IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 23, 2021, accepted April 27, 2021, date of publication April 29, 2021, date of current version May 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3076692

Design and Analysis of Lightweight Authentication Protocol for Securing IoD

SAEED ULLAH JAN[1], FAWAD QAYUM[1], AND HABIB ULLAH KHAN[2], (Member, IEEE)
[1]Department of Computer Science and IT, University of Malakand, Chakdara 18800, Pakistan
[2]Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Doha, Qatar

Corresponding author: Habib Ullah Khan (habib.khan@qu.edu.qa)

This work was supported in part by the Qatar National Library, Doha, Qatar, and in part by the Qatar University (QU) Internal Grant under Grant IRCC-2021-010.

ABSTRACT The Internet-of-drones (IoD) environment is a layered network control architecture designed to maintain, coordinate, access, and control drones (or Unmanned Aerial vehicles UAVs) and facilitate drones' navigation services. The main entities in IoD are drones, ground station, and external user. Before operationalizing a drone in IoD, a control infrastructure is mandatory for securing its open network channel (Flying Ad Hoc Networks FANETs). An attacker can easily capture data from the available network channel and use it for their own purpose. Its protection is challenging, as it guarantees message integrity, non-repudiation, authenticity, and authorization amongst all the participants. Incredibly, without a robust authentication protocol, the task is sensitive and challenging one to solve. This research focus on the security of the communication path between drone and ground station and solving the noted vulnerabilities like stolen-verifier, privileged-insider attacks, and outdated-data-transmission/design flaws often reported in the current authentication protocols for IoD. We proposed a hash message authentication code/secure hash algorithmic (HMACSHA1) based robust, improved and lightweight authentication protocol for securing IoD. Its security has been verified formally using Random Oracle Model (ROM), ProVerif2.02 and informally using assumptions and pragmatic illustration. The performance evaluation proved that the proposed protocol is lightweight compared to prior protocols and recommended for implementation in the real-world IoD environment.

INDEX TERMS Confidentiality, cryptography, drone, security, FANET, miniaturization.

I. INTRODUCTION

With the rapid invention, modification, miniaturization of embedded sensors, fast processing speed of CPU, and universal connectivity of wireless networks, drone technology can be used for different purposes to advance our life-styles. It is used in infrastructure inspection; fire monitoring, wildlife surveillance, cinematography, and agriculture-land monitoring. In addition, secure IoD architecture with physical security to the intersecting routs is obligatory in sensitive military missions. The severe challenges faced by drones now-a-days are security, privacy, and authentication and are an attractive area for research [1]. Before operationalizing a drone in IoD, its control infrastructure needs to secure its open network channel. Wireless network and computing technologies are attractive fields for enhancing quality of life [2]. Likely other computing technologies, Mobile Ad Hoc Network (MANET) contributed a vital role in providing numerous applications like wireless sensor networks (WSN), wireless medical sensor networks (WMSN), smart cities security surveillance, transportation system intelligence and physical phenomenon. A new idea currently came into being called flying ad hoc network (FANET) – which is similar to Mobile Ad Hoc Network (MANET) where nodes are drones, and stable infrastructure are communicating entities [3]. FANET is a subset of MANET, but the security features being developed for MANET cannot be applied to FANET. All the entities' synergy is mandatory in IoD, often missing for such a sensitive networking technology (FANETs) [4].

Furthermore, IoD is potentially vulnerable to several attacks, such as impersonation, drone capture, man-in-the-middle, password guessing, replay, and insider attacks. Before exchanging secrets and confidential information over an unreliable communication channel (FANETs), there is lack of coordination and collaboration of each communicating entity and suffering from not allowing a registered and

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed.

permitted entity to interact securely in IoD. Similarly, drones also have limited flight time and energy resources; therefore, it is vulnerable to many security threats. Without solving these issues correctly for drones would cause immense harm at any time [5]. It can be addressed only by designing a robust authentication protocol for IoD to effectively operationalize drones for both military and civilian domains. The major issues and challenges [6]–[8], faced by drone are:

i. Recently, UAVs (drones) face many security threats, i.e., eavesdropping, information injection, Denial-of-Services, forgery, and collation attacks, which disturb the normal flow of information, data integrity, availability, and confidentiality. There are 27 Satellites fixed in the upper Geostationary Orbit, also referred to as geosynchronous equatorial orbit (GEO) that cover each part of the world through Global Positioning System (GPS), which is a direction-finding system that delivers accurate velocity, location coordinates, and exact timing to a receiving station. GPS signal spoofing/jamming is a severe threat that stops a receiver from receiving a reliable GPS signal. Because an adversary gets and tracks the essential GPS signals used by FANETs for data transmission, it produces and regulates a fake signal using Ettus-USRP[1] of frequency and bandwidth equivalent to that of a GPS signal. It aligns fake and reliable signals, maximizes its strength to suppress the reliable signal, and then uses it for launching a GPS spoofing/jamming attack on both ground control stations and drones correspondingly.

ii. The IEEE 802.11/802.15 standards are commonly used for various networks, especially in civilian UAVs and base stations. Each associated device in wireless communications must become familiar with each other before starting transmission. Management frames can carry out this initial association between devices. If these frames are not adequately protected, the devices are easily exposed to an attacker for sending false frames or take control of either drones or ground control station, or both. Therefore, they should take preventive measures to make it secure from all types of attacks.

iii. As we know, UAVs can secretly catch a photograph of the suspected spot and privately communicate it with the centralized base station for onward decision. A UAV owner requires a robust authentication protocol to perform a useful function, and its flight becomes regulated in the warfare battle field. An Android software toolkit developed by SZ DJI Technology Co., Ltd. installed in the cameras of a drone are used to capture pictures containing invisible information like resolution, manufacturer, recording time, GPS coordinates, and shooting time, which in turn are used by many attackers for their purposes. The ground station's software contains all the secrets sent by drones, like video files, shared photographs, and the specific drone's name. It is a matter of fact that UAVs photos/videos taken and sent to the ground station contain much invisible information, which badly affects the security and privacy of UAV.

iv. As FANET is an infrastructureless network, so if a drone goes out of service, the network is required to reconfigure itself and hand over the communication session to another drone. This is a serious flaw which needs much attention of a soft hand-off methodology to support heterogeneous network applications for maintaining the broken communication session of IoD.

v. UAV communicates from a specified location. However, when an adversary generates a high-frequency signal, the communication session is broken and sensitive information forged. Therefore, a robust authentication protocol is much needed to improve tracking accuracy and reliability in a diverse environment.

vi. Sometimes, if IoD failure occurs, the hackers control the drone using frequency interference. AGCS (Allianz Global Corporate & Specialty) calculates the specific frequency interference that creates a significant risk; in the meantime, these occurrences can create serious security problems for IoD environment. A hacker might also use this frequency interference for malicious deeds.

vii. A leading security threat noted for UAVs is possible collisions with the airplanes and birds, as it flies at a low height and can easily take down planes and get crushed easily and the engine becomes destroyed.[2]

viii. The UAVs can fly for a limited time due to insufficient energy power in it. After completing its flight operation, it sits down for charging in the nearby stations where it can basically charge itself and take off again, which is not a good sign from a security and privacy point of view.

ix. UAVs must be flying within the area where its control towers are operating because it doesn't have a signal during flight like Wi-Fi or any other cellular connections, so they have in the full control of FANETs or must be on-board processing to fly in the area where signals are available for easy communication and data transmission.

x. Finally, adversary can launch a de-authentication attack on UAVs via activating aircrack-ng[3] to scan the coordinates' information from the stolen data packets, while

[1] Universal Software Radio Peripheral (USRP) while Ettus is the parent research company – a radio frequency family and software toolkit based on GNU radio - an open-source software having blockage functions and GPS signal processing modules implementing in Software Defined Radios (SDR) – to support a widespread transceiver front end and operate at any frequency.

[2] "National Aeronautics and Space Administration (NASA)" and "Federal Aviation Administration (FAA)" in cooperation with various other companies, such as Amazon and Google, have been developing the UAS Traffic Management (UTM) system for drones flying at low altitudes in between 200 and 500."

[3] A software application called packet sniffer, used by an attacker to find a route of the packet sent by drone to a centralized intelligence system.

airodump-ng[4] is used for detecting signal strength, particularly in open wireless network channels (FANET). The attacker quickly stores and filters it for necessary information. All the associated drones in that network channel easily detect and de-authenticate with airplay-ng[5], which is a serious security issue and challenge. The attacker now sends a disassociate data stream towards all the associated drones for disconnecting from the ground control station (server). If the attacker fails in such a task, they quickly jam the complete network by regularly sending disassociation packets to make it disturbing for its routine work.

A. SYSTEM MODEL

The embedded sensors inside drone can intelligently collect the physical conditions and relay it to the ground station through FANET. Due to limited battery power, the wireless communication (FANET) for drone technology, embedded sensors and installed applications can communicate seamlessly to right device. FANET provides back-end services, low-latency, fast and intelligent network features to UAVs in IoD environment. For example, i) visual sensor sensing visualize coordinates of tracking a location/spot, ii) pressure sensor on examining atmospheric wind pressure, iii) temperature sensor for examining environment heat, and iv) oxygen saturation sensor examining the amount of oxygen in air etc. Drone or UAV play the central role in IoD. Ground station allows and communicates with drones using FANET for real time condition monitoring like wild-life/forest fire surveillance, troop's movement, weather-forecasting, and war-fear battle field deployment. Certificate authority (CA) is a fully trusted entity which can issue/cancel certificate to/from both ground station and drone or user [9], [10].

Figure 1 shows the system model in this paper having four main participants: drone's service provider (CA), the ground station (gs), a set of drones, and external user. The certificate authority (CA) is considered to be a specialized company for providing connectivity, information processing support, and real-time problem-solving facilities. The ground station (gs) controls, monitors and supervises drone for navigation services. All drones must be equipped with the ground station (gs) and integrated with alternate network services like GPS, 5G, and wireless communication interface. Drones must be deployed in a specific flying zone, and their clusters also be operationalized in pre-determined flight zones. The external user can access a designated drone from some zone. When a drone is in the zone, ground station (server/gs) regulates its flight and authenticates its legitimacy. The confirmation of authenticity of a legitimate drone or the identification of unauthorized drone in the flying zone can also easily be detected due intermediary agent (server/gs).

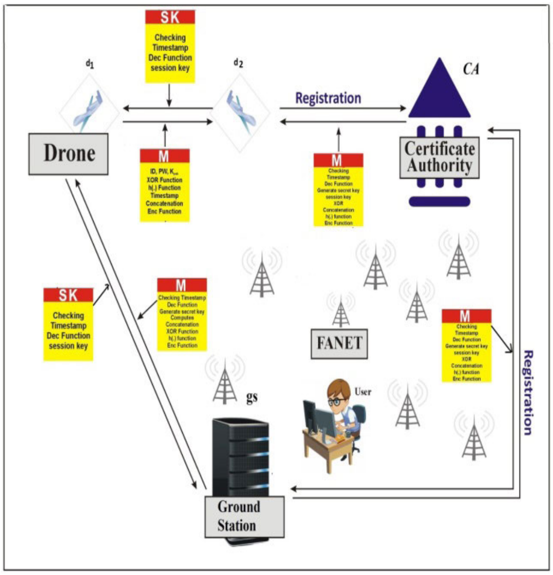


FIGURE 1. System model.

Garibi *et al.* [1] explained the flying zone strategy for a big geographical area in detail. We also consider their zone strategy for achieving impartiality, modularity, and standardization, so that a drone can disseminate information with the ground station and external user securely. Also, to cover a larger area such as a whole country, the ground stations need to be logically interacting with each other. This strategy will supervise the drones in a cluster at different flying zones, traffic, and drone switching from one flying zone to another and provide compulsory statistics. Gharibi *et al.* [1] also explained the handover strategies when a drone shifts its location from one to another flying zone.

Furthermore, the connection is focused to establish communication of drones with ground station in providing excellent data transmission for a tactical purpose. The synergy is mandatory for efficient and effective channel accessibility along with minimum communication overheads. It is worth mentioning that the said communication is synchronous, ground station must check every connection (drone→ to → drone or drone → to → user) so that to qualify for complex operation, otherwise cannot. Suppose there exists N number of drones, $\mathrm{N}^{/}$ is active drone involving for some current task. Ground station is denoted by E, and all other components is said to be C. Let the topology is a true mesh Z(Z-1)/2 where $\mathrm{Z}=\mathrm{N}^{/}+\mathrm{M}+|\mathrm{C}|$, which means the path is allocated to only authorize drone [11]. We have offered a dynamic drone addition phase to our protocol which several other researchers didn't in their protocols. By doing so, the network too dynamically changes its topology depending upon "*who access whom*"?

B. THREAT MODEL

According to this model, an attacker may alter, eavesdrop, or snoop data/information on any public networked-based communication. They might represent themselves as an authentic node (drone) at some location and starts

[4] A software application for capturing object coordinates and input to aircrack-ng.

[5] A software application capable of generating frames traffic that later on injects to the main aircrack-ng.

communication with the ground station; cannot enter the ground station for accessing the internal secret without permission. It can compromise some tags for obtaining the shared session key. Also, an adversary has full power to start negotiation with drone (d) or gs, can insert false tags with the legal message in public network channel during communication, delete the whole or some part of it, copy the message and replay it some other time.

This model was first presented by Dolev & Yao [12] and is called Dolev-Yao Model. Used by various protocols [9], [10], [46], this model tells the authority of an attacker between two communicating bodies through an open network channel. The threat model consists of the following possibilities with an adversary:

1) PRIVACY THREAT

If an adversary install aircrack-ng software for identifying drone's coordinates and other helpful information from the stolen data packets, airodump-ng software for detecting signal strength, stores and filters it for additional attacks and disturbed the synergy by de-authenticate using airplay-ng software. The attacker might jam the complete network by regularly sending disassociation packets to obscure its routine work.

