SecAuthUAV: A Novel Authentication Scheme for UAV-Ground Station and UAV-UAV Communication

Tejasvi Alladi, Naren, Gaurang Bansal, *Member, IEEE*, Vinay Chamola, *Senior Member, IEEE*, Mohsen Guizani, *Fellow IEEE*

***Abstract*—Unmanned Aerial Vehicles (UAVs) are becoming very popular nowadays due to the emergence of application areas such as the Internet of Drones (IoD). They are finding wide applicability in areas ranging from package delivery systems to automated military applications. Nevertheless, communication security between a UAV and its ground station (GS) is critical for completing its task without leaking sensitive information either to the adversaries or to unauthenticated users. UAVs are especially vulnerable to physical capture and node tampering attacks. Further, since UAV devices are generally equipped with small batteries and limited memory storage, lightweight security techniques are best suited for them. Addressing these issues, a lightweight mutual authentication scheme based on Physical Unclonable Functions (PUFs) for UAV-GS authentication is presented in this paper. The UAV-GS authentication scheme is extended further to support UAV-UAV authentication. We present a formal security analysis as well as old-fashioned cryptanalysis and show that our protocol provides various security features such as mutual authentication, user anonymity, etc, and is resilient against many security attacks such as masquerade, replay, node tampering, and cloning attacks, etc. We also compare the performance of our protocol with state-of-the-art authentication protocols for UAVs, based on computation, communication, and memory storage cost.**

***Index Terms*—UAVs, Internet of Drones (IoD), physical security, mutual authentication, security protocol, privacy, PUFs.**

I. INTRODUCTION

Around the world, Unmanned Aerial Vehicle (UAV) technology is being developed and deployed at a very rapid pace. From originally being used in military applications, these devices are being adopted across the government and corporate entities for providing a wide range of services including traffic management, goods delivery, inventory tracking, disaster management, wireless networks, etc [1–4]. With many governments easing regulations for UAV usage, the interest in this emerging technology is growing leaps and bounds leading to newer fields of applications being developed. The integration of drones into the IoT network is being termed the Internet of Drones (IoD), and has been discussed in many recent works [5, 6].

T. Alladi, Naren and V. Chamola are with the Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus, 333031, India. (e-mail: p20170433@pilani.bits-pilani.ac.in, f2015547@pilani.bits-pilani.ac.in, vinay.chamola@pilani.bits-pilani.ac.in).Vinay Chamola is also with APPCAIR, BITS-Pilani, Pilani Campus, 333031, India.

Gaurang Bansal is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077, Singapore (e-mail: e0622339@u.nus.edu)

Mohsen Guizani is with the Department of Computer Science and College of Engineering, Qatar University, 2713 Doha, Qatar (e-mail: mguizani@ieee.org).

Although IoD has been envisioned to provide various benefits, due to the UAVs being deployed in open environment and due to its communication being wireless in nature, they are prone to several security threats. Some of these security threats are man-in-the-middle attack, replay attack, node capture and tampering attack, etc. For example, in a military surveillance scenario, if an adversary UAV tries to impersonate itself as a legitimate UAV and authenticates itself to the ground station, it can get secret information about the legitimate UAV from the ground station or send malicious information to the ground station. This can cause serious disruptions to the surveillance services, leading to major economic or even human losses.

Node authentication is the first security aspect to be met for any IoT network and thus for IoD also, before beginning a secure communication session in the network. The existing Internet security techniques implement node authentication using stored cryptography keys. UAVs being deployed in the open air are prone to device capture attack, and thus to secret keys being exposed. Further, these UAVs generally have limited memory and computation capability, thus storing the secret keys and executing standard cryptography algorithms is also a challenge [7]. Keeping these issues in mind, this paper presents a light-weight mutual authentication protocol for authenticating UAVs with a ground station. Extending the UAV-GS authentication, an authentication scheme for UAV-UAV communication is also shown.

In several recent studies, Physical Unclonable Functions (PUFs) have been shown to be very promising in providing security in several IoT and embedded system based applications [8–10]. Due to the inherent randomness introduced during the manufacturing process of these devices, they provide unclonable and unique identities to their devices. Thus they are suitable to be employed as a hardware root of trust and as hardware security primitives. The operation of PUFs is based on challenge-response pair mechanism, i.e. PUF has a property that when an input stimulus called the challenge is applied to it, the device responds with an output called the response. Two different PUF devices manufactured using the same fabrication process and with the same configuration are expected to produce different responses given the same challenge. In the proposed protocol, the above property of PUF is exploited for generating a unique session key for each device, which is embedded with a PUF. The embedded PUF chip would function as a unique, unclonable, and non-reproducible fingerprint for each UAV in the network. It's behavior is similar to a function which takes a challenge as an input and produces a response as its corresponding output.

$$R = PUF(C)$$

In the above equation, C is the challenge, and R is the response, and both C, and R are binary strings.

The major contributions of this paper are as follows:

i. We propose a PUF-based mutual authentication protocol, SecAuthUAV which is capable of establishing a secure session between a UAV and the ground station without storing any secret information on the UAV.
ii. Extending the UAV-GS authentication, a secure UAV-UAV session can be established between any two UAVs in the network.
iii. We provide a formal security proof of the protocol using Mao and Boyd logic, and also perform cryptanalysis to guarantee the security and versatility of the protocol.
iv. A comparison of SecAuthUAV with state-of-the-art authentication protocols for UAVs, in terms of security features, computation, communication and storage cost is also provided.

The organization of the rest of the paper is as follows. Section II discusses the related works in securing UAV networks. The system model, attack model, security goals, and assumptions for the system model are discussed in Section III-A. In Section IV, we present our mutual authentication protocol (SecAuthUAV). In Section V, we subject our protocol to formal security analysis and informal cryptanalysis. In Section VI, we compare SecAuthUAV with several state-of-the-art security schemes for UAV-GS authentication in terms of security and computation performance. The paper is finally concluded in Section VII.

II. Related Works

UAV Networks differ greatly from the other wireless networks such as Mobile Ad-Hoc Networks (MANETs) [11–13] due to greater node mobility and a variety of security threats, and thus the existing security solutions for MANETs and other traditional wireless networks cannot be adapted to these networks. He et al. have explained the essential features of UAV communications and identified the requirements for UAV security protocols in [14]. They have shown that traditional security techniques such as anomaly detection induce significant delays and therefore cannot be used for time-critical applications such as those in UAV networks [15, 16]. Thus, they establish the need for UAV security protocols which take into account the trade-off between network performance and security strength.

