IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 August 2022, accepted 29 August 2022, date of publication 5 September 2022, date of current version 15 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204271

RESEARCH ARTICLE

A Verifiably Secure ECC Based Authentication Scheme for Securing IoD Using FANET

SAEED ULLAH JAN[1,2], IRSHAD AHMED ABBASI[3,5], (Member, IEEE), FAHAD ALGARNI[4], AND ADNAN SHAHID KHAN[5], (Senior Member, IEEE)

[1]Department of Computer Science, Government College Wari (Dir Upper), Wari, Khyber Pakhtunkhwa 18200, Pakistan
[2]Department of Computer Science and IT, University of Malakand, Chakdara 18800, Pakistan
[3]Department of Computer Science, Faculty of Science and Arts Belqarn, University of Bisha, Sabt Al-Alaya 61985, Saudi Arabia
[4]Faculty of Computing and Information Technology, University of Bisha, Bisha 67714, Saudi Arabia
[5]Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan 94300, Malaysia

Corresponding author: Irshad Ahmed Abbasi (aabasy@ub.edu.sa)

This work was supported in part by the University of Bisha; and in part by the Research, Innovation and Enterprise Centre (RIEC), Universiti Malaysia Sarawak.

ABSTRACT Perfect forward secrecy, cross-verification, and robust mutual authentication guarantee secure communication through unfavorable and unsafe channels. The speedy development in wireless communication and drone-assisted networking technology has miserable significance in many areas, including wildlife monitoring, sidewalk checking, infrastructure inspection, and smart city surveillance. But guaranteeing message integrity, non-repudiation, authenticity, and authorization for information transmission for these areas are still challenging for researchers when using Flying Ad Hoc Networks (FANETs). The FANET's existence for drone technology is more complicated due to dynamic changes in its topology and easily vulnerable to the adversary for numerous attacks. So far, before exhilarating a drone in the Internet-of-Drones (IoD) environment, controlled layered network architecture is indispensable to allow only legitimate drones to collaborate securely with each other and with the ground control station (GCS) for building the highest trust. A minor lapse creates a severe complication for communication security because an attacker might be trapping data from the open network channel and using it for their unusual deeds. Attentively, identification authentication and message authentication are necessary for such a sensitive environment. Therefore, in this research article, we have designed a verifiably secure Elliptic Curve Cryptographic (ECC)-based authentication scheme for IoD using FANET. The formal security proof of the scheme has been made using a programming verification toolkit ProVerif2.03, Random Oracle Model (ROM), and informally by pragmatic illustration. And the performance evaluation section of the article has been made by considering storage, computation, and communication costs. When comparing the proposed security mechanism with state-of-the-art schemes, it has been shown that the work done in this article is efficient and effective and is suitable for practically implementing in the IoD environment.

INDEX TERMS Manageability, sensors, drones, latency, integrity, authorization, cryptography.

I. INTRODUCTION

An environment where drones can provide secure access, controlled over the internet and operationalized for generic purposes, is termed Internet-of-Drones (IoD). A drone is also called Unmanned Aerial Vehicle (UAV). It can be operationalized in two ways: autonomously or through a pilot manually. The first type requires a communication link between the drone and the ground control station (GCS) for operating its flight and preprogrammed or automation systems. While in the second type, an operator manually controls, supervises, and manages the drone via Line-of-Sight (LoS) mode [1].

The rapid growth of drone technology in the past decades has led to the successful adoption of IoD in the military and civilian domains. In both areas, this adoption of IoD

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo.

is implemented for various purposes like infrastructure and pipeline inspection, filming movies, search and rescue operations, traffic monitoring, war reporting, troop movement, package delivery, cinematography, military mission, wildlife surveillance, and agricultural-land spraying [2].

Usually, the drone has a set of micro-electromechanical systems, low-capacity batteries, airframes, micro-processors, micro-radio devices, and a limited capacity and volume of payload. Due to its too fewer capabilities yet not qualified for complex operations, like aerial surveillance during natural disaster assessment, infrastructure inspection, reconnaissance mission, and other processes. However, multi-drone systems that operate across FANETs can allow drones and GCS to work collaboratively for such a complex mission completion. The synergy among all the participants is necessary to improve IoD's functionality. After achieving synergy, the drone can communicate with itself and the GCS through wireless and self-organizing networks called Flying Ad-hoc Networks (FANETs), a sub-type of Mobile Ad Hoc Networks (MANET). However, FANET causes networking problems that prevent a drone from communicating effectively with GCS. So far, considering all the basic features of FANET, message authentication and identification authentication are challenging tasks in providing successful path discovery, data transmission, and route maintenance services to all IoD participants [3].

Moreover, without presenting creative and sustainable solutions to the critical security features like integrity, non-repudiation, and confidentiality, the drone cannot show considerable protection against multiple attacks, including physical capture, clogging, DoS, and stolen-verifier, etc. And these issues and serious challenges can only be tackled by an innovative security design/framework to delegate it to various IoD participants and other entities for effective services. For this purpose, a cryptographic algorithm (cryptosystem) is crucial for making it practical for user accomplishments. However, according to the literature, standard cryptographic algorithms/protocols have been developed but do not offer efficient service [4]. For example, Ronaldo *et al.* [5] proposed an ECC key for encrypting symmetric cryptographic keys and Koblitz's Method for encoding decoding purposes. But due to heavy transmission time, their security scenario cannot offer efficient services to the drone. Even though there is a lot of research [6], [7], [8] done on drone communication security, still no systematic infrastructure or framework has been given for IoD's networking and communication difficulties. As a result, it is necessary to design an authentication scheme to fully use drones' potential in strengthening communication security, such as a mobile device to drone, drone to drone, drone to the mobile device, drone to GCS, and GCS to drone, and increase its performance.

A. MOTIVATION AND CONTRIBUTIONS

As the drone (D), mobile device (M), and ground-control-station (GCS) in IoD can exchange information via an open network channel (FANET), so the security of information broadcasting remains a noticeable concern. The attacker can catch data from the insecure channel and use it later for malicious deeds. Furthermore, suppose the integrity, confidentiality, and authorization of this sensitive information becomes leaked and exposed to the attacker; they can easily be launching reply, masquerade, man-in-the-middle, and drone physical capture attacks at any time. Also, the attacker can rebound it to an adversary (an algorithm, powerful computer or software program, etc.) for disrupting Ephemeral-Secret-Leakage (ESL) attack. Therefore, it is extremely required to build an authentication protocol that can provide access to an end-user at any time without interacting with the GCS. As a result, we suggested an ECC-Based protocol for IoD deployment drone using FANET. The major contributions of this research work are as under:

i. An ECC-based lightweight protocol is designed for an IoD deployment drone using FANET. The Computation Diffie-Hellman (CDHP) technique is used to securely exchange ECC keys among all the participants of IoD during the session key generation process.
ii. The proposed protocol guarantees to be secured against all known threats faced by IoD, incredibly privileged insider, stolen-verifier attacks, and mitigates the outdated data transmission and design flaws that are often noticed in state-of-the-art protocols.
iii. The randomized key over finite filed F_q has the capability of minimum computation costs, less communication and storage overheads, and strong security.
iv. The generation of 160-bit random keys, formal proof using [9], [10] and informal proof using [11] demonstrate the robustness of the proposed protocol.
v. The security and performance balancing approach has been accomplished, which is often lacking in previous protocols.

B. PAPER ORGANIZATION

The organization of the paper is structured as follows: Section 2 contains the foundation of this research, section 3 demonstrates the literature, section 4 proposes a lightweight ECC-based authentication protocol for IoD, and section 5 explains how to perform a security analysis using ROM [9], ProVerif2.03 [10], and proposition/pragmatic illustration [11] for the proposed protocol. Section 6 contains the proposed protocol's performance analysis regarding storage overheads, communication, and computation costs and comparative analysis. Finally, in section 7, we will conclude the paper and make some recommendations for future work.

II. PRELIMINARIES

This preliminary section of the paper will briefly explain the fundamental background and different security features considered to be the building blocks for the proposed verifiably secure ECC-based authentication and key agreement protocol for IoD using FANET.

A. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

The elliptic curve over points (x, y) can be defined by $y^2 = x^3+ax+b \bmod p$, where a, b$\in \mathbb{Z}p$ and whereas p is a prime. ECC is a widely used public-key cryptographic technique in which cryptographic primitives can be constructed on a cyclic group of finite filed (F_q) [12]. A standard methodology to automatically conceptualize E(F_q) is to understand the curve of the equation $y^2 = x^3+ax+b \bmod p$. It is to mention that the curve intersects three pints means point P is counted twice along with the point Q that can satisfy all these key pairs: The negation P is –P, P+(-P) = Q, P+(-P)+ Q= (P+(-P))+Q=Q+Q=Q. The point P is on the x-axis and is P(x, y), and -P=-(x, y)=(x, -y). Also, for two arbitrary points P1 and P2, the curve E/Fq P1+P2=-Q=Q [12].

B. DIFFIE-HELLMAN KEY EXCHANGE TECHNIQUE

The Diffie-Hellman key exchange method is a safe and efficient way for two parties to exchange a shared key without encrypting the data. It is important to note that Diffie-Hellman does not provide authentication. Diffie-Hellman algorithms can be used as part of an authentication protocol. We will use it for the secure exchange of keys, random numbers, identities, etc., among different peers that will be used for mutual authentication and cross-verification. In cryptosystems, elliptic curves are commonly occurred and defined over a finite field Fq forms a group G of order prime. So, there is no non-generic algorithm is available for solving the Computational Diffie-Hellman Problem (CDHP) on elliptic curves. By taking points (P, Q) from a curve and by applying CDHP will definitely make it hard to compute by anyone like aP, bQ, abP, abQ, or $(P, Q)^{ab}$ [13].

C. FLYING AD HOC NETWORK (FANET)

FANET is a type of network that can connect several small unmanned aerial vehicles (UAVs) in an ad hoc way and integrate into a team/group/cluster to achieve high-level targets. The essential characteristics of FANETs include mobility, absence of central control, self-organization, and ad hoc nature amongst the UAVs, which might enhance connection and communication range in infrastructure-less areas. On the one hand, FANETs can provide an immediate deployment, flexibility, self-configuration, and comparatively low-cost operating network in the development of a disaster where ordinary communication infrastructure is unavailable, and on the other way, it connects multiple UAVs in an ad hoc manner which is considered a big challenge. Therefore, in order to build a stable, reliable, and durable connection, this degree of coordination requires suitable communication-controlled architecture along with efficient and robust authentication protocols that can make UAVs functionalized for a complex tactical tasks in highly dynamic flying zones [14].

D. NETWORK MODEL

The proposed Internet-of-Drones (IoD) model consisted of the following main entities/participants:

Trusted Authority Center (TAC): The trusted authority center (TAC), also known as the certification authority, is a legitimate organization that verifies IoD services in order to determine with whom you interacted. It is an essential part of the designed security framework which is responsible for supplying real-time problem handling, data processing, and networking services to the IoD environment as a whole.

Ground Control Station (GCS): It is the centralized command and control center for drones' secure flight and direction services. It manages operational parameters, monitors drone sensors, and governs surveillance cameras. GCS creates flight separation operations and mission-critical activities and controls the drones' payload subsystems. GCS is also responsible for interpreting, gathering, and disseminating data gathered by the drone during a critical task.

Drone (D): A drone is a key participant in the IoD environment. Light materials like sensors, actuators, payloads, batteries, and GPS lasers reduce weight and enhance maneuverability. The navigational system and sensors are installed in the nose of a drone. Different companies selling drones are designed as highly complex and composite features that absorb vibrations, increase the communication module, and onboard small computer systems to supervise, monitor, and control the fundamental components of the drone.

Mobile-Device (M): Mobile devices (M) of different varieties are available now a day in the market. It must have the functionalities to control a drone remotely by operating someone (User). It must be powerful enough to prevent the drone flight from lagging, crashing during use and freezing, etc. Figure 1 shows the proposed network or system model in this article in which each entity first registers with TAC and then is deployed in IoD for the task.

E. ADVERSARY MODEL

This model is based on [15], suppose the proposed protocol is denoted by $\prod$, entities involved are Mobile-Device (M), Drone (D), ground-control-station (GCS) and many instances are $\mathcal{P}i$ means an ith instance of $\prod$. GCS has a confidential key *s*; let suppose the drone has its identity ID_D, nonce N_D, and public key R_D; mobile-device (M) has ID_M, nonce N_M, publicly known key R_M. Drone (D) stores (R_D, S_D, PK_D, SK_D), and Mobile-Device (M) stores (R_M, S_M, PK_M, SK_M) parameters in their memories. Adversary interacts $\prod$ to represent themselves as a malicious drone with D, M, or GCS in the following manner.

1. *A* extracts information for finding out the internal secrets in it
2. *A* can change the internal parameters stored in either drone's or GCS's memory, which later were used for malicious deeds.
3. *A* might erase some or whole data from the internal storage.
4. *A* upgrades the stored information to fulfill their desired work.
5. *A* may corrupt the internal secret information.

6. *A* can also falsify the secret parameters by injecting their own false credentials.
7. *A* copy message from the open network line and later used for a potential replay attack.
8. *A* using power analysis and reverse engineering techniques for extracting the valuable information from the stolen/take-down/crashed drone and later on use it for their own purpose.

Similarly, more generally, let suppose the static IoD's participant is denoted by f and dynamic by n whereas $n>$f, $n>2$f, or $n>3$f, so the power with adversary A in the stated situation is:

1) *A* corrupt either static or dynamic or both participants.
2) *A* gains some computational power for rebounding security parameters, perform a polynomial times calculation to figure out the internal credential by launching a possible attack.
3) *A* might construct some hash-rated values to match with the actual hash values for possible hash value collision.
4) *A* sees the different messages over the public network channel of all participants.
5) Finally, *A* can inject wrong things into an actual message by copying a message from the line, corrupting it, modifying it or deleting something from it, or dynamically taking extra round trips to decide what message needs to be corrupted and which one is un-corrupted, etc.

