Achieving Secure and Reliable UAV Authentication: A Shamir's Secret Sharing Based Approach

Gaurang Bansal, *Member, IEEE*, and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—In recent years, the rapid development of wireless communication-based technologies has been particularly prominent in the context of Unmanned Aerial Vehicle (UAV) applications. However, despite significant progress, the realization of the full potential of UAV-based applications is hindered by inherent security vulnerabilities, including the susceptibility of communication between drones and base stations to intrusion. This research addresses critical research gaps by proposing a novel authentication method that leverages Shamir's secret sharing. Unlike existing authentication protocols, which often rely on Physically Unclonable Functions (PUFs) that assume resistance to noise and theoretical ideals, our approach acknowledges the potential errors in PUF responses. This accommodation allows for the successful authentication of legitimate UAVs, even in the presence of external environmental factors introducing noise in PUFs. To evaluate the effectiveness of our proposed method, extensive simulations were conducted using NodeMCU and Raspberry Pi devices. We show that our approach outclasses other existing methods across multiple dimensions with communication cost of mere $450\mu s$ on NodeMCU. Additionally, it incurs lower communication costs (1600 bits) and storage costs (352 bits), showcasing superior efficiency. Moreover, it maintains robust security and scalability while consuming reasonable energy levels, making it a comprehensive solution for UAV-based applications.**

***Index Terms*—UAVs, Authentication, Fault tolerance, Protocol performance.**

I. Introduction

In recent years, the widespread adoption of Unmanned Aerial Vehicles (UAVs) has revolutionized wireless communication systems, creating a significant impact in the field. However, their deployment in open and unpredictable settings exposes them to potential threats from both human interference and environmental factors [1, 2]. To ensure secure and reliable communication between UAVs and the base station, it is crucial to establish a robust and protected communication channel [3, 4].

Figure 1 illustrates the system model that encompasses the UAVs, base stations, and potential adversaries. Human interference encompasses deliberate actions by adversaries to disrupt the communication between the transmitter and the receiver. Examples of such interference include data blocking, eavesdropping, and device capture, all of which can compromise the security of UAV communications. Therefore, it becomes essential to establish a secure channel to prevent unauthorized access, data corruption, and malicious interference [4].

The realm of UAVs is undergoing transformative advancements, but with this progress comes the heightened necessity for robust security measures. A paramount concern in this domain is the consistent and reliable authentication of these devices, ensuring that they are safeguarded against unauthorized access and potential operational breaches [5]. UAVs, by their very nature, are in constant motion, traversing diverse terrains and environments. This dynamism leads to frequent shifts in several operational parameters such as communication statuses and connectivity to base stations. As they navigate through different regions, changes in their operational state become inevitable. Thus, regular and meticulous device verification becomes indispensable to ensure that UAVs remain shielded from any malicious interference or tampering [6].

Over the years, the research community has proposed a multitude of authentication mechanisms tailored to address the unique challenges posed by UAVs. Each method brings its own set of prerequisites and trust foundations. Among the myriad of solutions, Physically Unclonable Functions (PUFs) based approaches have emerged as particularly promising [7]. The strength of PUFs lies in their ability to harness the innate randomness that surfaces during the fabrication of silicon devices. This inherent characteristic ensures that each PUF is distinct, making them a formidable challenge

Fig. 1. Isometric view of system model where ground stations, seamlessly communicate with a fleet of UAVs. Trusted drones, marked in green, transmit secure data, while black UAV, symbolizing malicious entities, trigger alerts.

Gaurang Bansal and Biplab Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077, Singapore (email: e0622339@u.nus.edu, bsikdar@u.nus.edu)

for adversaries to replicate. However, the very strength of PUFs—their sensitivity to environmental conditions—can also be a vulnerability. External factors can induce variations in PUF responses, making it crucial to craft protocols capable of differentiating genuine environmental variances from potential adversarial modifications. The evolving landscape of UAV operations and the challenges therein highlight the pressing need for authentication solutions that are both resilient and adaptive.

The novelty of our approach lies in its efficiency and fault tolerance. Initiating the authentication process, the base station transmits a set of challenges to the UAV, which then employs its PUF to generate corresponding responses. Instead of the traditional pairwise comparisons, our protocol employs Shamir's secret sharing algorithm [8]. This algorithm mandates each UAV to produce at least t accurate responses out of the total k challenges presented by the base station. By sidestepping the need for pairwise comparisons, our method drastically reduces the computational complexity, thereby ensuring a high degree of fault tolerance and setting a new benchmark in UAV authentication processes for the UAV community.

The main contributions of this paper are as follows:

- **Fault-Tolerant Authentication Mechanism:** The paper introduces a novel authentication method tailored for UAVs, leveraging PUF-based challenge-response pairs. This mechanism not only capitalizes on the inherent randomness of PUFs but also incorporates measures to discern genuine environmental variations from adversarial manipulations, ensuring secure UAV communications.
- **Efficiency through Shamir's Secret Sharing:** By employing Shamir's secret sharing algorithm in the authentication protocol, the paper presents an efficient approach that mandates each UAV to produce a minimum number of accurate responses. This eliminates the need for exhaustive pairwise comparisons, thereby reducing computational complexity from $O(k^2)$ to approximately $O(k)$ operations.
- **Setting a New Benchmark for UAV Community:** The paper offers a robust and efficient authentication protocol. This sets a new benchmark in UAV authentication processes, ensuring a higher degree of fault tolerance and security in the evolving UAV landscape.

The structure of this paper unfolds as follows. In Section II, we delve into existing UAV authentication methods, underscoring their constraints and laying the groundwork for our novel proposal. Section III introduces the concept of PUFs, elucidating their significance in UAV authentication. The intricacies of our innovative fault-tolerant authentication approach, encompassing the threshold-based verification procedure, are unveiled in Section IV. A comprehensive security examination of the introduced protocol, employing both Mao-Boyd logic and cryptanalysis methodologies, is presented in Section V. Section VI presents the efficacy of our approach against prevailing methods. Finally, Section VII presents the conclusions of the paper.

II. Related Work

UAVs exhibit specific characteristics that set them apart from other distributed network systems. These characteristics include their distinct topology, mobility, etc. These factors contribute to the unique nature of UAVs in the context of network systems. As a result, conventional security measures designed for distributed networks do not yield comparable outcomes when applied to UAVs [9, 10]. Consequently, the widespread adoption of UAVs has been hindered by various security concerns [11, 12]. In response to these challenges, considerable research efforts have been dedicated to implementing lightweight security provisions specifically tailored for UAVs [13–15].

Hooper et al. [16] was the first work highlighting the security vulnerabilities in UAV communication. The work gave an idea of a lightweight authentication framework, but the conceptualization was theoretical and weak. Also, there was no formal verification. This framework was further extended by Blazy et al. in [17]. It presented the first formal authentication protocol with formal proofs. The protocol in [17] has a disadvantage: the adversarial model is quite weak. The authors did not consider multiple scenarios such as replay attacks, physical attacks, etc. In their study [18], the authors propose the utilization of a secure channel incorporating a collection of random numbers to facilitate continuous authentication. Specifically, in the context of the protocol execution, the base station employs this array as a challenge to authenticate the UAVs. The approach recommended in [18] emphasizes the use of a secure channel and random number arrays to ensure ongoing and robust authentication throughout the protocol.