Hooper et al. have presented a multi-layer framework in [17] for defending commercial UAV systems from Address Resolution Protocol (ARP) cache poisoning, buffer overflow and Denial of Service (DoS) attacks. Physical layer security has also been an important area of research in UAV networks. Addressing this aspect, several works have been proposed in recent times to enhance security in UAV-aided non-orthogonal multiple access (NOMA) networks [18–20] by employing techniques such as precoding optimization and artificial jamming to secure against eavesdropping attacks. Blazy et al. have considered a strong adversarial model which can perform fault injection, side-channel and physical attacks on a UAV [21, 22]. In their scheme, they store a single initial key which is used to generate subsequent keys using a keyed hash function. They propose to use multiple streams of such key sequences to prevent an attack. But, if a powerful attacker gains just one key of each stream, he/she could very easily keep track of all the key streams simultaneously thereby rendering all further communication insecure.

Several authentication and other cryptographic frameworks have been proposed for UAV networks in recent times. Chen et al. have proposed the Direct Anonymous Attestation with Mutual Authentication (MA-DAA) scheme in [23] for use in network-connected UAV systems. Compared with the existing Direct Anonymous Attestation (DAA) schemes, MA-DAA scheme is well suited to the low bandwidth and computation capabilities of the UAV networks. However their solution is based on Trusted Platform Modules (TPMs), which are specialized expensive security co-processors that need to be integrated into the systems leading to higher costs. Abdallah et al. have presented a lightweight security scheme for surveillance UAV networks in [24], however their scheme ensures only one-sided authentication, and not mutual authentication. Another work [25] is a certificateless group key authentication protocol targeted at untrusted UAV networks. This work is based on bilinear pairing and elliptical curve cryptography (ECC) and thus is not lightweight. Another work [26] proposes a short certificate based signature distribution scheme which is also heavyweight due to usage of public key cryptographic techniques. Two more works proposed by authors in recent times [27, 28] have been proven to be sufficiently lightweight and secure against multiple attacks in Internet of Drones (IoD) deployment scenarios. The authors of another recent work [29] show that the work in [28] is not scalable and propose an improved protocol addressing the scalability issue. However, none of these protocols are secure against UAV node tampering and physical attacks. In these schemes, secret information required for authentication is stored in the UAV's physical memory, hence an adversary can easily extract this critical information and launch various attacks.

Recently, PUFs have been shown to be very effective in securing wireless and unmanned IoT environments [9, 30, 31]. Chen and Willems propose a secret key generation protocol based on PUF in [32]. In a recent work on UAV authentication using PUFs [33], the PUF's challenge-response pair is utilized as an initialization condition for a chaotic system, to secretly exchange random seeds and generate a session key.

We use a similar concept in our paper to do away with the storage of critical information such as secret keys in the device's memory. Our protocol supports generation of secret information on the fly based on the response generated from the inbuilt PUFs. Using this generated information, the proposed protocol ensures mutual authentication between the UAV node and the ground station.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3033060, IEEE Transactions on Vehicular Technology

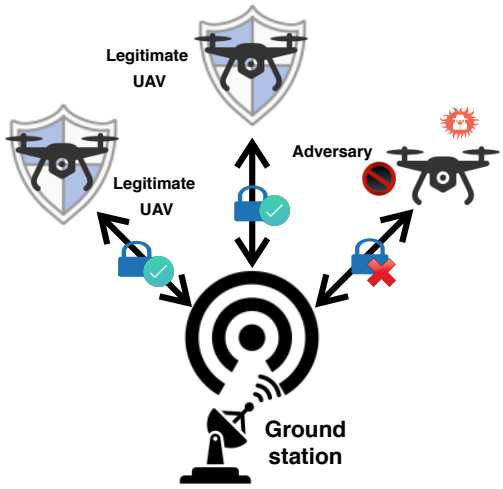

Fig. 1: System model.

III. System & Threat Model

A. System Model

The system model considered in this paper consists of legitimate UAVs and a ground station as shown in Fig. 1. UAVs have limited memory and computation capability as compared to the ground station. In this model, multiple UAVs may be connected to a single ground station. The objective of this paper is to establish a secure communication session between a UAV U_i and the ground station GS by performing mutual authentication between the two entities. It will further be shown that by extending the same mutual authentication scheme, a secure session can be established between any two UAVs namely, U_1 and U_2. Following the session establishment between U_1 and GS, U_1 requests GS for a secure session with the other UAV U_2. A similar authentication protocol is run between GS and U_2. Finally, a secure communication session is established between U_1 and U_2.

Every UAV U_i is equipped with a PUF, which is used for generating a response output for a challenge input to it. The response is split into two parts which are further used in the authentication protocol as discussed in Section IV. Before deployment, the registration of U_i with GS is carried out through a secure channel. Here, a challenge-response pair (C, R) of the $U_i's$ PUF is generated and securely added to $GS's$ database. Now, U_i is ready to be deployed. Further, GS starts with a single initial (C, R) pair for each U_i which was saved at the time of UAV registration. Every time U_i successfully authenticates with GS, a new challenge-response pair (C', R') is generated and securely transmitted to GS to replace the existing (C, R) pair.

Table I lists the notations used in this paper and their descriptions.

B. Attack Model

An adversary A may try to authenticate with GS by masquerading as a legitimate UAV U_i or by launching a man-in-the-middle attack. Since the communication channel is wireless and public, A may try to eavesdrop on the transmitted messages or modify these messages or replay them in the network. In this model, A may physically capture a U_i and try to extract secret information from its memory. Additionally, A may also try to clone U_i.

TABLE I: Notations used in the paper

Notation	Description
$U_i, TUID_i$	i^{th} UAV and its temporary ID
GS, GID	Ground station and its ID
(C, R)	PUF challenge response pair
$\|\|$	Concatenation operation
$\oplus, XOR$	XOR operation
N_X	Random nonce generated
$H()$	Hash function
Sk_i	Session key generated
Ack	Authentication acknowledgement
Req	Authentication request

C. Security Goals

This protocol has been proposed keeping the following security goals in view.

i. Achieving mutual authentication between the legitimate UAV U_i and the ground station GS.
ii. The protocol must be secure against common security attacks such as masquerade, man-in-the-middle, and replay attacks.
iii. The protocol must be capable of generating a unique session key for each authentication session.
iv. The protocol must be safe against cloning attacks as well as physical attacks such as UAV node capture and tampering.
v. If the communicated messages are tampered with, the receiver, either GS or U_i, must be able to detect the tampering and abort the authentication process.
vi. No unauthorized entity must be capable of tracking the temporary ID of any U_i, in other words, $U_i's$ anonymity must be maintained.