2) PHYSICAL CAPTURE THREAT

An adversary has the possibility to capture a drone physically, or if a drone's dropped down occurs or adversary can transcribe it or destroyed somewhere etc., adversary attack it to gain access to the stored information in the drone's memory. After that, he/she can disclose the encrypted data and start authentication with GCS or drone of the same cluster or any other.

3) TRAFFIC ANALYSIS THREAT

The adversary can carry out to analyze drone traffic to extract valuable data from IoD devices and networks. Packets exchanged between the drone and GCS make up the traffic. The forensic analysis of traffic packets exposes classified details. The drone is equipped with sensors for collecting data from the real-world environment in warfare battlefield containing helpful information in the packets. Adversary analyzed it for potential attacks.

4) ACCESS CONTROL THREAT

An attacker might understand all the rules, policies and how a legitimate entity can communicate? Afterwards, he/she gain access to control, alter privileges, permissions, authorization and authentication, which in turn can lead to considerable losses.

5) IDENTITY SPOOFING THREAT

Adversary can successfully masquerade a legitimate entity using a real drone's spoofed identity. Then he/she gain access to control the public communication channel.

C. MOTIVATION AND CONTRIBUTIONS

Cho *et al.* [45] proposed a protocol for small UAVs based on a hash-based message authentication code Secure Hash algorithmic (HMASHA1) function. The alternate of HMACSHA1 was Message Queuing Telemetry Transport (MQTT), intending to create a bandwidth-efficient, lightweight, and low-power consumption protocol. But slower transmission cycles, unencrypted design, restricted security, and lack of interoperability. MQTT also doesn't operate in open architectures, where multiple applications from various manufacturers are expected to work together seamlessly. Then Hash-based message authentication code Secure Hash algorithmic (HMASHA1) was launched, which are significant authentication results from a secret key. The hash function can work efficiently when applied to the body of a message and simultaneously verify both the data integrity and the authenticity of a message. Cho *et al.* [45] claimed that their protocol is fast and secure for small UAVs. However, the cryptanalysis result shows that Cho *et al.* [45] protocol suffers from a privileged insider, stolen verifier, and outdated data transmission flaw. The protocol failed to add dynamic drone addition and revocation phases. We then proposed an improved scheme for IoD deployment drones using Flying Ad Hoc Network (FANET). The same lightweight cryptographic technique (HMACSHA1) has been used in which a 160-bits random nonce has taken. The protocol consists of drone addition and revocation phases. We have proved its security using the widely used random oracle model (ROM)/ProVerif2.02 and informally using assumptions/lemmas. The main contributions of the research are as under:

i. We have designed authentication protocol for IoD and proved to be safe against the severe threats faced by drone especially privileged insider and stolen verifier attacks along with outdated data received by the ground station (gs) from a legitimate drone.
ii. The proposed authentication protocol is designed using HMACSHA1 which is lightweight, robust and feasible in IoD, as it resists all known attacks.
iii. The randomized key (nonce) generated has the capability of less computation cost, minimum storage overheads and robust/significant improvement in the security of the proposed protocol.
iv. The proposed protocol has been analyzed formally using Random Oracle Model (ROM) [13], and programming verification toolkit ProVerif2.02 [14] and informally putting pragmatics studies which show the robustness of the protocol.
v. The security and performance balancing strategy has been achieved in this work, which was often missing in the recent prior protocols [15].

D. PAPER ORGANIZATION

The rest of the paper is organized as in section 2, the literature review in a summarized form has been demonstrated, section 3 describes in detailed the review analysis of Cho *et al.* [45] protocol. In section 4, the proposed lightweight HMACSHA1 based authentication protocol for IoD has presented. Section 5 describes the security analysis both formally using ROM and informally using theorems and assumptions. We have validated the security of the protocol using the software toolkit ProVerif2.02, the code is given in appendix – A of the paper. In section 6, we assess the performance of the proposed protocol in terms of storage, message and time complexity or communication and computation costs. And then, we compare the performance of the proposed protocol with state of the art protocols, and finally, in section 7, we conclude the research and specify future work shortly.

II. LITERATURE REVIEW

In public key infrastructure (PKI), a digital signature is one of the most significant primitives. Knowing a signer's public key, anyone can check whether the signer's signature is legitimate. So that to allow signatures to be applied to one-to-one and one-to-many applications effectively requires data from leaf nodes to be collected by the root node, resulting in multi-to-one communication. The root node is very likely to be swamped in these applications when too many leave transmits simultaneously. Therefore, to provide normal validity, security and non-repudiation, signatures must be elegantly crafted to prevent the known problem of implosion in many-to-one authentication. For this, Xing *et al.* [16] proposed an identity-based signature authentication protocol based on cubic residues in which they claim that their protocol is the first one in the history of mankind for using cubic root in Eisenstein ring design, but later, failed to no resistance to existential forgery and identity attacks because of the non-usage of Diffie-Hellman Problem. He *et al.* [17] presented a certificateless public key cryptographic-based aggregate signature authentication protocol for eliminating the key-escrow problem and verified the protocol using a random oracle model, which proved to be safe against Type II adversary in random oracle model and highlighted the major drawbacks faced by authentication protocols of the time. They used the computational Diffie-Hellman (CDH) problem to improve protocol to be safe against forgery and collation attacks. Viet *et al.* [18] improved the security of certificateless aggregate signature-based protocols [16], [17] using mathematical lemmas.

As stated, security and privacy are critical concerns in FANET, such as the privacy, authentication and verification of messages before it is sent towards the recipient. Otherwise, the malicious node may alter the messages and even declare itself as legitimate one to send incorrect messages that can trigger a drone crash or deceive it to make an irrational plan for a wrong decision. To overcome the privacy issue, an attacker must not know its details such as real identity, location and session etc. On the contrary, traceability under certain circumstances is also necessary, e.g., using a pseudonym should not be avoided by a drone that sends fake messages. That is to say, FANET needs conditional privacy-preservation; therefore, Zhong *et al.* [19] proposed a certificateless signature-based aggregation protocol and demonstrated that it resists both Type I and II attacks using a random oracle model and Computational Diffie-Hellman Problem (CDHP). And Challa *et al.* [20] deliberated an improved signature-based protocol for network-enabled IoT to be applied for drone technology. After the successful authentication, participants create a secret session key for future communication. They used a fuzzy extractor for verifying user's specific credentials, like checking biometrics locally within the smart card. An ECC approach has also been used in this protocol for tackling the signature generation and verification mechanisms. Their protocol has passed from the new sensing device addition phase, smart card revocation phase and password/biometric update phase. However, the computation time complexity and communication cost compared to others are much more and couldn't be feasible for low power sensing devices.

The user's unique information can generate a unique key called identity, and the technique is called ID-based cryptography. The one party in the communicating network can generate a secret key and sends it to all corresponding users secretly for an encryption/decryption process. The private key is mostly exposed to a devastating attack, which leads to breaking the cryptographic protocol. To overcome such a big flaw for cryptography and to preserve the personal secret key recently, several attempts have been made for introducing an identity-based aggregate mechanism that not only guarantees the security of the user's private access but also delivers a delicate balance between safety and performance. Therefore, the first aggregate signature protocol was presented by Boneh *et al.* [21] in 2003 by aligning n signatures on n messages for n signers. The signature of [21] was worked for two parties, but it couldn't resist forgery attack when users' number increased. Lysyanskaya *et al.* [22] worked and presented three algorithmic-based sequential aggregate signature authentication protocols. They used RSA to secure the protocol, and mathematically, permutation/combination has also been used to construct the aggregate signature. Unfortunately, their protocol also doesn't resist a forgery attack. Paterson and Herranz [23], in 2005, proposed a deterministic identity-based signature authentication protocol. His protocol was a bit effective but failed as an attacker can quickly enter the internal credential of a legitimate user by running an extract algorithm.

Meanwhile, Paterson and Schuldt [24] proposed an efficient identity-based aggregate signature authentication protocol and claimed that their protocol is secure in the random oracle model, but when an attacker runs a query with the help of a challenger can successfully extract the secret identity. And Boldyreva *et al.* [25] constructed an identity-based sequential aggregate signature authentication protocol and was named Ordered Multi Signatures (OMS) protocol based on public-key primitives considered to be much secure protocol of the time. Still, due to the non-usage of Computational Diffie-Hellman Problem (CDHP), the adversary can easily reach the internal credentials by running the access algorithm.

Additionally, Haque *et al.* [26] designed an ID based protocol having i) low computation time complexity during encryption/decryption process, ii) no need to enhance the communication software among peers, iii) removes certificates cryptographic technique, and iv) more appropriate for a diverse environment. However, due to i) an easily compromise of the secret key which in turns disturb the entire messages based on public-private key pair, ii) the secret keys are generated for users, decrypted and signed any messages, iii) secret keys are often generated on the user's computer, which minimize load on server, and iv) SSL-like recommended for large-scale system. It is important to observe that all the users that hold accounts with the PKG must be able to verify themselves. In principle, this may be achieved through username, password or through public/privacy key pairs managed on smart cards. So, IBE solutions may rely on cryptographic techniques that are insecure against code breaking quantum computer attacks. Benzarti *et al.* [27] proposed signcryption, identity and aggregate signature based authentication protocol consisting of i) public key cryptography that simultaneously fulfills both the functions of digital signature and public key encryption in a logically single step, ii) reduction of computational cost and communication over- head, iii) static key management, and iv) it has a reduced computational cost compared to signature-then-encryption protocol which are two basic security properties of any Signcryption protocol. Such properties include integrity, non-repudiation, unforgeability and confidentiality. But digitally signing a message and then encrypting it, consumes more machine cycles and bloats the message by introducing extended bits to it.

Turkanovic *et al.* [28] proposed user authentication protocol in IoT environment can be utilized in the IoD environment. The mutual authentication along the user, sensor node, and the gateway node are achieved in their protocol. After the successful authentication process, both user and sensing node agree on a session key that can be used for future secure communication. Their protocol uses only one-way-hash and bitwise exclusive-OR (XOR) operations during the authentication and key agreement phase. However, their protocol is vulnerable to several attacks, such as man-in- the-middle, sensing node impersonation and stolen smart card attacks. In addition, their protocol also fails to maintain user untraceability, sensing node anonymity and session key security properties.

Recently, Tanveer *et al.* [29] proposed a lightweight protocol for IoD which utilize AE-algorithm, SHA256, and bitwise XOR operation. Their protocol consists of revocation, drone-deployment phases in addition with password updating phase. BAN Logic was used for formally analyzing the security of their protocol, while for simulation they used Scyther toolkit and mathematical assumptions were used for informal analysis. They claim that their protocol resists malicious node and replay attack. A unique methodology has been adopted by Pu and Li [30] for the design of a lightweight protocol for IoD. They used physical unclonable function for verification and validation of message among drone and server. They said that traditional cryptography is not enough for the security of such a sensitive data transmission, PCAP and PUF can guarantee for secure communication. Chaotic map was used for random key generation, PMNeT++ for simulation and compare PCAP with other protocols. The result shows a better attempt done by Pu and Li. Alladi *et al.* [31] also proposed a PUF based authentication protocol for UAV using FANET. Their protocol is computed two session keys for ensuring high security in UAVs' sensitive data transmission. They claim that the identities used in their protocol are protected from all known threats, ensure confidentiality, secrecy, and integrity. Pu [30] used Mao-Boyd logic for checking the security of protocol, and compared with recent protocols of the same domain.

The ECC is shown to be the right choice because, compared to other systems, it can obtain protection at higher levels while consuming less bandwidth and energy and incurring lower overhead computing compared to RSA. In regard to this, [32] proposed protocols based on ECC, in which they claim that their protocols are secure based on traceability assumptions of CDHP. But due to key escrow problem, their protocol is not suitable for practical implementation in IoD environment. Similarly, Ozmen *et al.* [33] said that IoD is crucial for coordinating drone in both civilian and military domains, but its security is a major concern which can be tackled by adopting standard cryptographic primitives. They also expressed that energy-efficient authentication can fulfill the requirements of battery-limited IoD. In this way, they proposed and ECC based cryptographic protocol and proved it using bit-AVR and 32-bit ARM of drone. They claimed that their protocol is secure, broadly encompass and provide efficient and effective result in the random oracle model. The cryptanalysis result shows that [33] is suffering from privacy issue and has many design flaws.

There are many restrictions on the drone's computing resources, due to which it is vulnerable to many security threats, such as replay attacks, forgery attacks, and man-in-the-middle attacks. Critically, the work performed by a drone in smart city' surveillance would cause a big harm at any time. Therefore, Li *et al.* [34] proposed a lightweight identity authentication protocol based on elliptic curve cryptography. But they forget to describe drone addition, revocation and password change phases. Hayat *et al.* [35] aimed at data transmission, instruction data leakage triggered by malicious drone in communication between server and drone. To ensure the identity authentication of drone and ground control station, authenticity and reliability of the transmission instructions obtained by a drone and to guarantee the privacy of a drone's identity details. Wazid *et al.* [36] proposed a protocol based on ECC, having i) a support to the security features, such as user authentication, key agreement, user revocation and non-repudiation, ii) it is an ECC based secure protocol for a drone as a hybrid encryption mechanism for multiple recipients in order to send user-specific information to a huge number of smart objects, iii) it has the characteristic of data gathering party (e.g. a drone) to collect privacy-related information from the smart objects, iv) combines the optimized batch verification method and ElGamal holomorphic encryption protocol, and v) a dual channel strategy which helps the drones to save their battery life. However, due to i) using Elliptic Curve Cryptography (ECC), symmetric-key encryption/decryption, batch verification and one-way-hash cryptographic functions it is not efficient in computation, and ii) low efficiency and high communication cost. Srinivas *et al.* [37] proposed temporal credential lightweight authentication protocol (TCLAS) for drone deployment IoD environment, but failed to restrict unauthorized access of drones. But Singh *et al.* [38] proposed a simple hash cryptographic function based authentication protocol for IoT environment, which is lightweight and balancing of security with performance but the opponent can easily control the already transmitted messages, figure-out nodes' secret values, and then impersonate. Zhang *et al.* [39] also proposed hash cryptographic function based authentication protocol for IoD, a much lightweight protocol, but additionally it needs control server for intervention. Also, [39] used timestamp in the first-round trip, and forget to use it on the other which leads to an outdated data transmission flaw and iTACLAS has been proposed by Ali *et al.* [40] and catered all the weaknesses of TCALAS of [37].