The authors of [19] introduced a pioneering distributed key authentication system for wireless mesh networks, which incorporated the use of a Certificate Authority (CA). Their study made a significant contribution to authentication mechanisms among entities within these networks. In [19], encryption operations were employed to generate public and private keys for all participating entities based on a unique identifier. The CA played a crucial role in generating new keys for authentication, updating the unique identifier during each verification cycle. However, this approach had drawbacks related to its reliance on centralized trusted parties and the computational demands it imposed. To address the issue of high computational overhead, subsequent works such as [20] and [21] introduced authentication techniques based on bilinear pairing and elliptic curve cryptography, respectively. Although these approaches mitigated the computational burden, they did not provide resilience against physical attacks or ensure security in the event of device failures. Furthermore, UAV-UAV authentication was not addressed in these works.

The aforementioned limitations were addressed in the work by Alladi et al. [22], which provided physical security and demonstrated remarkable computational efficiency. However, this solution lacked scalability. In separate studies, works such as [23] and [24] addressed scalability concerns using different techniques. Regrettably, none of these works offered resilience to faults, which are commonly encountered in UAV networks. A summary of the existing literature is presented in

Works	Author	Contributions	Scheme	Limitation
[16]	Hooper et al.	Highlighted the security vulnerabilities in UAV scenario.	HMAC	No formal protocol to support UAV authentication.
[17]	Blazy et al.	One of the initial authentication protocols with formal security analysis.	Encryption, HMAC	The adversarial model was robust but not complete. Replay attacks and physical attacks were not addressed.
[18]	Yoon et al.	Proposed additional encrypted communication channel hijacking network channel.	AES, SSL/TLS	Requires separate dedicated hardware.
[19]	He et al.	Distributed key authentication system that used a Certificate Authority (CA) for UAV.	Distributed Key Auth	Dependency on centralized trusted parties and the requirement of high computational computing.
[20]	Wazid et al.	Lightweight authentication technique based on bilinear pairing.	bilinear pairing	No physical security or fault-tolerance.
[21]	Jangirala et al.	Rapid authentication technique based on Elliptic curve cryptography (ECC).	ECC	No physical security, no fault tolerance, no support for peer-to-peer authentication.
[22]	T. Alladi et al.	UAV-GS authentication scheme is extended further to support UAV-UAV authentication.	PUF,ECC	Supports one-one authentication alone.
[23]	G. Bansal et al.	Scalable UAV authentication protocol using PUFs (resistant to physical attacks).	PUF, Public Key Infrastructure (PKI)	Does not support fault tolerance.
[24]	G. Bansal et al.	Lightweight, scalable and location centric based UAV authentication protocol.	PKI, PUF, AES	Does not support fault tolerance.

TABLE I
SUMMARY OF RELATED WORKS

Table I, providing an overview of the key characteristics and limitations of each study.

Thus, we propose a fault-tolerant authentication mechanism to address the challenges of fault tolerance and accommodate variations in PUFs caused by environmental factors. Our approach is based on Shamir's secret sharing scheme [8], which ensures both fault tolerance and robust security. In our protocol, the original secret is divided into multiple shares and distributed among shareholders. The secret key can be reconstructed when a specified threshold of shares is present. However, if the number of available shares is below the threshold, reconstructing the secret becomes computationally infeasible. In our authentication protocol, we adopt a different approach by selectively verifying a subset of PUF responses based on a predefined threshold, rather than validating all responses. This selective verification process improves the efficiency of the authentication process while maintaining the required level of security. This selective verification effectively mitigates the impact of environmental variations on PUFs. Additionally, we demonstrate that our proposed approach offers strong mathematical security guarantees and achieves completeness. By incorporating Shamir's secret sharing scheme, our protocol provides not only fault tolerance but also enhances the overall security of the authentication process.

III. Background Knowledge

A. Physically Unclonable Functions

PUFs are increasingly being used to secure IoT devices, data, and services as a hardware root of trust [25]. An integrated circuit's manufacturing process variations may be utilized directly as a source of randomization to create device-unique cryptographic keys to enable secure applications. The outputs (or responses) of a high-quality PUF are consistent and random. The implementation, behavior, and post-processing or key extraction of a PUF influence its dependability. PUFs are naturally unclonable and provide a one-of-a-kind response to each challenge. For example, the challenge in SRAM PUF might be an array of memory locations, with the answer or output being produced by concatenating the memory values of SRAM cells. This pattern may be used to generate a hardware-based device-specific answer [26].

PUFs are not always deterministic and ideal, even though this assumption is made in most current literature. Instead, real-life PUFs follow a probabilistic behavior. For example, analytically, each SRAM cell has two stable states, one representing a 1 and the other as 0. The resulting state when a cell is turned on is unknown, and the random sub-microscopic variations between the cell's transistors give each cell a preference to come up as a 0 or a 1. During SRAM power cycles, some closely balanced cells might become unstable and create inverted bit values (cell flipping) on the original pattern of zeros and ones.

Thus, every time the device is turned on, there might be a slightly different response for a given challenge. However, most of the bit-values in a PUF's output will be consistently generated based on the original silicon fingerprint and will be unique to the integrated circuit chip. The PUF noise is defined as the number of inverted bits divided by the number of bits in the pattern. Local temperature changes, supply voltage variations, and aging are the significant causes of cell flipping. As a result, the reliability of PUFs is an open research area that attracts considerable community attention. The effect of environmental and operating conditions on the PUF's operation is not limited to SRAM PUFs and can be seen in all types of PUFs [27, 28].

The details regarding the challenge-response correspondence of PUFs are mentioned below:

1) A response R_i (to a challenge C_i) is expected to provide no information about another response R_j (to a separate challenge C_j).
2) Without the right PUF on hand, it is impossible to develop the right response.
3) PUFs are considered to be tamper-proof. If an intruder tries to examine the PUF to learn more about its construction physically, the PUF will be destroyed.
4) A strong PUF has many challenge-response pairings (C_i, R_i). A PUF contains so many CRPs that an attack based on meticulously measuring the CRPs over a short period has little probability of succeeding.
5) Local temperature changes, local supply voltage variations, and aging can affect PUF responses.

B. Shamir's Secret Sharing

Shamir's Secret Sharing is a cryptographic technique developed by Adi Shamir in 1979 [8]. It is widely used for securing sensitive data by dividing it into multiple shares, such that a predefined number of shares are required to reconstruct the original secret. This approach offers a robust mechanism for distributing trust among multiple entities, preventing any single point of failure. In the context of our authentication protocol, we utilize Shamir's Secret Sharing to distribute the responsibility of generating and verifying authentication responses among multiple UAVs. This not only enhances fault tolerance but also reduces computational complexity by eliminating the need for pairwise comparisons among all responses.

C. Adversary Model

The adversary model as described in [29, 30] is highlighted below:

- The adversary cannot collect any information during the enrollment phase because this phase takes place in a protected environment.
- The adversary can access the interface of the device and the unprotected communication channel between the two entities of our system.
- We presume that an attacker has complete control over the whole network at all times (Dolev-Yao model) [31]. An attacker can access the receiver's and sender's unencrypted communication. Depending on the situation, an attacker may impersonate a legal UAV or interfere with the current communication exchanges.
- In addition to man-in-the-middle attacks, manipulation with communications is possible. An attacker may intercept, change, or replay communications while they are sent over the network.
- PUFs are thought to be tamper-proof.
- In the authentication step, the adversary has access to the device's database. It is also possible for an attacker to seize a UAV, interrupt any transmission, and decode confidential information using brute force.

D. Notations

Table II contains the notations to used in the protocol.