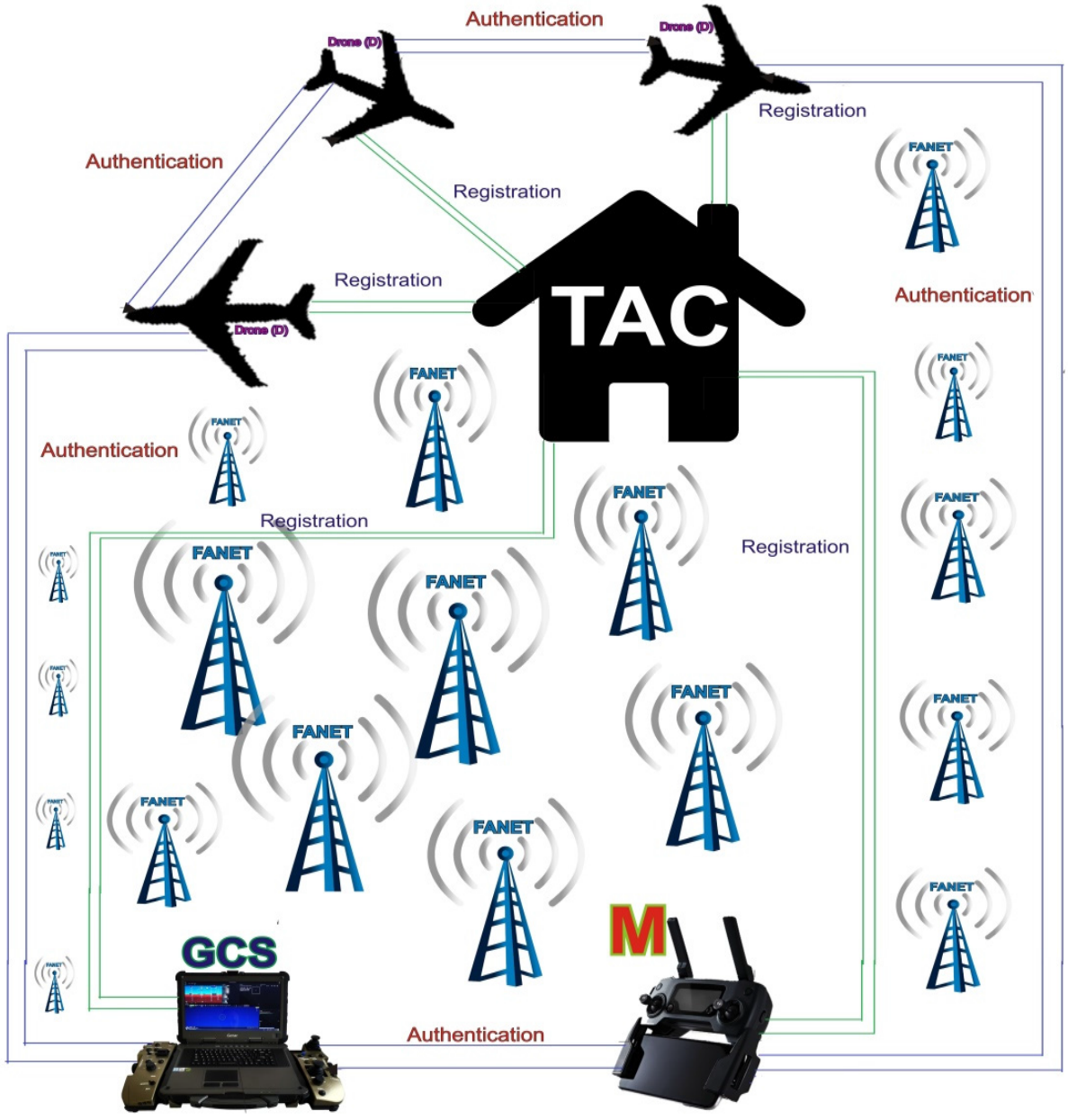

FIGURE 1. Network model.

F. THREAT MODEL

Dolev and Yao first introduced the threat model called DY [16] model, later elaborated by Myagmar *et al.* [17]. It is a practical risk assessment approach when using a system's security technique. This model must recognize and understand any possible threat to the system by formulating random tests to detect it and how the given security mechanism responds to such a threat. If someone understands how risk can affect systems, categorize it as either an active threat or passive threat, and then implement preventive measures or countermeasures for them like spoofing, privilege escalation, DoS, information leakage, tampering, and repudiation are all potential threats to our proposed security paradigm. We can quickly discover problems early by using DY [16] and [17], identifying design flaws, and specifying where the danger agents are located. We can also emphasize an attacker's strengths, maintain protocol ahead of internal and external attacks, consider risks that are not typical, and make a list of security requirements.

III. RELATED WORKS

This section of the paper demonstrates the various security protocols for IoD or IoT proposed by multiple researchers. Boneh *et al*. [18] proposed the first authentication method in 2003. Their protocol was adequate for the two parties, but when the number of users increased, it has vulnerable to forgery attack. Lysyanskaya *et al.* [19] offered an authentication protocol based on three algorithms, i.e., permutation, RSA, and combination, for securing the random numbers exchanged across different parties. Herranz [20] proposed a dynamic identity-based deterministic authentication system. However, an attacker can easily access a legitimate user's internal credentials when using an extract algorithm. Paterson and Schuldt [21] devised an efficient and effective protocol in the random oracle model compared to [20]. But intruder can successfully extract the hidden credentials by running an extract query with the help of a challenger. Boldyreva *et al*. [22] developed a public cryptographic-based authentication protocol based on dynamic identity. However, the adversary can quickly enter and extract the internal credentials by running the access algorithm due to the non-usage of the Computational Diffie-Hellman Problem (CDHP).

Zhang *et al.* [23] suggested a many-to-one certificateless authentication protocol for IoD, claiming that it is resistant to forgery attack and protected in restricted bandwidth. They used the CDHP for exchanging the public-private key pairs among all the participants. Despite CDHP, it still noticed several drawbacks, like the inability to withstand DoS and insider attacks. Xing *et al.* [24] proposed a protocol based on cubic residues, claiming that their method is the most effective due to working in Einstein cubic ring design. Xiong *et al.* [25], Tian *et al.* [26], and Viet *et al.* [27] explained a certificateless public key cryptographic-based aggregate signature authentication protocol. They established a system of small-scale drones that can operate in low-altitude areas and provide different services. They used the mathematical lemma and RSA to improve the security of their certificateless aggregate signature-based protocols. However, it does not offer efficiency during message broadcasting among participants. Won *et al.* [28] proposed a set of three security protocols for IoD implementation of drones for smart car parking, smart city infrastructure health monitoring, and infrastructure inspection after a severe earthquake. But due to the batch verification instead of one-to-one, the one protocol from three suits, i.e., CLDA (Certificateless Data Aggregation), is incompetent in computation. It does not secure under a random oracle model using CDHP. Zhong *et al.* [29] proposed an aggregate signature-based protocol and stated that it could withstand types I and II attacks in the random oracle model but later failed to resist a side-channel attack.

In the IoT environment, Challa *et al.* [30] developed a new user authentication and key exchange protocol that can also be used in the IoD environment. Haque *et al.* [31] suggested a protocol based on simple encryption/decryption with low computation complexity and efficiently running on a low-resource computer system. Benzart *et al.* [32] proposed a security mechanism that offered integrity, nonrepudiation, unforgeability, and confidentiality because they used encryption and aggregate signature concepts. But digitally signing and then encrypting a document takes more computer cycles and bloats the message by adding extra information. In an IoT environment, Turkanovic *et al.* [33] demonstrated a user authentication protocol that can also be used in an IoD environment. Farash *et al.* [34] proposed an improved authentication protocol and addressed all the security flaws identified in Turkanovic *et al.* [33] protocol. Farash *et al.* [34] cryptanalysis Turkanovic *et al.* [33] protocol and said that [33] is vulnerable to user impersonation, known temporary session key details, smart card problems, and off-line password guessing attacks.

Pu and Li [35] used a physical unclonable function (PUF) to verify and validate messages between drones and ground stations. They demonstrated that traditional cryptography is insufficient to protect sensitive data transmission; however, PUF may ensure communication. They also merged the Chaotic Map method for generating random keys, but it did not provide perfect forward secrecy. Alladi *et al.* [36] also suggested a PUF-based authentication protocol for UAVs using FANET. Their protocol computed two session keys to ensure high protection in UAVs' critical data transmission environment. Their protocol is unsafe, and confidentiality, privacy, and reliability are not guaranteed [38]. Jan *et al.* [37] proposed an HMACSHA1-based authentication protocol for securing IoD and have combined hash message authentication code with a secure hash algorithm (HMACSHA1) to offer a much more secure IoD environment for drone technology. Their scheme securely communicated between a drone to GCS, GCS to drone, drone to drone, and GCS. Nikooghadam *et al.* [38] demonstrated an ECC-based protocol, claiming that their protocol is stable and resists all known attacks under the random oracle model when using CDHP.

However, it still has traceability issues and is not appropriate for realistic implementation in the IoD setting due to important key-escrow issues. To implement IoD deployment drones for smart city surveillance and user-specific information to many smart objects, [39] suggested authentication protocols based on elliptic curve cryptography. [40] introduced a particular protection framework known as the demand response management authentication scheme (DRMAS) for grid computing. Jan *et al.* [41] proposed aggregate signature based pairing cryptographic key agreement protocol for IoD deployment military drone using UAVNs or FANET. They successfully mitigated the flaws found in literature like side-channel attacks, Unlink-ability, anonymity and traceability, forgery attack, and replay attacks. They have proved their scheme using mathematical lemma, logic, and ProVerif2.03; while the performance has been analyzed by considering three metrics. Their schemes have shown much more efficient and effective results, but instead of one-to-one authentication, aggregate signature minimizes the performance of the system.

Finally, Chen *et al.* [42] proposed a security protocol for small UAVs. Later, these small UAVs can be used for various applications, like entertainment, personal aerial photography, commercial markets, cargo transportation, military, police law enforcement activities, border control, disaster relief, and even agricultural and industrial applications. In addition, if implemented and installed for small UAVs, their protocol may provide smart city services such as traffic control and management, stock distribution, health, and emergency assistance. Their scheme is a hybrid cryptosystem-based privacy-preserving authentication protocol (e.g., digital signature, ECC, and cryptographic hash function). They claimed that their protocol is safe against malicious attacks and efficiently provides anonymity, confidentiality, and data integrity. However, after an extensive study of Chen *et al.* [42] scheme, the following drawbacks have been noticed. These are explained as under:

A. FORWARD SECRECY ISSUE

The attacker may put a fake request using a fake random number. The GCS understands that the said request was received from a legal drone, deriving a random number r_A, a large prime number P_A, calculates a master key $K_A = r_A P_A$, and computes $SE_{KA} = H_1(K_A, P_A)$. Also, if the secret is available to an adversary at any form/stage, they can efficiently compute the session key. Similarly, if a drone crashes or an adversary captures it physically or takedown and tries to recover the secrets from its memory, i.e., r_{GCS}, r_{UAV}, r_{PMD}, and $R_{UAV}=r_{UAV}P$, they can easily recover. After recovering these stored credentials, the adversary can launch any attack. Therefore, Chen *et al.* [42] scheme failed to deliver perfect forward secrecy.

B. PRIVILEGED INSIDER ATTACK

What happens if the offender uses internal channels to access the systems? What if it is someone whom you trust? What if it is an insider with a security clearance? Because several high-profile cases have arisen where a trusted individual caused great harm to an organization. It is unclear whether that person committed the sabotage willingly, but the payload only activated upon particular criteria. Therefore considering these questions in mind, the scheme presented by Chen *et al.* [42] contains a random number r_{UAV} and an identity ID_{UAV} during its initial flight; an operator can easily use it for malicious purposes.

Microsoft statistics tells us that 1.5 million users have used 6.5 identities and passwords for just 25 websites in just three months, implying that a single password is shared in 3.9 online accounts/applications. As a result, if GCS's privileged insider/administrator knows the identity (ID_{UAV}), he/she can easily impersonate that specific user by using it on another website for other reasons. In [42], a drone sends identity to GCS directly where the privileged insider can get and abuse it in some other place for accessing other applications. Therefore, Chen *et al.* [42] methodologies are vulnerable to privileged insider attacks.

C. STOLEN VERIFIER ATTACK

Suppose an attacker can steal the mobile device (M), in [42], they can easily figure out credentials from it using power analysis attack and reverse engineering. The attacker extracts the internally stored credentials by choosing an imaginary random number R_A, mobile-device (M) computes and multiply it with public key $r_A P$, $h_{PMD} = H_1(ID_{PMD}, R_{PMD})$, $S_{PMD} = r_{PMD}+h_{PMD}$ and confirms $SP_{MDP}?=R_{PMD}+H_1(ID_{PMD}, R_{PMD})PK_{TAC}$. Due to this, an attacker identifies R_{PMD}, S_{PMD}, PK_{PMD}, and SK_{PMD} parameters, which can then be used for launching any attacks at any time. Therefore, Chen *et al.* [42] protocol is weaker against stolen verifier attack.

D. OUT DATED DATA TRANSMISSION FLAW

In Chen *et al.* [42], it might send the previous session data in the upcoming session, and it has not mentioned how to deal with the already used credentials of other sessions. For example, during session key generation, the mobile device sent (ID_{PMD}, c_{PMD}, CHK_{UP}) without any time threshold and the system couldn't identify who sent it and when it was sent. The drone sends (ID_{UAV}, c_{UAV2}, CHK_{GU}) message towards GCS, and GCS sends (ID_{GCS}, c_{GCS2}, Sig_{GCS2}) message towards UAV which can misguide each one at any time. Also, if a drone is deployed to identify the temperature status, it is widely a chance to send the previous session's recorded temperature instead of a new one. Therefore, the outdated data transmission issue in Chen *et al.* [42] protocol has been observed.

E. MISSING UAV ADDITION PHASE

If the system needs another drone for some other task, it has never been explained how a drone can be added dynamically to the network. Therefore, Chen *et al.* [42] protocol doesn't

provide a dynamic UAV addition facility to the synchronous network topology.

F. MISSING UAV REVOCATION/REISSUE PHASE

If a drone goes out of a system, Chen *et al.* [42] never explain what happened to its stored record. Therefore [42] failed to offer a facility for the UAV revocation phase or cancellation/deletion of the previous drone (malfunctioned drone for some reason) record.

IV. PROPOSED SCHEME

This section presents the improved, lightweight, ECC-based authentication and key-agreement protocol for secure drone communication with the ground control station and other participants. The proposed authentication protocol consists of five phases, including the setup phase, registration phase, authentication phase, dynamic drone addition phase, and drone revocation/reissue phase; each of these phases is described one by one under the following headings, while different notations used for designing the protocol are shown in Table 1.

TABLE 1. Notations and its descriptions.

Symbol	Descriptions
ID_D	Drone's Identity
ID_M	Mobile-Device Identity
ID_G	Ground-Control-Station Identity
N_D, N_M, N_G	160-bits random nonce
Q	Finite prime
F_q	Finite prime
E/F_q	Elliptic Curve E over F_q
G	Cyclic additive group
r, a, b, c, d, e, f, g, t	Random points over curve
PK_C	A public key PK_C=sP
S_x	Elliptic curve signature group
M_{GPS}	The GPS message reported by a drone
SEK_{xy}	Session key
$E_x(m)$	Encryption
$D_x(m)$	Decryption
Sig_{xy}	Signed Signature
SK_x/PK_x	Private key/Pubic Key
$S_{SKx}(m)$	Signed Private key
$V_{PKx}(m)$	Signed public key
CHK_x	A verified message
A?=B	Determination
h(.)	Hash function
P	Generator from G
S	Secret key for TAC

A. SETUP PHASE

In the setup phase, the TAC generates the public parameters for the IoD as well as their own private key by choosing a large prime P, elliptic curve points $E_q(a, b)$, a base point $P \in E_q(a, b)$ of order prime q, one-way hash function $h:(0.\ 1)^* \rightarrow (0, 1)^l$, a secret key $s \in Z_q$ of the same order prime q, and publish system parameters ($E_p(a, b)$, P, s, h(.)).