Via different sensors, the Internet of Things (IoT) link massive objects to facilitate everyday life by interconnecting the knowledge space with decision-makers. However, due to the openness of communication networks and the existence of standard isolated sensors, its security, and privacy are considered to be the key concerns. To provide protection and ensure privacy for network enabled sensors' devices and uses it efficiently, Chaudhry *et al.* [41] suggested that the bilinear pairing cryptographic method is heavyweight and not good for fast and secure communication especially in IoT. They proposed a pairing free authentication protocol [41] for DIoT and formally verified its security using the random oracle model method. In addition, to supply additional power to grid networks, the electric vehicles must have the capability to consume less energy from the grid. A stable key establishment is critical in initiating the transmission of bidirectional power into and from the system. The authentication protocol must be free from cyber-attacks to enforce any Energy-Internet (EI)-based vehicle-to-grid (V2G) communication successfully. Therefore, Irshad *et al.* [42] highlighted the different drawbacks like desynchronization, replay, and man-in-the-middle attacks in various state of the art authentication protocols and presented an improved V2G framework which safe against much vulnerability. Their protocol delivers efficient and effective services for the end-user.

Moreover, the edge computing architecture has allowed many data to be processed at the edge of the network near the data generation source in the smart grid environment by many connected automated devices. Control of demand response is a fundamental necessity for an effective and secure intelligent grid environment that can deliver very often by exchanging data between smart devices and the Utility Center (UC) in a smart city. Many protocols have been presented for a grid environment to make it secure from potential attacks. In this regard, Chaudhry *et al.* [43] proposed a unique security mechanism named a scheme for demand response management (DRMAS). DRMAS offers all the essential security demand of the grid environment and exchange information in just two round trips, which means its performance, is better than other protocols.

Fog computing is suffering from privacy issues; without secure authentication and key management, it will never perform well for the end-user. Therefore, to ensure privacy, security and authentication issues and challenges of fog commuting for the end-user, Ali *et al.* [9], very recently proposed a scheme that resists the known attacks reported from time to time. They scrutinized their protocol using AVISPA software toolkit and BAN mathematical logic of authentication and informally using discussion. The communication and computation costs have also been compared with many schemes. The performance evolution result of their strategy is much better compared to other methods.

Desynchronization is a significant flaw now-a-days because millions of users are involved in information browsing from a different host. The attacker reaches internally to the server using some tags and desynchronizes the shared memory for the end-user. Remote users, in this regard, suffer from synchrony issue; therefore, to provide efficient services to the end-user, Jan *et al.* [44] proposed a scheme based on bilinear map technique mitigates this major flaw.

III. REVIEW ANALYSIS OF CHO *ET AL*. PROTOCOL

In 2020 Cho *et al.* [45] proposed an efficient and secure authentication protocol for UAVs in which they said that drone must suffer from privacy and security challenges. They named their framework as SENTINEL working in IoD environment. The communication cost of their protocol is efficient and effective due to symmetrically exchanging of certificate among the participants. They simulated their protocol using ECDSA, HMACSHA1 and FBKDF2. They designed a 5G data transmission path between drone and ground control station in IoD and which is also feasible for FANETs. Their system has four participants that are to play a central role in the IoD including GS, CA, UAV, and end user or operator. The hand-held mobile-device or remote control, first receive certificate from CA, obtained and install certificate for UAV and share its copy to CA. While the GS directly receive certificate form CA which means that all the participants are securely registered with each other.

The working scenario of Cho *et al.* [45] a UAV, before going for a mission, needs approval from GS as: shown in phase 1. The mutual authentication and cross-verification between UAV and GS have been performed in the 2^{nd} phase. The exchange of message which contains UAV's identity, HMAC, flying zone coordinates, and shared session key (secret one) is in 3^{rd} phase of their protocol. Upon receiving the said credentials, the GS repossesses the UAV's flight confirmation, given plan and share secret key and keep its record in the database. The GS then approve and checks weather the message exchange take place from an authentic drone or not. The different notations and its description used by Cho *et al.* protocol is shown in Table 1.

TABLE 1. Notations used by Cho *et al.* [45].

Notation	Description
$\|\|$	Concatenation Function
Enc	Encryption Function
Dec	Decryption Function
Sig	Signature Function
PBKDF	Password-based-key-derivation function
sk_d	Drone's secret key
sk_{gs}	gs secret key
pk_d	Drone's public key
pk_{gs}	gs public key
k_d	Drone's flight session key
k_{gs}	gs flight session key
HMAC	Hash-Message-Authentication-Code function
id_d	Drone's d identity
fp_d	Plan for drone's d flight
gs	Ground station
n_d	Drone's Nonce
N_{gs}	Nonce for gs nonce
$cert_d$	Drone's d certificate
cer_{gs}	gs certification
CA	Certificate authority
pms	Master secret key
H_d	Drone's hash function
$H^{/}_d$	gs has values computed by d
msg_d	Drone's message

A. KEY AGREEMENT PHASE

Let participants i.e. drone and GS have already registered with the third-party entity called CA. The GS authenticate the legitimacy of drone by confirming CA's issued certificate, and flight plan session secret key. The following steps are performed:

i. The drone *d* selects a random nonce n_d, id_d, $cert_d$, flight plan f_d and sends it towards ground station GS over a public network channel.

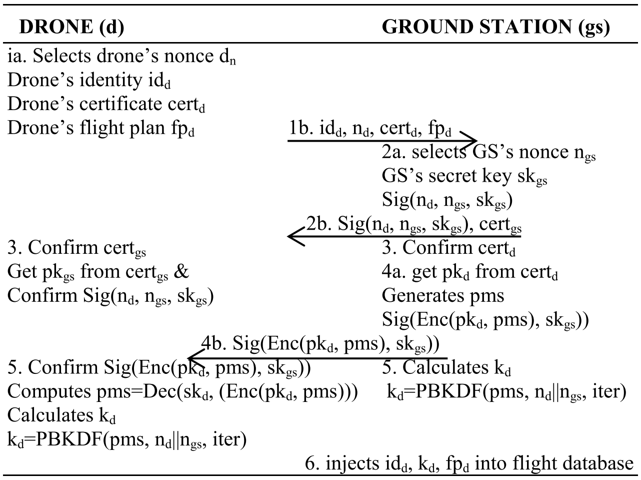
ii. Upon receiving $\{n_d, id_d, cert_d, f_d\}$ message, *GS* also selects a random nonce n_{gs}, and sign it using secret key sk_{gs} i.e. $Sig(n_d, n_{gs}, sk_{gs})$ and send back to drone's containing the gs's $cert_{gs}$.

iii. Bothe entities cross checked the certificates of each other, drone extract pk_{gs} from $cert_{gs}$ for confirming $Sig(n_d, n_{gs}, sk_{gs})$.

iv. Ground station extracts pk_d from $cert_d$, generates *pms* and encrypt it with pk_d i.e. $Enc(pk_d,\ pms)$ along with sk_{gs}, built $Sig(Enc(pms,\ pk_d),\ sk_{gs})$ and transmit towards drone over an open network channel.

v. The ground station performs $n_d||n_{gs}$, calculate the flight schedule key k_d along with *pms*. Drone also confirms $Sig(Enc(pms,\ pk_d),\ sk_{gs})$ on pk_{gs}, if found valid, decrypt it using sk_d.

vi. Upon confirming the session k_d, ground control station *gs* registers the legitimate drone's identity id_d. flight plan fp_d, secret key k_d and stored it in its database as shown in phase 1:


DRONE (d)
GROUND STATION (gs)
ia. Selects drone's nonce d_n
Drone's identity id_d
Drone's certificate $cert_d$
Drone's flight plan fp_d
1b. id_d, n_d, $cert_d$, fp_d
2a. selects GS's nonce n_{gs}
GS's secret key sk_{gs}
$Sig(n_d, n_{gs}, sk_{gs})$
2b. $Sig(n_d, n_{gs}, sk_{gs})$, $cert_{gs}$
3. Confirm $cert_{gs}$
Get pk_{gs} from $cert_{gs}$ &
Confirm $Sig(n_d, n_{gs}, sk_{gs})$
3. Confirm $cert_d$
4a. get pk_d from $cert_d$
Generates pms
$Sig(Enc(pk_d, pms), sk_{gs}))$
4b. $Sig(Enc(pk_d, pms), sk_{gs}))$
5. Confirm $Sig(Enc(pk_d, pms), sk_{gs}))$
Computes $pms=Dec(sk_d, (Enc(pk_d, pms)))$
Calculates k_d
$k_d=PBKDF(pms, n_d||n_{gs}, iter)$
5. Calculates k_d
$k_d=PBKDF(pms, n_d||n_{gs}, iter)$
6. injects id_d, k_d, fp_d into flight database

Phase 1: Key Agreement

B. AUTHENTICATION OF DRONE

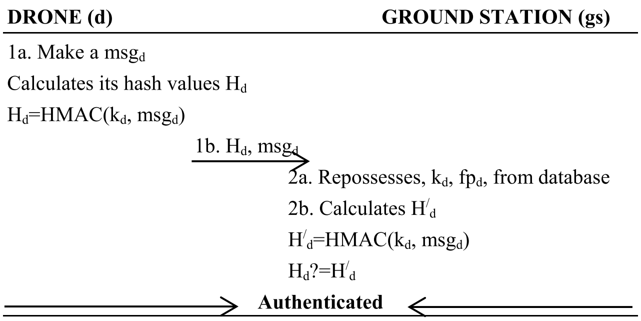
The session secret key k_d is used by a drone *d* to register with the ground control station gs. This phase of the protocol is competed in the following two steps:

1) DRONE → TO → GROUND STATION

This step involves the following computations:

i. First *d* creates msg_d that have all related information like coordinates, timestamp, location, identity (id_d), ground station identity (id_{gs}) and destination identity. Using HMAC function for calculating the drone's code H_d, k_d, msg_d and relays h_d along with msg_d to gs over a public network channel.

ii. Upon receiving the message from d, gs computes drone's identity id_d, extracts session key k_d and confirms H_d validity using $H^{/}_d = HMAC(k_d, msg_4)$. Compares H_d with $H^{/}_d$, if found valid, d is allowing for entering in the flying zone, else, a denied action is performed, as shown in phase 2.



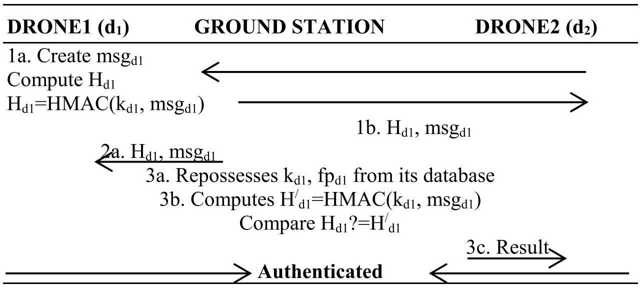
Phase 2: Drone to Ground Station Authentication

2) DRONE → TO → DRONE

If one drone desires to communicate with other drone, both must perform the following set of computations:

i. Let suppose drone1 of identity d_1 makes a message msg_1 having all information like location, coordinates, GPS, timestamp, id_{d1}, and the identity of destination drone id_{id2}. The first drone d_1 computes H_{d1} using HMAC using k_{d1}, msg_{d1} and relays it to d_2 over a public network channel.

ii. Upon receiving the message by d_2, d_2 sends the message to gs. On receiving the message of d_2 by gs, gs checks the identity of d_1 in its record i.e. id_{d1}, fp_{d1}, k_{d1} and validates H_{d1} by calculating $H^{/}_{d1} = HMAC(k_{d1}, msg_{d1})$. Compares $H_{d1} ?{=} H^{/}_{d1}$, if match, gs authenticate and tell d_2 that d_1 is correct according to their record as shown in phase 3.


DRONE1 (d_1)
GROUND STATION
DRONE2 (d_2)
1a. Create msg_{d1}
Compute H_{d1}
$H_{d1}=HMAC(k_{d1}, msg_{d1})$
1b. H_{d1}, msg_{d1}
2a. H_{d1}, msg_{d1}
3a. Repossesses k_{d1}, fp_{d1} from its database
3b. Computes $H^{/}_{d1}=HMAC(k_{d1}, msg_{d1})$
Compare $H_{d1}?=H^{/}_{d1}$
3c. Result
Authenticated

Phase 3: Drone to Drone Authentication

C. CRYPTANALYSIS OF CHO ET AL. [45] PROTOCOL

The cryptanalysis result of protocol [45] shows that it suffered from Privileged Insider Attack, Stolen Verifier Attack and Outdated Data Transmission flaw. These are explained as under:

1) PRIVILEGED INSIDER ATTACK

The only solution for the secure management of a privileged identity can provide significant access rights. The privileged identities can also help the management teams to identify and conclusively respond to possible insider threats or attacks before it damages the system. In this connection, Cho *et al.* [45] protocol, when a drone (d), initiate flight, it extracts a random number n_d, and identity id_d, sends it towards ground station gs on open channel which is a soft target for an operator to use it for launching some other attack on accessing other application. Nonetheless, it might not use the same identity everywhere, but according to [29], thousands of users have the habit of reusing the same identity and password. As per statistics received from Microsoft, in just three months, out of 1.5 million users used 6.5 identities and passwords for only 25 websites, means a single password is shared in 3.9 online accounts/applications. Therefore, if a privileged insider/administrator of a ground station (gs) knows the identity (id_d), they can easily impersonate it by using somewhere else. In [45] a drone sends identity to gs directly where the privileged insider can get and abuse it some other place for accessing other applications. Therefore, Cho *et al.* [45] is venerable to privileged insider attack.