Symbol	Description
id	Unique device identifier for each UAV
t	Degree of the Galois field polynomial and the number of correct challenge-response pairs
$f(x)$	Galois field polynomial
$GF(2^q)$	Galois Field
S	Secret value
a_i	Elements belonging to the Galois Field $GF(2^q)$
q	A constant selected by the network administrator
k	Number of random challenges
C_i	Challenges provided to the UAV
R_i'	Responses generated by the UAV's PUF
L	Set of computed values by the base station from responses
$H(S)$	Hashed form of the secret value S
N_A, N_B, N_C	Nonces used in the authentication process
C	Challenge set
R, R', R''	Different sets of responses in the authentication mechanism
S^*	Evaluated secret value
MAC	Message authentication code
PAD	Padding function
S_k	Session key
L'	Updated set of computed values by the base station from new responses
S'	Updated secret value
Q	Value generated using XOR operation between N_B and $H(S)$
C'	New challenge set generated by the UAV
P	Message created by the UAV
J	Another message created by the UAV
ϕ	Represents an empty set

TABLE II
NOTATION TABLE

IV. PROPOSED PROTOCOL

In this section, we will describe the working of our proposed protocol. The protocol is divided into two phases:

1) Enrollment Phase,
2) Authentication Phase.

A. Enrollment Phase

The step-by-step procedure of the enrollment phase (shown in Figure 2) is described as follows:

- In our system, the base station is responsible for generating a unique device identifier, denoted as *id*, for each UAV. This identifier is then securely stored in the UAV's one-time programmable (OTP) memory. The OTP memory is a non-volatile type of memory, meaning that it retains the stored information even when the power supply is disconnected or stopped. This ensures that the device identifier remains intact and accessible even in the event of power loss or system shutdown. By utilizing OTP memory, we guarantee the persistent storage of the UAV's unique identifier, enabling reliable identification and authentication processes.
- The base station generates a Galois field polynomial with degree t. Here, t is the number of correct challenge-response pairs that must be generated by a device to verify its authenticity. If t is small, UAVs can be authenticated even if there are significant variations in the PUF's output, but it increases the chances of masquerade attacks. Thus, the network administrator decides the value of t depending on how the system needs to be configured. The polynomial is given by:

$$f(x) = S + a_1 x + \cdots + a_{t-1} x^{t-1}.$$

In our authentication mechanism, we consider elements a_i belonging to the Galois Field $GF(2^q)$, where i ranges from 1 to $t-1$. The value of q is a constant selected by the network administrator. The secret value S is included as the constant term in this set of elements. Upon generating S, the base station securely stores it in its memory. During the authentication phase, devices must accurately determine the value of S to successfully authenticate themselves. By correctly identifying and providing the value of S, devices demonstrate their legitimacy within the system. This authentication process relies on the manipulation and verification of the defined elements a_i within the Galois Field, with S serving as the essential secret component for authentication.

- During the enrollment phase, the base station generates a set of k random challenges denoted as $C_1, C_2, \cdots, C_k$ for each UAV. These challenges are specifically designed

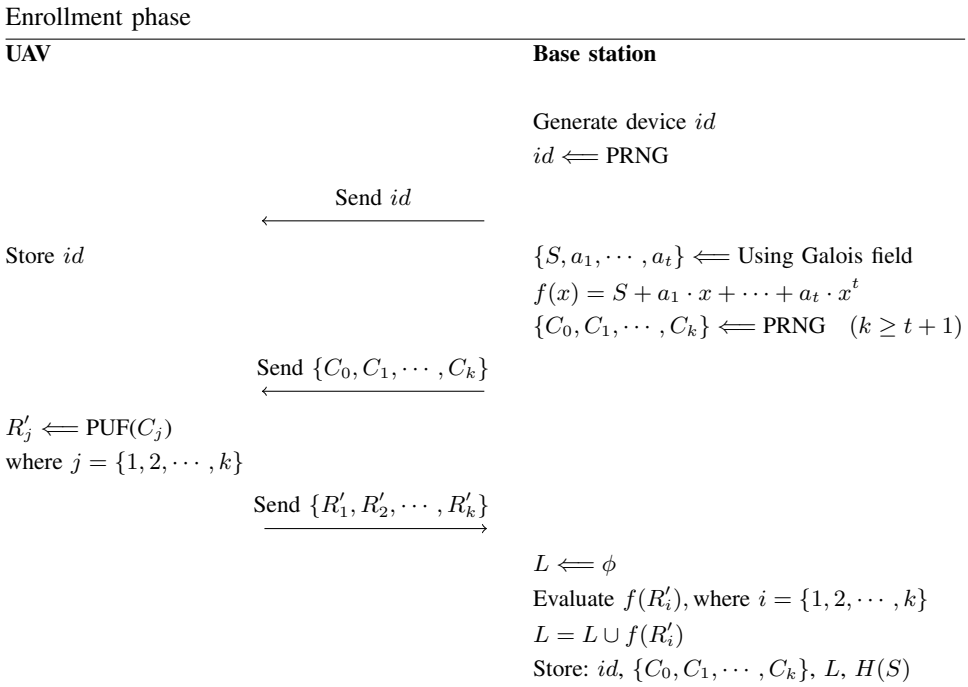

Fig. 2. Enrollment phase.

to evaluate the PUF present in the UAV's device. The challenges are then provided to the UAV. Using its PUF, the UAV generates a set of k responses, denoted as $R = R'_1, R'_2, \cdots, R'_k$, corresponding to the given challenges. These responses are generated by the UAV's PUF as a unique output for each challenge. The responses serve as the UAV's individualized and device-specific authentication information.

- Each challenge given to a UAV is associated with a single response. The response generated by the UAV's PUF is unique to that device. Even if two devices are presented with the same challenge, their responses will differ due to the inherent randomness introduced during the chip fabrication process.
- Upon receiving all the responses from a UAV, the base station performs a calculation on the set of responses. Specifically, it computes $f(R'_i)$ for each response R'_i where i ranges from 1 to k. The resulting values are then stored as the response set $L = \{f(R'_i)|i = 1, 2, \cdots, k\}$. The base station is considered to be secure entity.

B. Authentication Phase

In the authentication process, as illustrated in Figure 3, the verification of the UAV is based on the base station's ability to reconstruct the secret using the responses obtained from the UAV's PUF. This reconstruction process is transformed into a key threshold problem. In the protocol using the Shamir scheme, a threshold value t is set, and k is the total responses obtained from the UAV's PUF. The rationale behind this threshold is that if the base station possesses knowledge of any t correct responses, it can successfully authenticate the UAV. On the other hand, if fewer than t responses are correct, the secret remains indeterminate or provides no information about the actual secret. This scheme is commonly known as the Shamir (t, k) scheme.

The authentication process (shown as a flowchart in Figure 4) is carried out as follows:

- The authentication process is initiated by the UAV, which sends its device identifier (id) and a nonce (N_A) to the base station. The nonce is a 128-bit random number sent to ensure the freshness of the message. The base station checks if the UAV's id is present in its database or not. If it is present, the authentication process proceeds as follows.
- The base station extracts a challenge set C and $H(S)$ stored in memory. It generates a random nonce N_B using a pseudorandom generator function. Finally, it generates Q by using XOR operation between N_B and $H(S)$ as:

$$Q = N_B \oplus H(S).$$

- The base station sends the challenge set C, Q and L, along with the message authentication code (MAC) of aggregated message concatenation of C, $H(S)$, L and N_B, to the UAV.
- The UAV utilizes its PUF to generate k responses, denoted as $R'_1, R'_2, \cdots, R'_k$, in response to the challenges provided by the base station. However, it is important to note that due to environmental changes, the responses generated by the PUF during the authentication phase may differ from the original responses generated during the enrollment phase. Therefore, to account for these variations, we denote the responses generated by the PUF in the authentication phase as R instead of R' used in the