B. REGISTRATION PHASE

This phase completed in the following three sub-phases, while the different computation steps are shown in module I (a), (b), and (c).

1) DRONE'S REGISTRATION

A drone (D), must, first, registers with the trusted authority center (TAC). First, selects drone's identity ID_D, chooses a random nonce N_D, computes: $MID_D=h(ID_D||N_D)$ and sends MID_D towards trusted authority center (TAC). The TAC picks a random number r_D, computes: $R_D=r_DP$, $h_D=H_1(MID_D||R_D)$, $S_D=r_D \oplus h_D s$ and transmits (R_D, S_D, PK_D, SK_D) towards the drone over a secure channel. Upon receiving (R_D, S_D, PK_D, SK_D) message, the drone verify $S_DP?=R_D \oplus H_1(MID_D||R_D)||PK_C$, if confirmed, TAC stores (R_D, S_D, PK_D, SK_D) in the memory of a drone (D).

2) MOBILE-DEVICE REGISTRATION

The mobile device (M) chooses identity ID_M and picks a random nonce N_M, computes $MID_M= h(ID_M||N_M)$ and sends MID_M towards the trusted authority center. TAC selects a random number r_M, computes: $R_M= r_MP$, $h_M= H_1(MID_M||R_M)$, $S_M= r_M \oplus h_M||s$ and transmits (R_M, S_M, PK_M, SK_M) towards mobile device over a secure channel. The mobile device confirms $S_MP?=R_M \oplus H_1(MID_M||R_M)||PK_C$, if verified, stores ($R_M$, S_M, PK_M, SK_M) parameters in the memory of the mobile device (M).

3) GROUND CONTROL STATION REGISTRATION

The ground control station (GCS) chooses identity ID_G, random nonce N_G, computes $MID_G= h(ID_G||N_G)$ and sends MID_G towards trusted authority center. The TAC picks a random number r_G, computes $R_G= r_GP$, $h_G= H_1(MID_G||R_G)$, $S_G= r_G \oplus h_G||s$ and transmits (R_G, S_G, PK_G, SK_G) message towards GCS over a secure channel. Upon receiving (R_G, S_G, PK_G, SK_G) message, GCS confirms $S_GP?=R_G \oplus H_1(MID_G||R_G)||PK_C$, if verified stores ($R_G$, S_G, PK_G, SK_G) parameters in the memory of GCS.

C. AUTHENTICATION PHASE

This phase of the protocol comprising of Mobile-Device → Drone, Mobile-Device → Ground Control Station, Ground Control Station → Drone and Mobile-Device → Drone → Ground Control Station. Each of these sub-phases is described one-by-one under the following step of computations and shown in module II (a), (b), (c) and (d).

1) MOBILE-DEVICE AND DRONE AUTHENTICATION

By controlling the drone via a mobile device the following set of computations are to be performed:

i. The mobile-device chooses a random number a, calculates $X_M = aP||TS_1$, where TS_1 is the timestamp; and sends (ID_M, R_M, X_M, TS_1) message towards the drone (D).

ii. The drone first checks the timestamp with the current time $TS_c\text{-}TS_1 \leq \Delta TS$ and chooses a random number b, calculates $X_D = bP||TS_2$, $PK_M = R_M \oplus H_1(MID_M||R_M)||PK_C$, $K_{UP2} = b||X_M$, $K_{UP1} = S_D||(X_M \oplus b)||PK_M$, and computes session secret shared keys i.e. $SEK_{UP} = H_2(K_{UP1}||K_{UP2})$, and $CHK_{PU} =$

MODULE I (a): DRONE (D) REGISTRATION

Drone (Ds)		**Trusted Authrity Center (TAC)**
Choose ID_D and random nonce N_D $MID_D=h(ID_D\|\|N_D)$		
	$\xrightarrow{MID_D}$	
		Choose random number r_D $R_D=r_DP$ $h_D=H_1(MID_D\|\|R_D)$ $S_D=r_D\oplus h_Ds$
	$\xleftarrow{(R_D, S_D, PK_D, SK_D)}$	
$S_DP?=R_D\oplus H_1(MID_D\|\|R_D)\|\|PK_C$ Stores (R_D, S_D, PK_D, SK_D) parameters		

MODULE I (b): MOBILE (M) REGISTRATION

Mobile-Device (M)		**Trusted Authrity Center (TAC)**
Choose ID_M and random nonce N_M $MID_M=h(ID_M\|\|N_M)$		
	$\xrightarrow{MID_M}$	
		Choose a random number r_M $R_M=r_MP$ $h_M=H_1(MID_M\|\|R_M)$ $S_M=r_M\oplus h_M\|\|s$
	$\xleftarrow{(R_M, S_M, PK_M, SK_M)}$	
$S_MP?=R_M\oplus H_1(MID_M\|\|R_M)\|\|PK_C$ Stores (R_M, S_M, PK_M, SK_M) parameters		

MODULE I (c): GROUND CONTROL STATION (GCS) REGISTRATION

Ground Control Station GCS		**Trusted Authrity Center (TAC)**
Choose ID_G and random nonce N_G $MID_G=h(ID_G\|\|N_G)$		
	$\xrightarrow{MID_G}$	
		Choose a random number r_G $R_G=r_GP$ $h_G=H_1(MID_G\|\|R_G)$ $S_G=r_G\oplus h_G\|\|s$
	$\xleftarrow{(R_G, S_G, PK_G, SK_G)}$	
$S_GP?=R_G\oplus H_1(MID_G\|\|R_G)\|\|PK_C$ Stores (R_G, S_G, PK_G, SK_G) parameters		

$H_3(SEK_{UP}||X_{PMD})$, Build a message having $(MID_D, R_D, X_D, CHK_{PU}, TS_2)$ parameters and sends back towards the mobile device.

iii. There, the mobile-device, first checks the time threshold, by subtracting the received timestamp from its current time TS_c-$TS_2 \leq \Delta TS$, if not validates, it means the data is outdated, else, computes: $PK_D = R_D\oplus H_1(MID_D||R_D)||PK_C$, $K_{PU1} = S_M||(X_D\oplus a)||PK_D$, $K_{PU2} = aX_D$, and confirms $SEK_{UP} = H_2(K_{PU1}||K_{PU2})$, and $CHK_{PU}?=H_3(SEK_{UP}||X_M)$, if not matched, termination of the process performed, else, keep SEK_{UP} and CHK_{PU} are session secret shared keys.

2) MOBILE-DEVICE AND GCS AUTHENTICATION

Drone doesn't provide services without obtaining flight-path by an operator from the ground control station. To do so, authentication of mobile-device with the ground control station is necessary. For such a communication, the following computations are performed:

i. The mobile device extracts a random number c timestamp TS_1 and computes $X_{M2} = cP||TS_1$, sends $(MID_M, R_M, X_{M2}, TS_1)$ message towards GCS.

ii. The GCS upon receiving $(MID_M, R_M, X_{M2}, TS_1)$ message, checks whether the received data is fresh or outdated by subtracting the received timestamp from its current timestamp TS_c-$TS_1 \leq \Delta TS$, if not validated, the message is considered for potential replay attack, else, chooses a random number d and computes: $X_G = dP||TS_2$, $PK_M=R_M\oplus H_1(MID_M||R_M)))PK_C$, $K_{GP1}=S_G||(X_{M2}\oplus d)||PK_M$, $K_{GP2} = dX_{M2}$, and compute session secret keys i.e. $SEK_{GP}=H_2(K_{GP1}||K_{GP2})$, and $CHK_{PG}=H_3(SEK_{GP}||X_{M2})$. Build a message having $(ID_G, R_G, X_G, CHK_{PG}, TS_2)$ parameters and send back towards mobile device.

iii. The mobile device, first checks the freshness of the message TS_c-$TS_2 \leq \Delta TS$ and computes: $PK_G=R_G\oplus H_1(MID_G||R_G)||PK_C$, $K_{PG1}=S_M||(X_G\oplus c)||PK_G$, $K_{PG2} = cX_G$, confirms $SEK_{GP}=H_2(K_{PG1}||K_{PG2})$, and $CHK_{PG}?=H_3(SEK_{GP}||X_{M2})$, if matched, keep it session secret keys.

MODULE II (a): MOBILE DEVICE (M) AND DRONE (D) AUTHENTICATION

Mobile-Device (M)		Drone (D)
Choose a random number a		
$X_M=aP\|\|TS_1$		
	(MID_M, R_M, X_M, TS_1) →	
		$TS_c-TS_1\leq\Delta TS$
		Choose a random number b
		$X_D=bP\|\|TS_2$
		$PK_M=R_M\oplus H_1(MID_M\|\|R_M)\|\|PK_C$
		$K_{UP1}=S_D\|\|(X_M\oplus b)\|\|PK_M$
		$K_{UP2}=bX_M$
		$SEK_{UP}=H_2(K_{UP1}\|\|K_{UP2})$
		$CHK_{PU}=H_3(SEK_{UP}\|\|X_M)$
	← $(MID_D, R_D, X_D, CHK_{PU}, TS_2)$	
$TS_c-TS_2\leq\Delta TS$		
$PK_D=R_D\oplus H_1(MID_D\|\|R_D)\|\|PK_C$		
$K_{UP1}=S_M\|\|(X_D\oplus a)\|\|PK_D$		
$K_{UP2}=aX_D$		
$SEK_{UP}?=H_2(K_{UP1}\|\|K_{UP2})$		
$CHK_{PU}?=H_3(SEK_{UP}\|\|X_M)$		

MODULE II (b): MOBILE-DEVICE (M) AND GCS AUTHENTICATION

Mobile-Device (M)		Ground Control Station GCS
Choose a random number c		
$X_{M2}=cP\|\|TS_1$		
	$(MID_M, R_M, X_{M2}, TS_1)$ →	
		$TS_c-TS_1\leq\Delta TS$
		Choose a random number d
		$X_G=dP\|\|TS_2$
		$PK_M=R_M\oplus H_1(MID_M\|\|R_M)\|\|PK_C$
		$K_{GP1}=S_G\|\|(X_{M2}\oplus d)\|\|PK_M$
		$K_{GP2}=dX_{M2}$
		$SEK_{GP}=H_2(K_{GP1}\|\|K_{GP2})$
		$CHK_{PG}=H_3(SEK_{GP}\|\|X_{M2})$
	← $(MID_G, R_G, X_G, CHK_{PG}, TS_2)$	
$TS_c-TS_2\leq\Delta TS$		
$PK_G=R_G\oplus H_1(MID_G\|\|R_G)\|\|PK_C$		
$K_{PG1}=S_M\|\|(X_G\oplus c)\|\|PK_G$		
$K_{PG2}=cX_G$		
$SEK_{GP}?=H_2(K_{PG1}\|\|K_{PG2})$		
$CHK_{PG}?=H_3(SEK_{GP}, T_{PMD2})$		

MODULE II (c): GCS AND DRONE (D) AUTHENTICATION

Ground Control Station (GCS)		Drone (D)
Choose a random number $r\in Z^*q$		
$X_G=rP\|\|TS_1$		
$Sig_G=Enc_r(MID_D\|\|R_G)$		
	$(Sig_G, MID_G, R_G, X_G, TS_1)$ →	
		$TS_c-TS_1\leq\Delta TS$
		Confirms $(MID_D\|\|R_G)?=V_{PKG}(Sig_G)$
		Extracts t
		$X_D=tP\|\|TS_2$
		$Sig_D=Enc_t(MID_M\|\|R_D\|\|M_{GPS}\|\|cert_D)$
	← $(Sig_D, MID_D, R_D, X_D, TS_2)$	
$TS_c-TS_2\leq\Delta TS$		
Confirms $(MID_M\|\|R_D\|\|M_{GPS}\|\|cert_D)?=V_{PKD}(Sig_{D4})$		

3) GCS AND DRONE AUTHENTICATION

If a ground control station wants to know the status of a drone, flight-plan, GPS coordinates and scope of regulations, it needs to properly authenticate each other. In this regard, the following step will assure a successful communication between GCS and drone:

MODULE II (c): GCS AND DRONE (D) AUTHENTICATION

Ground Control Station (GCS)		Drone (D)
Choose a random number $r \in Z^*q$		
$X_G = rP \| TS_1$		
$Sig_G = Enc_r(MID_D \| R_G)$		
	$(Sig_G, MID_G, R_G, X_G, TS_1)$ →	
		$TS_c - TS_1 \leq \Delta TS$
		Confirms $(MID_D \| R_G)? = V_{PKG}(Sig_G)$
		Extracts t
		$X_D = tP \| TS_2$
		$Sig_D = Enc_t(MID_M \| R_D \| M_{GPS} \| cert_D)$
	← $(Sig_D, MID_D, R_D, X_D, TS_2)$	
$TS_c - TS_2 \leq \Delta TS$		
Confirms $(MID_M \| R_D \| M_{GPS} \| cert_D)? = V_{PKD}(Sig_{D4})$		

MODULE II (d): MOBILE-DEVICE (M), DRONE (D) AND GCS AUTHENTICATION

Mobile-Device (M)	Drone (D)	Ground Control Station (GCS)
$C_{M3} = E_{SEKup}(M_R \| cert_D)$		
$Sig_{M3} = S_{SKM}(M_R \| cert_D)$		
$(MID_M, c_{M3}, Sig_{M3})$ →		
	$(M_R \| cert_D) = D_{SEKup}(c_{M3})$	
	Confirms: $(M_R \| cert_D)? = V_{PKM}(Sig_{M3})$	
	Choose a random number e	
	$X_{D2} = eP \| TS_1$	
	$(MID_D, R_D, X_{D2}, TS_1)$ →	
		$TS_c - TS_1 \leq \Delta TS$
		$PK_D = R_D \oplus H_1(MID_D \| R_D) \| PK_C$
		Choose f
		$X_{G2} = fP \| TS_2$
		$K_{GU1} = S_G \| (X_{D2} \oplus f) \| PK_D$
		$K_{GU2} = fX_{D2}$
		$SEK_{GU} = H_2(K_{UG1} \| K_{UG2})$
		$CHK_{UG} = H_3(SEK_{UG} \| X_{D2})$
	← $(MID_G, R_G, X_{G2}, CHK_{UG}, TS_2)$	
	$TS_c - TS_2 \leq \Delta TS$	
	$PK_G = R_G \oplus H_1(MID_G \| R_G) \| PK_C$	
	$K_{UG1} = S_D \| (X_{G2} \oplus e) \| PK_G \| TS_3$	
	$K_{UG2} = eX_{G2}$	
	Confirms $SEK_{UG}? = H_2(K_{UG1} \| K_{UG2})$	
	$CHK_{UG}? = H_3(SEK_{GU} \| X_{D2})$	
	← $(MID_G, R_D, X_{G2}, CHK_{UG}, TS_3)$	
$TS_c - TS_3 \leq \Delta TS$		
$PK_D = R_D \oplus H_1(MID_D \| R_D) \| PK_C$		
Extracts random number g		
$K_{UG1} = gX_M$		
$K_{UG1} = S_M \| (X_D \oplus g) \| PK_D$		
Confirms: $SEK_{UG}? = H_2(K_{UG1} \| K_{UG2})$		
$CHK_{UG}? = H_3(SEK_{UG} \| X_{D2})$		

i. The ground-control-station, first chooses a random number $r \in Z^*q$ calculates $X_G = rP \| TS_1$ and build a signature containing drone identity, encrypted with a private key r i.e. $Sig_G = Enc_r(MID_D \| R_G)$ and sends $(Sig_G, MID_G, R_G, X_G, TS_1)$ message towards a drone (D).

ii. The drone firstly verifies the timestamp $TS_c - TS_1 \leq \Delta TS$ in X_G. if found outdated, shall be considered a potential replay attack, else decrypts and confirms $(MID_D \| R_G)? = V_{PKG}(Sig_G)$, Further drone extracts t, computes $X_D = tP \| TS_2$ and built a response signature containing ID_M, GPS values and drone certificate i.e.