D. Assumptions

We discuss here a few assumptions that are made in this paper.

i. Each legitimate UAV U_i is endowed with a unique PUF. In case U_i is capture, any attempt to tamper with the PUF will render the PUF unusable and U_i will not be able to authenticate with GS.
ii. Each U_i has limited computation power and is memory constrained while GS has no such constraints.
iii. U_i itself does not store any secret information, rather it relies on the response R generated using the challenge C on the PUF. This response serves as a secret to be used further in the authentication protocol.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3033060, IEEE Transactions on Vehicular Technology

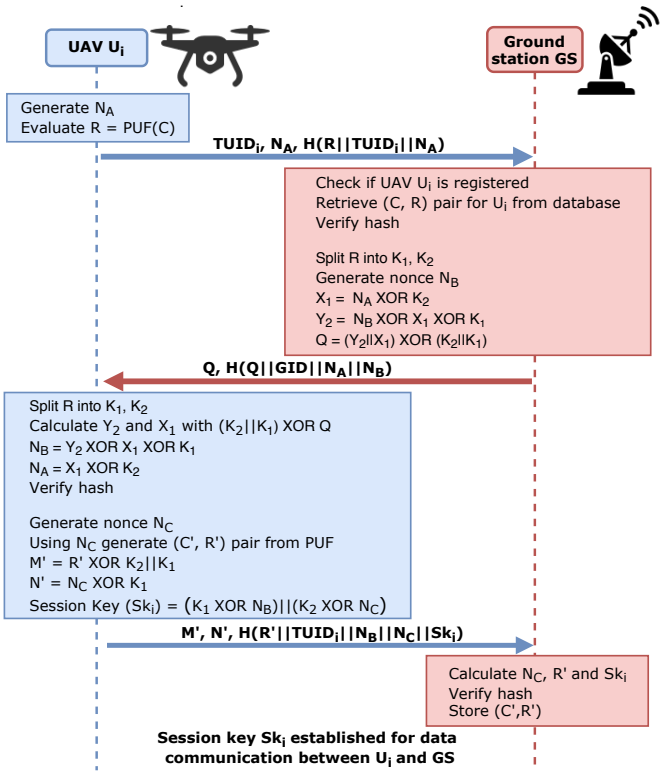

Fig. 2: SecAuthUAV protocol between a UAV and the ground station.

IV. Proposed Authentication Protocol (SecAuthUAV)

The proposed scheme is organized into three phases, namely the UAV registration phase, the UAV-GS authentication phase, and the UAV-UAV authentication phase.

A. UAV registration phase

1) Each UAV U_i is registered with the ground station GS before deployment.
2) During the registration process, a challenge-response pair (C, R) from $U_i's$ PUF is securely stored in the $GS's$ database.
3) A temporary identity $TUID_i$ is generated for each U_i at GS while the permanent identity GID is maintained by GS.
4) The set $\{TUID_i, C, GID\}$ is securely stored in the UAV's storage, while $\{TUID_i, C, R\}$ is stored in the $GS's$ database for UAV U_i.

B. UAV-GS authentication phase

Here, we present our protocol, SecAuthUAV to achieve mutual authentication between UAV U_i and the ground station GS. Fig. 2 depicts the various computations performed on U_i and GS, and the different messages exchanged in the protocol. This protocol ensures that only a legitimate U_i can authenticate with GS before it can begin a secure communication session with GS. In this process of authentication, a secure session key is established between U_i and GS, which can be used for further communication.

1) When a UAV U_i wants to authenticate with GS, it generates the response R by using the stored challenge C.
2) It sends its temporary ID $TUID_i$, a random nonce N_A, and a hash calculated as $H(R||TUID_i||N_A)$ to GS.
3) GS queries its database for any entry corresponding to the received ID $TUID_i$. It also determines the freshness of N_A (it must not match the N_A of the previous authentications). Unless both these conditions are satisfied, U_i's authentication request will not be handled. After verifying the hash, GS finds the corresponding challenge-response pair (C, R) from its database. It splits the response R into K_1 and K_2, generates a nonce N_B, and performs the following operations to generate Q.

$$X_1 = N_A \oplus K_2$$
$$Y_2 = N_B \oplus X_1 \oplus K_1$$
$$Q = (Y_2\|X_1) \oplus (K_2\|K_1)$$

4) GS then sends $Q, H(Q\|GID\|N_A\|N_B)$ to U_i.
5) U_i splits response R into two parts K_1, K_2 as was done on GS, to be used in the below mentioned operations.

$$(Y_2\|X_1) = (K_2\|K_1) \oplus Q$$
$$N_B = Y_2 \oplus X_1 \oplus K_1$$
$$N_A = X_1 \oplus K_2$$

The nonce N_A, N_B are retrieved and the hash is recomputed. Thus, the source of the message, its freshness and integrity are verified. In case hash verification does not succeed, U_i terminates the authentication of GS. If verification is successful, a random nonce N_C is generated, a substring of which acts as the new challenge C'. This depends on the type of PUF used in the device and the size of the challenge. The corresponding challenge-response pair (C', R') is generated by U_i using its PUF. This newly generated information is encoded into M', N' as follows.

$$M' = R' \oplus K_2\|K_1$$
$$N' = N'_C \oplus K_1$$

The session key Sk_i with which further communication will take place is computed as shown below.

$$Sk_i = (K_1 \oplus N_B)\|(K_2 \oplus N_C)$$

6) U_i sends M', N' and the hash $H(R'\|TUID_i\|N_B\|N_C\|Sk_i)$ to GS.
7) GS obtains N_C, R' and Sk_i as shown in the following equations.

$$N_C = N' \oplus K_1$$
$$R' = M' \oplus K_2\|K_1$$
$$Sk_i = (K_1 \oplus N_B)\|(K_2 \oplus N_C)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3033060, IEEE Transactions on Vehicular Technology