2) STOLEN VERIFIER ATTACK

If an attacker forges the previous or current session authenticated keys (k_d, sk_{gs}) and send towards ground station (gs), it forces gs as legal Drone for the upcoming authentication session. Because, in Cho *et al.* [45] protocol, the session key sk_{gs} is without encryption, is available in simple format in the memory of gs, so adversary can steal it to figure-out the internal credentials from it, which might harm the whole system in future. Similarly, on the other hand if an attacker A can steal $\{id_d, n_d, cert_d, fp_d\}$the message from the open network channel and transmits some other time towards gs. Ground station (gs) consider that it is sent by a drone (d), gs also chooses random number n_{gs} and computes $Sig(sk_{gs}, n_d, n_{gs})$, sends $\{Sig(sk_{gs}, n_d, n_{gs}), cert_d\}$message back towards drone (d). Upon receiving, drone (d) extracts n_{gs} from $cert_d$ and validate $Sig(sk_{gs}, n_d, n_{gs})$. Hence, disturbs the whole system for sensitive activity. Further, adversary can also calculate session key sk_{gs} and k_d. Therefore, Cho *et al.* [45] protocol is suffered from stolen-verifier attack due to lack of encryption function.

3) OUTDATED DATA TRANSMISSION FLAW

By granting approval of flying zone, the protocol doesn't explain in which time threshold will it use. Because, each drone, primarily, gets approval of flying zone/flight plan. Suppose, an attacker can prove the approval and grant flight session key, then it not only misguides drone for other task but can also disturb the whole system. They forgot to use timestamp in each message to make it for specific time. Not only in flight-plan/flying-zone, also in credentials of a previous session can also easily using by an attacker for sending towards ground station gs. Therefore, Cho *et al.* [45] protocol suffers from outdated data transmission flaw.

4) MISSING DYNAMIC DRONE ADDITION PHASE

If the system administrator desires to add new drone to its system for some other tasks; it has not been mentioned by [45], how to add a drone for the system? A dynamic drone addition phase is missing in [45].

5) MISSING DRONE REVOCATION/REISSUE PHASE

Similarly, in Cho *et al.* [45] protocol, the drone's revocation/reissue phase has not been specified. If a drone goes out or crashed, its credentials still present in the database of ground station (gs) in Cho *et al.* [45] protocol because it has not been mentioned by them that how to evocate/cancel/reactivate drone from/to the system.

IV. PROPOSED SOLUTION

The proposed protocol is divided into five phases i.e. registration, key-agreement, drone to drone authentication, dynamic drone's addition, and drone's revocation/reissue phases. These phases are described one by one under the following sub-headings while different notations used are shown in Table 2.

TABLE 2. Notations and its descriptions.

Notation	Description
$\|$	Concatenation Function
Enc	Encryption Function
Dec	Decryption Function
d	Drone
PBKDF	Password-based-key-derivation function
sk_d	Drone's secret key
sk_{gs}	Ground-station's secret key
pk_d	Drone's public key
pk_{gs}	Ground-station's public key
k_d	Drone's session key
k_{gs}	Ground-station's session key
HMAC	Hash-Message-Authentication-Code function
id_d	Drone's identity
fp_d	Drone's Plan
gs	Ground-station
n_d	Drone's Nonce
n_{gs}	Ground-station's Nonce
$cert_d$	Drone's *d* certificate
cer_{gs}	Ground-station's certification
CA	Certificate Authority
pms	Master secret key
H_d	Drone's hash function
msg_d	Drone's message

A. REGISTRATION PHASE

This phase of the protocol is accomplished in the following two sub phases:

1) GROUND STATION'S REGISTRATION

For the registration of ground station (gs) with the certificate authority (CA), it must perform the following set of operations:

i. The ground station chooses its certificate cer_{gs}, nonce n_{gs}, and secret key sk_{gs} and computes the master secret key pms= $sk_{gs}\|pk_{gs}$ and sends $\{cer_{gs}, n_{gs}, pms\}$ message towards CA through private channel.

ii. CA keeps $\{cer_{gs}, n_{gs}, pms\}$ in its memory.

iii. CA Calculates $A_{gs} = h(\{cer_{gs})$ and relays it towards the ground station through private channel as shown in module I.

GROUND STATION (gs) — **CERTIFICATE AUTHORITY (CA)**

Selects cer_{gs}, nonce n_{gs}, and sk_{gs}
Computes: pms=$sk_{gs}||pk_{gs}$

$\xrightarrow{cer_{gs},\ n_{gs},\ pms\}}$

Calculates: $A_{gs}=h(cer_{gs})$
Save $\{cer_{gs}, n_{gs}, pms\}$

$\xleftarrow{\{A_{gs}\}}$

Stores $\{h(.), n_{gs}, pms\}$

MODULE I
GROUND STATION'S REGISTRATION

2) DRONE'S REGISTRATION

In this second sub phase of registration phase, the registration of a drone is performed in the following steps:

i. Drone chooses id_d ,nonce n_d and $cert_d$.

ii. Drone calculates $A_d = (n_d \oplus id_d)$ and $PBKDF= h(cert_d\|A_d)\|n_d$ and transmits $\{A_d, PBKDF\}$ to CA via secure channel.

iii. CA calculates $B = h(A_d||PBKDF)$ and $C = h(A_d||sk_{gs})\oplus PBKDF$.

iv. CA keeps $\{B, C, h(\cdot)\}$ in its memory and relays $D = \{B, C, pk_d, h(\cdot)\}$ message towards g via secret channel.

v. $\{D, C, pk_d, h(\cdot)\}$ parameters are stored in Drone's memory as shown in module II.

DRONE (d) — **CERTIFICATE AUTHORITY (CA)**

Chooses id_d, nonce n_d and $cert_d$
Computes:
$A_d=(n_d\oplus id_d)$
$PBKDF=h(cert_d||A_d)||n_d$

$\xrightarrow{\{A_d,\ PBKDF\}}$

Computes:
$B=h(A_d||PBKDF)$
$C=h(A_d||sk_{gs})\oplus PBKDF$
Save $\{B, C, h(\cdot)\}$
$D=\{B, C, pk_d, h(\cdot)\}$

$\xleftarrow{\{D\}}$

Stores $\{D, C, pk_d, h(\cdot)\}$

MODULE II
DRONE'S REGISTRATION

B. KEY AGREEMENT PHASE

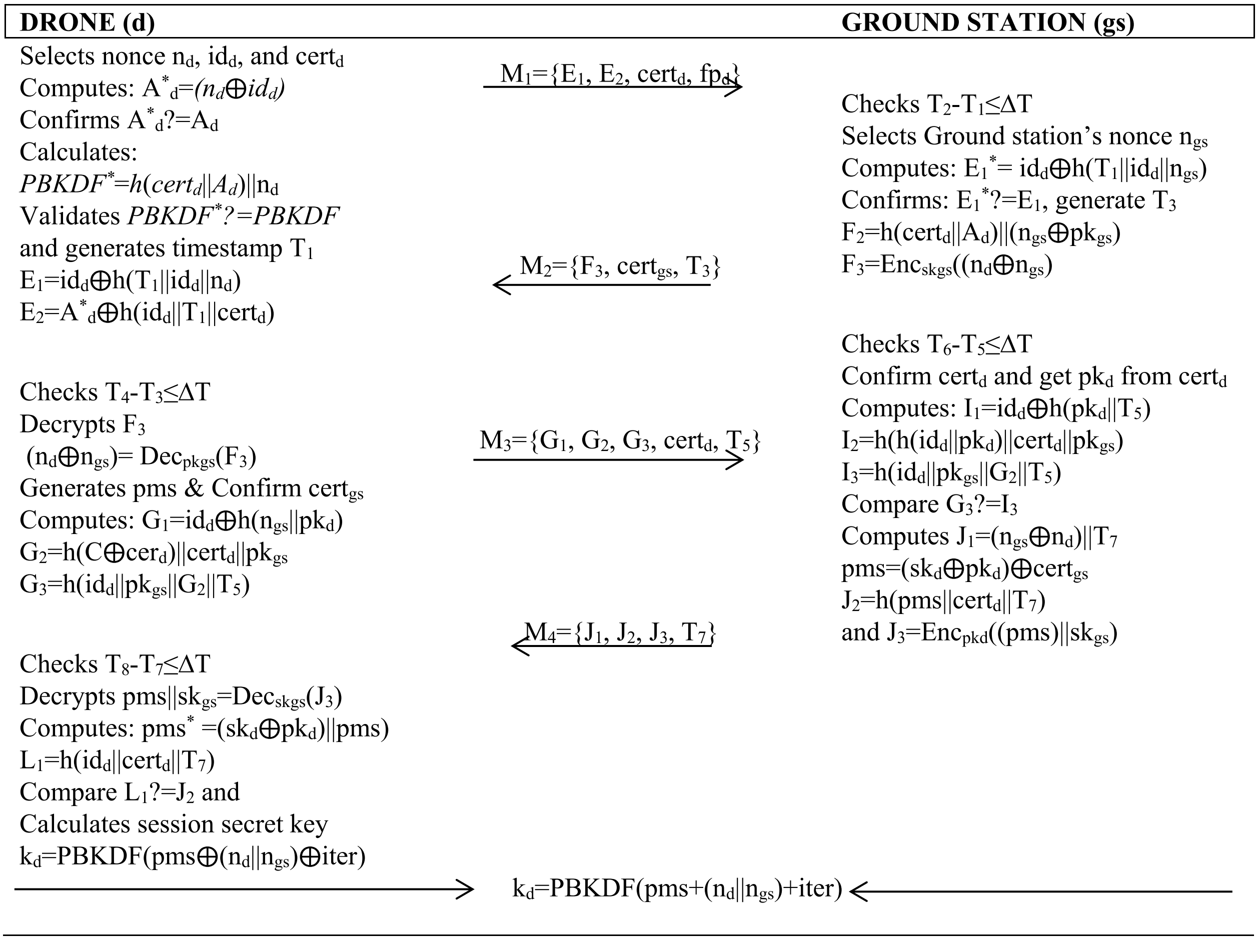
In this phase the ground station (gs) authenticates the legitimacy of Drone (d) by confirming CA's issued certificate, and session secret key. The following steps are performed:

i. The drone d selects a random nonce n_d, id_d, $cert_d$, plan f_d and computes: $A^*_d = (n_d\oplus id_d)$, validates it with the already stored values $A^*_d?{=}A_d$, if found not valid, the process terminates, else, calculates $PBKDF^* = h(cert_d\|A_d)\|n_d$, again validates $PBKDF^*? = PBKDF$. For successful confirmation of password-based-key-derivation operation (PBKDF), computes $E_1 = id_d\oplus h(T_1\|id_d\|n_d)$, $E_2 = A^*_d\oplus h(id_d\|T_1\|cert_d)$ and transmits $M_1 = \{E_1, E_2, cert_d, fp_d\}$towards gs over an open network channel.

ii. Upon receiving $M_1 = \{E_1, E_2, cert_d, fp_d\}$message, gs first checks timestamp T_2-$T_1 \leq \Delta T$ and selects a random nonce n_{gs}, secret key sk_{gs}; computes: $F_1 = E_1\oplus h(T_1\|id_d\|n_d)$, $F_2 = h(cert_d\|A_d)\|n_{gs}$, encrypt F_3 using gs secret key $F_3 = Enc_{skgs}((n_d\oplus n_{gs})\|T_2)$ and transmits $\{F_3, cert_{gs}, T_3\}$back to drone d.

iii. The d first check the timestamp and selects pk_{gs} from $cert_{gs}$, confirm its validity by decrypting $F_3 = Enc_{skgs}((n_d\oplus n_{gs})\|T_2)$ into $(n_d\oplus n_{gs})\|T_2 = Dec_{pkgs}(F_3)$. drone generates pms and confirms $cert_{gs}$ from $F_3 = Enc_{skgs}((n_d\oplus n_{gs})\|T_2)$ function; computes: $G_1 = id_d\oplus h(n_{gs}\|pk_d)$, $G_2 = h(C\oplus cer_d)\|cert_d\|pk_{gs}$ and $G_3 = h(id_d\|pk_{gs}\|G_2\|T_5)$. Finally d transmits $M_3 = \{G_1, G_2, G_3, cert_d, T_5\}$message towards gs over an open network channel.

iv. gs receives M_3 message, validate time interval by checks T_6-$T_5 \leq \Delta T$, confirms $cert_d$, get pk_d from $cert_d$, and computes: $I_1 = id_d\oplus h(pk_d\|T_5)$, $I_2 = h(h(id_d\|pk_d)\|cert_d\|pk_{gs})$, $I_3 = h(id_d\|pk_{gs}\|G_2\|T_5)$, and