Sig_D=Enc_t(MID_M||R_D||M_{GPS}||$cert_D$) and sends (Sig_D, MID_D, R_D, X_D) to GCS.

iii. GCS checks timestamp in X_D, if it was outdated, the GCS discard and terminate the process, else, confirms (MID_M||R_D||M_{GPS}||$cert_D$)?=V_{PKD}(Sig_{D4}) and authenticate each other.

4) M, D AND GCS AUTHENTICATION

The purchase approval, flight-path and other necessary authentication will be utilized by M, D, and GCS. Now, the operator can use D via his/her M for any operation. But mutual authentication among these (D, M and GCS) is necessary which can be performed in the following steps:

i. First, the M computes: c_{M3}=E_{SEKup}(M_R||$cert_D$), Sig_{M3}=S_{SKM}(M_R||$cert_D$) and sends (MID_M, c_{M3}, Sig_{M3}) message towards D.

ii. The drone decrypts (M_R||$cert_D$)=D_{SEKup}(c_{M3}), confirms (M_R||$cert_D$)?=V_{PKM}(Sig_{M3}) and extracts a random number e computes: X_{D2}=eP||TS_1 and sends (MID_D, R_D, X_{D2}, TS_1) message towards GCS.

iii. GCS checks timestamp in X_{D2}, TS_c-TS_1 $\leq$ ΔTS and computes: PK_D=R_D⊕H_1(MID_D||R_D)||PK_C, extracts another random number f and calculates X_{G2}=fP||TS_2. K_{GU1}=S_G||(X_{D2}⊕f)||PK_D, K_{GU2}=fX_{D2} and computes shared session secret keys i.e. SEK_{GU}=H_2(K_{UG1}||K_{UG2}), and CHK_{UG}=H_3(SEK_{UG}||X_{D2}) and sends (MID_G, R_G, X_{G2}, CHK_{UG}) message back towards drone.

iv. Drone checks timestamp in X_{G2}, TS_c-TS_2 $\leq$ ΔTS, computes PK_G=R_G⊕H_1(MID_G||R_G)PK_C, K_{UG2}=eX_{G2}, K_{UG1}=S_D||(X_{G2}⊕e)||PK_G, verify SEK_{UG}=H_2(K_{UG1}||K_{UG2}), and CHK_{UG}?=H_3(SEK_{GU}||X_{D2}). Build (MID_G, R_D, X_{G2}, CHK_{UG}, TS_3) message and transmits toward M.

v. The mobile device check timestamp in X_G, TS_c-TS_3 $\leq$ ΔTS, computes PK_D=R_D⊕H_1(MID_D||R_D) PK_C, extracts random number g, calculates K_{UG1}=gX_M, computes K_{UG1}=S_M||(X_D⊕g)||PK_D and confirms SEK_{UG}?=H_2(K_{UG1}||K_{UG2}) and CHK_{UG}?=H_3(SEK_{UG}||X_{D2}) and keeps it the session shared keys in M, D and GCS.

D. DYNAMIC DRONE ADDITION PHASE

The protocol which is presented in this paper provides the facility of adding a new drone to the system securely. Let a new drone (D^{new}) added to the system, the trusted authority center chooses a random nonce N_D^{new} and unique identity ID_D^{new} for the drone (D^{new}) and computes: MID_D^{new}=h(ID_D^{new}||N_D^{new}) and sends MID_D^{new} towards the trusted authority center (TAC) over a secure channel. The trusted authority center (TAC) chooses a random number r_D of 160-bits and computes the master key: R_D^{new}=r_D^{new}P; further calculates h_D^{new}=H_1(MID_D^{new}||R_D^{new}), S_D^{new}=r_D^{new}+h_D^{new}||s and sends (R_D^{new}, S_D^{new}, PK_D^{new}, SK_D^{new}) towards the drone over a secure channel. The D^{new} (drone) verifies S_D^{new}P?=R_D^{new}⊕H_1 (MID_D^{new}||R_D^{new})||PK_C, if validated, the TAC stores (R_D^{new}, S_D^{new}, PK_D^{new}, SK_D^{new}) in the memory of the drone (D^{new}). Now it is ready to be deployed in the IoD environment for the assigned task. It is worth mentioning that the topology of FANET is dynamically changed without losing values. This important phase has not been mentioned by Chen *et al.* [42] in their protocol.

E. DRONE REVOCATION/RE-ISSUE PHASE

If an authentic mobile-device (M) stolen by someone or lost somewhere from a legitimate user/operator and user desires to operationalize another mobile-device for operating a drone; the following steps must be performed:

i. The newly mobile device must create a random number $c^/$ computes: MID_M^{new}=h(ID_D^{New}||$c^/$) and presents (MID_M^{new}, h(.)) to TAC over a secure channel.

ii. TAC chooses a random number r_M^{new} and computes: R_M^{new}=r_M^{new}P, h_M^{new}=H_1(MID_M^{new}||R_M^{new}), S_M^{new}=r_M^{new}⊕h_M^{new}||s and dispatches (R_M^{new}, S_M^{new}, PK_M^{new}, SK_M^{new}) via a reliable line.

iii. The mobile-device (M), first checks and confirms S_M^{new}P?=R_M^{new}⊕H_1(MID_M^{new}||R_M^{new})||PK_C, if verified, replaces old (R_M, S_M, PK_M, SK_M) credentials with (R_M^{new}, S_M^{new}, PK_M^{new}, SK_M^{new}) in the memory of the mobile device.

iv. It is to mention that before using mobile-device (M), it must first securely authenticate with GCS for reliable communication. To do so, the following steps of computations are performed with the GCS:

v. The secret value at mobile device is X_{M2}^{new}=$c^/$P, and other stored values are MID_M^{new}, R_M^{new}. Now (MID_M^{new}, R_M^{new}, X_{M2}^{new}) parameters are transmitted towards GCS.

vi. When receiving (MID_M^{new}, R_M^{new}, X_{M2}^{new}) message, the GCS chooses a random number d^{new} and computes: X_G=d^{new}P, PK_M^{new}=R_M^{new}⊕H_1 (MID_M^{new}||R_M^{new})||PK_C, K_{GP1}=S_G||(X_{M2}^{new} ⊕d^{new})||PK_M^{new}, K_{GP2}=$d^{new}$$X_{M2}$, SEK_{GP}=H_2(K_{GP1}||K_{GP2}), and CHK_{PG}=H_3(SEK_{GP}||X_{M2}). The GCS send (ID_G, R_G, X_G, CHK_{PG}) message back towards mobile device (M).

vii. The mobile-device (M) calculates: PK_G=R_G⊕H_1 (MID_G||R_G)||PK_C, K_{PG1}=S_M^{new}|| (X_G⊕$c^/$)||PK_G, K_{PG2}=$c^/$$X_G$, and confirms: SEK_{GP}=H_2(K_{PG1}||K_{PG2}) with CHK_{PG}?=H_3(SEK_{GP}||X_{M2}). If matched with the previously stored values, the operation become successful, otherwise, a denied message will be displayed.

V. SECURITY ANALYSIS

In this section, the security analysis for the proposed ECC-based authentication protocol in the IoD environment is performed both informally using the general adversarial model and pragmatic discussions [42] and formally using the random oracle model (ROM) [13]/ProVerif2.03 [15]. Both of these methodologies (ROM and ProVerif2.03) for formal security analysis are given one by one as under:

A. ROM [13] ANALYSIS

We scrutinize the proposed security mechanism by a worldwide used technique ROM. This is a theoretical model investigating the adversary's advantage regarding breaking the proposed protocol for IoD using FANET. Suppose our protocol is denoted by $\mathcal{P}$, adversary by $\mathcal{A}$, and participants (M, D, GCS) by $\mathcal{P}$. Let i^{th} instances are available with $\mathcal{A}$ for breaking our protocol by launching attack(s) on M, D, and GCS. Before analyzing $\mathcal{P}$ in $\mathcal{P}$, the following actions are performed:

1) STATES & PARTICIPANTS

There exist three ROM states: 1-Accept, 0-Reject, and -denied. The i^{th} instance in P are $\mathcal{P}$, M, D, GCS and are denoted as IN_P, IN_D, IN_M, IN_{GCS}, respectively.

2) QUERIES

Let an adversary $\mathcal{A}$ and a responder $\mathcal{R}$ establish communication with GCS and D, and let E_i denotes both. In contrast, *i* indicate the i^{th} occurrence of GCS and D. Whereas E_{DS} means adversary action to impersonate D, M, or GCS by forging (R_D, S_D, PK_D, SK_D), (R_M, S_M, PK_M, SK_M) or (R_G, S_G, PK_G, SK_G). E_{SD} forges N_D or N_M, $_G$, a, r, b for impersonating any participant, and E_{SC} is considered to be an action of the adversary for semantic security of the proposed protocol is given as under:

1) If a challenger *C* runs ***Setup Query*** and returns system parameters to $\mathcal{A}$.
2) And then runs ***Hash Query*** and stores the output in a list of parameters, applies the one-way hash function, i.e., h(M_s, M_p, M_n, M_a, M_b, etc.), and generates a random number r_D, r_M, or r_G of order prime and stored with any of the given hash message and random numbers (M_a, r_G) or (M, r_D) or (M, R_M) and return it also to $\mathcal{A}$.
3) Next, *C* authenticates the message using ***MAC(M_i) Query***; if succeeded, return M_i to $\mathcal{A}$.
4) Now, *C* sends ***Send(E^i, M_i)*** towards GCS, acts as a legitimate drone, and the response received also returns to $\mathcal{A}$, but in our protocol, we have added an extra step, taking 160-bits of random keys in each round trip in which *C* cannot verify. Let *C* return the response to $\mathcal{A}$.
5) Using ***Execute (D_i^{∞}, GCS)*** Query, the proposed protocol returns R_D, R_M, or R_G to $\mathcal{A}$.
6) ***Reveal (E^i):*** The challenger *C* given message but not $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$/$CHK_{UG}?=H_3$ ($SEK_{UG}||X_{D2}$).
7) ***Test (E_i):*** In this step, $\mathcal{A}$ can flip a coin 1–Valid *M* (Win), 0 – Reject (Loss).

3) FRESHNESS

If $\mathcal{P}$ is in the state of Accept, ***Reveal (E^i)*** query is not executed while ***Corrupt (E^i)*** query executed at once which in turn means that the freshness of messages are confirmed.

4) SEMANTIC SECURITY

The proposed authentication protocol P involves three entities: the mobile device U, drone V and ground control station G. Each entity has several instances to connect with s after P is executed, which is referred to as an oracle. Let U^k represent the x^{th} instance of U, V^k is the y^{th} instance of V, and G^k is the z^{th} instance of G. However, I^k is known to be the instance of all three members, namely, U, V, and G; likely, an oracle has three outcomes, namely, accept, reject, $\perp$ and do nothing/no result; accept means receiving a message authentically, rejection means getting a wrong message, and do nothing/no result. Before execution, $\mathcal{U}$ has ((R_M, S_M, PK_M, SK_M)) parameters, $\mathcal{V}$ has ((R_D, S_D, PK_D, SK_D)) parameters and $\mathcal{G}$ has ((R_G, S_G, PK_G, SK_G)) parameters and supposes these are in the memory of each participant stored securely [37].

Suppose adversary $\mathcal{A}$ has complete control over the public network channel; they may initiate, cancel, and arbitrate the formed session among the participants for violating their privacy and tracking them. As a result, $\mathcal{A}$ can use Oracle to run these queries, which include: **i)** (MID_M, R_M, X_M), (MID_D, R_D, X_D, CHK_{PU}), **ii)** (MID_M, R_M, X_{M2}), (MID_G, R_G, X_G, CHK_{PG}), **iii)** (Sig_G, MID_G, R_G, X_G), (Sig_D, MID_D, R_D, X_D), iv) (MID_M, c_{M3}, Sig_{M3}), (MID_D, R_D, X_{D2}) and v) (MID_G, R_G, X_{G2}, CHK_{UG}), (MID_G, R_D, X_G, CHK_{UG}). $\mathcal{A}$ 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, ***Reveal (I^k)*** query for recognizing the session secret key SK, fraudulent $\mathcal{U}$ for apprehending the arguments stored in the $\mathcal{U}$ and ***Test (I^k)*** query for finding the shared secret session key. Each one of these participant, however, has a secretly encrypted unique identity, and it will consent to the creation of a session if and only if any message from I to v) is sent to any participant. It must confirms: $SEK_{UP}?=H_2(K_{UP1}||K_{UP2})$, $CHK_{PU}?=H_3(SEK_{UP}||X_M$, $SEK_{GP}?=H_2(K_{PG1}||K_{PG2})$, $CHK_{PG}?=H_3(SEK_{GP}$, $T_{PMD2})$, and $(MID_M||R_D||M_{GPS}||cert_D)?=V_{PKD}(Sig_{D4})$ for SK calculated by each participants. $\mathcal{A}$ has only the probability of breaking the security of $\mathcal{P}$ by flipping a coin Ω, and suppose $\mathcal{A}$ flip a coin and get Ω' output, the advantage is:

$$Adv_P^{Protocol}(\mathcal{A}) = \left|2Pr\left|\Omega = \Omega'\right| - 1\right| \tag{1}$$

Despite attempting polynomial times, $\mathcal{A}$ cannot compute the 160 bits arbitrary selection of key by the ground control station (GCS), drone (D), and mobile device (M) for each session. As a result, the proposed authentication protocol is reliable against all potential adversary attempts. Furthermore, if a hash oracle's performance is $q_{he}^2/2^{ths+1}$, $q_{he+1}^2/2^{ths+1}_{and}q_{he}^2/2^{ths}$ then the full probability of collision among hash-output is $(q_{send}+q_{receive})^2/2(p-1)$, we will get:

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

If $\mathcal{A}$ computes correct tuple without hash, $\mathcal{A}$ either forge (MID_M, c_{M3}, Sig_{M3}), (MID_D, R_D, X_{D2}) by doing so, he/she must know finite points/keys E_p, identities and random keys/numbers selected in different round trip like a, b, c, d, e, f, g, h, r, N, P, but $\mathcal{A}$ cannot catch $cert_D$, and also cannot check the inner secrets in (MID_M, c_{M3}, Sig_{M3}), (MID_D, R_D, X_{D2}). So, this attempt of $\mathcal{A}$ also looks to be failed, even $\mathcal{A}$ forge (MID_G, R_G, X_{G2}, CHK_{UG}), (MID_G, R_D, X_G, CHK_{UG}), still they cannot win. They have the knowledge of $K_{UG2} = eX_{G2}$, $K_{UG1} = gX_M$ and $K_{GU1} = S_G||(X_{D2} \oplus f)||PK_D$,$PK_D = R_D \oplus H_1(MID_D||R_D)||PK_C$ and points to be taken arbitrarily from the cure E/F_q which does not lie in the record of $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot realize in gaining fruitful information, as given as:

$$[Prob|Success_3 - Prob|Success_2|] \leq \frac{2q_{send} + 2q_{hs1}}{2^{l_{hs}}} \quad (3)$$

Also, if $\mathcal{A}$ wishes to obatin session secret key, they can try for calculating SK using:

$$[Prob|Success_3 - Prob|Success_2| \leq q_{receive}.Adv_A^{PTA}(X_{GCS}) \quad (4)$$

As PTA stands for polynomial-times-attempt, while W is the session secret key of $\mathcal{A}$, if the total chances with $\mathcal{A}$ is [1/D], then

$$Prob\,|Success_3| = \frac{1}{2} + \max\left(\frac{q_{hs1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right) \quad (5)$$

Combine all the possible calculations done by an adversary for impersonating, masquerading the legal peer(s), we get

$$\begin{aligned}
&Adv_P^{protocol}(A)\\
&= \text{Prob}|Success_0| - 1\\
&= 2\,|Prob|\,Success_0\,| - Prob|\,Success_4|\\
&\quad + \max\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\} \quad (6)\\
&\leq 2\,(Prob\,|Success_0| - Prob\,|Success_4|\\
&\quad + \max\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\}\Big) \quad (7)\\
&\leq 2\,(Prob|Success_0| - prob|Success_4|\\
&\quad + \max\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\}\Big) \quad (8)\\
&\leq 2\,(Prob|Success_1| - prob|Success_2|\\
&\quad + \max\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\}\Big) \quad (9)\\
&\leq \frac{q_{hs}^2 + q_{hs1}^2 + q_{hs2}^2}{2^{l_{hs}}} + \frac{(q_{send} + q_{receive})^2}{2\,(q-1)}\\
&\quad + 2q_{receive}.Adv_A^{PTA}(X_{GCS})\\
&\quad + 2\left\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{send}}{|D|}\right\} \quad (10)
\end{aligned}$$

B. PROVERIF2.03 [15] SIMULATION

In this section, a programming verification toolkit ProVerif2.03 [15], can be used for session key secrecy, confidentiality, and reachability. It is a widely used toolkit for confirming the secrecy, robustness, reachability, and authorization of the session key. To do so, we first define two communication channels, i.e., private and public; declare variables, events, constraints, functions, and equations, and write code for all the computations in each peer. After that, we will run the code to check whether the shared session key is safe against an adversary or not, as shown in Appendix A and Appendix B.

C. INFORMAL SECURITY ANALYSIS

This section of the paper presents a pragmatic illustration of the proposed protocol. This is the informal security analysis of the protocol. So far, we will discuss the proposed protocol for different security functionalities in the following manner.

1) RESISTS PRIVILEGED INSIDER ATTACK

In the registration phase the identities of drone (D), mobile-device (M) and ground control station (GCS) have not been exchanged in plain-text format i.e. MID_D=h($ID_D||N_D$), MID_M=h($ID_M||N_M$) and MID_G=h($ID_G||N_G$). The operator, administrator, privileged user, or manufacturer doesn't figure out any identity among these for creating any future hurdle for the system. Therefore, the proposed protocol, now in the registration phase, resists privileged insider attack.

2) SAFE AGAINST STOLEN VERIFIER ATTACK

This attack is subject to the stolen/lost mobile device (M) from a legal operator/user, and the attacker finds it elsewhere. After that, an attacker can attempt to expose identity, keys, and attached information related to the IoD from mobile device (M) memory. Either attacker uses power analysis/ reverse engineering techniques to extract ID_M or launches an offline identity/password guessing attack. In the first case, the attacker, let's suppose, reaches the different arguments MID_M, R_M, PK_{UP}, SK_{UP}, PK_{PU} SK_{PU}, h and tries to identify ID_M, ID_D, or ID_G as it is fully chained with 160-bits key and then hashed so that the attacker couldn't get success. To do so, an attacker must know X_M or N_M, SK_M, which is impossible for them. In the second case, an attacker must extract elliptic curve random points over a finite field E/F_q, which they cannot perform such a considerable calculation even in months.

Similarly, if an attacker stole a mobile device (M) or a legal user lost it somewhere, the attacker struggles to identify the identity and other credentials. They couldn't find it due to no knowledge of MID_M=h($ID_M||N_M$) and h_M=H_1($MID_M||R_M$). Also, they cannot pass through this SP_M?=$R_M \oplus H_1$ ($MID_M||R_M$)$||PK_C$ due to 160-bits random nonce, secret identity exchange, and frivolous hash values length. Therefore, the proposed protocol strongly resists stolen verifier attack.

3) UNTRACEABILITY OF PEER

If an adversary capture a message (MID_M, c_{M3}, Sig_{M3}), (MID_D, R_D, X_{D2}) or (MID_G, R_G, X_{G2}, CHK_{UG}) transmitted over an open channel. Mobile-device identity $ID_M$is in

(MID_M, c_{M3}, Sig_{M3}) message having 160-bits of big nonce value, drone's identity ID_D is in (MID_D, R_D, X_{D2}) message and ID_G in (MID_G, R_G, X_{G2}, CHK_{UG}) message. Adversary must first extract the random nonce or public-private keys pair, i.e., X_M, X_D, or X_G, to identify any identity. Also, each identity is different for a different session. Furthermore, all the other parameters in the mentioned messages are computed based on random numbers and TS (timestamps). So identity exposure or traceability is wholly hidden in the proposed protocol.

4) OUTDATED DATA TRANSMISSION FLAW

Before passing to the next round trip, each participant must check whether the data arrived in the pre-defined time threshold or not. If the answer is no, the participant considered it outdated data and discarded it for potential replay attack, else, smoothly performing onward operations. This means that our protocol removes Chen *et al.*'s [42] outdated data transmission flaw.

5) SPOOFING ATTACK

Suppose an adversary represents him/her(self) as a legitimate peer and sends false messages towards GCS or D. Each one first checks timestamp and verify $(M_R||certD)?=V_{PKM}(Sig_{M3})$, $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$, and $PK_D=R_D\oplus H_1(MID_D||R_D)||PK_C$, which cannot get/pass to the next phase. Adversary failed to spoof any peer at any stage due to the matching of parameters at each stage. Therefore, the proposed protocol is safe against spoofing attack.

6) REPLAY ATTACK

If an adversary capture any message (MID_M, R_M, X_M, TS_1), (MID_D, R_D, X_D, CHK_{PU}, TS_2), or (MID_M, R_M, X_{M2}, TS_1), (MID_G, R_G, X_G, CHK_{PG}, TS_2), or (Sig_G, MID_G, R_G, X_G, TS_1), (Sig_D, MID_D, R_D, X_D, TS_2), or (MID_M, c_{M3}, Sig_{M3}), (MID_D, R_D, X_{D2}, TS_1), (MID_G, R_G, X_{G2}, CHK_{UG}, TS_2), or (MID_G, R_D, X_{G2}, CHK_{UG}, TS_3) and replay it some other times towards GCS or drone (D) or mobile-device (M). Each peer first checks its time threshold with its current time; if found outdated, discard and consider a potential replay attack. Therefore, the proposed protocol is much more secure against replay attack.

7) DOS ATTACK

Suppose an attacker copies (MID_D, R_D, X_{D2}) message from the open network channel and repeatedly starts transmission towards GCS. Due to the selection of mutual random keys PK_G, SK_G, PK_D, and SK_D at both sides and the offering of signature function and its verification function, i.e. $Sig_{M3}=S_{SKM}(M_R||cert_D)$, $(M_R||cert_D)?=V_{PKM}(Sig_{M3})$, they couldn't pass to the next step. Also, key generation is different for different sessions at both ends, and the server cannot compute the session key without knowing the random number on both sides. For such an attempt, the GCS first checks mutually calculated identities, arbitrary numbers, identities in its database, and confirmation of freshness of every message; a DoS attack is impossible in the proposed protocol.

Similarly, upon authenticating M with D, if the adversary, for example, copied (MID_M, R_M, X_M, TS_1) message and sent towards D for getting helpful information and cannot succeed, alternately sent hundreds of thousand messages for disturbing the standard functionalities, such an attempt cannot accomplish, due to timestamp, the adversary requests will be discarded in the first phase; Or (MID_D, R_D, X_D, CHK_{PU}, TS_2) message towards M, due to timestamp, the M considered it outdated and discarded for potential DoS attack. Also, during the authentication of M with GCS, the GCS contains the predefined time threshold, if within the limit, accept, else, denied to prevent DoS attack and vice versa. Correspondingly, in the authentication of GCS with D, if the attacker copied (Sig_G, MID_G, R_G, X_G, TS_1) message, the D first check the time interval, and secondly confirms $(MID_D||R_G)?=V_{PKG}(Sig_G)$, while the GCS, also, first checks the timestamp, and then verifies $(MID_M||R_D||M_{GPS}||cert_D)?=V_{PKD}(Sig_{D4})$ which means to prevent denial-of-service (DoS) attack. Furthermore, upon authenticating M, D, and GCS, due to $(M_R||cert_D)?=V_{PKM}(Sig_{M3})$, $TS_c\text{-}TS_1 \leq \Delta TS$, $TS_c\text{-}TS_2 \leq \Delta TS$, $SEK_{UG}?=H2(K_{UG1}||K_{UG2})$, $TS_c\text{-}TS_3 \leq \Delta TS$, and $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$ messages can guarantee for DoS attack. Therefore, the proposed protocol is strong against a DoS attack.

8) CLOGGING ATTACK

If an attacker desires to launch a clogging attack, he/she has to send a fake message (MID_D, R_A, X_{D2}) towards GCS. For doing so, the attacker must first generate a public-private key pair X_D, and random number R_A and simulates it by calculating $PK_D=R_D\oplus H_1(MID_D||R_D)||PK_C$, $K_{GU1}=S_G||(X_{D2}\oplus f)||PK_D$, $SEK_{GU}=H_2(K_{UG1}||K_{UG2})$ and $CHK_{UG}=H_3(SEK_{UG}||X_{D2})$. For such a calculation, an attacker must identify the curve points by flipping a coin to win $(M_R||cert_D)?=V_{PKM}(Sig_{M3})$ or $(M_R||cert_D) \neq V_{PKM}(Sig_{M3})$ and $(M_R||cert_D)=D_{SEKup}(c_{M3})$. But doing such a complicated calculation requires the drone's identity MID_D, X_{D2}, and the previously computed value $eP||TS_1$.

Similarly, if the attacker transmits (MID_G, R_D, X_G, CHK_{UG}) message, he/she must correctly send the message in the pre-defined time threshold, which is not possible $TS_c\text{-}TS_2 \leq \Delta TS$. Also, the attacker must identify MID_D, random number R_D, and PK_C by calculating $PK_D=R_D\oplus H_1(MID_D||R_D)||PK_C$. Next, he/she has to extract two random points in the curve, compute $K_{UG1}=S_M||(X_D\oplus g)||PK_D$, and confirms: $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$ and $CHK_{UG}?=H_3(SEK_{UG}|\backslash X_{D2})$ which is not possible. The proposed protocol can detect clogging attack in both cases because the attacker couldn't pass from $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$ and $CHK_{UG}?=H_3(SEK_{UG}|\backslash X_{D2})$ authentication check. Therefore, the proposed protocol strongly resists a clogging attack.

9) PHYSICAL CAPTURE ATTACK

Due to the addition of drone dynamically to the network at any time, it is necessary to evaluate the proposed protocol for drone capture attack. Let an adversary captures a drone and extracts N_D, R_D, public key (PK_D), secret key (SK_D) and $S_D=r_D \oplus h_D s$. The attacker even attempted for months but couldn't calculate PK_D and SK_D as they have unknown points on the x-axis and y-axis of the elliptic curve over finite filed F_q. Similarly, he/she has to compute $MID_D=h(ID_D||N_D)$ and $R_D=r_D P$, $h_D=H_1(MID_D||R_D)$, and $S_D=r_D \oplus h_D s$; and identify the stored (R_D, S_D, PK_D, SK_D) parameters, confirming $S_{DP}?=R_D \oplus H_1(ID_D||R_D)||PK_C$. All these calculations are not possible for anyone at anytime and anywhere. These calculations also restrict attackers from deploying a drone in the network, and if deployed, for example, they cannot establish a secure session among ground control station due to several checks. Therefore, by capturing D, the attacker cannot settle a session with GCS, M and other participants of IoD. The compromised drone does not result in ensuring secure communications with ground-control-station (GCS) and D. As a result; our protocol is unconditionally secure against drone capture attack.

10) EPHEMERAL-SECRET-LEAKAGE (ESL) ATTACK

If an attacker obtained the ephemeral secret of drone like N_D and R_D using ESL attacks, he/she needs to verify $(MID_D||R_G)?=V_{PKG}(Sig_G)$ $X_D= tP||TS_2$and $Sig_D=Enc_t(MID_M||R_D||\ M_{GPS}||cert_D)$. Also, if he/she obtains N_G and R_G using power analysis or reverse engineering techniques, still he/she needs to authenticate/confirms $(MID_M||R_D||M_{GPS}||\ cert_D)?=V_{PKD}(Sig_{D4})$. In the next round, suppose, the attacker recovers X_{D2}; he/she has to pass $PK_D=R_D \oplus H_1(MID_D||R_D)||PK_C$. And, if they obtain PK_C or PK_D, he/she must has to solve $SEK_{GU}=H_2(K_{UG1}||K_{UG2})$, $CHK_{UG}=H_3(SEK_{UG}||X_{D2})$, $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$ and $CHK_{UG}?=H_3(SEK_{GU}||X_{D2})$. So far, without knowing the secret values of M, D, and GCS, an adversary cannot succeed in exacting secret values. Therefore, the proposed protocol resists ESL attack.

11) MODIFICATION ATTACK

Assume that an adversary obtains access to the confidential information (R_D, S_D, PK_D, SK_D) stored in the memory of a drone (D) or (R_M, S_M, PK_M, SK_M) stored in the mobile device (M) and diverts all messages sent between the Drone (D) and the GCS. After that, the adversary uses an offline dictionary attack to see if his/her guesses are correct. The adversary needed additional information in this attack, like knowing the stored credentials in the memories of different participants like N_D, $R_D=r_D P$, MID_D and N_M, $R_M=r_M P$, MID_M. However, without knowing N_D, $R_D=r_D P$, MID_D, and N_M, $R_M=r_M P$, the extra details cannot assist the adversary in correctly guessing the drone's/Mobile-identity. He/she must know the GCS secret key r_G, N_G and $R_G=r_G P$, which is impossible for him/her to identify. Therefore, the modification attack with the drone, mobile device, or GCS is invalid in the proposed protocol.

