validity, reliability, and privacy of a drone with the ground station have been addressed by [10] but do not offer perfect forward secrecy.

According to [11], low latency authentication plays a fundamental role on the internet of drones in a disaster environment where latency is between life and death. Furthermore, unauthorized access, energy consumption, and latency concern the internet of the drone's network [3]. The author [12] proposed a lightweight protocol that achieved performance but compromised security and could not resist most attacks. Another mutual authentication protocol was proposed in [13], and the scheme is based on PUF but failed to provide comprehensive security. The protocol proposed in [14] provides poor performance, leading to a fatal accident in IoD networks.

Furthermore, the author [15] cryptanalysis the scheme [16] and finds out that they cannot resist stolen authentication and traceability issues. According to [17], the scheme used in [18] suffers from session key leakage, inability to provide user anonymity, and scalability issues. Moreover, the scheme [19] used the same certificate in the authentication phase; thus, it does not provide anonymity.

In recent years, the idea of the Internet of Things (IoT) has been implemented for the IoD environment. The data, communication, and network technology are incorporated for drones in IoD because it is used for consumer conveniences like entertainment, toys, agricultural-land monitoring, high-value industries, and wide applications in the defence field shooter product [10]. Suppose improved battery power, sensing systems, communication security, and other technologies and incorporating them into drone technologies can become a top-rated product in recent years, advancing various fields and activities. In that case, small UAVs have enormous potential and have significant application versatility. In addition to personal aerial photography, entertainment, and commercial markets, they can be used in a range of surveillance activities, such as disaster relief, in diverse environments involving animals and plants, coasts and borders, in the transport of goods, military, and police enforcement tasks, and also in agricultural and industrial applications. Also, the smart city features like traffic monitoring and management, merchandise distribution, health and emergency services, and air taxi services, for example, will increase the efficiency, effectiveness, timeliness, reliability, and performance of these services and may help reduce the cost of delivering these services [20].

Small UAVs, however, can also pose many security threats when misused. Different researchers made several attempts to secure its data transmission. For example, Hussain et al. [21] proposed an elliptic curve cryptographic-based authentication scheme to secure the communication of external users and drones in the pre-defined flying zone. After successful information broadcasting, the drones can then be deployed for different applications like broadening IoT base IoD, smart cities surveillances, sidewalk monitoring, and stealth purposes. Yahuza et al. [22] identified flaws in some prior IoD-based protocols like switching drones from one flying zone to another needed a robust mechanism

for self-organizing its previous secure transmission path. They mitigated the flying zone flaw and proposed a provably secure protocol, and named it SLPAKA. Gope et al. [23] claimed that robust information authentication is necessary to successfully deploy UAVs in crop spraying, public safety, and critical infrastructure surveillance. For this, they proposed a privacy-aware edge-assisted UAVs protocol by taking into account the procedure for resistance of UAVs from physical capturing. Tian et al. [24] also proposed a security framework for edge-assisted IoD using the securely computed authenticated key in online and offline mode for efficient open-access communication. However, due to batch verification of the signature, the computation tie complexity of their framework is not good. Ever [25] demonstrated that the key features of drone-like mobility, energy consumption, reliability, and efficiency for an open network are fundamental because all the IoD participants are not designed with an integrated security phenomenon. Therefore, they proposed a security framework for IoD using WSN. They used the elliptic curve discrete logarithmic function to secure participants' computing keys. However, it still suffered from a key-escrow problem; [26] provision of secure and efficient communication between drone & ground station for smart city surveillance, [27] secured the confidential data transmission between drones in IoD environment, [28] presented protocol for public cloud data security in IoT enabled equipment using MANET, and [29] presented three-factor key-agreement protocol for network-enabled devices using WSN. Similarly, [30] demonstrated an authentication scheme for an e-health-care system using WMSN, and [31] published a homomorphic encryption-based authentication scheme for IoD environment in which innovative knowledge for the different environments has been presented. Also, [32] proposed a privacy protection protocol for grid computing has been presented in which guarantees secure communication between service providers and smart objects, and [33] presented a three-factor (password, smart-card and biometric) based authentication scheme, which works for Unmanned Aerial Vehicular Networks.

Zhang et al. [14] designed a one-way hash function based on authentication and key agreement scheme for the Internet of Drone in which they claim that their scheme can guarantee for cross verification of each participant during communication. They presented the scheme in three phases: setup, registration, and mutual authentication. After the extensive analysis, it has been noted that their scheme is suffering from the following drawbacks:

1. An attacker can intercept the first message sent between the user and the control server, which leads to forgery attacks. The intruder may then modify the timestamp ST1, but the CS would not detect this. Furthermore, if an intruder physically captures the drone [14], store security credentials in its memory to participate in the authentication protocol; as a result, an attacker can gain access to the memory or use side-channel attacks to obtain the stored credentials. It means the scheme is suffering from side-channel attacks.
2. If an attacker forges the previous or current session key SKij, as the verification data is without encryption, the attacker can then transmit it towards the control center (SC) and force it to declare himself/herself as a legal user for the upcoming authentication session. For example, let suppose an attacker A can steal $\{M_5, M_6, M_7\}$ message from the open network channel and transmit it towards drone. V_{ja} computes $r_1'' = M_5 \oplus h(PID_j||\alpha_j)$, $PID_i'' = M_6 \oplus h(PID_j||PID_s||\alpha_j||r_1'')$, $M_7' = h(PID_i''||PID_j||PID_s||\alpha_j||r_1'')$ and forced drone to confirmed: $M_7'? = M_7$. Next attacker A generates random number r_A and computes: $M_8 = h(PID_j||PID_i'') \oplus r_A$, and $M_9 = h(r_1''||r_A)$. Further he/she might calculate session key $SK_{ij} = h(PID_i''||PID_j||PID_s||M_9)$ which, then can be used for potential reply, DoS, insider and stolen-verifier attacks. Therefore, Zhang et al. [14] scheme is not safe against these attacks.

3. Zhang et al. [32] used ST_1 in the first round trip, while they forgot to use it in the next two round trips, which in turn does not guarantee the transmission of new data among drone and control centers (SC). Therefore, the scheme suffers from outdated data transmission flaws.
4. Since the scheme only uses a timestamp for the first-round trip and does not use a timestamp for any subsequent round trips, it suffers from a global time-synchronization issue.

4 Proposed Solution

To solve the weaknesses mentioned in Zhang et al. [14] scheme above, we, as a result of this, have proposed the following improved scheme consisting of 1) setup phase, 2) registration phase, 3) mutual authentication and cross-verification phase, 4) user's biometric/password update Phase, 5) dynamic drone addition phase, and 6) drone revocation/reissue phases, each of these are described one by one as under, while the different notations used for designing the scheme are shown in Tab. 1.