Device Authentication Phase

UAV		**Base Station**
$N_A \Longleftarrow$ PRNG		
	Send id, N_A $\longrightarrow$	
		$\{C_0, C_1, \cdots, C_k\}, H(S), L \Longleftarrow$ Extract from memory
		$N_B \Longleftarrow$ PRNG
		$Q \Longleftarrow N_B \oplus H(S)$
	Send $\{C_0, C_1, \cdots, C_k\}, Q, L$ $MAC(\{C_0, C_1, \cdots, C_k\}, H(S), N_B, L)$ $\longleftarrow$	
$\{R_0, R_1, \cdots, R_k\} \Longleftarrow$ PUF($\{C_0, C_1, \cdots, C_k\}$)		
Evaluate S^* using Shamir method		
$S^* = \sum_{i=1}^{t+1} L_i \prod_{j=1, j\neq i}^{t+1} \frac{R_j - X}{R_j - R_i}$		
If $H(S^*) = H(S)$		
$N_B \Longleftarrow Q \oplus H(S^*)$		
Verify $MAC(\{C_0, C_1, \cdots, C_k\}, H(S), N_B, L)$		
$N_C \Longleftarrow$ PRNG		
$P \Longleftarrow N_C \oplus H(S^*)$		
Generate new challenge set		
$C' \Longleftarrow \{C'_0, C'_1, \cdots, C'_k\}$		
$\{R''_0, R''_1, \cdots, R''_k\} \Longleftarrow$ PUF($\{C'_0, C'_1, \cdots, C'_k\}$)		
$J \Longleftarrow \{R''_0 \| R''_1 \| \cdots \| R''_k\} \oplus PAD(H(S^*))$		
$S_k \Longleftarrow N_C \oplus N_B$		
	Send $\{C'_0, C'_1, \cdots, C'_k\}, J, P$, $MAC(\{R''_0, R''_1, \cdots, R''_k\}, H(S^*), N_C)$ $\longrightarrow$	
		$\{R''_0 \| R''_1 \| \cdots \| R''_k\} \Longleftarrow J \oplus PAD(H(S))$
		$N_C \Longleftarrow P \oplus H(S)$
		Verify $MAC(\{R''_0, R''_1, \cdots, R''_k\}, H(S^*), N_C)$
		$S_k \Longleftarrow N_C \oplus N_B$
		$L' \Longleftarrow \phi$
		Evaluate $f(R''_i)$, where $i = \{1, 2, \cdots, k\}$
		$L' = L' \cup f(R''_i)$
		Evaluate S' using Shamir method
		$S' = \sum_{i=1}^{t+1} f(R''_i) \prod_{j=1, j\neq i}^{t+1} \frac{R''_j - X}{R''_j - R''_i}$
		Remove: id, $\{C_0, C_1, \cdots, C_k\}$, L, $H(S)$
		Store: id, $\{C'_0, C'_1, \cdots, C'_k\}$, $H(S')$, L'

Fig. 3. Authentication phase.

enrollment phase. This distinction acknowledges that the responses from the PUF during the authentication phase may not necessarily match the original responses obtained during enrollment.

- Then, the UAV generates S^* using L and applies the Shamir secret key generation algorithm as provided in [8] as:

$$S^* = \sum_{i=1}^{t+1} L_i \prod_{j=1, j\neq i}^{t+1} \frac{R_j - X}{R_j - R_i}.$$

The threshold parameter, denoted as t, plays a crucial role in our authentication mechanism. With a total of k responses to be generated by the UAV, if the UAV can produce t correct responses out of the k challenges, it ensures the successful generation of the secret value S. In other words, if the number of correct responses is less than the threshold value t, it renders the secret completely undetermined or reveals no information about the secret. Thus, the threshold value t serves as a minimum requirement for the UAV to authenticate itself by generating the

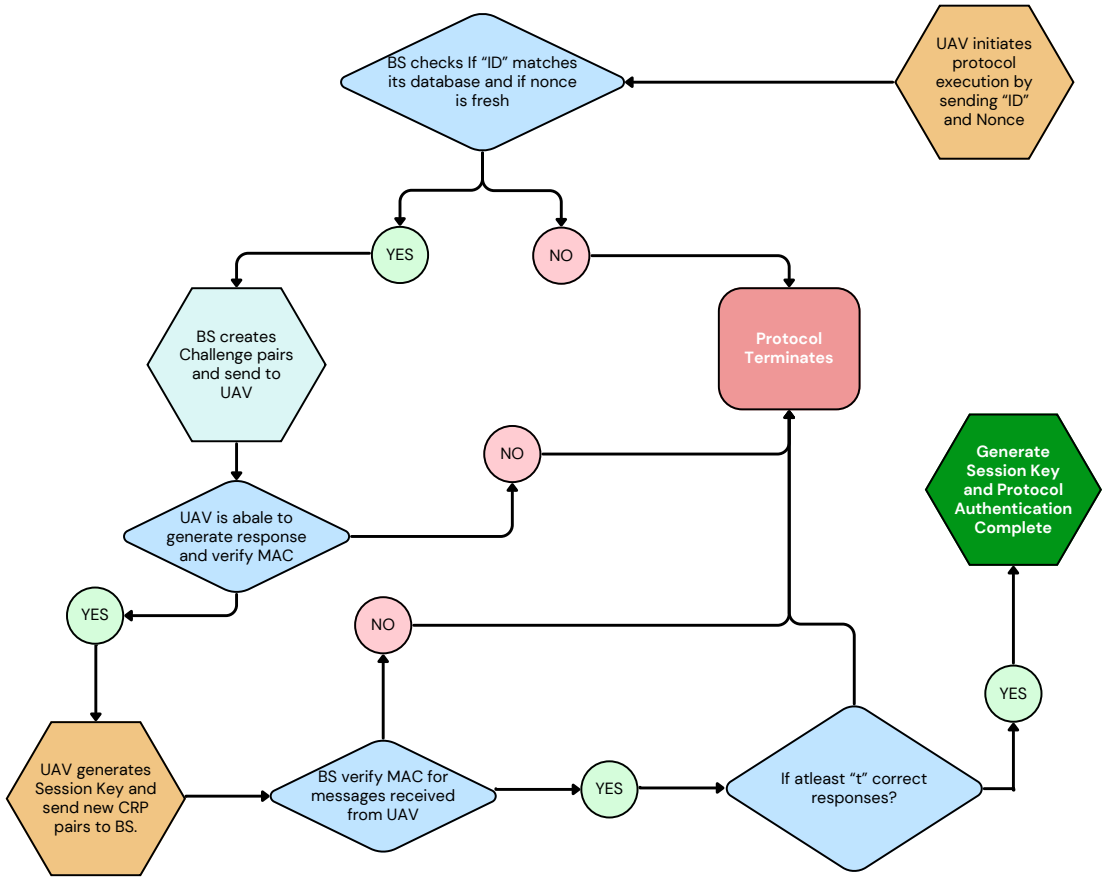

Fig. 4. Flowchart of protocol.

necessary number of correct responses.

- After evaluating $H(S^*)$, the UAV extracts N_B using Q as:

$$N_B \Longleftarrow Q \oplus H(S^*).$$

Then, the UAV verifies the MAC received from the base station and $MAC(\{C_0, C_1, \cdots, C_k\}, H(S^*), N_B, L)$. If the UAV can verify the MAC, it is assumed that the base station is authentic. Else, the protocol is aborted.