DRONE (d) | **GROUND STATION (gs)**

Selects nonce n_d, id_d, and $cert_d$
Computes: $A^*_d=(n_d \oplus id_d)$
Confirms $A^*_d?=A_d$
Calculates:
$PBKDF^*=h(cert_d||A_d)||n_d$
Validates $PBKDF^*?=PBKDF$
and generates timestamp T_1
$E_1=id_d \oplus h(T_1||id_d||n_d)$
$E_2=A^*_d \oplus h(id_d||T_1||cert_d)$

$\xrightarrow{M_1=\{E_1, E_2, cert_d, fp_d\}}$

Checks $T_2-T_1 \leq \Delta T$
Selects Ground station's nonce n_{gs}
Computes: $E_1^*= id_d \oplus h(T_1||id_d||n_{gs})$
Confirms: $E_1^*?=E_1$, generate T_3
$F_2=h(cert_d||A_d)||(n_{gs} \oplus pk_{gs})$
$F_3=Enc_{skgs}((n_d \oplus n_{gs})$

$\xleftarrow{M_2=\{F_3, cert_{gs}, T_3\}}$

Checks $T_4-T_3 \leq \Delta T$
Decrypts F_3
$(n_d \oplus n_{gs})= Dec_{pkgs}(F_3)$
Generates pms & Confirm $cert_{gs}$
Computes: $G_1=id_d \oplus h(n_{gs}||pk_d)$
$G_2=h(C \oplus cer_d)||cert_d||pk_{gs}$
$G_3=h(id_d||pk_{gs}||G_2||T_5)$

$\xrightarrow{M_3=\{G_1, G_2, G_3, cert_d, T_5\}}$

Checks $T_6-T_5 \leq \Delta T$
Confirm $cert_d$ and get pk_d from $cert_d$
Computes: $I_1=id_d \oplus h(pk_d||T_5)$
$I_2=h(h(id_d||pk_d)||cert_d||pk_{gs})$
$I_3=h(id_d||pk_{gs}||G_2||T_5)$
Compare $G_3?=I_3$
Computes $J_1=(n_{gs} \oplus n_d)||T_7$
$pms=(sk_d \oplus pk_d) \oplus cert_{gs}$
$J_2=h(pms||cert_d||T_7)$
and $J_3=Enc_{pkd}((pms)||sk_{gs})$

$\xleftarrow{M_4=\{J_1, J_2, J_3, T_7\}}$

Checks $T_8-T_7 \leq \Delta T$
Decrypts $pms||sk_{gs}=Dec_{skgs}(J_3)$
Computes: $pms^* =(sk_d \oplus pk_d)||pms)$
$L_1=h(id_d||cert_d||T_7)$
Compare $L_1?=J_2$ and
Calculates session secret key
$k_d=PBKDF(pms \oplus (n_d||n_{gs}) \oplus iter)$

$\longrightarrow k_d=PBKDF(pms+(n_d||n_{gs})+iter) \longleftarrow$

MODULE III
KEY AGREEMENT PHASE

compare $G_3?=I_3$. If found valid, ground station computes: $J_1 = (n_{gs} \oplus n_d)||T_7$, $pms=(sk_d \oplus pk_d) \oplus cert_{gs}$, $J_2 = h(pms||cert_d||T_7)$, and encrypts $J_3 = Enc_{pkd}((pms)||sk_{gs})$. Finally transmits, $M_4 = \{J_1, J_2, J_3, T_7\}$message towards drone over an insecure channel.

v. Drone checks timestamp, T_8-$T_7 \leq \Delta T$, decrypts $pms||sk_{gs} = Dec_{skgs}(J_3)$, computes: $pms^* = (sk_d \oplus pk_d)||pms)$, $L_1 = h(id_d||cert_d||T_7)$ and compare $L_1?=J_2$. If matches computes session secret key $k_d = PBKDF(pms \oplus (n_d||n_{gs}) \oplus iter)$ and cross checking the certificates of each other. Keeps $k_d = PBKDF(pms+(n_d||n_{gs})+iter)$ as a shared session secret key for future communication. Whereas iter represents the number of round trip used for calculating session shared key as shown in module III.

C. AUTHENTICATION OF DRONE WITH OTHER DRONE

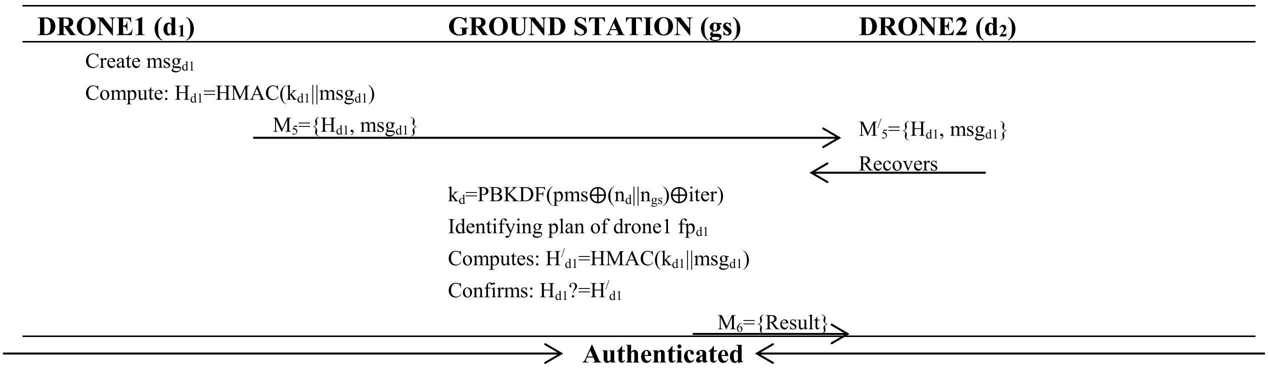
This phase of the protocol means that how a legal drone can communicate with other registered (legal) drone.

i. Suppose drone1 represented by d_1 of identity id_{d1} creates a message msg_1 containing user1's location, coordinates, GPS, and timestamp information. While drone2 denoted by d_2 of identity id_{id2}. Both d_1 and d_2 desire to communicate each other, the first drone1 (d_1) computes H_{d1} using HMAC using k_{d1}, msg_{d1} and send it to drone2 (d_2) over a public network channel.

ii. Upon receiving the message by drone2, it sends the message to ground station (gs) for validation. On receiving the message of d_2 by gs, it first checks the identity of drone1 id_{d1}, flight plan fp_{d1} and session shared key k_{d1} in its record. Then the gs validates hash code of drone1 i.e. H_{d1} by calculating $H^{/}_{d1} = HMAC(k_{d1}||msg_{d1})$ and compares $H_{d1}?=H^{/}_{d1}$, if matches, the ground station authenticate d_1 and tells d_2 that d_1 is correct according to their record as shown in module IV.

D. DRONE ADDITION PHASE

If the ground station desires to add a new drone, this protocol securely facilitates the dynamic addition of new drone. Suppose the new drone is represented by d^{new} its identity id^{new}_d. Before going to deploy for a critical task, it first register with Certificate Authority (CA) and then register with ground station (gs). The ground station (gs) generates a



MODULE IV
AUTHENTICATION OF DRONE WITH OTHER DRONE

matchless identity id_d^{new} by calculating $W^{new} = h(id_d^{new}||n_{gs})$, a master secret key pms^{new} for id_d^{new}, and calculates $X^{new} = h(id_d^{new}||pms^{new}||T^{new})$, where T^{new} means the registration timestamp for id_d^{new}. Also, *gs* creates sk_{gs}^{new}, pk_{gs}^{new} and computes $cert_{gs} = (sk_{gs}||id_d^{new})$. The *gs* stores $\{W^{new}, X^{new}, sk_d^{new}, pk_d^{new}, cert_d^{new}\}$in its database and injects $\{W^{new}, X^{new}, sk_{gs}^{new}, pk_{gs}^{new}, cert_{gs}^{new}\}$in the memory of the drone id_d^{new}. Now, the newly registered drone added dynamically, becomes a legitimate one and is fully authorized for delivering services securely.

E. DRONE REVOCATION/REISSUE PHASE

This is a much crucial phase of the protocol. In this phase, the revoked/departed drone's unique identities are recorded in a separate memory containing a list called ReL. The operator sitting for managing, supervising and monitoring all the activities of legal drones can add a secret key sk_i to ReL for departed drone. He/She then deletes the data of ReL using (sk_i, W_i, X_i, pms_i), whereas $W_i = h(id_{di}||n_i)$, $X_i = h(id_{di}||pms_i||T_i)$ and $pms_i = pk_i||sk_i$. Once a user becomes revoked, the ground station (gs) doesn't respond for any future request. The gs perform a revocation test using a record denoted by $sk_i \epsilon ReL$, gs calculates $A_i == (n_d \oplus id_d)$, $Z_d = h(cert_d||A_d)||n_d$ compares with the already stored values. If the ground station proves/matches $A_i?{=}A_d$ and $Z_i?{=}PBKDF$, it means the departed user record is still available, else, the tuple for sk_i has been successfully deleted/cancelled from the list and it is not authorized for future correspondence with the gs. When random nonce n_d and n_{gs} are generated there is also an acceptable timestamp T_1, T_3 and T_5 in each round trip of the proposed protocol which can consequently revoke a drone (either d_1 or d_2) after the predefined time threshold, which means each drone is in full control for renewing, when properly runs the protocol.

V. SECURITY ANALYSIS

The security analysis of the proposed protocol can be performed both formally and informally using ROM/ProVerif2.02 and assumptions. These are discussed as under:

A. SECURITY ANALYSIS USING ROM [13]

There are three entities involved in the proposed authentication protocol $\mathcal{P}$: the external user $\mathcal{U}$, drone (d) $\mathcal{V}$ and ground station (gs) $\mathcal{G}$. Upon executing $\mathcal{P}$, every entity has many instances to link with pk/sk or *pms* which is termed as oracle. Let $\mathcal{U}^k$ is the x^{th} instance of $\mathcal{U}$, $\mathcal{V}^k$ is the y^{th} instance of $\mathcal{V}$, and $\mathcal{G}^k$ is the z^{th} instance of $\mathcal{G}$. But I^k is considered to be the instance of all three participants i.e. $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{G}$, possibly, there exists three consequences of oracle i.e. accept, reject, $\perp$; accept means receiving message in an authentic manner, reject means accepting a wrong message and $\perp$ do nothing/no result. Before execution, $\mathcal{U}$ has $\{E_1, E_2, cert_d, fp_d\}$, $\mathcal{V}$ has $\{H_{d1}, msg_{d1}\}$and $\mathcal{G}$ has $\{F_3, cert_{gs}, T_3\}$and supposes these are in the memory in a secure manner.

Let the adversary $\mathcal{A}$ is having full control over the public network channel; he/she can initialize, terminate the session among the participants for violating the privacy by tracing and arbitrate the established session among them. By doing so, $\mathcal{A}$ can make these queries in oracle including i) $h(cert_d||A_d)||n_{gs} \oplus pk_{gs}$, ii) $id_d \oplus h(T_1||id_d||n_{gs})$, iii) $h(C \oplus cer_d)||cert_d||pk_{gs}$, iv) $h(id_d||cert_d||T_7)$, and v) $PBKDF(pms \oplus (n_d||n_{gs}) \oplus iter)$. $\mathcal{A}$ can also make Execute ($\mathcal{U}^x$, $\mathcal{V}^y$), Execute ($\mathcal{V}^y$, $\mathcal{G}^z$), Execute ($\mathcal{G}^z$, $\mathcal{V}^y$), and Execute ($\mathcal{V}^y$, $\mathcal{U}^x$) queries. Can also, reveal I^k query for identifying the known session key, Corrupt ($\mathcal{U}$) for capturing the arguments stored in the mobile device and Test (I^k) query for obtaining the shared session key SK.

However, each participants has its own unique identity that will be agreed on the development of session if and only if M_5 of d is equal to $M_5^{/}$ of gs, $E_1^*?{=}E_3$, $cert_{gs}?{=}cert_{gs}$ and $cert_d?{=}cert_d$. Same is the case in session shared keys computed by each participants i.e. $SK^{/}?{=}SK$ and $sk_{gs}?{=}k_d$. $\mathcal{A}$ has the probability to break the security of $\mathcal{P}$ by flipping a coin Ω, suppose $\mathcal{A}$ flip a coin and get $\Omega^{/}$ output, the advantage is:

$$Adv_{\mathcal{P}}^{Protocol}(\mathcal{A}) = \left|2Pr\left|\Omega = \Omega^{/}\right| - 1\right|$$

However, due to 160 bits random selection of key (pk_{gs}, pk_d, sk_{gs} and pk_d) by the ground control station for each session, $\mathcal{A}$ cannot compute it even though if he/she can attempt polynomial times. Therefore, the proposed authentication protocol is secure from all possible attempts of an adversary. Further, if the output of a hash oracle is $\mathrm{q}_{\mathrm{he}}^2/2^{\mathrm{ths}+1}$, $\mathrm{q}_{\mathrm{he}+1}^2/2_{\mathrm{and}}^{\mathrm{ths}+1}\mathrm{q}_{\mathrm{he}}^2/2^{\mathrm{ths}}$ then the maximum probability of collision among hash-output is $(\mathrm{q}_{\mathrm{send}}+\mathrm{q}_{\mathrm{receive}})^2/2(\mathrm{p}\text{-}1)$, we will get:

$$\frac{|Prob|Success_2| - Prob|Success_1| = q_{hs}^2 + q_{hs1}^2 + q_{hs2}^2}{q_{hs}^2 + 1} + \frac{(q_{send} + q_{receive})^2}{2(q-1)}$$

But, if the adversary calculates correct message without hash values, $\mathcal{A}$ either forge $\{\mathrm{E}_1, \mathrm{E}_2, \mathrm{cert}_\mathrm{d}, \mathrm{fp}_\mathrm{d}\}$by knowing $\mathrm{n_d}$, $\mathrm{id_d}$ and $\mathrm{n_d}\oplus\mathrm{cert_d}$, but $\mathcal{A}$ cannot find $\mathrm{cert_d}$, and he/she cannot check the internal secrets in $\{\mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_3, \mathrm{cert_d}, \mathrm{T}_5\}$. So, this query of an adversary also seems to be failed, or $\mathcal{A}$ forge $\{\mathrm{J}_1, \mathrm{J}_2, \mathrm{J}_3, \mathrm{T}_7\}$. He/she must be known $\mathrm{pk_{gs}}$, $\mathrm{cert_{gs}}$, $\mathrm{pk_d}$, $\mathrm{cert_d}$ and HMAC values which is not exists in the record of $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot succeed for obtaining useful information, as given as:

$$[Prob|Success_3 - Prob|Success_2|] \leq \frac{2q_{send} + 2q_{hs1}}{2^{l_{hs}}}$$

Similarly, if $\mathcal{A}$ desires to get session key SK, he/she can attempt for SK by calculating it using:

$$[Prob|Success_1 - Prob|Success_2|.] \leq q_{receive}\, Adv_{\mathcal{A}}^{PTA}(X_{GCS})$$

whereas PTA means polynomial times attempt, while W is adversary session key, if the probability is [1/D], then we get:

$$Prob\,|Success_3| = \frac{1}{2} + \max \frac{q_{hs1}}{2^{l_{hs}}} \frac{q_{send}}{|D|}$$

Now combine all the possible calculations done by an adversary for impersonation, masquerading legal peer(s), we get as shown at the bottom of the page.