Table 1: Notations and its description

Notation	Description	Notation	Description
U_{ia}	User	$\|\|$	Concatenation function
ID_s	CSS's identity	V_{ja}	Drone
ID_{ja}	Drone's identity	ID_{ia}	User's identity
l	Public key	s	Secret value
α_{ja}	Drone's master private key	n	Public value
ST_1	User's time stamp	α_{ia}	User's master private key
R_2	Drone's random number	R_1	User's random number
ΔT	Time threshold	ST_2	CSS timestamp

4.1 Setup Phase

Let the control station server (CSS) choose a random number l called a public key, s is a secret key and dispatches public parameters *pms*. Furthermore, CSS chooses collision-free one-way-hash function $h(.) \in Z_q*$, identity ID_s and calculates $PID_s = h(ID_s||s)$. The control station server (CSS) stores $\{l, s\}$ and issues $\{PID_s, h(.), pms\}$. The CSS uses l for encryption (Public key), s for decryption (private) in one session, and different keys for the next session.

4.2 Registration Phase

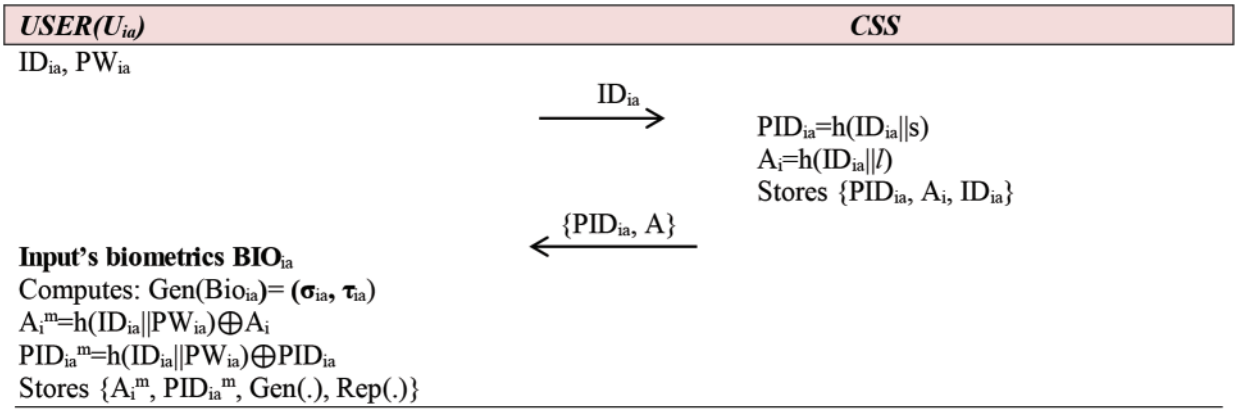
This phase of the proposed scheme is divided into two sub-phases:

4.2.1 User's Registration

This sub-phase of the scheme, completed in the following steps:

i. A legitimate user chooses his/her identity ID_{ia}, password PW_{ia} and sends a registration request towards the control station server (CSS)over a secure channel.
ii. Upon receiving the registration request, the control station server (CSS)computes $PID_{ia} = h(ID_{ia}||s)$, $A_i = h(ID_{ia}||l)$, store $\{PID_{ia}, A_i, ID_{ia}\}$ and transmit $\{PID_{ia}, A_i\}$ towards user over a secure channel.

iii. Upon receiving $\{PID_{ia}, A_i\}$, the user imprints his/her biometrics BIO_{ia}, and computes: $Gen(BIO_{ia}) = (\sigma_{ia}, \tau_{ia})$, $A_i^m = h(ID_{ia}||PW_{ia}) \oplus A_i$, and $PID_{ia}^m = h(ID_{ia}||PW_{ia}) \oplus PID_{ia}$ where $\sigma_{ia} \& \tau_{ia}$ are biometric keys associated with BIO_{ia}. Finally, U_{ia} stores $\{A_i^m, PID_{ia}^m, Gen(.), Rep(.)\}$ in its memory. Biometric Gen(.) and Rep(.) functions to concealed credentials from privileged user, as shown in Module I.

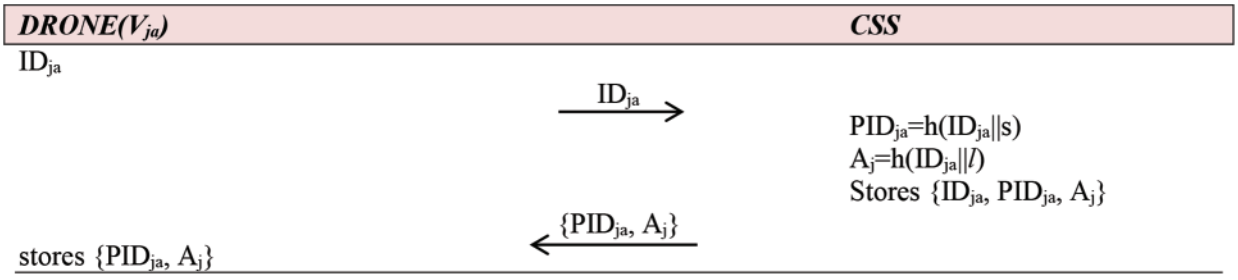


Module I: User's registration phase

4.2.2 *Drone's Registration Phase*

This sub-phase of registration phase is accomplished on the following steps:

i. A drone randomly selects ID_{ja} and transmits it to the control station server (CSS)over a secure channel.
ii. Upon receiving the request message, the control station server (CSS)computes $PID_{ja} = h(ID_{ja}||s)$, $A_j = h(ID_{ja}||l)$ stores $\{ID_{ja}, PID_{ja}, A_j\}$ and sends $\{PID_{ja}, A_j\}$ towards drone over a private channel.
iii. Upon receiving the message of CSS, the drone stores $\{PID_{ja}, A_j\}$ in its memory for future usage, as shown in Module II.



Module II: Drone's registration phase

4.3 Mutual Authentication & Cross Verification Phase

After registering user (U_i) and drone (V_j), they can communicate with each other subject to the condition that they can compute a secret shared session key SK. For such purpose, the following steps will perform:

i. The user first enters his/her ID_{ia}, PW_{ia}, imprints biometric BIO'_{ia} and computes $\sigma'_{ia} = Rep(BIO_{ia}, \tau_{ia})$, $PID_{ia} = PID_{ia}{}^{m} \oplus h(ID_{ia}||PW_{ia})$, $A_i = A_{ia}{}^{m} \oplus h(ID_{ia}||PW_{ia})$. Next generate a number $R_1 \in Z*_n$, present timestamp ST_1 and compute: $M_1 = h(PID_s||ST_1) \oplus PID_{ia}$, $M_2 = h(PID_{ia}||PID_s||A_i) \oplus R_1$, $M_3 = h(PID_{ia}||PID_s||A_i||R_1) \oplus PID_{ja}$, $M_4 = h(PID_{ia}||PID_{ja}||PID_s||A_i||R_1)$ and transmits $\{M_1, M_2, M_3, M_4, ST_1\}$ message towards control server over a public network channel. Here using public key for the encryption of $h(PID_{ja}||PID_{ia}''||ST_3) \oplus R_2$ message.

