A Lightweight and Privacy-Preserving Mutual Authentication and Key Agreement Protocol for Internet of Drones Environment

Cong Pu, *Member, IEEE*, Andrew Wall, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, Imtiaz Ahmed, *Member, IEEE*, and Sunho Lim, *Senior Member, IEEE*

***Abstract*—With accelerated advances in various technologies, drones, better known as unmanned aerial vehicles (UAVs), are increasingly commonplace and consequently have a more pronounced impact on society. For example, Internet of Drones (IoD), a new communication paradigm offering fundamental navigation assistance and access to information, has widespread applications ranging from agricultural drones in farming to surveillance drones in the COVID-19 pandemic. The increasingly prominent role of IoD in our society also reinforces the importance of securing such systems against various data privacy and security threats. Operationally, it can be challenging to adopt conventional off-the-shelf security products in an IoD system due to the underpinning characteristics of drones (e.g., dynamic and open communication channel). Therefore in this article, we propose a lightweight and privacy-preserving mutual authentication and key agreement protocol, hereafter referred to as PMAP. The latter uses a physical unclonable function (PUF) and chaotic system to support mutual authentication and establish a secure session key between communication entities in the IoD system. To be specific, PMAP consists of two schemes, namely: 1) PMAP^{D2Z} (that mutually authenticates drone and zone service provider (ZSP) and establishes secure session keys) and 2) PMAP^{D2D} (that mutually authenticates drones and establishes secure session keys). In addition, PMAP supports conditional privacy preserving so that the genuine identity of drones can only be revealed by trusted ZSPs. We evaluate the security of PMAP using automated validation of Internet security protocols and application (AVISPA), as well as provide formal and informal security analysis to show the resilience of PMAP against various security attacks. We also evaluate the performance of PMAP through extensive experiments and compare its performance with existing AKA and IBE-Lite schemes, whose findings show that PMAP achieves better performance in terms of computation cost, energy consumption, and communication overhead.**

***Index Terms*—Authentication and key agreement, chaotic system, drone, Internet of Drones (IoD), physical unclonable function (PUF).**

Manuscript received November 1, 2021; revised January 25, 2022; accepted March 27, 2022. Date of publication March 30, 2022; date of current version June 7, 2022. This work was supported in part by the NASA West Virginia EPSCoR under Grant 80NSSC20M0055, and in part by the West Virginia HEPC's Division of Science and Research under Grant dsr.20.1698-001. *(Corresponding author: Cong Pu.)*

Cong Pu and Andrew Wall are with the Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV 25755 USA (e-mail: cong.pu@ieee.org; wall48@marshall.edu).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Imtiaz Ahmed is with the Department of Electrical Engineering and Computer Science, Howard University, Washington, DC 20059 USA (e-mail: imtiaz.ahmed@howard.edu).

Sunho Lim is with the Department of Computer Science, Texas Tech University, Lubbock, TX 79409 USA (e-mail: sunho.lim@ttu.edu).

Digital Object Identifier 10.1109/JIOT.2022.3163367

I. Introduction

DRONES, originally built for military purposes, are increasingly found in civilian and commercial applications, such as hurricanes and tornadoes monitoring and tracking, enforcing stay-at-home orders during the COVID-19 pandemic, etc [1]. According to the ABI Research whitepaper [2], the drone industry is estimated to be worth U.S. $ 92 billion by 2030. In other words, drones are no longer a fad/hype, and the increasingly popularity of drones is due to advances in other supporting technologies, such as 5G and artificial intelligence (AI), as well as the lowering of costs and the movement toward Industry 4.0 (also referred to as the fourth industrial revolution) and Internet of Things (IoT) (eco)system [3].

An IoT (eco)system that comprises predominately drones can also be referred to as an Internet of Drones (IoD) system, which is a distributed network of drones communicating with each other, and collecting and distributing data (e.g., intelligence) to the supporting infrastructure. Such data can then be mined and analyzed to facilitate decision making. Generally, in an IoD setting, the airspace is virtually divided into a set of zones that are shared by drones. Each zone is directly under the administration of one or more zone service providers (ZSPs), and the latter serves as access points to support navigation assistance and access to other resources (e.g., information and services). For example, in a commercial delivery/courier application, drones can contact the nearest ZSP to obtain an optimal trajectory [4]. During the COVID-19 pandemic or street demonstration, a set of surveillance drones can survey an area of interest, observe crowds, and deliver the data to ZSP for modeling and forecasting (e.g., spread of disease and crowd movement) [5].

Considering the sensitivity of the data collected by drones, as well as the potential of drones being abused as physical weapons (e.g., improvised explosive devices), there is a need to secure such devices and their communications [6]. A telling example is the revelation that a 13-year-old teenager allegedly hacked a drone in a stunt to highlight security flaws of Web-connected devices at the 2019 global cybersecurity

conference [7]. There has been a number of studies focusing on the security and privacy risks of drones. For example, Nassi *et al.* [8] identified six major components in a typical drone's ecosystem, prior to proposing an approach to audit potential attacks and countermeasures.