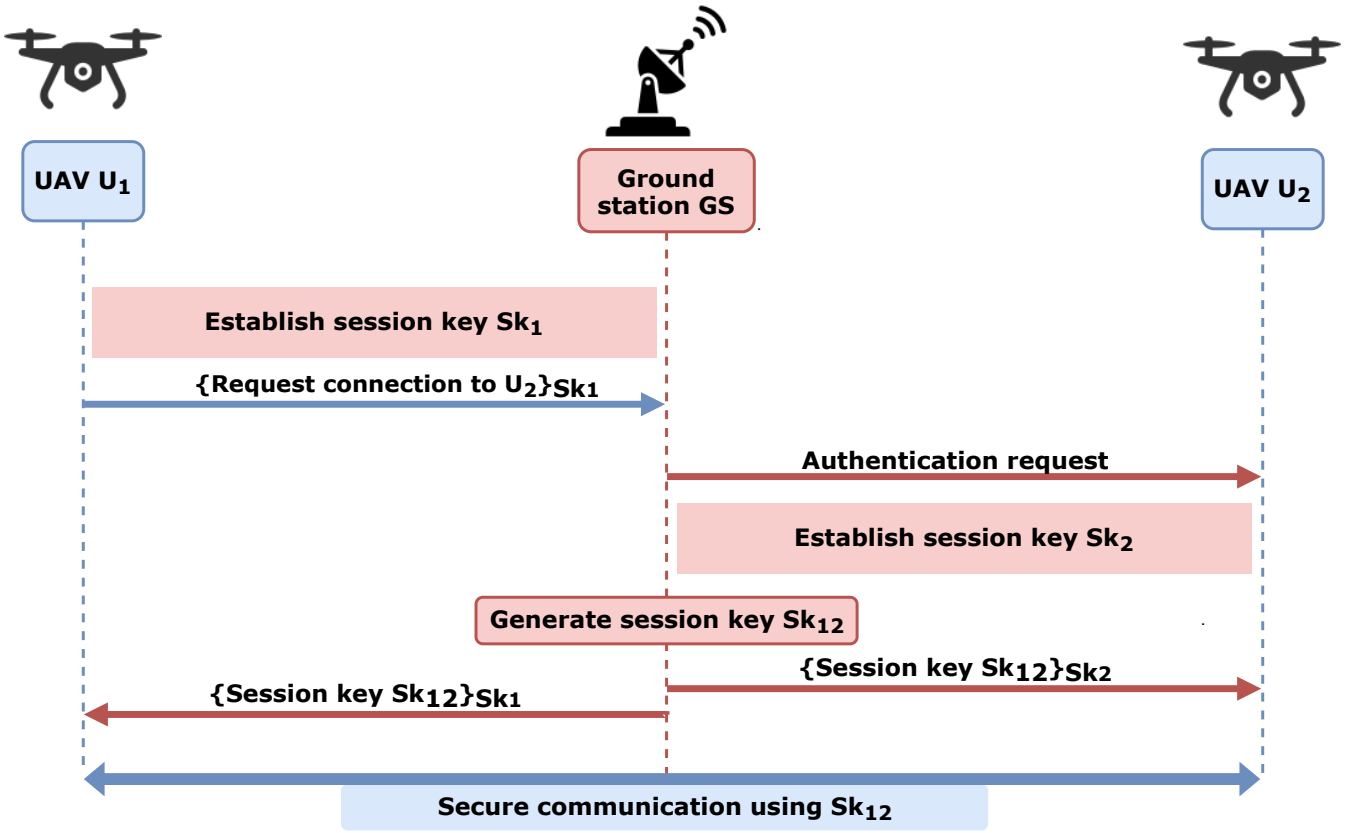

Fig. 3: UAV-UAV authentication.

GS thus has all the parameters to verify the hash. If the verification fails, authentication is terminated by GS. If verified, it stores a copy of the new challenge-response pair (C', R') for U_i in its database along with the previous pair (C, R). As mentioned earlier, C' is obtained from N_C.

8) Thus, mutual authentication between U_i and GS is achieved, and further communication is encrypted using the established session key Sk_i. Both U_i and GS compute the new $TUID_i'$ for U_i and update it as follows.

$$TUID_i' = H(K_2||TUID_i||K_1)$$

C. UAV-UAV authentication phase

Here, we briefly describe how any two UAVs, U_1 and U_2 will be able to establish a secure session using the above discussed UAV-GS authentication scheme. The steps of this authentication phase are depicted in Fig. 3.

1) Once a secure session is established between U_1 and GS with session key Sk_1, U_1 sends a request to GS for a secure session with another UAV. U_1 requires this session as part of its mission.
2) GS identifies a suitable UAV U_2 and forwards an authentication request containing a Req string as well as a hash computed as $\{Req, H(Req||TUID_2||GID)\}$ to U_2. U_2 verifies the hash and begins the same UAV-GS authentication described in Section IV-B to generate a session key Sk_2 thus establishing a secure session with GS.
3) GS generates a new secret session key Sk_{12} and distributes it to both U_1 and U_2 using the already established session keys Sk_1 and Sk_2. Both the UAVs obtain Sk_{12} and thus a secure communication session is established between U_1 and U_2.

V. Security Analysis

To evaluate the security of the proposed UAV-GS authentication phase (Section IV-B), we provide a formal security proof in Section V-A, as well as old-fashioned cryptanalysis in Section V-B. Such an approach is necessary because it guarantees both security and versatility while formal proof by itself is insufficient [34, 35]. We have used Mao Boyd logic [36] for formal verification of our protocol's security, and our old-fashioned cryptanalysis is based on 12 evaluation criteria (C1 - C12) as used by the authors in [37]. Since the UAV-UAV authentication phase (Section IV-C) involves two instances of UAV-GS authentication (Section IV-B), the proof for this phase is trivial.

A. Formal Proof

In our proof we invoke the ***authentication***, ***nonce-verification***, ***confidentiality***, ***super-principal***, ***intuitive*** **and** ***good-key inference rules*** of Mao Boyd logic in [36]. Throughout the verification process, we denote the UAV U_i by U and the ground station GS by G.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3033060, IEEE Transactions on Vehicular Technology

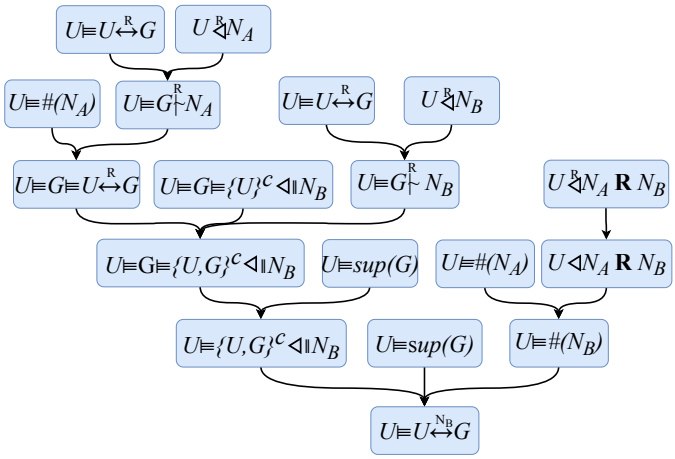

Fig. 4: U believes that N_B is a good secret known only to U and G.

We first prove the statement "U believes N_B is a good secret between U and G". The statements of this proof are written in Mao Boyd logic in equations (i) - (xv), and pictorial representation is shown in Fig. 4.