B. PROVERIF2.02 SIMULATION

This section is conducted to prove the robustness, security, reachability and integrity of the protocol using a verification toolkit ProVerif2.02 [14]. Also, to confirm the secrecy, reachability, and authorization of the proposed authentication protocol, a widespread programming toolkit ProVerif2.02 [14] has been used. Using this tool, we first define communication channels, and variables used during protocol designing and timestamp. Also, different events, constraints, functions, and equations, then calculation performed on user, drone and ground station sides and become ready to runs the code as given at the end in appendix – A.

C. INFORMAL SECURITY ANALYSIS

The proposed protocol is a lightweight cryptographic method based on Hash Message Authentication Code (HMAC) in combination with Secure Hash Algorithm (SHA1) for integrity and authentication of message among drone and ground station [46]. Some facts about message authentication are as under:

i. Sending Peer: HMACSHA1, random nonce *n*, identity, timestamp T, and original message m.
ii. Receiving Peer: Extracts random nonce *n*, hash-values and matching algorithm (Δ) for index finding, and computes original message if valid, accept, else, discard.

This means that HMACSHA1 is a keyed hash algorithm derived from the SHA1 hash function and used as a hash-based message authentication code. The HMAC method combines a secret key with message data, hashes the result with the hash function, combines the hash value with the private key once more, and then applies the hash function a third time. The length of the output hash is 160 bits. The following pseudocode demonstrates the working procedure of HMACSHA1.

Keeping in view these merits of HMACSHA1, the pragmatic illustration for the proposed protocol is demonstrated as under:

1) STOLEN VERIFIER ATTACK

There is no storage table in the ground station for password that could yield a chance for an adversary to capture it and later masquerade the ground station for wrong decision. Similarly, the session key computed previously for computation among all the peers is in encrypted form. If an adversary can steal it, they cannot figure out internal credentials from it. After the completion of the registration of the drone and gs their secrets are deleted from the memory of the certificate authority. Therefore, these secret values are not available to

$$\begin{aligned}
Adv_{\mathfrak{P}}^{protocol}(\mathcal{A}) &= \mathrm{Prob}|Success_0| - 1 \\
&= 2\,|Prob|\,Success_0\,| - Prob|\,Success_4| + \max\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\} \\
&\leq 2\left(Prob\,|Success_0| - Prob\,|Success_4| + \max\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\}\right) \\
&\leq 2\left(\leq 2\left(\begin{matrix} |\,Prob\,[|\,Success_1| - Prob|Success_2]\,| \\ +Prob\,[|\,Success_3| - Prob|Success_4] \end{matrix}\;|+\max\left\{\tfrac{q_{h1}}{2^{l_{hs}}}, \tfrac{q_{\mathrm{send}}}{|D|}\right\}\right)\right) \\
&\leq \frac{q_{hs}^2 + q_{hs1}^2 + q_{hs2}^2}{2^{l_{hs}}} + \frac{(q_{send} + q_{receive})^2}{2\,(q-1)} + 2q_{receive}.Adv_{\mathcal{A}}^{PTA}\,(X_{GCS}) + 2\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\}
\end{aligned}$$

```
Working Procedure of HMACSHA1
Start
function HMACSHA1
input:  key, message, and hash function
            blockSize: Integer
            outputSize: Integer
            if (length(key) >blockSize) then
            {
            key ← hash(input of key)
            if (length(key-input) <blockSize) then
                        {
            key ← Pad(key-input, blockSize)
            h[key_pad] ← key xor[blockSize]
            h(key_pad] ← key xor [blockSize]
            }
return hash(key_pad ||hash(key_pad ||message))
                        }
End
```

any authorized party. Hence such attacks are not feasible on the proposed scheme.

2) PRIVILEGED INSIDER ATTACK

In the proposed protocol, any type's identities are not transmitted openly with the certificate authority (an operator or manufacturing company). It is communicated securely in either concatenation with a nonce or by using collision-free one-way hash code. An operator or manufacturing company (CA) cannot find it in an accessible format. They also cannot figure out any other credentials for launching any future attack. The Insider-Threat Programme [47], which the CA is already equipped with, can trap and revoke any insider threat to make it free of sabotage, espionage, theft, and fraud. Therefore, the proposed model prevents unauthorized users to get authenticated and use the system resources.

3) RESISTANT TO TRACEABILITY ATTACK

Upon starting session, a Drone's nonce n_d, ground station nonce n_{gs} have been extracting randomly along with time stamp T_1, T_3 and T_5 and exchanged with each message duets M_1, M_2, M_3 and M_4 are categorical for each session. The attacker cannot trace two different sessions of ground station, drone or user. Therefore, our protocol shows strong resistance to traceability attack.

4) EPHEMERAL-SECRET-LEAKAGE (ESL) RESISTANCE

If an attacker come to have obtained the ephemeral nonce of drone n_d using ESL attacks [48], the adversary still needs to solve the $E_1 = id_d \oplus h(T_1 \| id_d \| n_d)$ and $E_2 = A_d^* \oplus h(id_d \| T_1 \| cert_d)$. Similarly, if they obtain n_{gs}, needs to $F_2 = h(cert_d \| A_d) \| (n_{gs} \oplus pk_{gs})$ and $F_3 = Enc_{skgs}((n_d \oplus n_{gs}))$. In the next round-trip, if the attacker recovers pms, let suppose, he/she has to pass $G_2 = h(C \oplus cer_d) \| cert_d \| pk_{gs}$ and $G_3 = h(id_d \| pk_{gs} \| G_2 \| T_5)$. Also, if they obtain pk_d or pk_{gs} sk_d or sk_{gs}, he/she has to solve $I_2 = h(h(id_d \| pk_d) \| cert_d \| pk_{gs})$, $I_3 = h(id_d \| pk_{gs} \| G_2 \| T_5)$, $J_2 = h(pms \| cert_d \| T_7)$ and $J_3 = Enc_{pkd}((pms) \| sk_{gs})$. So far without knowing the secret values of ground-statin (gs) or drone (d), adversary cannot succeed for computing exact values. Therefore, the proposed protocol withstands ESL attack.

5) DENIAL-OF-SERVICE ATTACK (DoS)

In the login and authentication phase of the proposed protocol a verification of password-based-key-derivation function with the already saved values $PBKDF^*? = PBKDF$, after successful confirmation onward computation performed, else, discarded. Similarly, in the 2nd receiving peer, confirms E_1^* with the received E_1 $(E_1^*?=E_1)$, which is not validated due to random nonce n_d and n_{gs}. Therefore, the proposed protocol shows resilience to DoS attack.

6) RESISTS REPLAY ATTACK

Suppose an attacker attempts to capture messages M_1, M_2, M_3 and M_4 in the login and authentication phase of the proposed protocol for launching replay attack some other time. But their capturing of messages fails due to timestamp and random nonce n_d, n_{gs} in it. Let an attacker, transmits any message of login in authentication protocol towards a receiver, first the peer check the time and validates the message, which is not possible in the proposed authentication protocol. Therefore, it resists replay attack.

7) SAFE AGAINST MAN-IN-THE-MIDDLE ATTACK

If an attacker attempt to modifies any message (M_1, M_2, M_3 or M_4) in the communication line. They cannot do so, due to lack of knowledge on $cert_d$, $cert_{gs}$ and other internal credentials. Also, each message is dynamic which is different in each session and has random nonce n_d, n_{gs} and time stamps. Thus, our protocol is safe against man-in-the-middle attack.

8) PREVENTION OF MALICIOUS USER

If an attacker shows themselves as a legitimate user and tries to communicate with the other peer, they cannot succeed due to not being registered with the certificate authority and thus failed for computing session shared key sk. No one can pretend themselves as a legitimate peer for sending false message towards another peer. Therefore, the proposed authentication protocol is safe against malicious user.

9) PREVENTION OF SPOOFING ATTACK

When a drone goes out of service, it must have the capability to handover the control to other drone, but when the network is interrupted, the communication is carried out through GPS signal. If an attacker tries to send fake message using GPS spoofing, they failed for obtaining session key due to complex computation process. They also cannot impersonate the ground station as well as another user. Thus, our protocol prevents spoofing attack.

10) RESISTS AGAINST CLOGGING ATTACK

If an attacker desires to launch a clogging attack by sending a fake message $M_1 = \{E_1, E_2, cert_d, fp_d\}$towards a ground station (gs). He/She has to first generates a random nonce n_d, and timestamp T_1 and simulates by calculating PBKDF= $h(cert_d \| A_d) \| n_d$, $E_1 = id_d \oplus h(T_1 \| id_d \| n_d)$ and $E_2 = A_d^* \oplus h(id_d \| T_1 \| cert_d)$. He/She can also $A_d^* = (n_d \oplus id_d)$ by flipping a coin to win $A_d^* ?= A_d$ or $A_d^* \neq A$ and $PBKDF^* ? = PBKDF$ or $PBKDF^* ? \neq PBKDF$. But doing such a difficult calculations require a drone's identity (Id_d), $cert_d$ and the previously computed value $A_d = (n_d \oplus id_d)$. Similarly, if the attacker transmits $M_3 = \{G_1, G_2, G_3, cert_d, T_5\}$towards ground station (gs), he/she has to passed from $G_1 = id_d \oplus h(n_{gs} \| pk_d)$, $G_2 = h(C \oplus cer_d) \| cert_d \| pk_{gs}$ and $G_3 = h(id_d \| pk_{gs} \| G_2 \| T_5)$. Doing so, the attacker needed 160 bits public key of ground-station (pk_{gs}), drone's identity Id_d and 160 bits drone's public-key (pk_d) in advance. Also, attacker must need timestamp T_5. As, the old credentials are available to the attacker (it is encrypted form), they couldn't figure out these credentials from it. The proposed protocol cannot detect clogging attack in both cases. Because, the attacker couldn't pass from $E_1^* ?= E_1$ and $G_3 ?= I_3$ authentication check. Therefore, the proposed protocol strongly resists clogging attack.

11) SECURITY AGAINST DRONE'S CAPTURE ATTACK

Due to the addition of drone dynamically to the network at any time, it is necessary to evaluate the proposed mechanism for drone capture attack. Let an adversary capture a drone, and extract n_d, id_d, $cert_d$, 160 bits secret key (sk_d) and public-key (pk_d) etc. They must make necessary arrangements for the calculation of two 160 bits long public-key which needs months for doing such calculation. Similarly, adversary must compute $A_d = (n_d \oplus id_d)$, and PBKDF= $h(cert_d \| A_d) \| n_d$; and figure out the stored credentials$\{B, C, pk_d, h(\cdot)\}$; computes $B = h(id_d || PBKDF)$, and $C = h(id_d || sk_{gs}) \oplus PBKDF$ and identifying $D = \{B, C, pk_d, h(\cdot)\}$ parameters. All these calculations in our protocol restrict attacker to deploy drone in the network, and if deployed, for example, cannot establish secure session with ground station due to several checks. Therefore, by capturing d, the attacker cannot settle a session with gs and others; it cannot agree to negotiate at any stage with gs or d. The compromised drone does not result in ensuring secure communications with ground-station (gs) and d. As a result, our protocol is unconditionally secure against drone capture attack.

12) RESISTANCE AGAINST BRUTE FORCE ATTACK

Suppose there exist two categories of an adversary, i.e. one attempt for the secret key hacking of ground station (gs) called type-I adversary. At the same time, another one tries to hack the private key or password-based-key-derivation function (PBKDF) of different drones is then said to be the adversary of type-II. And we have four round trips for secure session key generation. A security model for such a real-time intelligence system is safe if no probabilistic polynomial-time adversary of either type-I, type-II or both exists. In this connection, an adversary of any type is given a security parameter k by a challenger C, C runs the setup algorithm and exchanges the output/result with the adversary. The adversary can obtain the secret key in the first round if he/she is of type-II and must confirm $PBKDF = h(cert_d || A_d) || n_d$ with $PBKDF^* = h(cert_d \| A_d) \| n_d$ $(PBKDF? = PBKDF^*)$. This is possible only when maximum access power is granted to them. But our security protocol is much secure because the said calculation is needed that the adversary must know the drone's certificate $cert_d$, drone's nonce n_d and drone's identity Id_d, which is not possible. Also needs much time to compute. As we have defined the time threshold in each round trip of the protocol. The system discarded it by considering it is an outdated message or potential replay attack for any fake request. Therefore, the proposed protocol is secure against the brute force attack of an adversary.