ii. Upon receiving $\{M_1, M_2, M_3, M_4, ST_1\}$ message, the CSS checks the received timestamp with the current system time ($T_c - ST_1 \leq \Delta T$), if not found within the prescribed time threshold, the CSS consider it for potential replay attack and stops computation. But when found valid, CSS decrypts M_1 using private key s and computes $PID_{ia}' = M_1 \oplus h(PID_s||ST_1)$ and retrieves $A_i'$and calculates $R_1' = M_2 \oplus h(PID_{ia}'||PID_s||A_i')$, $PID_{ja}' = M_3 \oplus h(PID_{ia}'||PID_s||A_i'||R_1')$ and $M_4' = h\ (PID_{ia}'||PID_{ja}'||PID_s||A_i'||R_1')$. The control station server (CSS) confirms $M_4'? = M_4$, if found no confirmation, the process is terminated, else, it generates another timestamp ST_2 and computes $M_5 = h(PID_{ja}'||A_j'||ST_2) \oplus R_1'$, $M_6 = h(PID_{ja}'||PID_s||A_j'||R_1') \oplus PID_{ia}'$, $M_7 = h(PID_{ia}'||PID_{ja}'||PID_s||A_j'||R_1')$ and transmits $\{M_5, M_6, M_7, ST_2\}$ message towards drone over a public network channel.

iii. Upon receiving $\{M_5, M_6, M_7, ST_2\}$ message, drone first check the timestamp with system time ($T_c - ST_2 \leq \Delta\ T$), computes $R_1'' = M_5 \oplus h\ (PID_{ja}||A_j)$, $PID_{ia}'' = M_6 \oplus h\ (PID_{ja}||PID_s||A_j||R_1'')$ and $M_7' = h\ (PID_{ia}''||PID_{ja}||PID_s||A_j||R_1'')$, confirms $M_7'? = M_7$, if found no validation, the drone rejects the authentication request, else, it generates random number $R_2 \in Z*_n$, ST_3 and computes $M_8 = h\ (PID_{ja}||PID_{ia}''||ST_3) \oplus R_2$, $M_9 = h\ (R_1''||R_2)$, $SK_{iaja} = h\ (PID_{ia}''||PID_{ja}||PID_s||M_9)$, $M_{10} = h\ (PID_{ia}''||PID_{ja}||PID_s||R_1''||R_2||M_9)$ and sends $\{M_8, M_9, M_{10}, ST_3\}$ message towards CSS over a public channel. Here M_8 must perform encryption using l i.e., $M_8 = E_l(h\ (PID_{ja}||PID_{ia}''||ST_3) \oplus R_2)$.

iv. The CSS, when receiving $\{M_8, M_9, M_{10}, ST_3\}$ message, check the time, decrypt M_8, and computes, $R_2' = M_8 \oplus h(PID_{ja}|\backslash PID_{ia}||R_1)$, $M_9' = h(R_1||R_2')$, and $M_{10} = h(PID_{ia}||PID_{ja}||PID_s||R_1||R_2')$. It then Confirms: $M_{10}'? = M_{10}$, if matches, compute: $SK_{iaja} = h(PID_{ia}||PID_{ja}||PID_s||M_9')$, else, stop calculation. Finally, $\{M_8, M_9, M_{10}, ST_4\}$ message towards the user.

v. The user, upon receiving $\{M_8, M_9, M_{10}, ST_3\}$ message, checks drone time with its current time ($T_c - ST_3 \leq \Delta T$), if found no validation, the process is discarded, else, it decrypts M_8 using s and computes $R_2' = M_8 \oplus h(PID_{ja}|\backslash PID_{ia}||R_1)$, $M_9' = h(R_1||R_2')$, $M_{10} = h(PID_{ia}||PID_{ja}||PID_s||R_1||R_2')$, confirms $M_{10}'? = M_{10}$, if found no validation, the process once again be terminated, otherwise keeps $SK_{iaja} = h(PID_{ia}||PID_{ja}||PID_s||M_9')$ as the session shared key, as shown in Module III.

Module III: Mutual authentication phase

4.4 Dynamic Drone Addition Phase

Let us suppose a new drone denoted by V_j^{new} is required to add to the IoD environment. The control station server (CSS) initially generates a distinctive ID_{ja}^{new} and computes $PID_{ja}^{new} = h(ID_{ja}^{new}||s)$, where s is the secret key. Next, CSS chooses a 160-bits public key l, computes: $A_j^{new} = h(ID_{ja}^{new}||l)$ and stores $\{ID_{ja}^{new}, PID_{ja}^{new}, A_j\}$ in its memory and $\{PID_{ja}^{new}, A_j\}$ in drone's memory. The operator sitting on CSS informs all the previously registered drones from the newly added drone available on IoD for dynamic changing of its topology and deploy for the possible task.

4.5 User's Biometric/Password Update Phase

If a legitimate user desires to change his/her password or biometrics, our protocol offers changing facilities to him/her freely and securely. To do so, the user first enters his/her old identity ID_{ia}, old password PW_{ia}, and imprint biometric BIO'_{ia}; and computes: $\sigma'_{ia} = Rep(BIO'_{ia}, \tau_{ia})$, $A_i^m = h(ID_{ia}||PW_{ia}) \oplus A_i$, $PID_{ia}^m = h(ID_{ia}||PW_{ia}) \oplus PID_{ia}$, generates $R_1 \in Z*_n$ and computes $M_1' = h(PID_s||R_1) \oplus PID_{ia}$. If $M_1'? = M_1$, tells the user to fresh password PW_{ia}^{new}, or re-imprints biometrics BIO_{ia}^{new}, Locally the computations performed as: $PID_{ia} = h(ID_{ia}||s)$, $A_i = h(PW_{ia}^{new}||l)$, $Gen(BIO_{ia}^{new}) = (\sigma_{ia}^{new}, \tau_{ia}^{new})$, $A_i^{new} = h(ID_{ia}||PW_{ia}^{new}) \oplus A_i$, $PID_{ia}^{new} = h(ID_{ia}||PW_{ia}^{new}) \oplus PID_{ia}$ and replaces $\{A_i^m, PID_{ia}^m\}$ with $\{A_i^{new}, PID_{ia}^{new}\}$, as shown in Module IV.