The response R corresponding to the challenge-response pair (C, R) of U is stored in G, therefore "U believes R is a good secret between U and G" (i). Using R, the UAV U is able to decipher Q in the second message of the protocol $\{Q, H(Q||GID||N_A||N_B)\}$ to get N_A and N_B. Therefore, "U sees N_A with decipher key R" (ii) and "U sees N_B with decipher key R" (iii).

$$U \models U \overset{R}{\leftrightarrow} G \tag{i}$$

$$U \overset{R}{\triangleleft} N_A \tag{ii}$$

$$U \overset{R}{\triangleleft} N_B \tag{iii}$$

Applying the ***authentication rule*** to statements (i) and (ii), we get "U believes G encrypted N_A using R" (iv). On applying the same rule to (i) and (iii), we get "U believes G encrypted N_B using R" (v). Since U generates a new nonce N_A each time, "U believes N_A is fresh" (vi). Next, we apply the ***nonce-verification rule*** to statements (iv) and (vi) to get "U believes that G believes that R is a good secret between U and G" (vii).

$$U \models G \overset{R}{\sim} N_A \tag{iv}$$

$$U \models G \overset{R}{\sim} N_B \tag{v}$$

$$U \models \#(N_A) \tag{vi}$$

$$U \models G \models U \overset{R}{\leftrightarrow} G \tag{vii}$$

Since G is known to generate a new nonce N_B each time, U is aware of the fact that no one other than G could have seen N_B. Thus, "U believes that G believes that no one other than U has access to N_B" (viii). Applying the ***confidentiality rule*** to (v), (vii) and (viii) we get "U believes that G believes that no one other than U and G has access to N_B" (ix).

$$U \models G \models \{U\}^c \triangleleft \| N_B \tag{viii}$$

$$U \models G \models \{U, G\}^c \triangleleft \| N_B \tag{ix}$$

It is assumed in the protocol that U believes that the ground station G is secure and trusted or in other words, "U believes that G is the super-principal" (x). Applying the ***super-principal rule*** to statements (ix) and (x), we get "U believes that no one other than U and G has access to N_B" (xi).

$$U \models sup(G) \tag{x}$$

$$U \models \{U, G\}^c \triangleleft \| N_B \tag{xi}$$

To proceed further we need to understand a few definitions and rules of message idealization from [36] which are as follows:

- If a message does not have any symbols, it is an *atomic message (AM)*.
- An *AM* which is sent in one line and received in another by the origin node is called a *challenge*.
- If a *challenge* is present in a message sent to the origin node, it is a *replied challenge (RC)*.
- A *response* is an *AM* and an *RC* sent together by the node which sends the *response*.
- If an *AM* is neither a *challenge* nor a *response*, it is *nonsense*. All *nonsense* is discarded.
- If an *AM* behaves as a *challenge* as well as a *response* in a line, it is treated as a *response*.
- "*Response* **R** *RC*" is the combination of a *replied challenge* and its *response*.

In the first message of the protocol $\{TUID_i, N_A, H(R||TUID_i||N_A\}$, N_A is sent to G. In response to this, G sends N_B encrypted in Q in the second message. On decryption of Q, U obtains N_A and N_B. Therefore, according to the above mentioned definitions and rules, N_A is a challenge and N_B is the response. Note that these are not the same challenge-response pair (C, R) of the PUF. Thus, "U can see the replied challenge N_A and the response N_B with decipher key R" (xii). Applying the ***intuitive rule*** to (xii), we get "U can see the replied challenge N_A and the response N_B" (xiii).

$$U \overset{R}{\triangleleft} N_A \ \mathbf{R} \ N_B \tag{xii}$$

$$U \triangleleft N_A \ \mathbf{R} \ N_B \tag{xiii}$$

On applying the ***fresh rule*** to (vi) and (xiii), we get "U believes N_B is fresh" (xiv). Applying the ***good-key rule*** to statements (x), (xi) and (xiv) we have proved the statement "U believes that N_B is a good secret known only to U and G" (xv).

$$U \models \#(N_B) \tag{xiv}$$

$$U \models U \overset{N_B}{\leftrightarrow} G \tag{xv}$$

The statement "G believes that N_B is a good secret known only to U and G" can also be proved similarly as is shown in Fig. 5. In this figure, the logical AND operation between two

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3033060, IEEE Transactions on Vehicular Technology

$$\frac{\dfrac{\mathbf{G}\models\mathbf{U}\overset{R}{\leftrightarrow}\mathbf{G}\ \bigwedge\ \mathbf{G}\models\mathbf{U}^{c}\triangleleft\|}{\mathbf{G}\models\{\mathbf{U},\mathbf{G}\}^{c}\triangleleft\|N_B}\bigwedge\mathbf{G}\models\#(N_B)}{\mathbf{G}\models\mathbf{U}\overset{N_B}{\leftrightarrow}\mathbf{G}}$$

Fig. 5: G believes that N_B is a good secret known only to U and G

statements is represented by a '$\bigwedge$'. Thus, we have shown that N_B cannot be accessed by an adversary.

Similarly, it can be proved that an adversary cannot gain access to N_C and R'. If an adversary cannot obtain N_B, N_C, and R', it cannot make sense of the communicated data. This secrecy of N_B, N_C, and R' is regardless of the kind of attack used by the adversary such as a masquerade attack, a man-in-the-middle attack, or a replay attack.

B. Cryptanalysis

This analysis is based on a list of security criteria put forward by the authors in [37], which removes redundancies and ambiguities which were prevalent in the area of security protocols. We have omitted the discussion on the criteria C1-C4, C6, and C9 since they involve smart cards and user-input passwords which do not apply to our protocol. Several works such as [38, 39] have already adopted this state-of-the-art security evaluation technique. For the following cryptanalysis, we consider the UAV with ID U_i, the ground station with ID GS, and adversary A.

[C5] Resistance to known attacks: The proposed scheme is secure against masquerade attack (C5.1), man-in-the-middle (MITM) attack (C5.2), replay attack (C5.3), node tampering attack (C5.4), cloning attack (C5.5), and de-synchronization (C5.6). Adversary A cannot masquerade as U_i since it does not have the same PUF, and it cannot masquerade as GS since it does not have the correct (C, R) pairs. Thus, masquerade and MITM attacks will fail. In the proposed protocol, attempts by A to replay old messages to either GS or U_i will fail since a new (C, R) pair is used for achieving authentication in each session and A cannot gain access to $U_i's$ (C, R) pair. Since PUFs are inherently unclonable, A cannot successfully clone U_i, thus the scheme is secure against cloning attacks. The proposed protocol is secure against node tampering attacks for the following reasons: even if A physically captures U_i, any attempt to tamper with the PUF will render the PUF unusable. Thus, A cannot resort to nefarious activities such as node tampering even in the event of a device capture; also, no critical information related to the authentication process is stored in U_i. We address de-synchronization attacks in detail in Section V-C.

[C7] Provision of key agreement: In Section V-A, the secrecy of random nonce N_B and N_C has been proven. In addition, K_1 and K_2 are derived from R. Since a new (C, R) pair is used for achieving authentication in each session, a different R is used for each session. Thus, in the proposed protocol, a new and secure session key calculated as $Sk_i = (K_1 \oplus N_B)\|(K_2 \oplus N_C)$ is established after every successful authentication.