12) DENNING-SACCO ATTACK

Suppose an adversary gets the previous session key. The adversary cannot deduce the drone's identity from the old session key because the session key is made up of three random integers chosen separately by the mobile device (M), drone (D), and GCS and are unrelated to the identity MID_D, GCS's secret key r_G. But even if the adversary compromises an old session key, they can't find the R_D, MID_D, PK_D, SK_D for drone D or the private key r_G for the GCS.

In addition, a new session key is created for each session based on the integer chosen by the Drone, Mobile Device and GCS. As a result, even though the adversary compromises an old session key, they will be unable to acquire new session keys because the session key $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$ is not connected in any manner. Therefore, the proposed protocol can resist Denning-Sacco attacks.

13) MAN-IN-THE-MIDDLE ATTACK

In the authentication phases, let an adversary struggle to inject, delete, insert or modify the messages (MID_M, R_M, X_M, TS_1), (MID_D, R_D, X_D, CHK_{PU}, TS_2), or (MID_M, R_M, X_{M2}, TS_1), (MID_G, R_G, X_G, CHK_{PG}, TS_2) or (Sig_G, MID_G, R_G, X_G, TS_1), (Sig_D, MID_D, R_D, X_D, TS_2) in believing the participant for receiving the messages from a legitimate participant. To do so by an adversary means man-in-the-middle attack, let adversary modifies (MID_M, R_M, X_M, TS_1) message or (MID_D, R_D, X_D, CHK_{PU}, TS_2) message, (MID_M, R_M, X_{M2}, TS_1) message or (MID_G, R_G, X_G, CHK_{PG}, TS_2), and (Sig_G, MID_G, R_G, X_G, TS_1), (Sig_D, MID_D, R_D, X_D, TS_2) messages. Adversary failed for such an attempt due to no knowledge of MID_M, MID_D, MID_G, CHK_{PU}, CHK_{PG}, and Sig_D, Sig_G. Similarly, for (MID_M, c_{M3}, Sig_{M3}), (MID_D, R_D, X_{D2}, TS_1), (MID_G, R_G, X_{G2}, CHK_{UG}, TS_2) and (MID_G, R_D, X_{G2}, CHK_{UG}, TS_3) messages, an adversary's attempt cannot be successful due to the involvement of randomness in messages, timestamps, secrets and 160-bits ECC keys. *A* cannot make an independent connection for computing session shard key due to no knowledge of secret credentials, identities, and random numbers. Therefore, the proposed key agreement protocol withstands man-in-the-middle attack.

14) PERFECT FORWARD SECECY

The 160-bit long keys of mobile-device (a, c, and g), drone (b, t, and e), and ground-control-station (d, r, and f) are computed randomly for each session. Suppose an adversary can extract these keys from the previous session key; he/she needs to extract a form $K_{UP1} = S_M||(X_D \oplus a)||PK_D$, c from $K_{PG1}=S_M||(X_G \oplus c)||PK_G$, and g from $K_{UG1}=S_M||(X_D \oplus g)||PK_D$ and vice versa. However, an adversary cannot extract any of these keys from the captured information without knowing $K_{UP2}=aX_D$, $X_D= bP||TS_2$, $X_{M2}=cP||TS_1$, $K_{GP2}=dX_{M2}$, $X_G=rP||TS_1$, $X_D= tP||TS_2$ and secret information $X_{D2}=eP||TS_1$, $X_{G2}=$

$fP||TS_2$, $K_{UG2}= eX_{G2}$, and $K_{UG1}= gX_M$. Also, the adversary cannot obtain $(X_D \oplus a)$, $(X_G \oplus c)$, and $(X_D \oplus g)$ form $SEK_{UP}=H_2(K_{UP1}||K_{UP2})$, $CHK_{PU}=H_3(SEK_{UP}||X_M)$, $SEK_{GP}=H_2(K_{GP1}||K_{GP2})$, $CHK_{PG}=H_3(SEK_{GP}||X_{M2})$, and $Sig_D=Enc_t(MID_M||R_D||M_{GPS}||cert_D)$ as these are fully protected in collision free one way hash functions, and is in cipher form. Even if A extracts $(X_D \oplus a)$, $(X_G \oplus c)$, and $(X_D \oplus g)$, he/she cannot compute the session key without knowing hash values. It means that the secrecy of the previous session is not affected, even if the adversary can identify the long-term secret key, but still, A cannot succeed for hashed and encrypted values. Therefore, the proposed key-agreement protocol satisfies the feature of perfect forward secrecy.

15) INSIDER THREAT

In the registration phase, drone submits $MID_D=h(ID_D||N_D)$, mobile-device submits $MID_M=h(ID_M||N_M)$, and GCS submits $MID_G=h(ID_G||N_G)$ to TAC, whereas N_D, N_M, and N_{GCS} are the random values. The administrator doesn't get ID_D, ID_M, and ID_{GCS}. Also, suppose an adversary reached internally to GCS and tried to figure out the internally stored secret. In that case, he/she cannot access it due to many long-term ECC keys, identities, and signature/verification functions. Therefore, any insider attempt at the proposed protocol is strongly unsuccessful. Hence, the proposed protocol withstands insider threat.

16) KEY SECRECY

Each participants cross-verify messages like $(M_R||cert_D)?=V_{PKM}(Sig_{M3})$, $SEK_{UG}?=H_2(K_{UG1}||K_{UG2})$, $(MID_D||R_G)?=V_{PKG}(Sig_G)$, and $(MID_M||R_D||M_{GPS}||cert_D)?=V_{PKD}(Sig_{D4})$ for confirming session shared keys $CHK_{UG}?=H_3(SEK_{UG}||X_{D2})$ $SEK_{UP}?=H_2(K_{UP1}||K_{UP2})$, $CHK_{PU}?=H_3(SEK_{UP}||X_M)$, $SEK_{GP}?=H_2(K_{PG1}||K_{PG2})$, and $CHK_{PG}?=H_3(SEK_{GP}, T_{PMD2})$ which in turn means key secrecy.

17) DESYNCHRONIZATION ATTACK

In the proposed security mechanism, R_D, S_D, PK_D, SK_D are updated R_D^{new}, S_D^{new}, PK_D^{new}, and SK_D^{new} on the drone side and calculated at the mobile side. Even though if adversary intercepts, create no hurdle for SK in the upcoming session key among all the three participants. Therefore, the proposed scheme is safe against desynchronization attack.

18) UNLINKABILITY AND ANONYMITY

A drone's identity ID_D or any other message transmitted over public channels is much more complicated for an adversary to compute. Because any identity is linked with several other parameters, the adversary first struggles to compute the associated credentials for reaching the identity of participants. Similarly, each session extracts new random numbers of size 160-bits, records a separate timestamp and then concatenates with the identity to ensure anonymity and Unlink-ability. As these credentials are rugged for an adversary to trace; therefore, the proposed security scheme for each session is anonymously started and guarantees Unlink-ability.

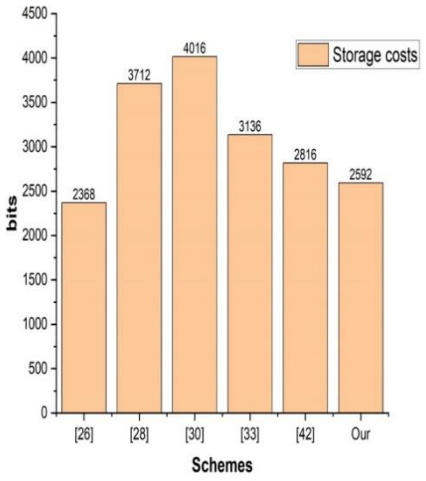

FIGURE 2. Storage overheads comparsions graph.

TABLE 2. Storage overheads analysis.

Protocol	Storage in Bits
Tian et al. [26]	2668
Won et al. [28]	3712
Challa et al. [30]	4016
Turkanovic et al. [33]	3136
Chen et al. [42]	2816
Our	2592

19) KNOWN KEY ATTACK

Suppose A knows the session key, as each session starts with a different SK; therefore, A doesn't launch any attack; therefore, knowing the session key creates no problem for the IoD.

VI. PERFORMANCE ANALYSIS

This section evaluates the proposed protocol's performance and compares it with state-of-the-art protocols for satisfying all the security functionalities in detail. These security functionalities are discussed one by one as under:

A. STORAGE OVERHEADS ANALYSIS

The storage overheads mean the parameters stored in the memory during the registration phase of the proposed scheme. According to [44], identity requires 64 bits of space, timestamp 56 bits, ECC key is 160 bits, the output of hash map using SHA-1 is 160 bits, while E_{nc}/D_{ec} functions need 192 bits of space. So far, the parameters stored in the memory of the drone (D), mobile device (M), and ground control station (GCS) are shown in Table 2.

The proposed protocol has much better storage overheads compared to Tian *et al.* [26], Won *et al.* [28], Challa *et al.* [30], Turkanovic *et al.* [33], and Chen *et al.* [42], the graphical representation of storage overhead is shown in figure 2.

B. COMMUNICATION OVERHEADS ANALYSIS

To evaluate communication cost or estimate this feature, we will consider [45] in which the identity,

TABLE 3. Communication overheads comparison.

Protocol	Messages	Costs in Bits
Tian et al. [26]	2	5856
Won et al [28]	1	2252
Challa et al. [30]	3	2408
Turkanovic et al. [33]	4	2960
Nikooghadam et al. [38]	3	2336
Chen et al. [42]	6	3072
Ever [43]	3	2272
Odelu et al. [46]	3	1376
Kumar et al. [47]	3	2368
Our	4	2240

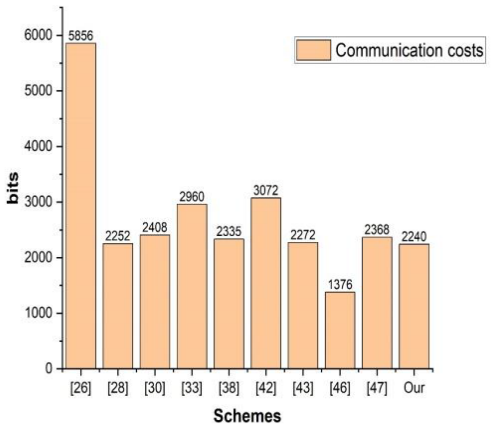

FIGURE 3. Communication cost comparsion graph.

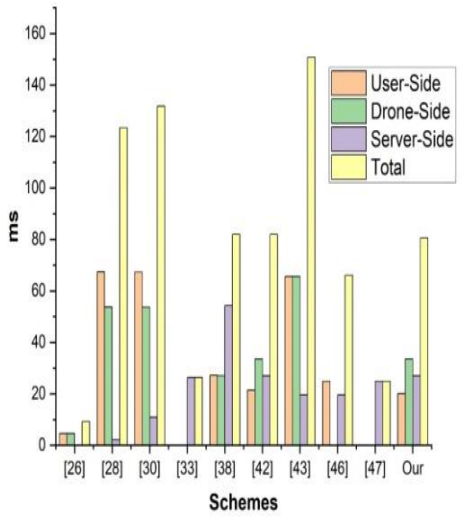

FIGURE 4. Computation time comparsion graph.

ECC key, timestamp, and Encryption/Decryptions occupy 64 bits, 160 bits, 56 bits, and 160 bits, and 192 bits of memory space, respectively. Figure 3 demonstrates the communication cost in bits of the proposed protocol with state-of-the-art protocols like Tian *et al.* [26], Won *et al.* [28], Challa *et al.* [30], Turkanovic *et al.* [33], Nikooghadam *et al.* [38], Chen *et al.* [42] Ever [43], Odelu *et al.* [46], and Kumar *et al.* [47] at the authentication key generation process.

TABLE 4. Execution time comparison in milliseconds.

Protocol	User-side (M)	Drone-side (D)	Server-Side (GCS)	Total
[26]	4.605	4.605	000	9.21
[28]	67.417	53.732	2.172	123.321
[30]	67.305	53.641	10.853	131.799
[33]	XXX	XXX	26.34	26.34
[38]	27.146	27.09	54.292	54.292
[42]	21.417	33.494	27.004	81.915
[43]	65.594	65.594	19.577	150.765
[46]	24.875	XXX	41.26	66.135
[47]	XXX	XXX	24.876	24.876
Our	20.081	33.494	26.994	80.569

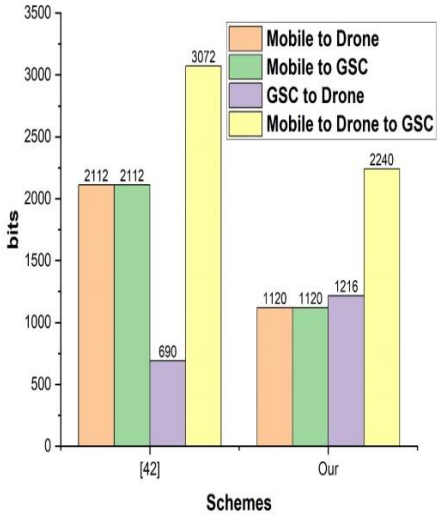

FIGURE 5. Comparison with [42] graph.

The result in table 3 shows that our proposed authentication protocol has a better performance compared to Tian *et al.* [26], Won *et al.* [28], Challa *et al.* [30], Turkanovic *et al.* [33], Nikooghadam *et al.* [38], Chen *et al.* [42] and Ever [43]. The difference in the proposed protocol with Tian *et al.* [26], Won *et al.* [28], Challa *et al.* [30], Turkanovic *et al.* [33], Nikooghadam *et al.* [38], Ever [43], Odelu *et al.* [46], and Kumar *et al.* [47] in term of communication costs in bits is also shown graphically in Figure 3. But it is to mention that [43] is for network-enabled IoT using Elliptic Curve cryptographic techniques.

C. COMPUTATION OVERHEADS ANALYSIS

We use the experimental findings from [38] and [45], [46] to determine the execution time complexity or computation costs. They [38] used a Samsung Galaxy S5 (CPU: 2.45GHz Quad-core; RAM: 2BG; Android OS version 4.4.2) and a Dell PC (Processor: Intel Core i5-4460S, CPU: 2.90GHz; RAM: 4GB, and Windows OS of version 8.1), for various cryptographic operations. The PC is regarded as the GCS, the cell phone as for the D or M. The result obtained for different processes on using these resources are given as under:

- T_{PA} means the execution time for Elliptic Curve Point Addition=0.081ms in drone/mobile device and 0.013ms in GCS.

TABLE 5. Comparison of necessary security functionalities.