One viable security solution is to deploy mutual authentication schemes to verify the genuine identity of communication entities before sharing any sensitive information via an insecure wireless channel. In other words, communication entities (i.e., drones and ZSPs) in the IoD should have the capability to mutually authenticate each other and establish a secure session key for subsequent communications. However, we cannot assume that drones have the computational capabilities to deploy full-fledged security solutions, and we have to bear in mind that the deployed security solutions should not adversely impact the performance of the drones (e.g., significant battery consumption will result in reduced performance). This highlights the importance of designing lightweight security protocols to achieve basic security services, such as confidentiality, integrity, authentication, authorization, and nonrepudiation, in drone deployments. There are also applications that require privacy-preserving feature to support entity anonymity. In addition, an adversary may capture the drone and attempt to probe its integrated circuit to extract secret information. Therefore, we posit that tamper-resistance is an essential feature to minimize the risks of the compromise of cryptographic security parameters (e.g., regeneration of secret information such as session key).

Motivated by the above discussion, we propose a secure communication protocol for IoD, and analyze and measure its security resiliency and performance tradeoff through security analysis and experiments. Our major contribution is briefly summarized in the following.

1) We propose a lightweight and privacy-preserving mutual authentication and key agreement protocol (hereafter referred to as PMAP),[1] based on the physical unclonable function (PUF) and chaotic system to achieve mutual authentication and establish a secure session key between communication entities in the IoD. PMAP consists of two schemes, namely: 1) PMAP^{D2Z}, which mutually authenticates drone and ZSP and 2) PMAP^{D2D}, which mutually authenticates drone and drone.
2) PMAP is designed to support privacy-preserving. In our approach, a different pseudonym of drone is created for each communication session so that any entity except the trusted ZSP cannot reveal drone's real identity by sniffing on any communication or capturing any messages.

We also conduct extensive experiments and evaluate the performance of PMAP in terms of computation cost, energy consumption, and communication overhead. The experiment results show that PMAP can achieve lower computation cost, energy consumption, and communication overhead compared to prior security protocols while meeting all security requirements, indicating a viable and competitive approach for securing communications in the IoD.

We remark that this is an extension of an earlier work that appeared in the 2020 IEEE International Symposium on local and metropolitan area networks (LANMANs) [9]. Specifically, we made the following significant extensions (with over 70% new materials).

1) We investigate the most recent security protocols in the IoD environment, analyze their strengths and weaknesses, and rewrite the related work section.
2) We adopt the most studied example of dynamical systems, Henon map [10], to implement the operations of mutual authentication and session key establishment.
3) We propose a mutual authentication and key agreement scheme, called PMAP^{D2D}, to authenticate the communication and establish a secure session key between drone and drone. PMAP^{D2D} complements PMAP^{D2Z} [9], which mutually authenticates drone and ZSP.
4) We implement PMAP in high-level protocol specification language (HLPSL) [11], and then evaluate its security using automated validation of internet security protocols and applications (AVISPAs).
5) We provide a formal security analysis based on the formal security protocol analysis approach [12]. In addition, we present an informal security analysis to show PMAP is secure against various security attacks.
6) We develop a real-world testbed consisting of one HP ENVY laptop [13], one Latte Panda development board [14], and one power bank for experimental study.
7) We implement two new benchmarks, AKA [15] and IBE-Lite [16], in Java, and deploy them in the real-world testbed for performance evaluation and analysis.

The remainder of this article is organized as follows. Existing literature and recent studies are summarized in Sections II and III gives a brief introduction to PUF and chaotic system. Section IV describes the network and adversary models, followed by security requirements. The proposed security protocol is presented in Section V. Section VI presents the security verification, and formal and informal security analysis. Section VII focuses on experiment results and their analysis. In Section VIII, we further discuss the proposed security solution. Finally, concluding remarks are provided in Section IX.

II. Related Work

In [15], an authentication and key agreement protocol is proposed to achieve the goal of a secure communication in the IoD. The traditional one-way hash function and bitwise XOR are adopted and combined in different ways to authenticate communication entities and establish a session key. To be specific, the proposed protocol consists of three steps, such as setup, registration, as well as mutual authentication. Master private key and other public information are generated by a control server during setup. In the phase of registration, registered user, and drone are assigned with secret key via a secure channel. Finally, user and drone establish a session key and communicate. Srinivas *et al.* [17] proposed a security protocol

[1]The source code of PMAP and the security verification programs are publicly available at the https://github.com/congpu/PMAP.

(TCALAS) to protect drones in the IoD system. TCALAS prevents users from receiving services from remote drones before they are registered at the ground station server (GSS). In addition, all legitimate drones are required to sign in at the GSS. After registration, a secret credential that is exclusively known to drone and GSS is provided to drone so that it can communicate with registered users. TCALAS also allows registered users to update their passwords and/or biometrics without the involvement of GSS.

In [18], a privacy-preserving authentication framework is proposed for the IoD environment. In order to meet the resource-constrained requirement of drones, a lightweight online/offline signature technique is adopted in the framework. Additionally, to reduce the authentication cost, a predictive authentication technique integrated with edge computing is investigated. Lastly, the authors design a buffer pseudonym and public key update strategy to protect drones' identity, location, and flying routes. Alladi *et al.* [19] proposed to mutually authenticate drones and users over an insecure channel using a hash function and bitwise XOR operations. During the setup phase, a control server generates a master private key and other public system parameters. In the following registration phase, users and drones register at the control server and obtain their secret keys. Finally, users and drones authenticate each other and establish a session key.