[C8] No clock synchronization: In our protocol we ensure message freshness by using random nonces instead of timestamps. Thus, the proposed security scheme is free from the problems of time delay and clock synchronization, i.e., GS does not need to synchronize its clock with U_i and vice versa.

[C10] Mutual authentication: In the proposed scheme, a single PUF (C, R) pair of U_i is stored on GS prior to $U_i's$ deployment. Due to this, a session key can be established only between U_i and GS. Therefore, if a session key Sk_i is established, it means that the authenticating parties are legitimate, i.e., mutual authentication has been achieved.

[C11] User anonymity: In our protocol, the temporary ID of U_i is updated at the end of each authentication as $TUID_i' = H(K_2||TUID_i||K_1)$, which guarantees user anonymity in the proposed scheme. Both U_i and GS can compute the same temporary ID without requiring any message exchange. No entity other than U_i or GS can know the PUF response R for a given challenge C to compute the temporary ID.

[C12] Forward secrecy: In the proposed scheme, even if A manages to guess the current session key Sk_i, this does not compromise the security of any future session. This is because the response for the next session, R' (which is required to establish the session key for the next session) cannot be obtained by A as already proven in Section V-A. Since the proposed protocol guarantees C11, A will not even be able to track U_i, let alone compromise any future sessions. Thus, the proposed protocol achieves perfect forward secrecy. A will not be able to get the session key of any previous session as well since it cannot gather the required secrets to generate them. Thus, perfect backward secrecy is also guaranteed by this protocol.

C. Addressing de-synchronization attacks

After the UAV-GS authentication phase, i.e., once GS calculates the session key Sk_i, it sends an acknowledgment string Ack along with a hash of Ack computed as $H(Ack||GID||N_C)$ to U_i. When U_i does not receive this acknowledgement message, it assumes that GS has not received the third message $\{M', N', H(R'||TUID_i||N_B||N_C||Sk_i)\}$ in Fig. 2, and will repeatedly send this third message to GS. Even after repeated attempts, if U_i does not receive the acknowledgment message, it does not update its $TUID$ and discards the (C', R') pair sent to GS. Even if the third message of the protocol is blocked or not received by GS, U_i will not receive any acknowledgment from GS, which will result in the same outcome discussed above.

In case GS does receive the third message, it will compute and store $TUID'$ and (C', R'). It will also retain a copy of $TUID$ and (C, R). At a later time, when U_i again initiates the UAV-GS authentication, GS will receive either $TUID$ or $TUID'$ depending on whether U_i has updated its temporary ID or not. If $TUID'$ is received, then GS will delete $TUID$ and (C, R), while if $TUID$ is received it will delete $TUID'$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3033060, IEEE Transactions on Vehicular Technology

TABLE II: Comparison of security features

Scheme	C5.1	C5.2	C5.3	C5.4	C5.5	C5.6	C7	C8	C10	C11	C12
[27]	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	✓
[28]	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	✓
[33]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
[29]	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	✓
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

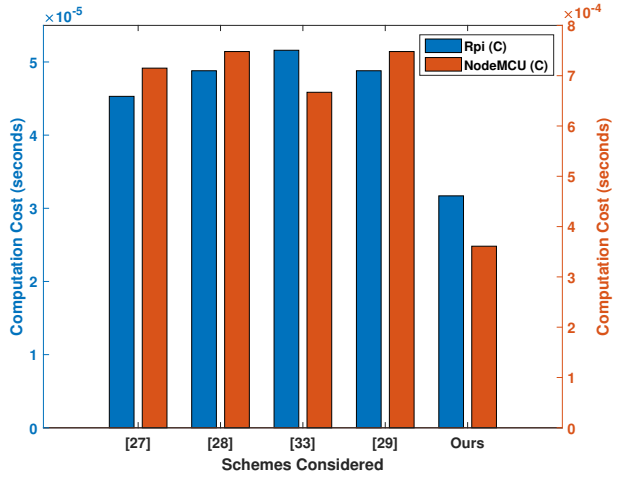

Fig. 6: Computation cost comparison- Rpi (C) vs NodeMCU (C).

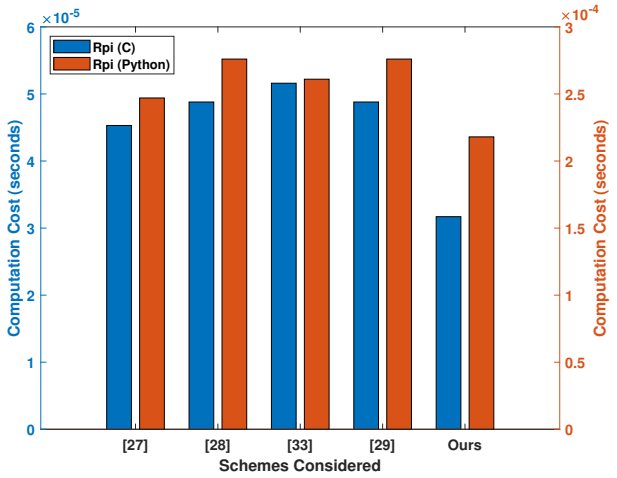

Fig. 7: Computation cost comparison- Rpi (C) vs Rpi (Python).

and (C', R'). This mechanism prevents de-synchronization and ensures the availability of the system when the third message of the protocol is not received by GS.

VI. Security and Computation Performance Comparison

A. Security Comparison

This section provides a comparative analysis of our protocol with other existing security schemes available for UAV-GS authentication [27–29, 33]. Table II presents this comparison based on the criteria discussed in the previous section. In the table, '✓' and '✗' respectively indicate whether a protocol satisfies or does not satisfy a criterion. As depicted in Table II, our proposed protocol fares well in all the criteria. All considered protocols protect against masquerade (C5.1), MITM (C5.2), replay (C5.3), and de-synchronization attacks (C5.6), along with provision for session key establishment (C7), mutual authentication (C10), and forward secrecy (C12). By using PUFs on the UAVs, both our protocol and [33] account for security against node tampering (C5.4) of individual UAVs, and protection against cloning attacks (C5.5). The schemes of [27–29] require all network entities to have synchronized clocks and thus do not satisfy the feature of no clock synchronization (C8). The scheme [33] does not provide user anonymity (C11) since the ID of the UAV is always communicated in clear text and is not updated for subsequent sessions.