Enters ID_{ia}, PW_{ia} and BIO_{ia}
Computes: $\boldsymbol{\sigma'_{ia}}$=Rep($BIO'_{ia}$, $\boldsymbol{\tau_{ia}}$)
$A_i^m = h(ID_{ia}||PW_{ia}) \oplus A_i$
$PID_{ia}^m = h(ID_{ia}||PW_{ia}) \oplus PID_{ia}$
Generates $R_1 \in Z^*_n$
Computes: $M_1' = h(PID_s||R_1) \oplus PID_{ia}$
Confirms: $M_1'? = M_1$
Enters: PW_{ia}^{new}, BIO_{ia}^{new}
Computes: $PID_{ia} = h(ID_{ia}||s)$, $A_i = h(PW_{ia}^{new}||l)$
$Gen(BIO_{ia}^{new}) = (\boldsymbol{\sigma_{ia}^{new}}, \boldsymbol{\tau_{ia}^{new}})$
$A_i^{new} = h(ID_{ia}||PW_{ia}^{new}) \oplus A_i$
$PID_{ia}^{new} = h(ID_{ia}||PW_{ia}^{new}) \oplus PID_{ia}$
Replaces $\{A_i^m, PID_{ia}^m\}$ with $\{A_i^{new}, PID_{ia}^{new}\}$

Module IV: Password/Biometric change phase

4.6 Drone Revocation/Reissue Phase

If a drone goes out of service or is physically captured by an attacker or taken down/crashed, its data is available in the CSS poses a severe threat. This can, in turn, be used by the unauthorized entity, which means the danger of IoD. Therefore, we suggested that the CSS have a list/database table consisting of unique identities of compromised drones. Personal values can be added and removed from the record correspondingly, i.e., $A_{ja} = ID_{ja}||s$, $A_{ja}^{del} = ID_{ja}||s$, $ID_{ja}? = ID_{ja}^{del}$, if it confirms, delete ID_{ja}^{del} and completely remove the record of such drone from the CSS.

5 Security Analysis

Security analysis for any protocol is considered an essential task. Because security analysis uses system engineering ideas and trust to scrutinize and examine the strength of a cryptographic-based designed protocol, this section identifies the protocol's credibility, authenticates the IoD environment

protocol's stability, shared authentication, and integrity. The protocol mentioned above analyzed formally using BAN Logic [34] and, ProVerif2.02 [35], which are as under:

5.1 Formal Security Analysis

The formal security analysis of the proposed authentication protocol will be conducted using the following different methods used by different researchers from time to time like using BAN Logic [34] and, ProVerif2.02 [35], are as under:

5.1.1 BAN Logic

Before analyzing the proposed protocol using BAN logic [34], let's, define a few concepts and different notation defined are shown in Tab. 2, below:

Table 2: BAN logic notations and its description

Notation	Description
W|≡X	This statement describes believes rule like W believes message X
W◁X	This statement describes seeing rule like W sees message X
W|∼X	This statement describes the Once-Said rule like W once said X
#x	This statement describes freshness rule like x is fresh
$W \overset{K}{\leftrightarrow} X$	This statement describes shared key rules like W and X communicate through key K
W⇒X	This statement describes jurisdiction rules like W control over X
< A > B	This statement describes combine rule like A combines with B
$\{M\}_k$	This statement describes encryption rules like M encrypted by key K
$\{M\}_K$-1	This statement describes decryption rule like M decrypted by key K

Different rules defined are as under:

Rule 1: **Message Meaning**

$$\frac{Uia| \equiv \ Uia \overset{S_K}{\leftrightarrow} CSS, \ \triangleleft \{X\}}{Uia| \equiv \ CSS| \sim \ X} \tag{1}$$

If U_{ia} believes that U_{ia} and CSS share SK sees message X, then U_{ia} believes CSS once said.

Rule 2: **Nonce Verification**

$$\frac{\text{Uia} \mid\equiv \ \#(\text{X}), \ CSS| \sim \ X}{\text{Uia} \mid\equiv \ CSS| \equiv \ X} \tag{2}$$

If U_{ia} believes that message X is fresh that CSS once said X, then U_{ia} believes that CSS trust X

Rule 3: **Jurisdiction Rule**

$$\frac{\text{Vja} \mid\equiv \ CSS| \Rightarrow (X), \ \text{Vja} \mid\equiv \ CSS \mid \equiv \ X}{CSS| \equiv \ X} \tag{3}$$

If V_{ja} believes CSS control X because it is under the jurisdiction of both and Vjabelieves that CSS believes X, then Vja believes X.

Rule 4: **Freshness Rule**

$$\frac{\mathrm{Vja}\mid\equiv\ \#\ (X)\,,\ \mathrm{Vja}\mid\equiv CSS\mid\equiv\ X!}{\mathrm{Vja}\mid\equiv\ X\overset{S_K}{\leftrightarrow}\ CSS} \tag{4}$$

If V_{ja} believes that message X is fresh, and CSS believes X, then V_{ja} believes they sharedkey.

Rule 5: **Belief Rule**

$$\frac{\mathrm{Uia}\mid\equiv(\mathrm{X})}{CSS\mid\equiv(\mathrm{X},\ \mathrm{Vja})} \tag{5}$$

If U_{ia} believes that X, then CSS believe in message X and V_{ja}

1) Goals

The following goals are demonstrated for the proposed authentication protocol.

G_1: $U_{ia}\mid\equiv(R_1)$

G_2: $CSS\mid\equiv(R_1)$

G_3: $CSS\mid\equiv(R_1')$

G_4: $V_{ja}\mid\equiv(R_1')$

G_5: $V_{ja}\mid\equiv(R_2)$

G_6: $U_{ia}\mid\equiv(R_2)$

2) Idealized Form

The following idealized form is as a result of this described for the proposed authentication scheme:

$\mathrm{Message}_1$: $U_{ia}\rightarrow CSS$: $\{M_1,\ M_2,\ M_3,\ M_4,\ ST_1\}\,R_1$

$\mathrm{Message}_2$: $CSS\rightarrow V_{ja}$: $\{M_5,\ M_6,\ M_7,\ \ ST_2\}\,R_1'$

$\mathrm{Message}_3$: $V_{ja}\rightarrow U_{ia}$: $\{M_8,\ M_9,\ M_{10},\ \ ST_3\}\,R_2$

3) *Assumption*

The following assumptions will prove our protocol:

A_1: $CSS\mid\equiv\#(R_1)$, A_7: $CSS\mid\equiv(CSS^{A_i}\ \leftrightarrow\ U_{ia})$

A_2: $U_{ia}\mid\equiv\#(R_1)$, A_8: $U_{ia}\mid\equiv(CSS^{A_i}\ \leftrightarrow\ U_{ia})$

A_3: $V_{ja}\mid\equiv\#(R_1')$, A_9: $CSS\mid\equiv(CSS\ \leftrightarrow^{A_j}\ V_{ja})$

A_4: $CSS\mid\equiv\#(R_1')$, A_{10}: $V_{ja}\mid\equiv(CSS\ \leftrightarrow^{A_j}\ V_{ja})$

A_5: $V_{ja}\mid\equiv\#(R_2)$, A_{11}: $V_{ja}\mid\equiv(V_{ja}\ \leftrightarrow^{K_{iaja}}\ U_{ia})$