Feng *et al.* [21] criticized the centralized security system for being vulnerable to a single point of failure as well as inadequate for cross-domain identity verification. Motivated by the above argument, they design a cross-domain authentication scheme using the blockchain technique for the 5G-based IoD, where the threshold-multi-party signatures scheme helps to establish a federated identity across domains. Moreover, the smart contract is used to achieve the goal of authentication among drones from different domains. Tanveer *et al.* [20] adopted the dedicated authenticated encryption algorithm, elliptic curve cryptography, and hash function to implement an authentication scheme for the IoD system. Specifically, during seven steps, the identity of user is first verified, and then a secret key is established between the user and the drone for the follow-up communications. The authors claim that the proposed security scheme not only satisfies the predefined security requirements, but also achieves better performance. Nonetheless, the proposed security scheme does not guarantee dynamic privacy preservation.

In the preliminary version of this work [9], we propose a mutual authentication protocol to protect the communications between drone and ground station. In [9], a chaotic system is implemented as a random shuffling operation to authenticate the identity of drone and ground station. In order to prevent an adversary from regenerating the same shuffling result, the challenge-response pair (CRP) of PUF is adopted as the initial condition of the chaotic system. Similar to our previous work [9] on mutual authentication and session key establishment, both rely on the chaotic system and PUF. However, there are several significant differences between these two works. First, PMAP adopts a more efficient chaotic system, i.e., Henon map, to implement the random shuffling operation. Second, PMAP proposes a mutual authentication and key agreement scheme to authenticate the communication between drone and drone, which complements our previous work [9]. Third, in order to prove that PMAP is a secure authentication and key agreement protocol, we verify the security of PMAP using AVISPA and provide formal and informal security analysis. Finally, we develop a real-world testbed, implement two new benchmark schemes, and conduct new experiments for performance evaluation and analysis.

Compared to the abovementioned schemes, our approach PMAP is novel in terms of four aspects. First, PMAP adopts lightweight operations, such as PUF, chaotic system, random shuffling, bitwise XOR, and hash function to realize mutual authentication and session key establishment between two communication entities in the IoD. Thus, the deployed security scheme PMAP will not negatively impact the performance of drones. This is because drones are usually resource-constrained and significant energy consumption of security solutions can harm the lifetime of drones. Second, in the IoD environment, the adversary might physically capture a drone and extract credentials stored in the memory through memory disclosure attacks. However, many existing schemes failed to provide physical attack protection. To defend against both software-based and physical memory disclosure attacks, PMAP adopts PUF as a tamper-resistant module to safeguard information stored in the electronic circuitry of drones. Third, PMAP supports conditional privacy preserving so that the genuine identity of drones can only be revealed by trusted ZSPs. Most importantly, PMAP will guarantee that a different pseudonym of drone is created for each communication session. Fourth, many recently developed security protocols for IoD environment only consider a few security primitives; most importantly, they have some inherent vulnerabilities. However, the security of PMAP has been carefully evaluated through formal and informal security analysis, as well as security verification, which prove that PMAP is a secure protocol for IoD environments. In summary, over the last several years, several security protocols and techniques have been proposed for IoD and similar environments. However, little attention has been paid to a privacy-preserving mutual authentication and key agreement protocol using PUF and chaotic system. Finally, we compare PMAP with existing schemes in Table I.

III. Preliminary Background

A. Physical Unclonable Function

Since the slight physical difference is introduced on each integrated circuit during the process of manufacturing, a PUF is widely believed to be an electronic identity, analogous to biometrics, such as hand geometry, palm print, iris, etc. [9]. In other words, a PUF is regarded as an integrated circuit taking an input and producing an output according to its unique physical characteristics. When an input query, called *challenge*, is fed into a PUF, a challenge-specific output, called *response*, is produced. The challenge together with the response is called a CRP. Formally said, a PUF is represented as a function P in the following, $R = P(C)$, where C and R are the input challenge and the output response of PUF, respectively. For a PUF, the same response will result in the same challenge. However,

TABLE I
COMPARISON OF EXISTING SECURITY SOLUTIONS

Scheme	Adopted Techniques	Security Evaluation / Features / Weaknesses
[15]	hash and XOR	formal security analysis; does not support drone-to-drone authentication
[16]	ECC, hash, and XOR	no security analysis and verification; vulnerable to security attacks and does not provide anonymity
[17]	hash, XOR, and fuzzy extractor	formal security analysis and verification; vulnerable to traceability and stolen verifier attacks
[18]	AES, hash, and XOR	informal security analysis; does not provide anonymity
[19]	hash and XOR	formal security analysis; does not provide drone anonymity
[20]	ECC, hash, and XOR	formal and informal security analysis; does not provide drone anonymity
PMAP	PUF, chaotic system, hash, and XOR	thorough security evaluation; secure against various well-known attacks and low computation cost

when the same challenge is provided to different PUFs, totally different responses will be generated. In summary, a PUF exudes two attractive features. First, secret information can be reproduced using publicly available information. Second, inherent tamper-resistant capability can defend against various physical attacks.