- After successful verification of the base station by the UAV, the UAV must prove its authenticity to the base station. It generates a random nonce N_C using PRNG (Psedo Random Generator Function), a new challenge set C', and a response set where:

$$N_C \Longleftarrow \text{PRNG},$$
$$C' \Longleftarrow \{C'_0, C'_1, \cdots, C'_k\},$$
$$\{R''_0, R''_1, \cdots, R''_k\} \Longleftarrow \text{PUF}(\{C'_0, C'_1, \cdots, C'_k\}).$$

- The UAV creates two messages, P and J, to be transmitted to the base station as:

$$P \Longleftarrow N_C \oplus H(S^*),$$
$$J \Longleftarrow \{R''_0||R''_1||\cdots||R''_k\} \oplus PAD(H(S^*))$$

where PAD is a function that adds padding bits to $H(S^*)$ for proper XOR operation.

- Finally, the UAV evaluates the session key as:

$$S_k \Longleftarrow N_C \oplus N_B$$

and sends $\{C'_0, C'_1, \cdots, C'_k\}$, J, and P, along with integrity check, to the base station.

- On receiving the message with $\{C'_0, C'_1, \cdots, C'_k\}$, J, and P from the UAV, the base station evaluates the response set and N_C using its stored shared secret key:

$$\{R''_0||R''_1||\cdots||R''_k\} \Longleftarrow J \oplus PAD(H(S)),$$
$$N_C \Longleftarrow P \oplus H(S).$$

If the base station is able to successfully verify $MAC(\{R''_0, R''_1, \cdots, R''_k\}, H(S^*), N_C)$, the base station is sure that the UAV has correctly evaluated the secret $H(S)$.

- The base station evaluates the session key as:

$$S_k \Longleftarrow N_C \oplus N_B.$$

- Finally it updates its stored values by replacing L with L', S with S' and $\{C_0, C_1, \cdots, C_k\}$ with $\{C'_0, C'_1, \cdots, C'_k\}$. The evaluation of these variables is as follows:

$$L' \Longleftarrow \phi,$$
$$\text{Evaluate } f(R''_i), \text{where } i = \{1, 2, \cdots, k\},$$
$$L' = L' \cup f(R''_i),$$

$$S' = \sum_{i=1}^{t+1} f(R_i'') \prod_{j=1, j\neq i}^{t+1} \frac{R_j'' - X}{R_j'' - R_i''}.$$

V. Security Analysis

This section presents a formal security proof and conventional cryptanalysis to evaluate the proposed authentication phase's security. This approach is necessary since formal evidence alone does not offer sufficient security or flexibility [4]. We use Mao Boyd logic [32] and conventional cryptanalysis as stated by the authors in [33] to establish the protocol's security.

A. Formal proof

A comprehensive security analysis of our protocol is presented in a formal manner by employing Mao-Boyd logic [32] to model the communication within the protocol. The notation for symbols utilized in Mao-Boyd logic follows the conventions established in the original work [32].

Claim. The UAV knows that N_B is a valid shared and secure message between the UAV and the base station.

Proof. The proof has the inherent assumption that PUF is secure and untameable. Any change in the PUF will result in a response different from the initial response when a challenge is given. Also, during the registration phase, communication among the entities is secure and private. During the enrolment phase, the UAV evaluates the responses from the PUF corresponding to the challenges generated by the base station. The base station does not store the response in plain text. Instead, it saves them in the form of L, an aggregate set of $f(R)$. Thus, the secret $H(S)$ generated after the enrolment phase is safe and known only to the base station. Additionally, we presume that the base station is trustworthy and cannot be hacked.

The base station stores the secret value S in its memory during the registration phase as $H(S)$. The UAV (denoted by α) knows that it can regenerate the secret S, by using PUF responses $\{R_0, R_1, \cdots, R_k\}$ for challenges $\{C_0, C_1, \cdots, C_k\}$ given by the base station (denoted by β) using Shamir secret key sharing as:

$$S^* = \sum_{i=1}^{t+1} f(R_i) \prod_{j=1, j\neq i}^{t+1} \frac{R_j - X}{R_j - R_i}.$$

If the UAV generates more than t correct responses, then it is guaranteed that S^* will be same as S. Thus, the statement "*UAV knows that $H(S)$ is a shared secret between UAV and BS*" can be expressed as:

$$\alpha \models \alpha \overset{H(S)}{\leftrightarrow} \beta. \tag{i}$$

From Fig. 3, we can observe that the UAV can extract the correct value of nonce N_B generated by the base station from Q. The UAV can perform an XOR operation between Q and $H(S)$ as:

$$N_B \Longleftarrow Q \oplus H(S).$$

Thus, Mao Boyd logic equivalent of "*UAV obtaining N_B using $H(S)$*" can be written as:

$$\alpha \overset{H(S)}{\triangleleft} N_B. \tag{ii}$$

Using the authentication rule (ref. [32]) on (i) and (ii), we get "*the UAV knows the base station encrypted N_B using the key $H(S)$*" as

$$\alpha \models \beta \overset{H(S)}{\mid\sim} N_B. \tag{iii}$$

The base station generates a nonce N_B, and thus, the statement "*base station super principle for the UAV with respect to N_B*" can be formulated as:

$$\alpha \models sup(\beta). \tag{iv}$$

Foe each protocol execution, the nonce is generated randomly using a pseudorandom number generator function. Thus, the nonce N_B generated by the base station is guaranteed to be fresh. So, "*UAV also knows the freshness of nonce*", which can be formulated as:

$$\alpha \models \#(N_B), \tag{v}$$

The base station sends message Q to the UAV, and N_B can be extracted by performing an XOR operation as:

$$N_B \Longleftarrow Q \oplus H(S).$$

Given that the secret $H(S)$ is known to the base station and the UAV, it is established that "U_j is aware that no one other than BS knows N_B," which can be represented as:

$$\alpha \models \beta^c \triangleleft \parallel N_B. \tag{vi}$$

By applying the confidentiality rule using Equations (i), (iii), and (vi), we can conclude that the UAV is convinced that no one else except itself and the base station knows the secret nonce N_B:

$$\alpha \models \alpha, \beta^c \triangleleft \parallel N_B. \tag{vii}$$

Finally, the statement "*UAV is convinced of the shared secret N_B between UAV and the base station*" can be proved by applying the good-key rule to (v), and (vii) as:

$$\alpha \models \alpha \overset{N_B}{\leftrightarrow} \beta. \tag{viii}$$

Similarly, we can prove the secrecy of session key Sk and the nonce generated by the UAV (N_C). This completes the Mao Boyd logic-based proof of secure communication between the UAV and the base station. □

B. Cryptanalysis

For the cryptanalysis in this paper, we use the set of security criteria presented in [33] that eliminates redundancies and ambiguities often seen in security procedures.

[C1] Resistance to masquerade and MITM attacks: The proposed protocol offers resistance to MITM (Man-In-the-Middle) and masquerade attacks. An adversary A is unable to masquerade as a UAV due to the absence of the matching PUF, and it is also unable to masquerade as BS due to the

Scheme	C1	C2	C3	C4	C5	C6	C7
[20]	✓	✓	✗	✓	✓	✗	✗
[34]	✓	✓	✗	✓	✓	✗	✗
[35]	✓	✓	✓	✓	✓	✓	✗
[36]	✓	✓	✗	✓	✓	✗	✗
Ours	✓	✓	✓	✓	✓	✓	✓

TABLE III
COMPARISON OF SECURITY FEATURES

absence of the matching secret value $H(S)$ at the BS. As a consequence, masquerade and MITM attacks are rendered ineffective.

[C2] Protection against replay attacks: Attempts by an attacker to replay any previous messages to either the base station or the UAV would fail under the proposed protocol since each session has a new nonce, and the adversary does not have access to this new nonce. The nonce is randomly generated for each session. To address this concern effectively, we have employed N_A and N_B as nonces, ensuring that the issue of replay attacks is mitigated.