TABLE III: Running time for various operations

Sym.	Description	Computation time		
		RPi(C)	**NMCU(C)**	**RPi(Py)**
T_x	Bitwise XOR	$1.26\mu s$	$3.63\mu s$	$11.8\mu s$
T_n	PRNG	$0.253\mu s$	$4.09\mu s$	$5.90\mu s$
T_h	Hash	$4.63\mu s$	$91\mu s$	$18.3\mu s$
T_{hm}	HMAC	$23\mu s$	$309\mu s$	$99.9\mu s$
T_p	PUF	$0.4\mu s$ (common for all)		
T_c	Concatenation	$0.587\mu s$	$4.58\mu s$	$5.09\mu s$

TABLE IV: Cost comparison for authentication protocol

Scheme	UAV computation cost
[27]	$4T_x + 1T_n + 7T_h + 0T_{hm} + 0T_p + 13T_c$
[28]	$3T_x + 1T_n + 7T_h + 0T_{hm} + 0T_p + 21T_c$
[33]	$0T_x + 5T_n + 0T_h + 2T_{hm} + 2T_p + 6T_c$
[29]	$3T_x + 1T_n + 7T_h + 0T_{hm} + 0T_p + 21T_c$
Ours	$8T_x + 2T_n + 3T_h + 0T_{hm} + 2T_p + 11T_c$

TABLE V: Comparison of communication cost

	Scheme				
	[27]	**[28]**	**[33]**	**[29]**	**Ours**
Comm. cost (bits)	1696	1536	1952	1696	1600

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3033060, IEEE Transactions on Vehicular Technology

TABLE VI: Comparison of storage cost

	Scheme				
	[27]	**[28]**	**[33]**	**[29]**	**Ours**
Storage cost (bits)	480*	640*	320	640*	352

$*$ represents minimum storage

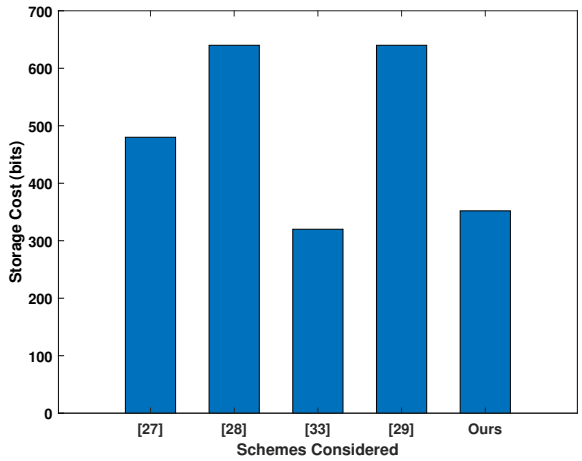

Fig. 8: Storage cost comparison.

B. Simulation and Computation Performance

We used NodeMCU v3.0 and Raspberry Pi 3B as two different simulation environments for the UAV in our system model, to run commonly used mathematical and cryptographic operations such as XOR, pseudo-random number generation (PRNG), Hash (SHA-1), HMAC (SHA-1), and concatenations. The operations have been simulated in both C and Python programming languages. We consider a recent PUF proposed in [40] to be deployed in the UAVs for both our protocol and the scheme in [33]. This PUF has been demonstrated to give a response of at least 256 bits, with a 320-bit response generated from the PUF with a response time of 0.4 μs when a 32-bit challenge is input to it. Notations of these operations and their running times on Raspberry Pi 3B (in C and Python), as well as on NodeMCU v3.0 (in C) are provided in Table III. A cost comparison of these above mentioned operations for the UAV-GS authentication process in the works of [27, 28, 33] and [29] is provided in Table IV. Based on the time taken for each operation, the total time taken corresponding to the computation cost for all the operations on a UAV is plotted in Fig. 6 and Fig. 7. Fig. 6 shows the computation cost comparison for the schemes considered above on Raspberry Pi and NodeMCU (both in C language), and Fig. 6 shows the computation cost comparison on Raspberry Pi (C and Python languages).

From Fig. 6, it is clearly evident that SecAuthUAV outperforms all other works both on NodeMCU and Raspberry Pi platforms (in C language). While [27], [28], [33] and [29] have computation costs of $715\mu s$, $748\mu s$, $667\mu s$ and $748\mu s$ respectively on NodeMCU (C), our protocol has a cost of only $361\mu s$. On Raspberry Pi (C), while [27], [28], [33] and [29] show a cost of $45.3\mu s$, $48.8\mu s$, $51.6\mu s$ and $48.8\mu s$ respectively, ours shows just $31.7\mu s$. From Fig.7, it can be observed that the computation costs are higher on Python when compared with C. Using Python also, SecAuthUAV outperforms all other works. While [27], [28], [33] and [29] show a cost of $247\mu s$, $276\mu s$, $261\mu s$ and $276\mu s$ respectively on Raspberry Pi (Python), ours shows $218\mu s$.

We further do a comparison of the proposed protocol with other protocols in terms of total communication cost between the UAV and ground station. In our comparison, we set standard sizes for the different fields communicated across the entities such as hash digest, HMAC output, nonce, and device ID to be 160 bits each, and timestamp to be 32 bits. These sizes have been chosen to be consistent in comparison with other protocols [28, 29] which have also considered the same sizes. As shown in Table V, based on these sizes the communication overhead in [27], [28], [33] and [29] is 1696 bits, 1536 bits, 1952 bits and 1696 bits respectively, while in our protocol the communication overhead is 1600 bits. We also do a comparison of the proposed protocol with other protocols in terms of storage cost in the number of bits for storing various data fields in the UAV's memory. As shown in Table VI and Fig. 8, the memory storage cost in [27], [28], [33] and [29] are at least 480, 640, 320 and 640 bits respectively. Our scheme has a storage cost of 352 bits since only the identities $TUID$ and GID, and the challenge C needs to be stored in the UAV's memory. Although our scheme has 9% more storage cost compared to [33], the communication and computation costs in [33] are much higher compared to ours. The communication cost (1952 bits) in [33] is 22% higher than our scheme (1600 bits). Their computation cost on NodeMCU ($667\mu s$) is 85% higher than ours ($361\mu s$), while the cost on Raspberry Pi in C language ($51.6\mu s$), and in Python ($261\mu s$) is 63% and 20% higher than ours in C language ($31.7\mu s$) and in Python ($218\mu s$) respectively. Thus, we have shown that our scheme is a very good choice compared to the existing authentication schemes for UAVs in terms of computation capabilities of UAVs, and storage and communication costs involved.