In a noisy environment, the response generated by the PUF might have a slight variation despite the same challenge is provided. Said diplomatically, the PUF is not noise-resistant by default, which might lead to the unavailability of secret information (i.e., a cryptographic value) for critical operations. In recent years, various designs of noise-resistant and reliable PUF [22] have been investigated, where almost 0% bit error rates in a noisy environment with voltage fluctuations and wide temperature ranges can be achieved. Thus, in this article, we assume that an ideal and noise-resistant PUF is deployed in drones [23].

B. Chaotic System

The chaotic system is a deterministic system that exhibits nonlinear behavior and pseudo-randomness that are highly sensitive to initial conditions [24]. During the past several years, many pseudo-random number functions in cryptography are designed based on the chaotic system, where the control parameter and the initial condition of the chaotic system are adopted as their seeds. As a result, the correct/original pseudo-random number sequence cannot be restored in the absence of the right initial condition. The Henon map [10] is one of the chaotic systems that exhibits chaotic behavior. It is widely known as a 2-D discrete-time and dynamical system that displays chaos for certain parameter values and initial conditions. Specifically, the Henon map accepts a 2-D point on the plane, denoted as (x_n, y_n), and maps it to a new point, denoted as (x_{n+1}, y_{n+1}), according to

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n. \end{cases} \tag{1}$$

Here, both a and b are system parameters. The sequence of points generated by the Henon map is totally determined by the initial point, denoted as (x_0, y_0). In order to exhibit chaotic behavior, the following set of system parameters is always adopted, $a = 1.4$ and $b = 0.3$. The rationale for setting $a = 1.4$ and $b = 0.3$ is that other values might cause the Henon map to be chaotic, intermittent, or converge to a periodic orbit. Ignoring the initial point (x_0, y_0), the sequence of points is distributedly located around the Henon map attractor in a random way. It is shown in Fig. 1 that any change in the initial point (x_0, y_0) will make the Henon map output a different sequence of points. For example, the sequence of points located inside the dashed rectangles in Fig. 1(a) and (b) are totally different.

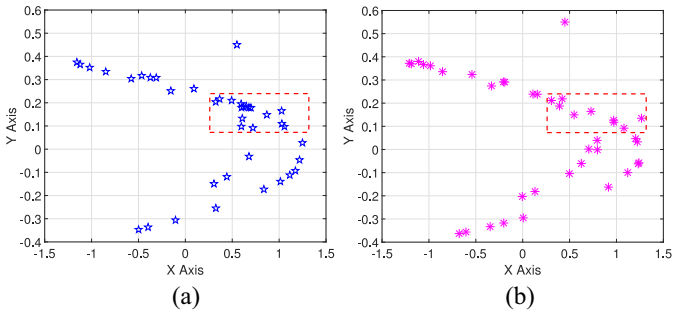


Fig. 1. Henon map with different initial points (x_0, y_0) after 40 iterations, where we highlight a different sequence of points inside the dashed rectangle. (a) $x_0 = 0.55$ and $y_0 = 0.45$. (b) $x_0 = 0.45$ and $y_0 = 0.55$.

C. Random Shuffling

In PMAP, the to-be-communicated message is represented as an array, and the PUF and Henon map are integrated together to implement the operation of random shuffling. The basic idea is that the Henon map first takes the CRP pair of PUF as the initial condition to output a unique sequence of points. Each point in the sequence is mapped to the corresponding element in the array (e.g., the first point is mapped to the first array element and so on.). And then, the first point in the sequence is converted into a unique integer which indicates the new location of the first array element in the output array. The same idea will be applied to the rest of array elements. When the last array element is rearranged to a new location, the output array is considered as the randomly shuffled message.

IV. SYSTEM MODEL

A. Network Model

The network model of PMAP is presented in Fig. 2, where there are two communication entities: 1) drones and 2) ZSP. Without loss of generality, the following two scenarios are considered for privacy-preserving mutual authentication and key agreement: 1) drone ID_1 wants to establish a communication with ZSP Z_s through PMAP^{D2Z} and 2) drone ID_2 wants to establish a communication with drone ID_n through PMAP^{D2D}. For example, ZSP Z_s can provide navigation information and commands to coordinate drone ID_1 in the predetermined area,

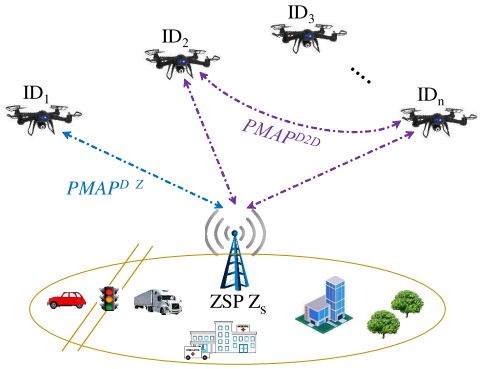


Fig. 2. Network model, where PMAP^{D2Z} authenticates drone ID_1 and ZSP Z_s and PMAP^{D2D} authenticates drone ID_2 and ID_n.

or drone ID_2 wants to exchange the collected data of interest with drone ID_n for further processing and analyzing the behavior of interest. We assume that each drone is equipped with an integrated circuit consisting of a PUF. However, the design of PUF is out of the scope of this article. According to [25], we implement the PUF as a 256-bit hash function [26]. ZSP Z_s is considered as a trusted entity and has no limitation of resources. However, drones are not trusted and have limited resources. The communication between entities occurs over an insecure wireless channel and therefore it should be secured.