[C3] Defense against cloning attacks and node manipulation attacks: Due to PUFs' inherent security against copying, an adversary cannot successfully clone the UAV or its PUF. Thus, this guarantees the proposed scheme's security against cloning attacks. In addition, any attempt to tamper with the PUF makes it inoperable and ineffective. Thus, device acquisition and attacks such as node tampering do not provide critical information about the authentication process.

[C4] Authentication: Each session is characterized by randomly generated nonces, namelyN_B and N_C, from the base station and UAV, respectively. This randomness ensures that every session is distinct, thereby thwarting any replay attacks. Augmenting this, the protocol's challenge-response mechanism mandates that for each unique challenge C presented, there is a valid corresponding response R. This dynamic interplay ensures that both the sender and receiver can only proceed upon successful validation, thus reinforcing the authentication process. Moreover, the derived session key, represented as S_k, is renewed for each session. Its derivation is a secure function of the nonces and the challenge-response pairs, ensuring its uniqueness and security. This session key is a testament to mutual authentication, as both the base station and UAV can only possess the same S_k upon successful validation of each other.

[C5] Provision of key agreement: Our protocol is designed to establish a secure session key, denoted as (S_k), which is consistent and recognized by both the base station and the UAV. The generation of this session key involves mutual cryptographic challenges and responses from both parties, ensuring that both the base station and the UAV have contributed to its creation. This mutual participation ensures that an eavesdropper or a Man-in-the-Middle attacker cannot predict or deduce the session key without access to the private challenge-response mechanics of the involved parties. Furthermore, the session key's consistent regeneration ensures that even if a particular session key is compromised, future sessions remain secure.

[C6] No clock synchronization: The protocol guarantees message freshness by utilizing random nonces. Thus, the proposed security method is free of time delay and clock synchronization issues.

[C7] Fault Tolerance: We employ Shamir's secret key sharing, where multiple collaborative responses generate a secret, $H(S)$. Using Shamir's secret key sharing, we ensure that the system maintains a k degree of resilience. In the proposed protocol, if a UAV has more than k correct responses for the challenge, the UAV can generate the secret $H(S)$ correctly. Thus, the proposed protocol ensures fault tolerance.

After discussing the security features of the proposed protocol and the mechanism through which they are achieved, Table II compares our protocol to several current security protocols: [20, 34, 36] and [35]. The '✓' and '✗' in the table show if a protocol meets or does not meet a requirement. As previously stated, our proposed approach meets all of the requirements. All protocols evaluated defend against masquerade attacks (C1), MITM attacks (C1), and replay attacks (C2), as well as establish session keys (C4) and provide mutual authentication (C5). Both the proposed protocol and [35] provide security against node manipulation and cloning threats by using PUFs in the UAVs (C3). Works such as [20, 34, 36] need the synchronisation of all network entities' clocks, and hence do not fulfil the characteristic of no clock synchronisation (C6). None of the preceding state-of-the-art procedures ensures fault tolerance (C7).

The developed authentication protocol also robustly secures the UAV against critical cyber-attacks. It adeptly addresses routing misbehavior by validating the authenticity of communication entities, ensuring no unauthorized interventions disrupt routing. In the face of flooding attacks, the protocol's reliance on PUFs and Shamir's secret sharing permits only authenticated drones to transmit data, thereby mitigating potential overloads. Furthermore, the scheme's periodic verification acts as a safeguard against selective forwarding attacks, identifying and flagging drones that intentionally drop packets. Collectively, these measures underscore the protocol's commitment to preserving the integrity and security of the UAV against prevalent cyber threats.

C. Security Analysis using AVISPA

In the development of our protocol, we employed AVISPA (Automated Validation of Internet Security Protocols and Applications) [14] to ensure a robust and secure design. The implementation process began with the definition of two

Operation	Notation	Time Taken (s)	
		NMCU	RPI
Bitwise XOR	T_a	3.63E-06	1.18E-05
PRNG	T_b	4.09E-06	5.90E-06
Hash	T_c	9.10E-05	1.83E-05
HMAC	T_d	3.09E-04	9.90E-05
PUF	T_e	4.00E-07	4.00E-07
Concatenation	T_f	4.50E-06	5.09E-06

TABLE IV
TIME TAKEN (IN SEC) FOR OPERATIONS ON NODEMCU AND RPI.

Operation / Time (s) [RPi]	[20]	[34]	[36]	[35]	Our
T_a	4.72E-05	3.54E-05	3.54E-05	0.00E+00	4.72E-05
T_b	5.90E-06	5.90E-06	5.90E-06	2.95E-05	1.18E-05
T_c	1.28E-04	1.28E-04	1.28E-04	0.00E+00	1.83E-05
T_d	0.00E+00	0.00E+00	0.00E+00	1.98E-04	9.90E-05
T_e	0.00E+00	0.00E+00	0.00E+00	8.00E-07	4.00E-07
T_f	6.62E-05	1.07E-04	1.07E-04	3.05E-05	3.05E-05
Total time [RPi]	**2.47E-04**	**2.76E-04**	**2.76E-04**	**2.59E-04**	**2.07E-04**

TABLE V
TIME TAKEN (IN SEC) FOR DIFFERENT OPERATIONS IN DIFFERENT PROTOCOLS (RPI).

Operation / Time (s) [NMCU]	[20]	[34]	[36]	[35]	Our
T_a	1.45E-05	1.09E-05	1.09E-05	0.00E+00	1.45E-05
T_b	4.09E-06	4.09E-06	4.09E-06	2.05E-05	8.18E-06
T_c	6.37E-04	6.37E-04	6.37E-04	0.00E+00	9.10E-05
T_d	0.00E+00	0.00E+00	0.00E+00	6.18E-04	3.09E-04
T_e	0.00E+00	0.00E+00	0.00E+00	8.00E-07	4.00E-07
T_f	5.85E-05	9.45E-05	9.45E-05	2.70E-05	2.70E-05
Total time [NMCU]	**7.14E-04**	**7.46E-04**	**7.46E-04**	**6.66E-04**	**4.50E-04**

TABLE VI
TIME TAKEN (IN SEC) FOR DIFFERENT OPERATIONS IN DIFFERENT PROTOCOLS (NODEMCU).

principal roles within the protocol: UAV and BS. Each role was meticulously modeled to emulate the specific behaviors and cryptographic capabilities necessary for secure communication.

The UAV role was equipped functions like a Pseudo Random Number Generator (PRNG), a Physical Unclonable Function (PUF), and an implementation of the Shamir Secret Sharing method. These functions were integral to the UAV's role in nonce generation, challenge handling, and execution of critical cryptographic operations. Similarly, the BS role was crafted to respond appropriately to the UAV's communications, generate its challenges and responses, and perform necessary security verifications. A detailed sequence of message exchanges was constructed to mirror the intricate logic of the authentication protocol. Each message was defined with essential components such as unique identifiers, nonces, and cryptographic elements like MAC.

Security Verification: The verification of the protocol's security was a pivotal aspect of our implementation. We delineated specific goals pertaining to the secrecy and authentication aspects of the protocol. These goals were critical in ensuring the confidentiality of sensitive data and the proper authentication of the communicating entities. Utilizing AVISPA's simulation environment, we subjected the protocol to a comprehensive array of potential attack scenarios. This process was instrumental in testing the protocol's resilience against common and sophisticated security threats. The results obtained from the AVISPA simulations were crucial in assessing the effectiveness of our protocol design.

"SAFE" outcome from the tool indicated that our protocol could successfully withstand the tested attack scenarios, effectively maintaining the integrity of both secrecy and authentication goals. This outcome served as a significant validation of the protocol's robustness in a controlled, simulated environment.