A_6: $U_{ia}\mid\equiv\#(R_2)$, A_{12}: $U_{ia}\mid\equiv(V_{ja}\ \leftrightarrow^{K_{iaja}}\ U_{ia})$

4) *Proof*

Now to verify each statement, take $\mathrm{message}_1$, and assumption 2, i.e.,

Seeing1: $U_{ia}\lhd\{M_1,\ M_2,\ M_3,\ M_4,\ ST_1\}\,R_1$

From seeing$_1$ and assumption 1, we get

Seeing$_2$: $CSS|\equiv U_{ia}|\sim \{M_1,\ M_2,\ M_3,\ M_4,\ ST_1\}\ R_1$

Now, taking freshness and assumption 1, we get

Seeing$_3$: $CSS|\equiv U_{ia}|\equiv \#(\{M_1,\ M_2,\ M_3,\ M_4,\ ST_1\}\ R_1)$

From S_2 and S_3 along with nonce verification rule

Seeing$_4$: $CSS|\equiv U_{ia}|\equiv(\{M_1,\ M_2,\ M_3,\ M_4,\ ST_1\}\ R_1)$

Taking S_4 along with the belief rule

Seeing$_5$: $CSS|\equiv(R_1)$ **G_1 Achieved**

From S_5, assumption 5, and jurisdictional rule

Seeing$_6$: $U_{ia}|\equiv(R_1)$ **G_2 Achieved**

Taking message$_2$, assumption 3, we get

Seeing$_7$: $V_{ja} \lhd \{M_5,\ M_6,\ M_7,\ \ ST_2\}\ R_1^{/}$

From S_7 and assumption 1, we get

Seeing$_8$: $V_{ja}|\equiv CSS|\equiv\{M_5,\ M_6,\ M_7,\ \ ST_2\}\ R_1^{/}$

From S_8, assumption 2 and freshness rule

Seeing$_9$: $V_{ja}|\equiv CSS|\equiv \#(\{M_5,\ M_6,\ M_7,\ \ ST_2\}\ R_1^{/})$

From S_9 and nonce verification rule

Seeing$_{10}$: $V_{ja}|\equiv CSS|\equiv(\{M_5,\ M_6,\ M_7,\ \ ST_2\}\ R_1^{/})$

From S_{10} and belief rule

Seeing$_{11}$: $CSS|\equiv(R_1{}^{/})$ **G_4 Achieved**

From S_{10}, belief rule, and assumption 3, we get

Seeing$_{12:}$ $V_{ja}|\equiv(R_1{}^{/})$ **G_3 Achieved**

Taking message$_3$ and assumption 5, we get

Seeing$_{13}$: $V_{ja} \lhd \{M_8,\ M_9,\ M_{10},\ \ ST_3\}\ R_2$ and $U_{ia} \lhd \{M_8,\ M_9,\ M_{10},\ \ ST_3\}\ R_2$

From S_{13}, along with the belief rule, we get

Seeing$_{14}$: $V_{ja}|\equiv U_{ia}|\equiv\{M_8,\ M_9,\ M_{10},\ \ ST_3\}\ R_2$

From S_{14}-, assumption 6, along with the freshness rule, we get

Seeing$_{15}$: $V_{ia}|\equiv U_{ia}\ \#(\{M_8,\ M_9,\ M_{10},\ \ ST_3\}\ R_2)$

From $S_{15\text{-}}$, along with nonce verification

Seeing$_{16}$: $V_{ia}|\equiv U_{ia}|\equiv(\{M_8,\ M_9,\ M_{10},\ \ ST_3\}\ R_2)$

From S_{16}, assumption 5, along with the belief rule

Seeing$_{17}$: $V_{ia\text{-}}|\equiv(R_2)$ **G_5 Achieved**

From S_{17} and belief rule

Seeing$_{18}$: $U_{ia}|\equiv(R_2)$ **G_6 Achieved**

Therefore, from this proof, it has been cleared that the keys exchanges between the user, control station server (CSS), and the drone are fully authenticated by each peer, and no one can compromise its integrity at any stage during communication.