B. Adversary Model

According to the widely adopted adversary model in [27], any two entities who are communicating over an insecure wireless channel are assumed to be untrustworthy. Thus, an adversary can overhear, duplicate, corrupt, alter, replay, or delete the transmitted messages. A drone may accidentally move to an unattended hostile area with the collected sensitive information, and hence there are possibilities that the drone could be easily captured by an adversary. However, any adversary that attempts to probe or alter the integrated circuit of captured drone will irreversibly modify the slight physical variations in the integrated circuit, which in turn changes the challenge-response mapping of PUF, or even destroys the PUF. The goal of the adversary is to establish an authentication with ZSP or any uncompromised drone without being detected, and then cause serious damages to individuals or organizations. For example, if the drone is communicating with ZSP for navigation information and the adversary plans to authenticate itself to the drone as a ''legitimate'' ZSP, this scenario can pose a threat to the government, national institutions, and assets, such as nuclear power plants and historical sites (e.g., using drone as an improvised explosive device).

C. Security Requirements

In the light of well-established essential security objectives for network and computer services [27], we design PMAP to meet the following security requirements.

1) *Authentication:* PMAP shall assure that two communicating entities are authentic, that is, each is the entity that it claims to be. In addition, PMAP shall assure that a third party cannot masquerade as one of the two legitimate entities for unauthorized transmission or reception.
2) *Integrity:* PMAP shall assure that messages are verified for their source of origin and are received as sent with no duplication, modification, reordering, or replays.
3) *Confidentiality:* PMAP shall assure that messages are confidentially shared between communicating entities, safe from adversary after a session key is established.
4) *Anonymity:* PMAP shall assure that a different pseudonym of drone is created for each communication session. In addition, PMAP shall assure that any entity except trusted ZSPs cannot reveal other entities' real identities by sniffing on any communication.
5) *Session Key Agreement:* PMAP shall assure that a secure session key will be established between communicating entities for subsequent communication after authentication and other unauthorized entities cannot retrieve any useful information from the obtained session key.
6) *Immune Against Various Attacks:* PMAP shall be resilient and immune against various attacks, such as drone impersonation attack, ZSP spoofing attack, message modification attack, drone capture attack, replay attack, known session key attack, and man-in-the-middle attack.

V. Proposed Security Protocol

During the system deployment phase, a drone chooses its real identity ID_i, obtains its initial CRP (C_i^t, R_i^t), and computes its initial pseudonym $PID_i^t = H(ID_i \| R_i^t)$. Then, the real identity ID_i, the initial CRP (C_i^t, R_i^t), and the initial pseudonym PID_i^t of drone ID_i would be securely shared with ZSPs using the time-based OTP algorithm (TOTP) [28]. When the system deployment phase is complete, ZSPs store each drone's real identity, initial CRP, and initial pseudonym, while drones only store their real identities and challenges of initial CRP. Note that the challenge of CRP is embedded in drone's PUF, and any attempt that probes or alters the integrated circuit will change the challenge-response mapping of PUF, or even destroy the PUF. Table II lists all notations used in this article.

A. Mutual Authentication and Key Agreement Between Drone and ZSP

PMAP^{D2Z} for the scenario when drone ID_i wants to establish a communication with ZSP Z_s is shown in Fig. 3. The basic idea is that drone ID_i and ZSP Z_s first exchange an encrypted message which is generated through random shuffling with drone ID_i's CRP pair to verify each other's identity. If the verification succeeds, drone ID_i and ZSP Z_s proceed to exchange two encrypted messages in order to share two unique random numbers. Finally, drone ID_i and ZSP Z_s use these two unique random numbers to calculate drone ID_i's new pseudonym and CRP pair, and establish a secure session key. The detailed steps are as follows.

1) Drone ID_i first computes its pseudonym $PID_i^t = H(ID_i \| R_i^t)$ using its real identify ID_i and PUF response R_i^t. Then it generates a random number N_i^t and calculates

TABLE II
NOTATIONS

Notation	Meaning
Z_s	Identity of the sth ZSP
ID_i	Real identity of the ith drone
t	Timestamp
PID_i^t	Pseudonym of the ith drone
N_i^t	Random number generated by ID_i
N_s^{t+1}	Random number generated by Z_s for $PMAP^{D2Z}$
$N_{i,s}^{t+1}$, $N_{j,s}^{t+1}$	Random numbers generated by Z_s for $PMAP^{D2D}$
(C_i^t, R_i^t)	PUF CRP of ID_i
$S(\cdot)_{(C_i^t, R_i^t)}$	Random shuffling with (C_i^t, R_i^t)
$S^{-1}(\cdot)_{(C_i^t, R_i^t)}$	Reverse process of random shuffling with (C_i^t, R_i^t)
$C(\cdot)$	Message Authentication Code (MAC) function
$H(\cdot)$	Secure one-way hash function
$\oplus$	Bitwise XOR operation
$\|$	Concatenation operation
M_i	The ith message
MAC_i	MAC of the ith message
$SK_{i,s}$	Session key between drone ID_i and ZSP Z_s
$SK_{i,j}$	Session key between drone ID_i and drone ID_j