Finally, the protocol presented in this article is suitable for two parties; it is a bit weaker when the number of drones increased; other limitations are listed as under:

i. Let's look at some potential issues with the Hash-based Message Authentication Code (HMACSHA1) based security protocols that employ a symmetric key, the sender and receiver both use the same key, how will the key be securely exchanged between the sender and receiver? Still a challenge for the researchers.

ii. Similarly, if we share the symmetric key with several parties, how would the receiver know that the message was prepared and sent by the sender and not by the receivers? There is a risk that one of the receivers will send false messages. Also another challenge for the researchers?

VI. PERFORMANCE EVALUATION

In this section of the paper, we analyze the performance of our protocol in terms of storage overheads, computation, and communication costs by considering the already experiment done by [32], [50], and [51] for various cryptographic operations. They have conducted an experiment using cell-phone namely Samsung Galaxy S5 of Quad-core 2.45G processor, 2GB of RAM and Android Operating System of version 4.4.2. They also used a Dell PC of CPU 2.90GHz, 4GB RAM and Windors8.1 OS. The results of ''Dell PC'' are considered for the ground station (gs) and ''Samsung Galaxy S5'' for the drone (d).

A. STORAGE OVERHEADS ANALYSIS

The storage overheads analysis of the proposed protocol can be described on the basis of [15], [32], [50] and [51] and as given as: Nonce for drone and ground station n_d, n_{gs} occupy 160 bits of memory space, identities id_d and id_{gs} 64 bits, PBKDF-SHA1 160 bits, timestamp T_2, and T_4 56 bits, secrete keys (sk_d drone secret key, sk_{gs} ground station secret key pk_d drone public key, pk_{gs} ground station public key, k_d drone session key, k_{gs} ground station session key) 32 bits,

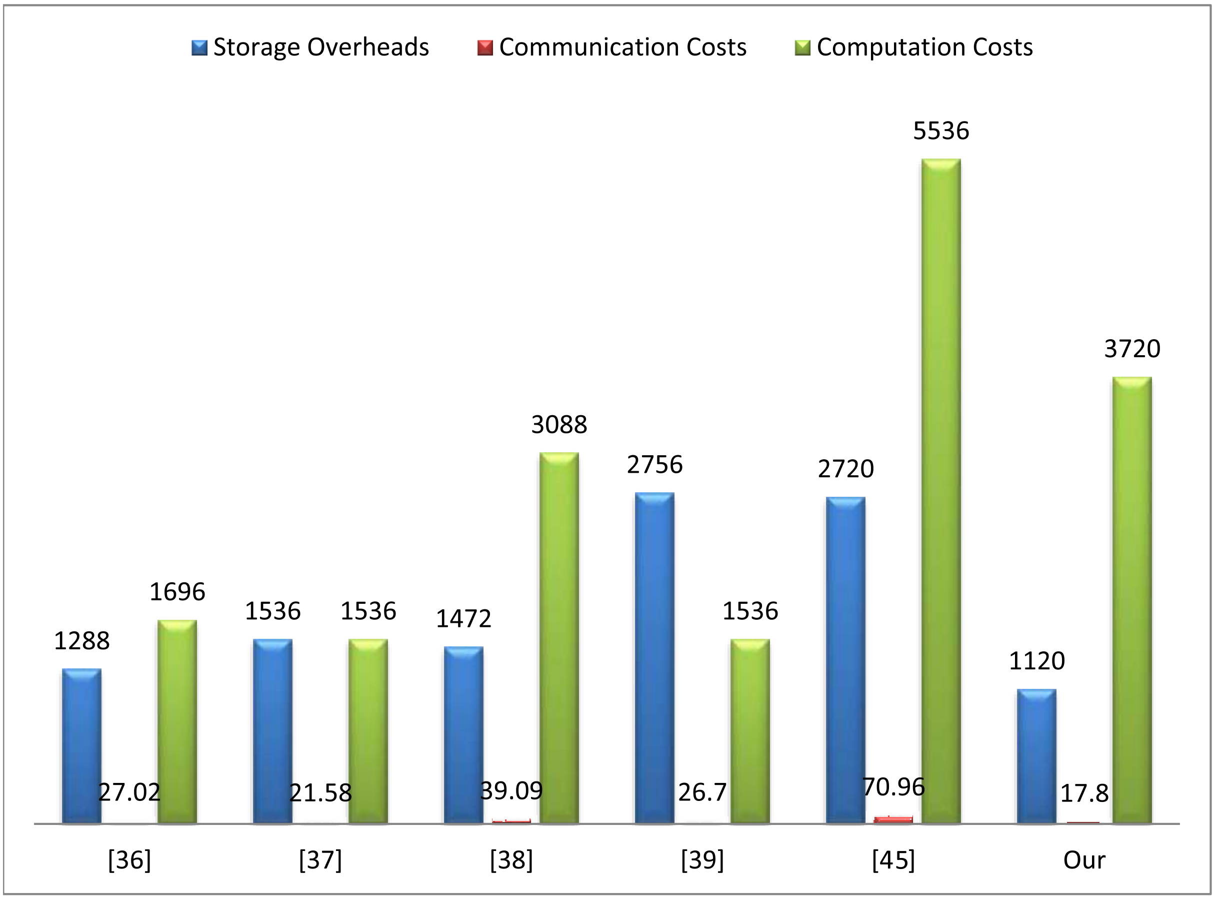


FIGURE 2. Comparison analysis graph.

TABLE 3. Storage overheads analysis in bits.

Argument	Size in bits	Argument	Size in bits
Random number (nonce)	160	Identity	64
Timestamp	56	Secret key	32
HMAC	256	Encryption	512
Decryption	512	PBKDF	160
Certificates	64	**Total**	**1120**

TABLE 4. Computation time complexity in milliseconds.

Peer	Operations	Total
d	$1T_{Mul}+7T_H+1T_{Nonce}+8T_{XOR}+2T_{Dec}$.	8.6475
gs	$6T_H+2T_{Nonce}+7T_{XOR}+2T_{Enc}$	9.1464
Total cost in ms during session key computation		17.7939

TABLE 5. Communication cost in bits.

Message	Arguments Passed	Values in Bits	Total
M_1	E_1, E_2, $cert_d$, fp_d	512 + 256 + 64 + 32	864 Bits
M_2	F_3, $cert_{gs}$, T_3	512 + 64 + 56	632 Bits
M_3	G_1, G_2, G_3, $cert_d$, T_5	512 + 256 + 256 + 64 + 56	1144 Bits
M_4	J_1, J_2, J_3, T_7	512 + 256 + 256 + 56	1080 Bits
		Total	**3720 Bits**

TABLE 6. Comparison analysis.

Protocol	Storage cost	Computation cost	Communication cost
[36]	1288	27.02ms	1696
[37]	1536	21.58ms	1536
[38]	1472	39.09ms	3088
[39]	2756	26.70ms	1536
[45]	2720	70.96ms	5536
Our	1120	17.79ms	3720

collision free on-way HMACSHA1 functions 256 bits, drone certificate, $cert_d$ ground station certificate $cert_{gs}$ 64 bits and encryption/decryption functions 512 bits. It is to mention that encryption/decryption function is not used in the registration phase of the proposed protocol, so never consider it in the storage overheads analysis. Therefore, the storage overheads in the registration phase is considered for those arguments/credentials whose values are stored in the memory like id_d, id_{gs}, n_d, n_{gs}, $cert_d$, $cert_{gs}$, pk_{gs}, sk_{gs}, pk_d, sk_d, PBKDF-SHA and HMAC-SHA1 $(64 + 64 + 160 + 160 + 64 + 64 + 32 + 32 + 32 + 32 + 160 + 256)$ is 1120 bits as shown in Table 3.

B. COMPUTATION COST ANALYSIS

According to [15], [32], [50], and [51] computation time complexity for different operations performed during session key computation are given as:

- T_H: computation time for collision free one-way hash function (HMAC) is ≈ 0.0023 ms.
- T_{Enc} execution time for Encryption Function is $\approx$ 3.85 ms.

TABLE 7. Security features functionalities analysis.

Feature→ Schemes↓	SFF1	SFF2	SFF3	SFF4	SFF5	SFF6	SFF7	SFF8	SFF9	SFF10	SFF11	SFF12	SFF13	SFF14
[36]	✘	✓	✘	✘	✘	✓	✓	✓	✓	✘	✓	✘	✘	✘
[37]	✘	✓	✓	✓	✘	✘	✘	✓	✓	✘	✘	✘	✘	✓
[38]	✘	✘	✘	✓	✓	✓	✘	✘	✘	✓	✘	✘	✓	✓
[39]	✓	✓	✘	✘	✘	✘	✓	✓	✘	✘	✘	✘	✘	✘
[45]	✘	✘	✘	✓	✓	✓	✘	✓	✘	✘	✘	✘	✘	✘
Our	✘	✘	✘	✘	✘	✘	✘	✓	✘	✘	✘	✘	✘	✘

- T_{Dec} execution time for Decryption Function is ≈ 3.85 ms.
- T_{XOR} CPU time required for bitwise exclusive-OR operation is ≈ 0.0288 ms.
- T_{Nonce} CPU execution time needed for the extraction of random nonce (n) is ≈ 0.539 ms.
- T_{Mult} Multiplication execution time complexity is ≈ 0.0171 ms.

Keeping in view, the computation cost for drone (d) and ground-station (gs) are $1T_{Mul}+7T_H+1T_{Nonce}+8T_{XOR}+2T_{Dec}$ and $6T_H+2T_{Nonce}+7T_{XOR}+2T_{Enc}$ respectively which is for the whole process/system during session key computation is $1T_{Mul}+13T_H+3T_{Nonce}+15T_{XOR}+2T_{Enc}+2T_{Dec}$ and as shown in Table 4.

C. COMMUNICATION COST ANALYSIS

The communication cost for the proposed authentication protocol in the login and authentication phase means the transmission of messages among drone and ground station are as $M_1 = \{E_1, E_2, cert_d, fp_d\}$of cost $\{512 + 256 + 64 + 32\}=$ 864 bits, $M_2 = \{F_3, cert_{gs}, T_3\}= \{512+64+56\}=$ 632 bits, $M_3 = \{G_1, G_2, G_3, cert_d, T_5\}=\{512 + 256 + 256 + 64 + 56\}=$ 1144 bits and $M_4 = \{J_1, J_2, J_3, T_7\}= \{512 + 256 + 256 + 56\}=$ 1080 bits. Therefore, the total communication cost of the proposed authentication protocol is 3720 bits as shown in Table 5.

D. COMPARISON ANALYSIS

In this section, we compare the proposed protocol with [45], [41], [42], [43] and [9] in terms of storage overheads, computation and communication cost. The result shows that the proposed protocol is better than that of Wazid *et al.* [36], Srinivas *et al.* [37], Singh *et al.* [38], Zhang *et al.* [39] and Cho *et al.* [45], as show in Table 6. The comparative study can also be represented graphically in Fig 2.

Similarly, comparing the proposed protocol for different attacks with [36], [37]–[39] and [45], we will get result which is shown in Table 7 along with our protocol, which is much stronger than these protocols. In the given table impersonation Attack is denoted by SFF1, Anonymity-violation SFF2, traceability-attack SFF3, Outdated Data Transmission SFF4, privileged insider attack SFF5, Stolen-verifier attack SFF6 and Spoofing Attack is SFF7, Mutual Authentication SFF8, DoS Attack SFF9, Replay Attack SFF10, Man-in-Middle Attack SFF11, Masquerade Attack SFF12, Clogging Attack SFF13, and Drone Capture Attack SFF14.

VII. CONCLUSION

In this paper, a lightweight authentication protocol for IoD is presented. We have used simple hash cryptographic functions for protecting data from a strong adversary. This protocol is free of the privileged insider, stolen-verifier attacks. It doesn't have the outdated data transmission flaw. The timestamp identifies each transmitted message with a pre-defined time threshold before communicating with the ground station, which leads to dynamicity. Similarly, a malicious node cannot misguide a drone for a wrong decision. In the end, the security analysis and performance evaluation result shows that the proposed protocol is much lightweight, secure, and ensures perfect forward secrecy. Therefore, it can be used for implementation in a real-world IoD environment.

In future, we have planned to implement the proposed robust and lightweight security protocol for IoD deployment military drones (reconnaissance drone and attacking drone) and examine for warfare battlefield using NS3 simulation.