5.1.2 ProVerif2.02 Simulation

To check the proposed protocol's security, a verification software toolkit [35] is now used to confirm its reachability and authorization. The coding is explained in different parts as given as:

```
(*======= CHANNELS =======*)
free MySecCh:channel [private].
free MyPubCh:channel.
(*======= CONSTANTS & VARIABLES =======*)
free SK:bitstring [private].
free IDia :bitstring.
free PWia :bitstring [private].
free CR :bitstring.
free BIOia :bitstring.
free pia :bitstring.
free siia :bitstring.
free s :bitstring [private].
free l :bitstring.
free IDja :bitstring.
free R1 :bitstring.
free R1dash :bitstring.
free R2 :bitstring.
free ST1 :bitstring.
free ST2 :bitstring.
free ST3 :bitstring.
free Aidash:bitstring.
free Tc :bitstring [private].
free deltaT :bitstring [private].
free BIOiadash :bitstring.
free piiadash :bitstring.
free siadash :bitstring.
free PIDiam :bitstring.
free PIDs :bitstring.
free IDs :bitstring.
free Ai :bitstring.
```

```
free PIDja :bitstring.
free Ajdash :bitstring.
free Aj :bitstring.
(*======= EVENTS & QUERIES=======*)
event start_Uia(bitstring).
event end_Uia(bitstring).
event start_CSS(bitstring).
event end_CSS(bitstring).
event start_Vja(bitstring).
event end_Vja(bitstring).
query attacker(SK).
query id:bitstring; inj-event(end_Uia(id)) ==>inj-event(start_Uia(id)).
query id:bitstring; inj-event(end_CSS(id)) ==>inj-event(start_CSS(id)).
query id:bitstring; inj-event(end_Vja(id)) ==>inj-event(start_Vja(id)).
(*======= CONSTRUCTORS & FUNCTIONS =======*)
fun h(bitstring):bitstring.
fun Concat(bitstring,bitstring):bitstring.
fun Concat3(bitstring,bitstring,bitstring):bitstring.
fun Concat4(bitstring,bitstring,bitstring,bitstring):bitstring.
fun Enc(bitstring,bitstring):bitstring.
fun Dec(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun Encr1(bitstring):bitstring.
fun Encr2(bitstring):bitstring.
fun Decl(bitstring):bitstring.
fun Decll(bitstring):bitstring.
fun Gen(bitstring):bitstring.
fun Rep(bitstring,bitstring):bitstring.
(*======= EQUATIONS =======*)
equation forall a:bitstring, b:bitstring; XOR(XOR(a,b),b)=a.
equation forall m:bitstring, key:bitstring; Dec(Enc(m, key), key)=m.
(*=====USER Uia=====*)
let Uia=
eventstart_Uia(IDia);
let piiiadash =Rep(BIOiadash, siia) in
let PIDia=XOR(PIDiam, (h(Concat(IDia, PWia)))) in
```

```
   let M1=XOR(PIDia, (h(Concat(PIDs, ST1)))) in
   let M2=XOR(R1, (h(Concat3(PIDia, PIDs, Ai)))) in
   let M3=h(Concat4(PIDia, PIDs, Ai,XOR(R1,PIDja))) in
   let M4=h(Concat4(PIDia, PIDja, PIDs, Concat(Ai, R1))) in
   out(MyPubCh,(M1, M2, M3, M4, ST1));
   in(MyPubCh,(M8:bitstring,M9:bitstring,M10:bitstring,ST33:bitstring));
   let R2dash=XOR(M8, (h(Concat3(PIDja, PIDia, R1)))) in
   let M9dash=h(Concat(R1, R2dash)) in
   let M10dash=h(Concat4(PIDia, PIDja, PIDs, Concat(R1, R2dash))) in
   if M10dash=M10 then
   let SKiaja=h(Concat4(PIDia, PIDja, PIDs, M9dash)) in
   event end_Uia(IDia)
   else
   0.
   (*=====CONTROL SERVER STATION (CSS)=====*)
   let CSS=
   eventstart_CSS(IDs);
   in(MyPubCh,(M1:bitstring,M2:bitstring,M3:bitstring,M4:bitstring,ST11:
   bitstring));
   let PIDiadash=XOR(M1,(h(Concat(PIDs,ST1)))) in
   let R11dash=XOR(M2, (h(Concat3(PIDiadash, PIDs, Aidash)))) in
   let PIDjadash=XOR(M3, (h(Concat4(PIDiadash, PIDs, Aidash, R1dash))))
   in
   let M4dash=h(Concat4(PIDiadash, PIDjadash, PIDs, Concat(Aidash, R1dash)))
   in
   if M4dash=M4 then
   let M5=h(Concat3(PIDjadash, Ajdash,XOR(ST2,R1dash)))    in
   let M6=h(Concat4(PIDjadash, PIDs, Ajdash, XOR(R1dash,PIDiadash))) in
   let M7=h(Concat4(PIDiadash, PIDjadash, PIDs, Concat(Ajdash, R1dash)))
in
   out(MyPubCh,(M5, M6, M7,ST2));
   event end_CSS(IDs)
   else
   0.
   (*=====DRONE (Vja)=====*)
   let Vja=
   eventstart_Vja(IDja);
```

```
in(MyPubCh,(M5:bitstring,M6:bitstring,M7:bitstring,ST111:bitstring));
let R1dash2=XOR(M5, (h(Concat(PIDja, Aj)))) in
let PIDiadash2=XOR(M6, (h(Concat4(PIDja, PIDs, Aj, R1dash2)))) in
let M7dash=h(Concat4(PIDiadash2, PIDs, Aj, R1dash2)) in
if M7dash=M7 then
let M8=h(Concat3(PIDja, PIDiadash2, XOR(ST3, R2))) in
let M9=h(Concat(R1dash2, R2)) in
let SKiaja=h(Concat4(PIDiadash2, PIDja, PIDs, M9)) in
let M10=h(Concat4(PIDiadash2, PIDja, PIDs, Concat3(R1dash2, R2, M9)))
in
out(MyPubCh,(M8,M9,M10,ST3));
event end_Vja(IDja)
else
0.
process ((!Uia) | (!CSS) | (!Vja))
```

SIMULATION RESULT

Upon running the code, the following result shows that the attacker could not figure out the secret session key at any stage during communication.

```
————————————————————-
Completing equations...
Completing equations...
– Process 1– Query not attacker(SK[]) in process 1
Translating the process into Horn clauses...
Completing...
Starting query not attacker(SK[])
RESULT not attacker(SK[]) is true.
RESULT inj-event(end_Uia(id)) ==>inj-event(start_Uia(id)) is true.
RESULT inj-event(end_Vja(id)) ==>inj-event(start_Vja(id)) is true.
————————————————————
Verification summary:
Query not attacker(SK[]) is true.
Query inj-event(end_Uia(id)) ==>inj-event(start_Uia(id)) is true.
Query inj-event(end_CSS(id)) ==>inj-event(start_CSS(id)) is true.
Query inj-event(end_Vja(id)) ==>inj-event(start_Vja(id)) is true.
————————————————————
(*==============================================*)
```

5.2 Informal Security Analysis

Suppose an adversary has full power by entering the open channel for eavesdropping, altering, deleting, or updating the message exchange between participants. Then how the proposed authentication protocol can resist such known flaws [36]. We will discuss such suppositions one by one here in this section of the paper.

5.2.1 Resists Privileged Insider Attack

Firstly, the control station server (CSS) chooses a big random number l of 160-bits, and a 160-bits secrete number s, collision-free one-way hash function h(.): $\{0, 1\}* \in Z_q*$ and public parameters pms. Secondly, messages exchanged between $U_{ia} \rightarrow$ CSS, CSS $\rightarrow$ $V\text{-}_{ja}$ and $V_{ja} \rightarrow U_{ia}$, i.e., $\{M_1, M_2, M_3, M_4, ST_1\}$, $\{M_5, M_6, M_7, ST_2\}$ and $\{M_8, M_9, M_{10}, ST_3\}$ are in encrypted form in which an insider (let be a privileged one) cannot figure out the internal credentials. Because l is used for encryption, s for decryption purposes and insiders cannot identify anything from it. Also, the exchange among participants is entirely unreadable, so he/she failed to identify the identity or password from the stored information. Therefore, the proposed protocol shows resistance to privileged insider attacks.

5.2.2 Stolen Verifier Attack

Suppose an adversary steals the information from the shared memory and tries to compute identity and other information. Due to exchanging of random numbers R_1, R_1', R_2, R_1'' on each communication, the attacker failed to do so. Similarly, it is hard for him/her to find the big 160-bits random numbers, as these numbers are linked with ID_{ia}, ID_{ja}, ID_s, PID_{ia}, PID_{ja}, PID_s, biometric (BIO_{ia}), and password (PW_{ia}). Therefore, the proposed protocol resists stolen verifier attacks.

5.2.3 Replay Attack

Let suppose if an attacker copies message $\{M_1, M_2, M_3, M_4, ST_1\}$ from a communication channel and desires to replay it at some other time. At this stage, due to the involvement of random numbers and timestamps, the system can quickly identify the replay message and discards such request. The attacker can do the same also for other messages, i.e., $\{M_5, M_6, M_7, ST_2\}$ and $\{M_8, M_{-9}, M_{10}, ST_3\}$. Therefore, the proposed authentication protocol shows resilience to replay attacks.

5.2.4 Untraceability

The drone or user starts each session with different session keys; let us suppose, if an adversary can record the session of a user/drone and tries to record another session at some other time, he/she may find a different session key. We can say that the adversary cannot figure out the same credentials from these session keys for which he/she can identify the exact location or trace user/drone. Therefore, the proposed authentication scheme is untraceable.