Attribute	[26]	[28]	[30]	[33]	[38]	[42]	[43]	[46]	[47]	Our
SF1	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
SF2	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES
SF3	NO	YES	YES	YES	NO	NO	YES	YES	YES	YES
SF4	YES	YES	YES	YES	YES	NO	NO	NO	YES	YES
SF5	YES	YES	NO	YES	YES	YES	YES	NO	YES	YES
SF6	N/A	YES	YES	YES	NO	YES	YES	YES	YES	YES
SF7	YES	YES	NO	NO	NO	NO	YES	YES	YES	YES
SF8	YES	N/A	YES	YES	YES	NO	NO	YES	NO	YES
SF9	YES	NO	YES	YES	YES	YES	YES	YES	NO	YES
SF10	NO	YES	NO	NO	YES	YES	YES	YES	YES	YES
SF11	YES	YES	N/A	N/A	YES	YES	YES	YES	NO	YES
SF12	YES	YES	YES	YES	NO	NO	NO	NO	YES	YES

TABLE 6. Communication cost comparison with chen et al. [42] protocol.

Peer	Total No. of Messages	Size of each Message in Bits	Total Size of Messages in Bits	Peer	Total No. of Messages	Size of each Message in Bits	Total Size of Messages in Bits
CHEN ET AL. [42] PROTOCOL				PROPOSED PROTOCOL			
I	4	480 640 512 480	2112	I	2	480 640	1120
II	4	480 640 512 480	2112	II	2	480 640	1120
III	2	480 480 480	690	III	2	608 608 480	1216
IV	6	480 640 512 480 480	3072	IV	4	480 640 640	2240

I: Mobile-Device (M) →Drone (D), II: Mobile-Device (M) →Ground-Control-Station (GCS), III: Ground-Control-Station (GCS) → Drone (D), IV: Mobile-Device (M) →Drone (D) →Ground-Control-Station (GCS).

- T_{SM} means the Elliptic Curve Scalar Multiplication Operation = 13.405ms in drone/mobile device and 2.165ms in GCS
- T_{hash} means the Hash Function Execution Time = 0.056ms 405ms in drone/mobile device and 0.007ms in GCS.
- T_{σ} means the time require for signature generation/verification = 0.081ms 405ms in drone/mobile device and 0.027ms in GCS.
- T_{Mul} means the execution time of multiplication operation = 0.008ms 405ms in drone/mobile device and 0.001ms in GCS.
- T_{Enc} means the running time for encryption Operation = 3.2500ms 405ms in drone/mobile device and 3.2500ms in GCS.
- T_{Dec} means the running time for decryption Operation = 3.2500ms 405ms in drone/mobile device and 3.2500ms in GCS.

Therefore, keeping in view the aforementioned results/findings, in the proposed protocol, mobile-device (M) executes $1T_{SM}$, $3T_{hash}$, $1T_{Mul}$ which is collectively equal to $2T_{Enc}$, $1T_{MS}+3T_{hash}+1T_{Mul}+2T_{Enc} \approx 20.081$ms; drone (D) executes $2T_{MS}$, $3T_{hash}$, $2T_{Mul}$ and $2T_{Dec}$, equal to $2T_{MS}+3T_{hash}+2T_{Mul}+2T_{Dec} \approx 33.494$ms and ground-control-station (GCS) executes $2T_{MS}$, $3T_{hash}$ and $2T_{Mul}$, equal to $2T_{MS}+3T_{hash}+2T_{Mul} \approx 26.994$ms. Therefore, keeping in view all these computations, the total computation/execution time in millisecond for the proposed protocol is 80.569 ms as shown in table 4.

Figure 4 shows the comparison of the proposed protocol in terms of execution time complexity with [26], [28], [30], [33], [38], [42], [43], [46], and [47].

D. SECURITY AND FUNCTIONALITY COMPARISON

Let suppose SF1 represents session key agreement, SF2 formal verification, SF3 Mutual Authentication, SF4 resists to known session key attack, SF5 resists replay attack, SF6 resists impersonation attack, SF7 resists stolen-verifier attack, SF8 support forward secrecy, SF9 support of anonymity, SF10 withstands ESL attack, SF11 resists drone physical capture attack and SF12 safe against privileged insider attack. Table 5 shows that the proposed protocol fulfills all the given necessary security functionalities comparing with [26], [28], [30], [33], [38], [42], [43], [46], and [47].

E. COMPARISON WITH CHEN ET AL. [24]

If we compare the proposed scheme with Chen *et al.* [42] protocol only, the results show a clear difference in communication cost. The messages exchanged among different participants, parameters, message values in bits and total cost as shown in table 6. The communication cost of [42] is much higher than that of the proposed scheme. For example, in the messages exchanged among different participants (Mobile-Device (M) → Drone (D) → GCS), which is denoted as IV, the communication cost in [42] is 3072 and in the proposed is 2240 bits. The overall result shows that the proposed protocol is lightweight and robust and has achieved balancing of ''security with performance'' compared to Chen *et al.* [42] protocol, as shown in figure 5.

VII. CONCLUSION

This research article uses the infrastructure-less, resourceless, and self-organizing network FANET for the IoD environment, which requires a robust security mechanism. So far, we have successfully achieved the goal of designing an ECC-based robust and lightweight scheme for IoD deployment drone. We have first designed a controlled infrastructure for IoD (network model, defined different entities involved in it like mobile-device, ground-control-station, and drone), specified the role of adversary, and demonstrated possible threats to the system. After all these efforts, next, we have designed a protocol that can guarantee fast and secure communication in IoD and can resist all known threats. The proposed scheme's security analysis and performance assessment sections have been efficiently tackled. The result obtained in this article based on the ECC technique shows that it is a much more secure, efficient, and effective method. When comparing it with other protocols, we conclude that it is stunningly efficient and can be recommended for operationalizing in IoD for infrastructure surveillance after severe damage due to flood or earthquake. In the future, we plan to present a survey paper for our previous work regarding the security and performance trade-off, sum up all the security schemes and give their combined effect on the system.

APPENDIX-A

ProVerif2.03 is used to confirm the session key secrecy, confidentiality, and reachability among mobile-device (M) and drone (D) and mobile-device (M) and ground-control-station (GCS).

PROVERIF2.03 SIMULATION (MOBILE-DEVICE (M) AND DRONE (D), MOBILE-DEVICE (M) AND GROUND-CONTROL-STATION (GCS))

CODE: MOBILE DEVICE & DRONE

```
(*=======Channels======*)
free ChSec:channel [private].
free ChPub:channel.
(*=======Constants and Variables=======*)
free IDM:bitstring.
free IDD:bitstring.
free MIDm:bitstring.
free MIDd:bitstring.
free PKc:bitstring [private].
free SEKup:bitstring [private].
free TS:bitstring.
free TS2:bitstring.
free CHKpu:bitstring.
free Sm:bitstring.
free Sd:bitstring.
free Rm:bitstring.
free Rd:bitstring.
const P: bitstring.
(*=======Queries=======*)
query attacker(SEKup).
query id:bitstring; inj-event(end_M(IDM)) ==>
inj-event(start_M(IDM)).
query id:bitstring; inj-event(end_D(IDD)) ==>
inj-event(start_D(IDD)).
(*=====*Events*=====*)
event start_M(bitstring).
event end_M(bitstring).
event start_D(bitstring).
event end_D(bitstring).
(*========Constructors=======*)
fun h(bitstring): bitstring.
fun Concat(bitstring,bitstring): bitstring.
fun XOR(bitstring,bitstring): bitstring.
fun ECPM(bitstring, bitstring): bitstring.
fun Add(bitstring, bitstring): bitstring.
(*======Equations=======*)
equation forall a: bitstring, b: bitstring;
XOR(XOR(a,b),b)=a.
(*=======Login and Authentication=======*)
(*=======User=======*)
let pM=
event start_M(IDM);
new a: bitstring;
let Xm=Concat((ECPM(a,P)),TS) in
out(ChPub,(MIDm, Rm, Xm));
in(ChPub,(MIDd:bitstring, Rd:bitstring, Xd:bitstring,
CHKpu:bitstring));
let PKd=XOR(Rd,h(Concat(MIDd,(Rd,PKc)))) in
let Kup1=Concat(Sm,(Add(Xd,a),PKd)) in
let Kup2=ECPM(a,Xd) in
if SEKup=h(Concat(Kup1,Kup2)) then
if CHKpu=h(Concat(SEKup,Xm)) then
event end_M(IDM)
else
0.
(*=======Login and Authentication=======*)
(* =====*Drone* ======*)
let pD=
event start_D(IDD);
in(ChPub,(MIDm:bitstring, Rm:bitstring,Xm:bitstring));
new b:bitstring;
let Xd=ECPM(b,P) in
let PKm=XOR(Rm,(h(Concat(MIDm,(Rm,PKc))))) in
let Kup1=Concat(Sd,(Add(Xm,b),PKm)) in
let Kup2=Concat(ECPM(b,Xm),TS2) in
if SEKup=h(Concat(Kup1,Kup2)) then
if CHKpu=h(Concat(SEKup,Xm)) then
out(ChPub,(MIDd,Rd, Xd, CHKpu));
event end_D(IDD)
else 0.
process ( (!pD) | (!pM))
```

RESULT: MOBILE DEVICE & DRONE

```
--------------------------------------------------------------
Verification summary:
Query not attacker(SEKup[]) is true.
Query inj-event(end_M(IDM[])) ==>
```

```
inj-event(start_M(IDM[])) is true.
Query inj-event(end_D(IDD[])) ==>
inj-event(start_D(IDD[])) is true.
------------------------------------------------------------
```

CODE: MOBILE-DEVICE AND GCS

```
(*=======Channels=======*)
free ChSec:channel [private]. (*secure channel between
MD and GCS*)
free ChPub:channel.(*publicchannel between MD and GCS*)
(* ========Constants and Variables=======*)
free IDMD:bitstring.
free IDGCS:bitstring.
free MIDm:bitstring.
free MIDg:bitstring.
free PKc:bitstring [private].
free PKd:bitstring [private].
free SEKgp:bitstring [private].
free TS1:bitstring.
free TS2:bitstring.
free Tpmd2:bitstring.
free CHKpg:bitstring.
free Sm:bitstring.
free Sg:bitstring.
free Xd:bitstring.
free Rm:bitstring.
free Rm2:bitstring.
free Rg:bitstring.
const P: bitstring.
(* ===-=Queries=====*)
query attacker(SEKgp).
query id:bitstring; inj-event(end_MD(IDMD)) ==>
inj-event(start_MD(IDMD)).
query id:bitstring; inj-event(end_GCS(IDGCS)) ==>
inj-event(start_GCS(IDGCS)).
(* =====*Events* =====*)
event start_MD(bitstring).
event end_MD(bitstring).
event start_GCS(bitstring).
event end_GCS(bitstring).
(* ========Constructors=======*)
fun h(bitstring): bitstring.
fun Concat(bitstring,bitstring): bitstring.
fun XOR(bitstring,bitstring): bitstring.
fun ECPM(bitstring, bitstring): bitstring.
fun Add(bitstring, bitstring): bitstring.
(* ======Equations=======*)
equation forall a: bitstring, b: bitstring;
XOR(XOR(a,b),b)=a.
(*=======Login and Authentication=======*)
(*=======Mobile Device=======*)
let pMD=
event start_MD(IDMD);
new c: bitstring;
let Xm2=Concat((ECPM(c,P)),TS1) in
out(ChPub,(MIDm, Rm2, Xm2));
in(ChPub,(MIDg:bitstring, Rg:bitstring, Xg:bitstring,
CHKpg:bitstring));
let PKg=XOR(Rg,h(Concat(MIDg,(Rg,PKc)))) in
let Kpg1=Concat(Sm,(Add(Xg,c),PKd)) in
let Kpg2=ECPM(c,Xg) in
if SEKgp=h(Concat(Kpg1,Kpg2)) then
if CHKpg=h(Concat(SEKgp,Tpmd2)) then
event end_MD(IDMD)
else
0.
(*=======Login and Authentication=======*)
(* =====*Drone* =======*)
let pGCS=
event start_GCS(IDGCS);
in(ChPub,(MIDm:bitstring,Rm2:bitstring,Xm2:bitstring));
new d:bitstring;
let Xg=ECPM(d,P) in
let PKm=XOR(Rm,(h( Concat(MIDm,(Rm,PKc))))) in
let Kgp1=Concat(Sg,(Add(Xm2,d),PKm)) in
let Kgp2=Concat(ECPM(d,Xm2),TS2) in
if SEKgp=h(Concat(Kgp1,Kgp2)) then
if CHKpg=h(Concat(SEKgp,Xm2)) then
out(ChPub,(MIDg,Rg, Xd, CHKpg));
event end_GCS(IDGCS)
else 0.
process ( (!pGCS) | (!pMD))
```

RESULT: MOBILE-DEVICE AND GCS

```
------------------------------------------------------------
Verification summary:
Query not attacker(SEKgp[]) is true.
Query inj-event(end_MD(IDMD[])) ==>
inj-event(start_MD(IDMD[])) is true.
Query inj-event(end_GCS(IDGCS[])) ==>
inj-event(start_GCS(IDGCS[])) is true.
------------------------------------------------------------
```

APPENDIX-B

Also, to verify mutual authentication and the secrecy, confidentiality and reachability of secret session shared key between ground-control-station (GCS) and drone (D) as well as in all three participants, again, we have used the ProVerif2.03 software toolkit. The code below confirms that attackers couldn't reach for any session key, and their attack cannot affect the SK. Therefore, the proposed protocol is provable and secure against all known attacks.