VI. RESULTS AND DISCUSSION

To evaluate the performance of the proposed protocol, this section presents simulation results of its operation. We employed two simulation settings to model the UAV operations in our system model. We use NodeMCU v3.0 and Raspberry Pi 3B for the evaluation of the computational performance of our protocol.

Table IV describes the time taken by different cryptographic operations performed on NodeMCU (NMCU) and Raspberry Pi 3B (RPi). The operations considered are bitwise XOR (T_a),

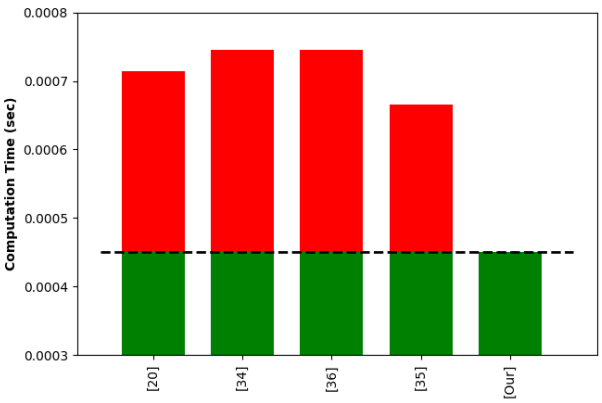

Fig. 5. Comparison of total computational time on NMCU (in sec).

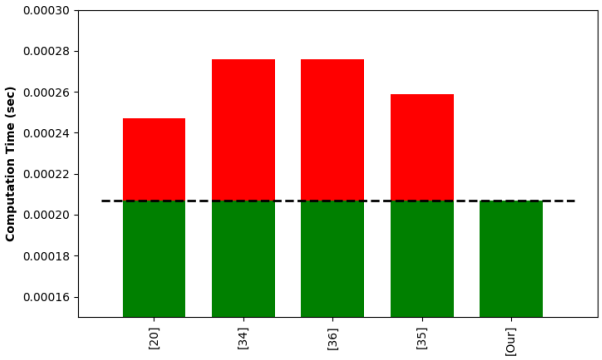

Fig. 6. Comparison of total computational time on RPi (in sec).

PRNG (T_b), hash (T_c), HMAC (T_d), PUF (T_e) and concatenation (T_f). For both scenarios, the time taken is evaluated in seconds. We can observe that operations such as hash and HMAC take more time on NMCU than RPi. On the other hand, in the NMCU environment, lighter operations such as XOR, PRNG, and concatenation exhibit faster execution times. To ensure hardware-based security, we employ an SRAM PUF that generates a 320-bit response with an operation time of 0.4 μs [37]. The comparison of total execution times for the proposed protocol and the protocols in [20, 34–36] in both the RPi and NMCU environments is presented in Table IV and V, respectively.

In Fig. 5, we provide a comparison of the total operation time on UAVs with previous state-of-the-art works such as [20, 34–36] on NMCU. While [20], [34], [36] and [35] have computation costs of $714\mu s$, $746\mu s$, $746\mu s$ and $666\mu s$ respectively, our protocol has a cost of only $450\mu s$. The red color bar shows the extra time that other protocols take in comparison to our approach. Similarly, in Fig. 6, we compare the total operation time on UAVs on RPi. While [20], [34], [36] and [35] have computation costs of $247\mu s$, $276\mu s$, $276\mu s$ and $259\mu s$ respectively, our protocol has a cost of only $207\mu s$. The primary reason for improved performance is a difference in the number of operations required in the proposed protocol and the state-of-the-art. Also, we can see that our protocol has much greater improvement than other protocols when run on low constraint devices such as NMCU instead of RPi. Since the protocol uses less computationally intensive operations, the proposed protocol is much faster. The use of PUF based challenge-response pairs is also responsible for the faster authentication protocol.

TABLE VII
COMPARISON OF COMMUNICATION AND STORAGE COSTS

Scheme	[20]	[34]	[36]	[35]	**Ours**
Communication cost (bits)	1696	1536	1952	1696	1600
Storage cost (bits)	480*	640*	320	640*	352

* *represents minimum storage*

In addition, we evaluate the storage cost of our proposed protocol by considering the number of bits needed to store

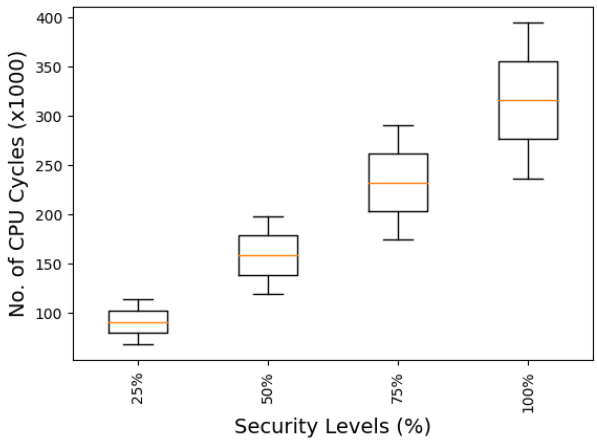

Fig. 7. Variation of number of CPU cycles vs security levels.

various data fields in the UAV's memory. As detailed in Table VII, the memory storage cost in the protocols [20], [34], [36], and [35] amounts to at least 480 bits, 640 bits, 320 bits, and 640 bits, respectively. In contrast, our scheme incurs a storage cost of 352 bits. This higher storage cost in our scheme compared to [36] is by design, as we utilize a larger number of challenge-response pairs, resulting in increased communication and storage expenses. This trade-off is made to enhance reliability against errors and mitigate environmental factors affecting the Physical Unclonable Function (PUF).

Despite the associated cost increases, our approach demonstrates superior performance when compared to all other schemes [20, 34–36]. Specifically, when implemented on Raspberry Pi (Rpi) and NodeMCU, our scheme surpasses the alternatives by 25% and 48%, respectively, as illustrated in Figures 5 and 6. This highlights the effectiveness of our approach in real-world scenarios.

Figure 7 gives an insight into the average number of execution cycles consumed by our protocol by varying the level of security. Level of security is defined as the percentage of CRPs that needs to be verified before deeming a device to be authentic. The network administrator decides the level of security. In Fig. 7, we provide a comparison of the time

taken for three scenarios by varying the number of threshold CRP pairs required to authenticate UAVs using the Shamir secret key. As mentioned in Section IV-B, the number of CRPs required to authenticate the device is denoted by t. The total number of CRPs used during the enrolment phase is k. So, based on our definition of the level of security, the level of security can be evaluated as $t*100/k\%$. For instance, consider a case where if 3 out of 5 CRP pairs are correct and generate correct secrets using Shamir's secret, the device is considered to be genuine and authentic. In this case, the level of security is evaluated as $3/5$ or 60%. In the case of 100% level of security, all the CRP must be correct and required to form a shared key. We fixed k as 8 and varied the value of t as 2, 4, 6 and 8. Thus, the level of security in each of the scenarios is given as 25%, 50%, 75%, and 100%, respectively. For 25% level of security, the number of cycles required for execution was $91,125$. It increased to $166,055$ for achieving a 50% level of security and further increased to $240,985$ for 75% level of security. The total number of cycles for 100% level of security took 315917 execution cycles. The number of cycles increases linearly, following the asymptotic order of $\mathcal{O}(k)$ where k is the number of CRPs used in the protocol. Thus, the proposed protocol is scalable, robust, and ensures different degrees of security.

Building on our previous analysis of execution cycles, we can deduce the energy consumption for the proposed protocol on the NodeMCU. Utilizing the average current draw during active transmission mode (170 mA) and a common voltage level of 3.3V for NodeMCU, the power consumption is approximately $P = 170 \times 10^{-3} \times 3.3 = 561 \times 10^{-3}$W or 561 mW.