5.2.5 Anonymity

Due to the dynamic identities, random numbers, and timestamps, each time a message transmission over a public channel can be performed dynamically. If an adversary desires to copy one message in T_A and another message from the same line on time T_A', he/she cannot identify the surrounding of a user/drone because different messages are communicated between the participants each time. Therefore, the proposed protocol preserves anonymity security features.

5.2.6 DoS Attack

If an adversary copy {M_5, M_6, M_7, ST_2} and chooses timestamp ST_A and the CSS passed T_c-$ST_A \leq \Delta T$, computes $PID_{ia}' = M_1 \oplus h(PID_s||ST_A)$, retrieves A_i' and calculates $R_1' = M_2 \oplus h(PID_{ia}'||PID_s||A_i')$,$PID_{ja}' = M_3 \oplus h\ (PID_{ia}'||PID_s||A_i'||R_1')$ and $M_4' = h(PID_{ia}'||PID_{ja}'||PID_s||A_i'||R_1')$. Next server has to match $M_4'? = M_4$, which is not possible. So, in such situations, the process is terminated and stops further computations. Similarly, if an adversary selects T_A, catch message {M_8, M_9, M_{10}, ST_A} and transmits it towards drone (V_{ja}). Next drone suppose can successfully perform T_c-$ST_A \leq \Delta T$ and computes $R_1'' = M_5 \oplus h(PID_{ja}||A_j)$,$PID_{ia}'' = M_6 \oplus h\ (PID_{ja}||PID_s||A_j||R_1'')$, and $M_7' = h(PID_{ia}''||PID_{ja}||PID_s||A_j||R_1'')$. Now, check $M_7'$with M_7 and if not match then the connection will terminate. Therefore, the proposed protocol is strong against DoS attack.

5.2.7 Drone Capture Attack

Due to unique credentials stored in the memory of a remote drone and distinct session key established among drone, user, and CSS in the network, attackers at any stage cannot capture or divert a drone towards itself. Therefore, the proposed protocol resists drone captures attacks. Similarly, a drone might be required to engage in dangerous situations where it is abandoned during military mission delivery, making it vulnerable to physical capture and traditional cyber threats. An adversary uses white-box attack capabilities to completely control the internal credentials, figure outing identity, and execute the cryptography modules in static and dynamic ways, including all side-channel information. Therefore, the proposed protocol had guaranteed not to disclose any parameters when someone takedown/captured a drone physically.

5.2.8 Resists Side-Channel Attack

Due to being less dependent on fundamental values, confirmation of values at different stages of the protocol, and computing the session shared key randomly for each session which leads to the sequence of operations changing, can generally make the proposed protocol better to resist side-channel attack.

6 Performance and Comparison Analysis

In this section, the performance analysis/evaluation of the proposed authentication protocol can be performed from the perspective of storage overheads, computation, and communication costs by keeping the already experiment conducted by [37], which are as under:

6.1 Storage Overheads Analysis

The storage overheads mean the parameters stored during the registration phase of the proposed scheme. In this regard, ID_{ia}, ID_{ja}, ID_s, and PW_{ia} are stored in 64 bits, each of a total sum of 256 bits in memory space. Biometric keys (σ_{ia}, τ_{ia}) are in 128-bit space; timestamp takes 56 bits space, R_1, R_1', R_1'', R_2 needs $160 + 160 + 160 + 160 = 640$ bits, l, and s needs 320 bits space. Encryption/Decryption functions require every 192 bits, a total sum of 384 bits. Therefore, the storage overheads cost of the proposed authentication protocol is 1784 bits.

6.2 Computation Cost Analysis

The computation cost can be analyzed by keeping in view the experiment done by [37]. According to [24], during the selection of a random numbers the CPU consume 0.539 ms (total random

numbers 6 ($6 \times 0539 = 3.234$ ms)), public-key encryption 3.8500 ms ($3 \times 3.8500 = 11.55$ ms), decryption 3.8500 ms ($3 \times 3.8500 = 11.55$ ms), hashing 0.0023 ms ($24 \times 0.0023 = 0.0552$ ms), multiplication 2.226 ms ($2 \times 2.226 = 4.452$ ms), and addition 0.0288 ms ($11 \times 0.0288 = 0.3168$ ms). The estimated cost for hash-based message authentication is 0.0056 ms. Therefore, the final computation cost for the proposed authentication scheme is 31.158 ms.

Table 3: Comparison analysis

Protocol					
Parameters	[20]	[26]	[27]	[14]	Our
Storage-Overheads in bits	4256	2756	1472	1656	1784
Computation cost in ms	39.092	26.70	31.001	44.794	31.158
Communication cost in Bits	4256	1536	3088	2292	2728

6.3 Communication Cost Analysis

Based on [37], the messages exchanged among all the participants over the public network channel can be considered communication costs. The communication cost for the first message is 1080bits ($\{M_1, M_2, M_3, M_4, ST_1\} = 256 + 256 + 256 + 256 + 56 = 1080$ bits), second and 3rd are 824 each ($\{M_5, M_6, M_7, ST_2\} = 256 + 256 + 256 + 56 = 824$ bits, $\{M_8, M_9, M_{10}, ST_3\} = 256 + 256 + 256 + 56 = 824$ bits). Therefore, the total communication costs for the proposed authentication protocol are 2728 bits.

6.4 Comparison Analysis

Comparing the proposed authentication protocol with state of the art protocols like Challa et al. [20], Seo et al. [26], Farash et al. [27], and Zhang et al. [14], the communication cost is slightly higher than [26], but it is much better in computation cost. The results are shown in Tab. 3, followed by a graph in Fig. 3.

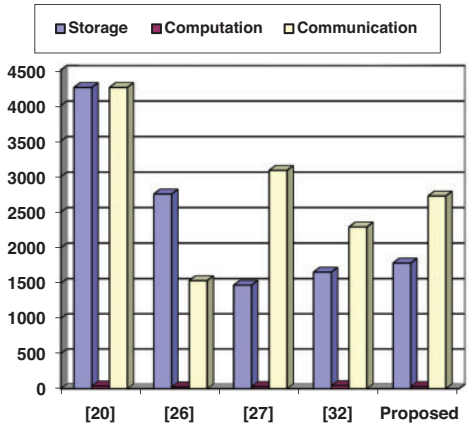


Figure 3: Comparison Chart with state-of-the-art protocols

Furthermore, the various sections in Tab. 4 represent the comparison of the proposed solutions with different security functionalities as given in Challa et al. [20], Seo et al. [26], Farash et al. [27], and

Zhang et al. [14]. For example, in Tab. 4, I represent the Physical Security of the Drone, II represents its security against Impersonation attack, III shows resistance to location threat, IV represents safe against stolen verifier attack, and V represents secure privileged insider threat.