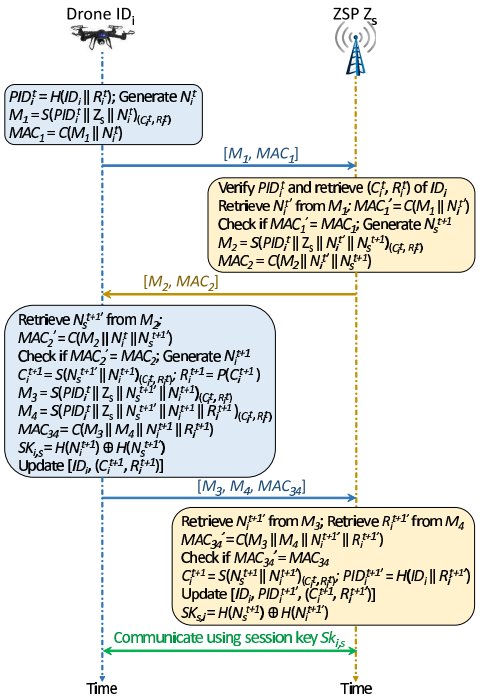


Fig. 3. Mutual authentication and secret session key establishment between drone ID$_i$ and ZSP Z_s, where vertical dash-dotted line with arrow indicates the time and horizontal solid line with arrow represents the communication between two entities.

an encrypted message M_1, where the plaintext, ($\text{PID}_i^t \| Z_s \| N_i^t$), will be arranged in a random shuffle using the Henon map with the CRP (C_i^t, R_i^t) as the initial condition

$$M_1 = S\left(\text{PID}_i^t \| Z_s \| N_i^t\right)_{(C_i^t, R_i^t)}.$$

It also calculates MAC MAC_1 as follows:

$$\text{MAC}_1 = C\left(M_1 \| N_i^t\right).$$

Finally, it sends authentication request message [M_1, MAC_1] to ZSP Z_s.

2) ZSP Z_s first tries to locate PID_i^t in the database. If PID_i^t is not found, the authentication request is rejected. Otherwise, it fetches the entry [ID$_i$, PID_i^t, (C_i^t, R_i^t)] for drone ID$_i$. Then, it retrieves $N_i^{t'}$ from M_1 through $S^{-1}(M_1)$, which is the reverse process of random shuffling with the CRP (C_i^t, R_i^t) as the initial condition. With $N_i^{t'}$, it can calculate $\text{MAC}_1^{'} = C(M_1 \| N_i^{t'})$ and check it with the received MAC_1. If $\text{MAC}_1^{'} = \text{MAC}_1$, the message verification succeeds. Otherwise, it discards the message. Next, it generates a random number N_s^{t+1}, and calculates an encrypted message M_2 and MAC MAC_2 as follows:

$$M_2 = S\left(\text{PID}_i^t \| Z_s \| N_i^{t'} \| N_s^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$\text{MAC}_2 = C\left(M_2 \| N_i^{t'} \| N_s^{t+1}\right).$$

Finally, it sends message [M_2, MAC_2] to drone ID$_i$.

3) Drone ID$_i$ first retrieves $N_s^{t+1'}$ from M_2 through $S^{-1}(M_2)$ and calculates $\text{MAC}_2^{'}$ as follows:

$$\text{MAC}_2^{'} = C\left(M_2 \| N_i^t \| N_s^{t+1'}\right).$$

If $\text{MAC}_2^{'} = \text{MAC}_2$, the message verification succeeds. Otherwise, it discards the message. Then, it generates a random number N_i^{t+1} and computes its new CRP as follows:

$$C_i^{i+1} = S\left(N_s^{t+1'} \| N_i^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$R_i^{t+1} = P\left(C_i^{i+1}\right).$$

After that, it calculates the following:

$$M_3 = S\left(\text{PID}_i^t \| Z_s \| N_s^{t+1'} \| N_i^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$M_4 = S\left(\text{PID}_i^t \| Z_s \| N_s^{t+1'} \| N_i^{t+1} \| R_i^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$\text{MAC}_{34} = C\left(M_3 \| M_4 \| N_i^{t+1} \| R_i^{t+1}\right).$$