APPENDIX – A

Proverif2.02 Simulation Code

```
(* ——— CHANNELS ———-*)
free ChSec:channel [private]. (*secure channel between GS and CA*)
free ChPub:channel. (*public channel between d, GS*)
(*——— CONSTANTS AND VARIABLES ———*)
free skgs:bitstring [private].
free skd:bitstring [private].
free idd:bitstring.
free nd:bitstring.
free kd:bitstring [private].
free certd:bitstring.
free ngs:bitstring.
free pkd:bitstring.
free pkgs:bitstring.
free fpd:bitstring.
free pms:bitstring.
free C:bitstring.
free Ad:bitstring.
free iter:bitstring.
free certgs:bitstring.
free T1:bitstring.
free T2:bitstring.
free T3:bitstring.
free T4:bitstring.
free T5:bitstring.
free T7:bitstring.
free IDd:bitstring.
free IDGS:bitstring.
(*——-QUERIES——*)
query attacker(kd).
query id:bitstring; inj-event(end_d(IDd)) == >inj-event(start_d(IDd)).
query id:bitstring; inj-event(end_GS(IDGS)) ==>inj-event(start_GS(IDGS)).
(*=====*EVENTS*=====*)
event start_d(bitstring).
event end_d(bitstring).
event start_GS(bitstring).
event end_GS(bitstring).
(*========CONSTRUCTORS=======*)
fun h(bitstring):bitstring.
fun Concat(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun Encskgs(bitstring):bitstring.
fun Encpkd(bitstring):bitstring.
fun Decpkgs(bitstring):bitstring.
fun Decskgs(bitstring):bitstring.
fun PBKDF(bitstring):bitstring.
(*======EQUATIONS=======*)
equation forall a:bitstring, b:bitstring;
XOR(XOR(a,b),b)=a.
(*—————LOGIN AND AUTHENTICATION——————-*)
(*————-DRONE AND GS—————*)
let pd=
event start_d(IDd);
let Ad=XOR(nd,idd) in
if Ad=Ad then
let PBKDF=h(Concat(certd,Ad)) in
if PBKDF= PBKDF then
let E1=XOR(idd,h(Concat(T1,(idd,nd)))) in
let E2=XOR(Ad,h(Concat(idd,(T1,certd)))) in
out(ChPub,(E1,E2,certd,fpd));
in(ChPub,(F3:bitstring,certgs:bitstring,T3:bitstring));
let G1=XOR(idd,h(Concat(ngs,pkd))) in
let G2=h(Concat((XOR(C,certd)),(certd,pkgs))) in
let G3=h(Concat(idd,(pkgs,G2,T5))) in
out(ChPub,(G1,G2,G3,certd,T5));
in(ChPub,(J1:bitstring,J2:bitstring,J3:bitstring,
T7:bitstring));
let pms=Concat(skd,(pkd,pms)) in
let L1=h(Concat(idd,(certd,T7))) in
if L1=J2 then
let kd=XOR(pms,(nd,ngs,iter)) in
event end_d(IDd)
else
0.
(*————-GROUND STATION—————*)
let pGS=
event start_GS(IDGS);
in(ChPub,(E1:bitstring,E2:bitstring,certd:bitstring,
fpd:bitstring));
let F1=XOR(E1,(h(Concat(T1,(idd,ngs))))) in
let F2=h(Concat(certd,(Ad,ngs))) in
let F3=Encskgs(Concat((nd,ngs),T2)) in
out(ChPub,(F3,certgs,T3));
in(ChPub,(G1:bitstring,G2:bitstring,G3:bitstring,
certd:bitstring,T5:bitstring));
let I1=XOR(idd,(h(Concat(pkd,T5)))) in
let I2=h(Concat((h(idd)),(certd,pkgs))) in
let I3=h(Concat(idd,(pkgs,G2,T5))) in
if G3=I3 then
let J1=Concat(T7,(XOR(ngs,nd))) in
let pms=XOR(skd,(pkd,certgs)) in
let J2=h(Concat(pms,(certd,T7))) in
let J3=Encpkd(Concat(pms,skgs)) in
out(ChPub,(J1,J2,J3,T7));
event end_GS(IDGS)
else
0.
process ((!pGS) |(!pd))
```

After successfully running the code, the following result is displayed which shows that the session shared key is much secure from any attacker that confirms the confidentiality, authorization and reachability of the protocol.

```
————————————————————————————————
Completing equations...
Completing...
Starting query not attacker(kd[])
RESULT not attacker(kd[]) is true.
Completing...
inj-event(start_d(IDd[])) is true.
Completing...
RESULT inj-event(end_GS(IDGS[])) ==>inj-event
(start_GS(IDGS[])) is true.
————————————————————————————————
Verification summary:
Completing...
Query not attacker(kd[]) is true.
Completing...
Query inj-event(end_d(IDd[])) ==>inj-event
(start_d(IDd[])) is true.
Completing...
Query inj-event(end_GS(IDGS[])) ==>inj-event
(start_GS(IDGS[])) is true.
————————————————————————————————
```

REFERENCES

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

[2] P. K. Valavanis, P. Kimon, and J. V. George, *A Handbook of Unmanned Aerial Vehicles*, vol. 1. Dordrecht, The Netherlands: Springer, 2015.

[3] G. Tuna, V. M. Tarik, G. V. Kayhan, G. Cagri, and E. H. Erturk, "Unmanned aerial vehicle-aided wireless sensor network deployment system for post-disaster monitoring," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2012, pp. 298–305.

[4] M. W. Lewis and E. Crawford, "Drones and distinction: How IHL encouraged the rise of drones," *Georgetown J. Int. Law*, vol. 44, no. 3, pp. 1127–1166, 2013.

[5] A. Bello, "Radio frequency toolbox for drone detection and classification," M.S. thesis, Elect./Comput. Eng., Old Dominion Univ., Norfolk, VA, USA, 2019, doi: 10.25777/9gkm-jd54.

[6] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun, "FANET: Communication, mobility models and security issues," *Comput. Netw.*, vol. 163, Nov. 2019, Art. no. 106877.

[7] N. A. Khan, N. Ali, N. Z. Jhanjhi, S. N. Brohi, and A. Nayyar, "Emerging use of UAV's: Secure communication protocol issues and challenges," in *Drones in Smart-Cities*, 2020, pp. 37–55, doi: 10.1016/B978-0-12-819972-5.00003-3.

[8] F. Abdi, C.-Y. Chen, M. Hasan, S. Liu, S. Mohan, and M. Caccamo, "Preserving physical safety under cyber attacks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6285–6300, Aug. 2019.

[9] Z. Ali, S. A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, and Y. B. Zikria, "A clogging resistant secure authentication scheme for fog computing services," *Comput. Netw.*, vol. 185, Feb. 2021, Art. no. 107731.

[10] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, and N. Kumar, "Amassing the security: An ECC-based authentication scheme for Internet of drones," *IEEE Syst. J.*, early access, Mar. 1, 2021, doi: 10.1109/JSYST.2021.3057047.

[11] K. P. Valavanis, and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*. Dordrecht, The Netherlands: Springer, 2015.

[12] D. Dolev and A. C. Yao, "On the security of public key protocols—An information theory," *IEEE Trans.*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[13] M. Bellare, A. Boldyreva, and A. Palacio, "An uninstantiable random-oracle-model protocol for a hybrid-encryption problem," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 171–188.

[14] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif 2.00: Automatic cryptographic protocol verifier, user manual and tutorial," *Version*, pp. 5–16, May 2018.

[15] H. H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1005–1023, Oct. 2013.

[16] D. S. Xing, Z. F. Cao, and X. L. Dong, "Identity based signature scheme based on cubic residues," *Sci. China Inf. Sci.*, vol. 54, no. 10, pp. 2001–2012, 2001. [Online]. Available: https://link.springer.com/content/pdf/10.1007/s11432-011-4413-6.pdf

[17] D. He, M. Tian, and J. Chen, "Insecurity of an efficient certificateless aggregate signature with constant pairing computations," *Inf. Sci.*, vol. 268, pp. 458–462, Jun. 2014.

[18] N. O. Viet, N. Quoc, and W. Ogata, "Certificateless aggregate signature protocols with improved security," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 98, no. 1, pp. 92–99, 2015.

[19] Zhong, S. Hong, S. Han, J. Cui, J. Zhang, and Y. Xu, "Privacy-preserving authentication scheme with full aggregation in VANET," *Inf. Sci.*, vol. 476, pp. 211–221, Feb. 2019.

[20] 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.

[21] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2003, pp. 416–432.

[22] M. Lysyanskaya and S. Reyzin, "Sequential aggregate signatures from trapdoor permutations," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 74–90.

[23] J. Herranz, "Deterministic identity-based signatures for partial aggregation," *Comput. J.*, vol. 49, no. 3, pp. 322–330, Dec. 2005.

[24] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Berlin, Germany: Springer, 2006, pp. 207–222.

[25] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum, "Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 276–285.

[26] Haque, M. Samsul, and M. U. Chowdhury, "A new cyber security framework towards secure data communication for unmanned aerial vehicle (UAV)," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.* Cham, Switzerland: Springer, 2017, pp. 113–122.

[27] S. Benzarti, B. Triki, and O. Korbaa, "Privacy preservation and drone authentication using ID-based signcryption," in *SoMeT*. Hammam Sousse, Tunisia: Univ. Sousse, MARS Lab, ISITCom, 2018, pp. 226–239.

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

[29] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of drone environment," *IEEE Access*, vol. 8, pp. 155645–155659, 2020.

[30] C. Pu and Y. Li, "Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN)*, Jul. 2020, pp. 1–6.

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

[32] M. Nikooghadam, H. Amintoosi, S. H. Islam, and M. F. Moghadam, "A provably secure and lightweight authentication scheme for Internet of drones for smart city surveillance," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 101955.

[33] M. O. Ozmen and A. A. Yavuz, "Dronecrypt—An efficient cryptographic framework for small aerial drones," 2019, *arXiv:1903.12301*. [Online]. Available: https://arxiv.org/abs/1804.00742

[34] Y. Li, X. Du, and S. Zhou, "A lightweight identity authentication scheme for UAV and road base stations," in *Proc. Int. Conf. Cyberspace Innov. Adv. Technol.*, New York, NY, USA, Dec. 2020, pp. 54–58.

[35] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, Apr. 2016.

[36] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[37] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[38] J. Singh, A. Gimekar, and S. Venkatesan, "An efficient lightweight authentication scheme for human-centered industrial Internet of Things," *Int. J. Commun. Syst.*, p. e4189, Nov. 2019.

[39] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for Internet of drones," *Comput. Commun.*, vol. 154, pp. 455–464, Mar. 2020.

[40] 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.

[41] 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 Syst. J.*, early access, Dec. 10, 2020, doi: 10.1109/JSYST.2020.3036425.

[42] 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 Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4425–4435, Aug. 2020.

[43] S. A. Chaudhry, H. Alhakami, A. Baz, and F. Al-Turjman, "Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure," *IEEE Access*, vol. 8, pp. 101235–101243, 2020.

[44] S. U. Jan and F. Qayum, "Mitigating the desynchronisation attack in multiserver environment," *IET Commun.*, vol. 14, no. 13, pp. 2210–2221, Aug. 2020.

[45] G. Cho, J. Cho, S. Hyun, and H. Kim, "SENTINEL: A secure and efficient authentication framework for unmanned aerial vehicles," *Appl. Sci.*, vol. 10, no. 9, p. 3149, Apr. 2020.

[46] S. A. Chaudhry, "Correcting 'PALK: Password-based anonymous lightweight key agreement framework for smart grid,'" *Int. J. Electr. Power Energy Syst.*, vol. 125, Feb. 2021, Art. no. 106529.

[47] M. D. Guido and M. W. Brooks, ''Insider threat program best practices,'' in *Proc. 46th Hawaii Int. Conf. Syst. Sci.*, Wailea, HI, USA, Jan. 2013, pp. 1831–1839.

[48] C.-L. Liu, W.-J. Tsai, T.-Y. Chang, and T.-M. Liu, ''Ephemeral-secret-leakage secure ID-based three-party authenticated key agreement protocol for mobile distributed computing environments,'' *Symmetry*, vol. 10, no. 4, p. 84, Mar. 2018.

[49] Y.-P. Liao and S.-S. Wang, ''A secure dynamic ID based remote user authentication scheme for multi-server environment,'' *Comput. Standards Interfaces*, vol. 31, no. 1, pp. 24–29, Jan. 2009.

[50] L. Wu, J. Wang, K.-K.-R. Choo, and D. He, ''Secure key agreement and key protection for mobile device user authentication,'' *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 319–330, Feb. 2019.

[51] B. Bera, D. Chattaraj, and A. K. Das, ''Designing secure blockchain-based access control scheme in IoT-enabled Internet of drones deployment,'' *Comput. Commun.*, vol. 153, pp. 229–249, Mar. 2020.

SAEED ULLAH JAN received the M.Phil. degree in network security from the University of Malakand, in 2016, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and IT. He is currently working as a Lecturer in computer science with Higher Education Achieves & Libraries Department, Government of Khyber Pakhtunkhwa, Pakistan. He is also working as a Coordinator for 09 BS Disciplines, Government College Wari (Dir Upper), a far-flung remote area of the province, where most of the youngsters have no access to universities/institutions for higher education. Furthermore, he has conducted research in many areas, including green computing, distributed computing, privacy-preserving parallel computation, and drone security and authentication. He has published over ten research articles in prestigious conferences and journals and written an introductory Book in Computer Science for beginners. For the academic year 2019–20, the Government of Khyber Pakhtunkhwa awarded him the Best Teacher Award out of 11000 college teachers in 309 public sector colleges in the province.

FAWAD QAYUM received the Ph.D. degree from the University of Leicester, U.K., in 2012. He is currently working as an Assistant Professor with the Department of Computer Science and IT, University of Malakand, Pakistan. His research interests include model-driven software evolution and re-engineering, quality controlled refactoring using graph transformation systems, image processing, and network security. He also did research in applied cryptography and identification authentication of 5G-enabled IoT.

HABIB ULLAH KHAN (Member, IEEE) received the Ph.D. degree in management information systems from Leeds Beckett University, U.K. He is currently working as a Professor of information systems with the Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Qatar. He has more than 20 years of industry, teaching, and research experience. He is an Active Researcher, and his research work has published in leading journals of the MIS field. His research interests include IT security, online behavior, IT adoption in supply chain management, Internet addiction, mobile commerce, computer-mediated communication, IT outsourcing, big data, cloud computing, and e-learning. He is a member of leading professional organizations, such as DSI, SWDSI, ABIS, FBD, and EFMD. He is a reviewer of leading journals of his field and also working as an editor for some journals.