PROVERIF2.03 SIMULATION (GROUND-CONTROL-STATION (GCS), DRONE (D), MOBILE-DEVICE (M), DRONE (D) AND GROUND-CONTROL-STATION (GCS))

CODE: GCS AND DRONE

```
(* ----- Channels ----- *)
free ChSec:channel [private]. (*secure channel between
GCS and D*)
free ChPub:channel. (*public channel between GCS and D*)
(*=======Constants and Variables=======*)
free IDGCS:bitstring.
free IDD:bitstring. free MIDm:bitstring.
free MIDd:bitstring. free Mgps:bitstring.
free MIDg:bitstring. free PKc:bitstring.
free certd:bitstring. free TS1:bitstring.
free TS2:bitstring. free Rg:bitstring.
free Rd:bitstring.
free VpkgSigd4:bitstring [private].
const P: bitstring.
(*=======Queries=======*)
query attacker(VpkgSigd4).
query id:bitstring; inj-event(end_GCS(IDGCS)) ==>
inj-event(start_GCS(IDGCS)).
query id:bitstring; inj-event(end_D(IDD)) ==>
inj-event(start_D(IDD)).
(*=====*Events*=====*)
event start_GCS(bitstring).
event end_GCS(bitstring).
event start_D(bitstring).
event end_D(bitstring).
```

```
(* ========Constructors=======*)
fun h(bitstring): bitstring.
fun Concat(bitstring,bitstring): bitstring.
fun XOR(bitstring,bitstring): bitstring.
fun Encr(bitstring): bitstring.
fun ECPM(bitstring,bitstring): bitstring.
fun Enct(bitstring): bitstring.
fun Add(bitstring, bitstring): bitstring.
(*======Equations=======*)
equation forall a: bitstring, b: bitstring;
XOR(XOR(a,b),b)=a.
(*-Login and Authentication-*)
(*=======GCS=======*)
let pGCS=
event start_GCS(IDGCS);
new r: bitstring;
let Xg=Concat((ECPM(r,P)),TS1) in
let SIGg=Encr(Concat(MIDd,Rg)) in
out(ChPub,(SIGg,MIDg, Rg, Xg));
in(ChPub,(SIGg:bitstring,MIDd:bitstring,Rd:bitstring,
Xd:bitstring));
let PKd=XOR(Rd,h(Concat(MIDd,(Rd,PKc)))) in
if VpkgSigd4=h(Concat(MIDm,(Rd,Mgps,certd))) then
event end_GCS(IDGCS)
else 0.
(*=======Login and Authentication=======*)
(*=====*Drone*======*)
let pD=
event start_D(IDD);
in(ChPub,(SIGg:bitstring, MIDg:bitstring,
Rg:bitstring, Xg:bitstring));
if VpkgSigd4=Concat(MIDd,Rg) then
new t:bitstring;
let Xd=Concat(TS2,ECPM(t,P)) in
let SIGd=Enct(Concat(MIDm,(Rd,Mgps,certd))) in
out(ChPub,(SIGg,MIDd,Rd, Xd));
event end_D(IDD)
else 0.
process ( (!pD) | (!pGCS))
```

RESULT: GCS AND DRONE

```
--------------------------------------------------------
Verification summary:
Query not attacker(VpkgSigd4[]) is true.
Query inj-event(end_GCS(IDGCS[])) ==> inj-
event(start_GCS(IDGCS[])) is true.
Verification summary:
Query inj-event(end_D(IDD[])) ==> inj-
event(start_D(IDD[])) is true.
--------------------------------------------------------
```

CODE: MOBILE-DEVICE, DRONE AND GCS

```
(*=======Channels=======*)
free ChSec:channel [private]. (*secure channel between
M,GCS and D*)
free ChPub:channel. (*public channel betweenM, GCS and
D*)
(*=======Constants and Variables=======*)
free IDGCS:bitstring. free IDD:bitstring. free
IDM:bitstring.
free g:bitstring. free f:bitstring. free
MIDm:bitstring.
free Mr:bitstring. free Kug2:bitstring. free
 Xm:bitstring.
free Sm:bitstring. free Sd:bitstring. free
 Sg:bitstring.
free Xd:bitstring. free Xg:bitstring. free
 MIDd:bitstring.
free Mgps:bitstring. free MIDg:bitstring. free
PKc:bitstring.
free certd:bitstring. free TS1:bitstring. free
TS2:bitstring.
free Rg:bitstring. free Rd:bitstring. free
 SEKgu:bitstring [private].
const P: bitstring.
query attacker(SEKgu).
query id:bitstring; inj-event(end_M(IDM)) ==> inj-
event(start_M(IDM)).
query id:bitstring; inj-event(end_D(IDD)) ==> inj-
event(start_D(IDD)).
query id:bitstring; inj-event(end_GCS(IDGCS)) ==>
inj-event(start_GCS(IDGCS)).
(*=====*Events*=====*)
event start_M(bitstring). event end_M(bitstring).
event start_D(bitstring). event end_D(bitstring).
event start_GCS(bitstring).event end_GCS(bitstring).
(*========Constructors=======*)
fun h(bitstring): bitstring.
fun Concat(bitstring,bitstring): bitstring.
fun XOR(bitstring,bitstring): bitstring.
fun ESEKup(bitstring): bitstring.
fun ECPM(bitstring,bitstring): bitstring.
fun Mul(bitstring,bitstring): bitstring.
fun SSKM(bitstring): bitstring.
fun Add(bitstring, bitstring): bitstring.
equation forall a: bitstring, b: bitstring;
XOR(XOR(a,b),b)=a.
let pM=
event start_M(IDM);
let CM3=ESEKup(Concat(Mr,certd)) in
let Sigm3=SSKM(Concat(Mr,certd)) in
out(ChPub,(MIDm, CM3, Sigm3));
in(ChPub,(MIDg:bitstring,Rd:bitstring,Xg:bitstring,
CHKug:bitstring));
let PKd=XOR(Rd,h(Concat(MIDd,(Rd,PKc)))) in
let Kug1=Mul(g,Xm) in
let Kug1=Concat(Sm,(Xd,g,PKd)) in
if SEKgu=h(Concat(Kug1,Kug2)) then
event end_M(IDM)
else
0.
(*=======Login and Authentication=======*)
let pD=
event start_D(IDD);
in(ChPub,(MIDd:bitstring, CM3:bitstring,
Sigm3:bitstring));
new e:bitstring;
let Xd2=Concat(TS1,ECPM(e,P)) in
out(ChPub,(MIDd,Rd, Xd2));
in(ChPub,(MIDg:bitstring, Rg:bitstring, Xg2:bitstring,
CHKug:bitstring));
let PKg=XOR(Rg,(h(Concat(MIDg,(Rg,PKc))))) in
let Kug1=Concat(Sd,(Xg2,e,PKg)) in
let Kug2=Mul(e,Xg2) in
if SEKgu=h(Concat(Kug1,Kug2)) then
if CHKug=h(Concat(SEKgu,Xd2)) then
out(ChPub,(MIDg,Rd, Xg,CHKug));
event end_D(IDD)
else
0.
let pGCS=
event start_GCS(IDGCS);
in(ChPub,(MIDd:bitstring,Rd:bitstring,Xd2:bitstring));
let PKd=XOR(Rd,(h(Concat(MIDd,(Rd,PKc))))) in
let Xg2=Concat(TS2,(Mul(f,P))) in
let Kug1=Concat(Sg,(Xd2,f,PKd)) in
let Kug2=Mul(f,Xd2) in
let SEKgu=h(Concat(Kug1,Kug2)) in
let CHKug=h(Concat(SEKgu,Xd2)) in
```

```
event end_GCS(IDGCS)
else 0.
process ( (!pGCS) | (!pD)| (!pM))
```

RESULT: MOBILE-DEVICE, DRONE AND GCS

```
--------------------------------------------------------
Verification summary:
Query not attacker(SEKgu[]) is true.
Query inj-event(end_M(IDM[])) ==>
inj-event(start_M(IDM[])) is true.
Query inj-event(end_D(IDD[])) ==>
inj-event(start_D(IDD[])) is true.
Query inj-event(end_GCS(IDGCS[])) ==>
inj-event(start_GCS(IDGCS[])) is true.
--------------------------------------------------------
```

REFERENCES

[1] S. U. Jan, I. A. Abbasi, and F. Algarni, "A mutual authentication and cross verification protocol for securing Internet-of-Drones (IoD)," *Comput., Mater. Continua*, vol. 72, no. 3, pp. 5845–5869, 2022.

[2] Q. Galvane, C. Lino, M. Christie, J. Fleureau, F. Servant, F.-L. Tariolle, and P. Guillotel, "Directing cinematographic drones," *ACM Trans. Graph.*, vol. 37, no. 3, pp. 1–18, Aug. 2018.

[3] L. Mottola, "Real-world drone sensor networks: A multi-disciplinary challenge," in *Proc. 6th ACM Workshop Real World Wireless Sensor Netw.*, Nov. 2015, p. 1.

[4] J. O. Uchidiuno, J. Manweiler, and J. D. Weisz, "Privacy and fear in the drone era: Preserving privacy expectations through technology," in *Proc. Extended Abstr. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2018, pp. 1–6.

[5] F. Ronaldo, D. Pramadihanto, and A. Sudarsono, "Secure communication system of drone service using hybrid cryptography over 4G/LTE network," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2020, pp. 116–122.

[6] J. Chakareski, "Drone networks for virtual human teleportation," in *Proc. 3rd Workshop Micro Aerial Vehicle Netw., Syst., Appl. (DroNet)*, 2017, pp. 21–26.

[7] S. U. Jan, I. A. Abbasi, and F. Algarni, "A key agreement scheme for IoD deployment civilian drone," *IEEE Access*, vol. 9, pp. 149311–149321, 2021.

[8] C. Lin, D. He, N. Kumar, K.-K. R. Choo, A. Vinel, and X. Huang, "Security and privacy for the Internet of Drones: Challenges and solutions," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 64–69, Jan. 2018.

[9] R. Canetti, O. Goldreich, and S. Halevi, "The random Oracle methodology, revisited," *J. ACM*, vol. 51, no. 4, pp. 557–594, Jul. 2004.

[10] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif2.03: Automatic cryptographic protocol verifier, user manual and tutorial," INRIA, Paris, France, Tech. Rep. EP/D076625/2, 2020.

[11] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *J. ACM*, vol. 58, no. 3, pp. 1–37, May 2011.

[12] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Heidelberg, Germany: Springer, 2012.

[13] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably secure authenticated group Diffie–Hellman key exchange," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, p. 10, 2007.

[14] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, 2013.

[15] Q. Do, B. Martini, and K.-K. R. Choo, "The role of the adversary model in applied security research," *Comput. Secur.*, vol. 81, pp. 156–181, Mar. 2019.

[16] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[17] S. Myagmar, A. J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Proc. Symp. Requirements Eng. Inf. Secur. (SREIS)*, 2005, pp. 1–8.

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

[19] 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.

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

[21] 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.

[22] B. Boldyreva, C. Gentry, A. O. Neill, and S. H. Yum, "Ordered multisignatures and identity-based sequential aggregate signatures," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, Atlanta, GA, USA, 2007, pp. 276–285.

[23] L. Zhang, B. Qin, Q. Wu, and F. Zhang, "Efficient many-to-one authentication with certificateless aggregate signatures," *Comput. Netw.*, vol. 54, no. 14, pp. 2482–2491, 2010.

[24] D. S. Xing, Z. F. Cao, and X. L. Dong, "An identity-based signature protocol based on cubic residues," *Sci. China Inf. Sci.*, vol. 54, no. 10, pp. 2001–2012, 2011.

[25] H. Xiong, Z. Guan, Z. Chen, and F. Li, "An efficient certificateless aggregate signature with constant pairing," *Comput. Inf. Sci.*, vol. 126, pp. 225–235, Jan. 2013.

[26] Y. Tian, J. Yuan, and H. Song, "Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones," *J. Inf. Secur. Appl.*, vol. 48, Oct. 2019, Art. no. 102354.

[27] 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.

[28] J. Won, S.-H. Seo, and E. Bertino, "Certificateless cryptographic protocols for efficient drone-based smart city applications," *IEEE Access*, vol. 5, pp. 3721–3749, 2017.

[29] H. Zhong, 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.

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

[31] M. S. Haque 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.

[32] S. Benzart, B. Triki, and O. Korbaa, "Privacy preservation and drone authentication using ID-based signcryption," in *Proc. SoMeT*, 2018, pp. 226–239.

[33] M. Turkanovic, B. Brumen, and M. H?lbl, "A novel user authentication and key agreement protocol for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw*, vol. 20, no. 1, pp. 96–112, 2014.

[34] M. S. Farash, M. Turkanovic, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement protocol for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Netw.*, vol. 36, no. 1, pp. 152–176, 2016.

[35] 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*, Orlando, FL, USA, Jul. 2020, pp. 1–6.

[36] 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.

[37] S. U. Jan, F. Qayum, and H. U. Khani, "Design and analysis of lightweight authentication scheme for securing IoD," *IEEE Access*, vol. 9, pp. 69287–69306, 2021.

[38] M. Nikooghadam, A. H. Amintoosi, S. K. H. Islam, and M. F. Moghadam, "A provably secure and lightweight authentication protocol for Internet of Drones for smart city surveillance," *J. Syst. Archit.*, May 2020, Art. no. 101955.

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

[40] 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.

[41] S. U. Jan and H. U. Khan, "Identity and aggregate signature-based authentication protocol for IoD deployment military drone," *IEEE Access*, vol. 9, pp. 130247–130263, 2021.

[42] C. L. Chen, Y. Y. Deng, W. Wensg, C. H. Chen, Y. J. Chiu, and C. M. Wu, "A traceable and privacy-preserving authentication for UAV communication control system," *Electron.*, vol. 9, no. 1, pp. 1–32, 2020.
[43] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the Internet of Drones applications," *Comput. Commun.*, vol. 155, pp. 143–149, Apr. 2020.
[44] D. Abbasinezhad-Mood and M. Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended Chebyshev chaotic maps," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4815–4828, Nov. 2018.
[45] *ArduinoLibs: Cryptographic Library*, 2018. [Online]. Available: https://rweather.github.io/arduinolibs/crypto.html
[46] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2017.
[47] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4349–4359, Jul. 2019.

SAEED ULLAH JAN received the B.S., M.Phil., and Ph.D. degrees from the University of Malakand, in 2007, 2016, and 2021, respectively. He is currently working as a Lecturer in computer science at the Higher Education, Achieves and Libraries Department, Government of Khyber Pakhtunkhwa, Pakistan. He is also working as a Controller of Examinations at the 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. He has published over 25 research papers in prestigious conferences and journals and written an introductory Book in *Computer Science* for beginners. His research interests include information security, cloud computing, distributed computing, privacy-preserving parallel computation, and drone security and authentication. The Government of Khyber Pakhtunkhwa, Pakistan, awarded him the Best Teacher Award for the year 2019–2020 out of 11 000 College Teachers in 309 public sector colleges in the Province.

IRSHAD AHMED ABBASI (Member, IEEE) received the M.S. degree in computer science from COMSATS University Islamabad, Abbottabad Capmus, Pakistan, and the Ph.D. degree in computer science from Universiti Malaysia Sarawak, Malaysia. He worked as a Senior Lecturer at King Khalid University, Saudi Arabia, from 2011 to 2015. He is currently a Postdoctoral Research Fellow at Universiti Malaysia Sarawak. He is also working as an Assistant Professor with the Department of Computer Science, University of Bisha, Saudi Arabia. He has over 12 years of research and teaching experience. He is the author of many articles published in top quality journals. His research interests include networks, VANETs, MANETs, FANETs, mobile computing, the IoT, cloud computing, cybersecurity, cryptography, soft computing, and drone security and authentication. He has received multiple awards, scholarships, and research grants. He is serving as an editor. He is also acting as a reviewer for many well reputed peer-reviewed international journals and conferences.

FAHAD ALGARNI received the bachelor's degree (Hons.) from the Department of Computer Science, King Abdulaziz University, the M.I.T. degree in computer networks from La Trobe University, Melbourne, Australia, and the Ph.D. degree from the Clayton School of Information Technology, Monash University, Melbourne. He is currently the Dean of the College of Computing and Information Technology, University of Bisha, Saudi Arabia. His research interests include wireless sensor networks, cloud computing, systems, design, and reliability, the IoT, and cyber security.

ADNAN SHAHID KHAN (Senior Member, IEEE) received the B.Sc. degree (Hons.) in computer science from the University of the Punjab, Lahore, Pakistan, in 2005, and the master's, Ph.D., and Postdoctoral degrees in networks and information security from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 2008, 2012, and 2013, respectively. He is currently a Senior Lecturer with the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak. His research interests include wireless communication, cloud computing, the Internet of Things, software-defined networking, cryptography, networks, and information security.