Finally, it sends message [M_3, M_4, MAC_{34}] to ZSP Z_s, updates its CRP (C_i^{t+1}, R_i^{t+1}), and calculates the secret session key as follows:

$$SK_{i,s} = H\left(N_i^{t+1}\right) \oplus H\left(N_s^{t+1'}\right).$$

4) ZSP Z_s first retrieves $N_i^{t+1'}$ and $R_i^{t+1'}$ from M_3 and M_4 through $S^{-1}(M_3)$ and $S^{-1}(M_4)$, respectively. Then, it calculates $\text{MAC}_{34}^{'}$ as follows:

$$\text{MAC}_{34}^{'} = C\left(M_3 \| M_4 \| N_i^{t+1'} \| R_i^{t+1'}\right).$$

If $\text{MAC}_{34}^{'} = \text{MAC}_{34}$, the message verification succeeds. Otherwise, it discards the message. After that, it computes drone ID$_i$'s new challenge C_i^{i+1} and new pseudonym $\text{PID}_i^{t+1'}$, and then updates the entry [ID$_i$, $\text{PID}_i^{t+1'}$, $(C_i^{t+1}, R_i^{t+1'})$] in the database

$$C_i^{i+1} = S\left(N_s^{t+1} \| N_i^{t+1'}\right)_{(C_i^t, R_i^t)}$$
$$\text{PID}_i^{t+1'} = H\left(\text{ID}_i \| R_i^{t+1'}\right).$$

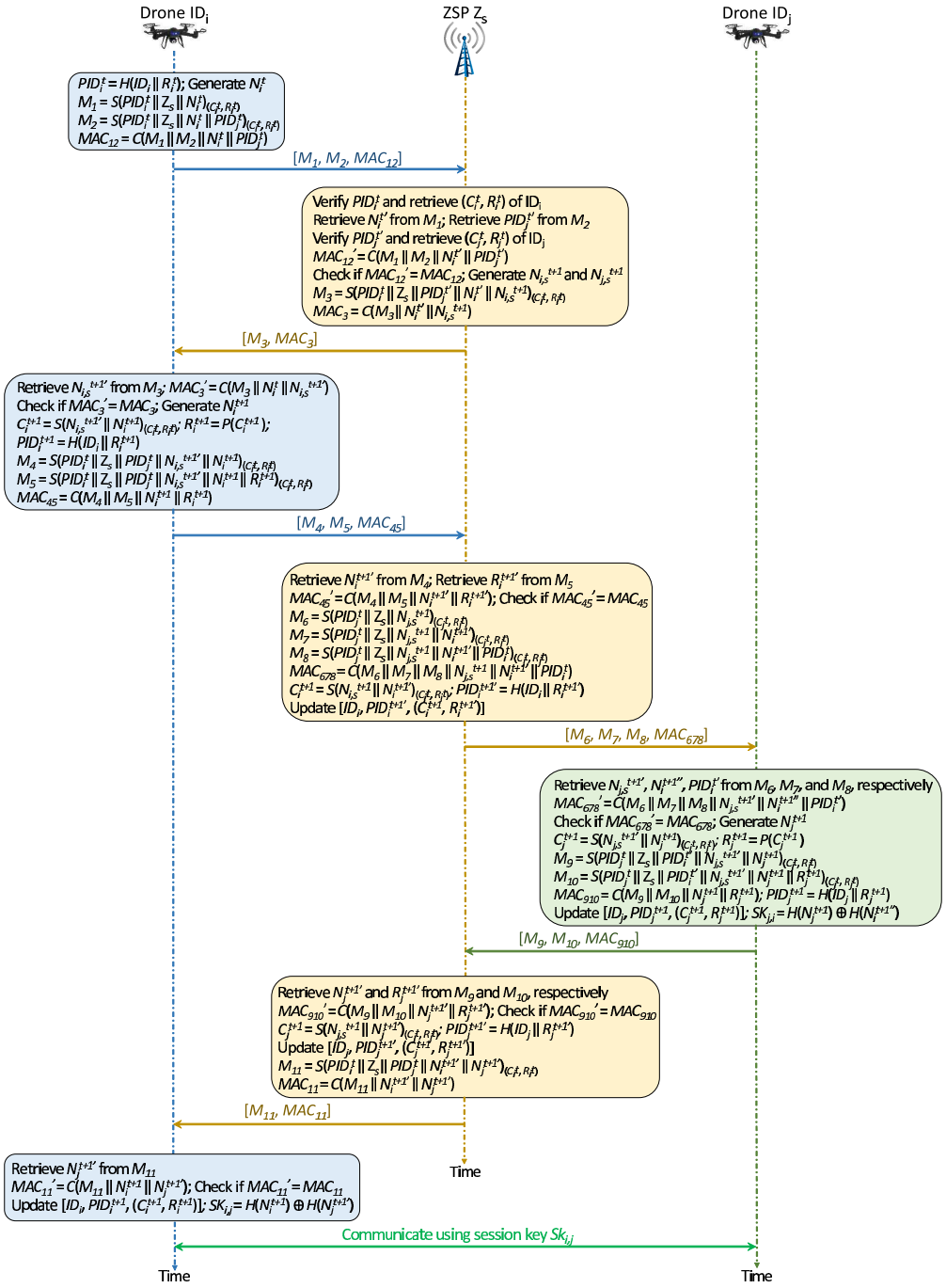


Fig. 4. Mutual authentication and secret session key establishment between drone ID$_i$ and drone ID$_j$, where vertical dash-dotted line with arrow indicates the time and horizontal solid line with arrow represents the communication between two entities.