For different security levels, the energy consumption is calculated based on the previously determined execution cycles. At a **25% Level of Security**, given 91,125 cycles, the energy expended is represented as $E_{25\%} = 561 \times 10^{-3} \times 91,125 \times T_{cycle}$. Moving to a **50% Level of Security**, with 166,055 cycles, the energy utilization can be expressed as $E_{50\%} = 561 \times 10^{-3} \times 166,055 \times T_{cycle}$. Similarly, for a **75% Level of Security**, based on 240,985 cycles, the energy consumption is $E_{75\%} = 561 \times 10^{-3} \times 240,985 \times T_{cycle}$. Lastly, at a **100% Level of Security**, encompassing 315,917 cycles, the energy computation equates to $E_{100\%} = 561 \times 10^{-3} \times 315,917 \times T_{cycle}$.

VII. Acknowledgement

This work is supported by National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research and Development Programme under project FCP-NUS-TG-2022-001.

VIII. Conclusion

This research paper introduces a secure approach for authenticating UAVs in a formal manner. The proposed methodology leverages Shamir's secret key sharing mechanism to ensure the protocol's functionality even in challenging environments, where the operation of a PUF may be susceptible to external factors. By utilizing PUFs, the protocol provides a guarantee of physical security and exhibits resilience against man-in-the-middle attacks, replay attacks, and denial-of-service attacks. The study demonstrates that the proposed protocol surpasses existing state-of-the-art protocols in terms of computational efficiency, while also being the sole solution that offers customizable security levels to meet the requirements of network administrators.

References

[1] G. Bansal, N. Naren, and V. Chamola, "Rama: Real-time automobile mutual authentication protocol using puf," in *Proceedings of IEEE International Conference on Information Networking (ICOIN), Barcelona, Spain*. IEEE, 2020.

[2] M. N. Aman, U. Javaid, and B. Sikdar, "A privacy-preserving and scalable authentication protocol for the internet of vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1123–1139, 2020.

[3] 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.

[4] 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.

[5] S. Benzarti, B. Triki, and O. Korbaa, "Privacy preservation and drone authentication using id-based signcryption." in *SoMeT*, 2018, pp. 226–239.

[6] D. Rudinskas, Z. Goraj, and J. Stankūnas, "Security Analysis of Uav Radio Communication System," *Aviation*, vol. 13, no. 4, pp. 116–121, 2009.

[7] G. Bansal and V. Chamola, "Lightweight authentication protocol for inter base station communication in heterogeneous networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 871–876.

[8] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[9] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[10] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[11] C. Zhong, J. Yao, and J. Xu, "Secure uav communication with cooperative jamming and trajectory control," *IEEE Communications Letters*, vol. 23, no. 2, pp. 286–289, 2018.

[12] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[13] A. Birk, B. Wiggerich, H. Bülow, M. Pfingsthorn, and S. Schwertfeger, "Safety, security, and rescue missions with an unmanned aerial vehicle (uav)," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 1, pp. 57–76, 2011.

[14] L. Vigano, "Automated security protocol analysis with the avispa tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 61–86, 2006.

[15] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019.

[16] 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*. IEEE, 2016, pp. 1213–1218.

[17] 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)*. IEEE, 2017, pp. 1–21.

[18] K. Yoon, D. Park, Y. Yim, K. Kim, S. K. Yang, and M. Robinson, "Security authentication system using encrypted channel on uav network," in *2017 First IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2017, pp. 393–398.

[19] S. He, Q. Wu, J. Liu, W. Hu, B. Qin, and Y.-N. Li, "Secure communications in unmanned aerial vehicle network," in *International Conference*

on Information Security Practice and Experience. Springer, 2017, pp. 601–620.
[20] 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.
[21] S. Jangirala, A. K. Das, N. Kumar, and J. Rodrigues, "Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Transactions on Vehicular Technology*, 2019.
[22] T. Alladi, N. Naren, G. Bansal, V. Chamola, and M. Guizani, "Secauthuav: A novel authentication scheme for uav-base station scenario," *IEEE Transactions on Vehicular Technology*, 2020.
[23] G. Bansal and B. Sikdar, "S-maps: Scalable mutual authentication protocol for dynamic uav swarms," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 088–12 100, 2021.
[24] ——, "Location aware clustering: Scalable authentication protocol for uav swarms," *IEEE Networking Letters*, vol. 3, no. 4, pp. 177–180, 2021.
[25] S. Katzenbeisser, Ü. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Pufs: Myth, fact or busted? a security evaluation of physically unclonable functions (pufs) cast in silicon," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2012, pp. 283–301.
[26] U. Rührmair, H. Busch, and S. Katzenbeisser, "Strong pufs: models, constructions, and security proofs," in *Towards hardware-intrinsic security*. Springer, 2010, pp. 79–96.
[27] U. Rührmair and D. E. Holcomb, "Pufs at a glance," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2014, pp. 1–6.
[28] U. Rührmair and M. van Dijk, "Pufs in security protocols: Attack models and security evaluations," in *2013 IEEE symposium on security and privacy*. IEEE, 2013, pp. 286–300.
[29] 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.
[30] B. Bera, A. K. Das, and A. K. Sutrala, "Private blockchain-based access control mechanism for unauthorized uav detection and mitigation in internet of drones environment," *Computer Communications*, vol. 166, pp. 91–109, 2021.
[31] I. Cervesato, "The dolev-yao intruder is the most powerful attacker," in *16th Annual Symposium on Logic in Computer Science—LICS*, vol. 1. Citeseer, 2001, pp. 1–2.
[32] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *[1993] Proceedings Computer Security Foundations Workshop VI*, 1993, pp. 147–158.
[33] 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.
[34] J. Srinivas, A. K. Das, N. Kumar, and J. J. Rodrigues, "Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6903–6916, 2019.
[35] 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 (LANMAN*. IEEE, 2020, pp. 1–6.
[36] 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.
[37] 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.

Gaurang Bansal is pursuing his Ph.D. in Electrical and Computer Engineering from National University of Singapore (NUS). His exceptional expertise in the domains of Privacy and Security earned him the prestigious Google PhD Fellowship award. Additionally, he has been honored with the NUS President Fellowship Award in recognition of his outstanding academic achievements. Previously, he had completed his Master's and Bachelor's from BITS Pilani in 2020 & 2018, respectively. His research interests span a wide spectrum, including cryptography, security, algorithms, blockchains, and the Internet of Things (IoT). He boasts an impressive portfolio of 36 extensive publications, featured in renowned conferences and journals such as IEEE Network Magazine, IEEE Transactions on Vehicular Technology, and IEEE INFOCOM, among others. His research has garnered significant recognition, evident through more than 600 citations on Google Scholar.

Biplab Sikdar (S'98–M'02–SM'09) received the B.Tech. degree in electronics and communication engineering from North-Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He was a faculty at the Rensselaer Polytechnic Institute, from 2001 to 2013, first as an Assistant Professor and then as an Associate Professor. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He serves as the Vice-dean of Graduate program and as the Area Director of Communications and Networks lab at NUS. His current research interests include wireless network and security for the Internet of Things and cyber-physical systems. Dr. Sikdar served as an Associate Editor for the IEEE Transactions on Communications from 2007 to 2012. He currently serves as an Associate Editor for the IEEE Transactions on Mobile Computing. He has served as a TPC in various conferences such as IEEE LANMAN, GLOBECOM, BROADNETS and ICC to name a few. He is a member of Eta Kappa Nu and Tau Beta Pi.