VII. Conclusion

This paper proposed SecAuthUAV, a lightweight mutual authentication protocol for UAV-ground station authentication using physically unclonable functions (PUFs) which eliminates the need for storing any secrets on the UAVs. This protocol is extended further to support UAV-UAV authentication. In the process of authenticating with the ground station, a secure session key is established between them using the secret information generated using PUFs. SecAuthUAV assures essential security features such as mutual authentication, UAV anonymity, and forward secrecy. It is also secure against common security attacks such as masquerade, man-in-the-middle, replay, cloning, and de-synchronization attacks in addition to UAV capture and tampering attacks. Despite not storing any secret keys, it is capable of generating a unique session key for every session. Moreover, SecAuthUAV

fares competitively well in comparison with the other existing authentication schemes for UAVs in terms of computation, communication, and storage costs. Hence, we argue that the proposed protocol, SecAuthUAV is very efficient and viable for UAV-GS authentication.

REFERENCES

[1] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "A comprehensive review of the covid-19 pandemic and the role of iot, drones, ai, blockchain, and 5g in managing its impact," *IEEE Access*, vol. 8, pp. 90 225–90 265, 2020.

[2] X. Liu, Z. Li, N. Zhao, W. Meng, G. Gui, Y. Chen, and F. Adachi, "Transceiver design and multihop d2d for uav iot coverage in disasters," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1803–1815, 2018.

[3] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. R. Yu, and K.-K. Wong, "Uav-assisted emergency networks in disasters," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 45–51, 2019.

[4] T. Alladi, V. Chamola, N. Sahu, and M. Guizani, "Applications of blockchain in unmanned aerial vehicles: A review," *Vehicular Communications*, p. 100249, 2020.

[5] M. Singh, G. S. Aujla, and R. S. Bali, "A deep learning-based blockchain mechanism for secure internet of drones environment," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[6] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5g-based iot-enabled internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9097–9111, 2020.

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

[8] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for v2g using physical unclonable function," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7234–7246, 2020.

[9] B. Chatterjee, D. Das, S. Maity, and S. Sen, "Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 388–398, 2018.

[10] T. Alladi, V. Chamola, N. Kumar *et al.*, "Parth: A two-stage lightweight mutual authentication protocol for uav surveillance networks," *Computer Communications*, vol. 160, pp. 81–90, 2020.

[11] B. A. Alzahrani, S. A. Chaudhry, A. Barnawi, A. Al-Barakati, and M. H. Alsharif, "A privacy preserving authentication scheme for roaming in iot-based wireless mobile networks," *Symmetry*, vol. 12, no. 2, p. 287, 2020.

[12] F. Al-Turjman, Y. K. Ever, E. Ever, H. X. Nguyen, and D. B. David, "Seamless key agreement framework for mobile-sink in iot based cloud-centric secured public safety sensor networks," *IEEE Access*, vol. 5, pp. 24 617–24 631, 2017.

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

[14] D. He, S. Chan, and M. Guizani, "Communication Security of Unmanned Aerial Vehicles," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 134–139, 2016.

[15] V. Hassija, V. Chamola, D. N. G. Krishna, and M. Guizani, "A distributed framework for energy trading between uavs and charging stations for critical applications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5391–5402, 2020.

[16] V. Hassija, V. Saxena, and V. Chamola, "Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory," *Computer Communications*, vol. 149, pp. 51–61, 2020.

[17] M. Hooper, Y. Tian, R. Zhou, B. Cao, A. P. Lauf, L. Watkins, W. H. Robinson, and W. Alexis, "Securing commercial wifi-based uavs from common security attacks," in *MILCOM 2016-2016 IEEE Military Communications Conference*, 2016, pp. 1213–1218.

[18] N. Zhao, X. Pang, Z. Li, Y. Chen, F. Li, Z. Ding, and M.-S. Alouini, "Joint trajectory and precoding optimization for uav-assisted noma networks," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3723–3735, 2019.

[19] W. Wang, J. Tang, N. Zhao, X. Liu, X. Y. Zhang, Y. Chen, and Y. Qian, "Joint precoding optimization for secure swipt in uav-aided noma networks," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 5028–5040, 2020.

[20] N. Zhao, Y. Li, S. Zhang, Y. Chen, W. Lu, J. Wang, and X. Wang, "Security enhancement for noma-uav networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3994–4005, 2020.

[21] O. Blazy, P.-F. Bonnefoi, E. Conchon, D. Sauveron, R. N. Akram, K. Markantonakis, K. Mayes, and S. Chaumette, "An efficient protocol for uas security," in *2017 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, 2017, pp. 1–21.

[22] T. Alladi, V. Chamola, B. Sikdar, and K.-K. R. Choo, "Consumer iot: Security vulnerability case studies and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 17–25, 2020.

[23] L. Chen, S. Qian, M. Lim, and S. Wang, "An enhanced direct anonymous attestation scheme with mutual authentication for network-connected UAV communication systems," *China Communications*, vol. 15, no. 5, pp. 61–76, 2018.

[24] A. Abdallah, M. Z. Ali, J. Mišić, and V. B. Mišić, "Efficient security scheme for disaster surveillance uav communication networks," *Information*, vol. 10, no. 2, p. 43, 2019.

[25] B. Semal, K. Markantonakis, and R. N. Akram, "A certificateless group authenticated key agreement protocol for secure communication in untrusted uav networks," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018, pp. 1–8.

[26] G. K. Verma, B. Singh, N. Kumar, and D. He, "Cb-ps: An efficient short-certificate-based proxy signature scheme for uavs," *IEEE Systems Journal*, vol. 14, no. 1, pp. 621–632, 2019.

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

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

[29] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43 711–43 724, 2020.

[30] D. Choi, S.-H. Seo, Y.-S. Oh, and Y. Kang, "Two-factor fuzzy commitment for unmanned iot devices security," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 335–348, 2018.

[31] T. Alladi, V. Chamola, and S. Zeadally, "Industrial control systems: Cyberattack trends and countermeasures," *Computer Communications*, vol. 155, pp. 1–8, 2020.

[32] B. Chen and F. M. Willems, "Secret key generation over biased physical unclonable functions with polar codes," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 435–445, 2018.

[33] C. Pu and Y. Li, "Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system," in *2020 IEEE International Symposium on Local and Metropolitan Area Networks*, 2020, pp. 1–6.

[34] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

[35] D. Wang, H. Cheng, D. He, and P. Wang, "On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices," *IEEE Systems Journal*, vol. 12, no. 1, pp. 916–925, 2016.

[36] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *[1993] Proceedings Computer Security Foundations Workshop VI*, 1993, pp. 147–158.

[37] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE transactions on dependable and secure computing*, vol. 15, no. 4, pp. 708–722, 2016.

[38] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[39] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. Rodrigues, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 457–468, 2018.

[40] X. Zhao, Q. Zhao, Y. Liu, and F. Zhang, "An ultracompact switching-voltage-based fully reconfigurable rram puf with low native instability," *IEEE Transactions on Electron Devices*, vol. 67, no. 7, pp. 3010–3013, 2020.