Table 4: Functionalities comparison analysis

Security features	I	II	III	IV	V
Protocol					
Challa et al. [20]	X	✓	✓	X	✓
Seo et al. [26]	✓	X	X	✓	✓
Farash et al. [27]	✓	✓	X	X	✓
Zhang et al. [14]	✓	✓	X	X	X
Our	✓	✓	✓	✓	✓

7 Conclusion

The widespread usage of IoD technology and the non-availability of foolproof secure authentication protocols for the IoD environment motivates us to design a mutual authentication and cross-verification protocol. The current research work deeply examined different protocols available in the literature and highlighted the various flaws in Zhang et al. protocol. We then presented a PKI, XOR, and simple hash function-based protocol used for checksum at both ends. This cyclic checksum of hash functions has the capability of less storage and high security. Its performance is better than any other method because it allows mutual processing of public network channels between Drone-CSS, CSS-Drones, User-Drone, and CSS-Drone without loss of security. The proposed scenario's security has been verified formally using BAN logic of authentication. While the key secrecy, confidentiality, and reachability have been verified using the ProVerif2.02 toolkit. Moreover, the strength of the scheme has been discussed pragmatically in the informal analysis section of the paper. At the end of the article, the performance analysis section has been completed by considering three metrics storage, communication, and computation costs. Upon comparing the proposed scheme with state-of-the-art protocol, it has been shown that it is efficient and effective and can be recommended for practical implementation in the IoD environment.

Acknowledgement: The authors would like to express their sincere thanks to the University of Bisha, Bisha, Saudi Arabia, for the support provided during the research.

Funding Statement: No funding has been received for conducting this research.

Conflicts of Interest: The authors declared that they have no conflict of interest.

References

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

[2] A. Chriki, H. Touati, H. Snoussi and F. Kamoun, "FANET: Communication, mobility models and security issues," *Computer Networks*, vol. 163, pp. 106877, 2019.

[3] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, G. Srivastava, S. Mohan *et al.,* "Cost optimization of secure routing with untrusted devices in software defined networking," *Journal of Parallel and Distributed Computing*, vol. 143, pp. 36–46, 2020.
[4] J. R. Vacca, *Computer and information security handbook*, Waltham, Massachusetts, USA, Newnes, 2012.
[5] D. He, Y. Qiao, S. Chan and N. Guizani, "Flight security and safety of drones in airborne fog computing systems," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 66–71, 2018.
[6] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.
[7] W. Stallings, *Cryptography and Network Security: Principles and Practice, 4th Edition*, India: Pearson Education, 2006.
[8] M. Abdalla, P. A. Fouque and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," *IEE Proceedings-Information Security*, vol. 153, no. 1, pp. 27–39, 2006.
[9] S. Hayat, E. Yanmaz and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.
[10] 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.
[11] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa and D. Dutra, "Network slice mobility in next generation mobile systems: Challenges and potential solutions," *IEEE Network*, vol. 34, no. 1, pp. 84–93, 2020.
[12] V. Chamola, V. Hassija, V. Gupta and M. Guizani, "A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," *Ieee Access*, vol. 8, pp. 90225–90265, 2020.
[13] C. Pu and Y. Li, "Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system," in *2020 IEEE Int. Symp. on Local and Metropolitan Area Networks (LANMAN)*, Orlando, FL, USA, IEEE, pp. 1–6, 2020.
[14] Y. Zhang, D. He, L. Li and B. Chen, "A lightweight authentication and key agreement scheme for internet of drones," *Computer Communications*, vol. 154, pp. 455–464, 2020.
[15] 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.
[16] 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.
[17] Z. Ali, S. Hussain, R. H. U. Rehman, A. Munshi, M. Liaqat *et al.,* "ITSSAKA-MS: An improved three-factor symmetric-key based secure AKA scheme for multi-server environments," *IEEE Access*, vol. 8, pp. 107993–108003, 2020.
[18] S. Barman, H. P. Shum, S. Chattopadhyay and D. Samanta, "A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme," *IEEE Access*, vol. 7, pp. 12557–12574, 2019.
[19] B. Bera, D. Chattaraj and A. K. Das, "Designing secure blockchain-based access control scheme in IoT-enabled internet of drones deployment," *Computer Communications*, vol. 153, pp. 229–249, 2020.
[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. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan *et al.,* "Amassing the security: An ECC-based authentication scheme for internet of drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4431–4438, 2021.
[22] M. Yahuza, M. Y. I. Idris, A. W. A. Wahab, T. Nandy, I. B. Ahmedy *et al.,* "An edge assisted secure lightweight authentication technique for safe communication on the internet of drones network," *IEEE Access*, vol. 9, pp. 31420–31440, 2021.
[23] P. Gope and B. Sikdar, "An efficient privacy-preserving authenticated key agreement scheme for edge-assisted internet of drones," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13621–13630, 2020.

[24] Y. Tian, J. Yuan and H. Song, "Efficient privacy-preserving authentication framework for edge-assisted internet of drones," *Journal of Information Security and Applications*, vol. 48, pp. 102354, 2019.
[25] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the internet of drones applications," *Computer Communications*, vol. 155, pp. 143–149, 2020.
[26] S. H. Seo, J. Won, E. Bertino, Y. Kang and D. Choi, "A security framework for a drone delivery service," in *Proc. of the 2Nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, Singapore, pp. 29–34, 2016.
[27] 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.
[28] F. Al-Turjman, Y. K. Ever, E. Ever, H. X. Nguyen and D. B. David, "Seamless key agreement framework for mobile-sink in IoT based cloud-centric secured public safety sensor networks," *IEEE Access*, vol. 5, pp. 24617–24631, 2017.
[29] 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.
[30] Y. K. Ever, "Secure-anonymous user authentication scheme for e-healthcare application using wireless medical sensor networks," *IEEE Systems Journal*, vol. 13, no. 1, pp. 456–467, 2018.
[31] J. H. Cheon, K. Han, S. M. Hong, H. J. Kim, J. Kim *et al.,* "Toward a secure drone system: Flying with real-time homomorphic authenticated encryption," *IEEE Access*, vol. 6, pp. 24325–24339, 2018.
[32] L. Zhang, L. Zhao, S. Yin, C. -H. Chi, R. Liu, and Y. Zhang, "A lightweight authentication scheme with privacy protection for smart grid communications," *Future Generation Computer Systems*, vol. 100, pp. 770–778, 2019.
[33] L. Teng, M. Jianfeng, F. Pengbin, M. Yue, M. Xindi *et al.,* "Lightweight security authentication mechanism towards UAV networks," in *2019 Int. Conf. on Networking and Network Applications (NaNA)*, Daegu, South Korea, IEEE, pp. 379–384, 2019.
[34] M. Burrows, M. Abad and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.
[35] B. Blanchet, B. Smyth, V. Cheval and M. Sylvestre, "ProVerif 2.02-automatic cryptographic protocol verifier," User Manual and Tutorial, 2020.
[36] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
[37] L. Wu, J. Wang, K. -K. R. Choo and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 319–330, 2018.