Finally, it calculates the secret session key as follows:

$$SK_{s,i} = H\left(N_s^{t+1}\right) \oplus H\left(N_i^{t+1'}\right).$$

By this time, the mutual authentication between drone ID$_i$ and ZSP Z_s is finally succeeded and the secret session key $SK_{i,s}$ has been securely established for the subsequent communications.

B. Mutual Authentication and Key Agreement Between Drone and Drone

PMAP^{D2D} for the scenario when drone ID$_i$ wants to establish a communication with drone ID$_j$ is shown in Fig. 4. The basic idea is that drone ID$_i$ first contacts ZSP Z_s and requests to communicate with drone ID$_j$. After verifying the identity of drone ID$_i$ and drone ID$_j$, ZSP Z_s, as a trusted third party, helps to exchange secret information (i.e., unique random numbers) between drone ID$_i$ and drone ID$_j$ through encrypted messages. Finally, drone ID$_i$ and drone ID$_j$ calculate their new pseudonym and CRP pair, and establish a secure session key for communication. The detailed steps are as follows.

1) Drone ID$_i$ first computes its pseudonym $\text{PID}_i^t = H(\text{ID}_i \| R_i^t)$ using its real identify ID$_i$ and PUF response R_i^t, and generates a random number N_i^t. Then, it calculates

the following:

$$M_1 = S\left(\mathrm{PID}_i^t \| Z_s \| N_i^t\right)_{(C_i^t, R_i^t)}$$
$$M_2 = S\left(\mathrm{PID}_i^t \| Z_s \| N_i^t \| \mathrm{PID}_j^t\right)_{(C_i^t, R_i^t)}$$
$$\mathrm{MAC}_{12} = C\left(M_1 \| M_2 \| N_i^t \| \mathrm{PID}_j^t\right).$$

Finally, it sends authentication request message [M_1, M_2, MAC_{12}] to ZSP Z_s.

2) ZSP Z_s first tries to locate PID_i^t in the database. If the search fails, the authentication request is rejected. Otherwise, it fetches [ID_i, PID_i^t, (C_i^t, R_i^t)] for drone ID_i. Then, it retrieves $N_i^{t'}$ and $\mathrm{PID}_j^{t'}$ from M_1 and M_2 through $S^{-1}(M_1)$ and $S^{-1}(M_2)$, respectively. It also tries to locate $\mathrm{PID}_j^{t'}$ in the database and fetches the corresponding entry. With $N_i^{t'}$ and $\mathrm{PID}_j^{t'}$, it can calculate $MAC_{12}^{'}$ as follows:

$$\mathrm{MAC}_{12}^{'} = C\left(M_1 \| M_2 \| N_i^{t'} \| \mathrm{PID}_j^{t'}\right)$$

and verifies the message. If $\mathrm{MAC}_{12}^{'} = \mathrm{MAC}_{12}$, the message verification succeeds. Otherwise, it discards the message. Next, it generates two random numbers, $N_{i,s}^{t+1}$ and $N_{j,s}^{t+1}$, and calculates the following:

$$M_3 = S\left(\mathrm{PID}_i^t \| Z_s \| \mathrm{PID}_j^{t'} \| N_i^{t'} \| N_{i,s}^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$\mathrm{MAC}_3 = C\left(M_3 \| N_i^{t'} \| N_{i,s}^{t+1}\right).$$

Finally, it sends message [M_3, MAC_3] to drone ID_i.

3) Drone ID_i first retrieves $N_{i,s}^{t+1'}$ from M_3 through $S^{-1}(M_3)$ and calculates $\mathrm{MAC}_3^{'}$ as follows:

$$\mathrm{MAC}_3^{'} = C\left(M_3 \| N_i^t \| N_{i,s}^{t+1'}\right).$$

If $\mathrm{MAC}_3^{'} = \mathrm{MAC}_3$, the message verification succeeds. Otherwise, it discards the message. Then, it generates a random number N_i^{t+1} and computes its new CRP (C_i^{i+1}, R_i^{t+1}) and pseudonym PID_i^{t+1} as follows:

$$C_i^{i+1} = S\left(N_{i,s}^{t+1'} \| N_i^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$R_i^{t+1} = P\left(C_i^{i+1}\right)$$
$$\mathrm{PID}_i^{t+1} = H\left(\mathrm{ID}_i \| R_i^{t+1}\right).$$

After that, it calculates encrypted messages M_4 and M_5, and MAC MAC_{45}, which are shown as follows:

$$M_4 = S\left(\mathrm{PID}_i^t \| Z_s \| \mathrm{PID}_j^t \| N_{i,s}^{t+1'} \| N_i^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$M_5 = S\left(\mathrm{PID}_i^t \| Z_s \| \mathrm{PID}_j^t \| N_{i,s}^{t+1'} \| N_i^{t+1} \| R_i^{t+1}\right)_{(C_i^t, R_i^t)}$$
$$\mathrm{MAC}_{45} = C\left(M_4 \| M_5 \| N_i^{t+1} \| R_i^{t+1}\right).$$

Finally, it sends message [M_4, M_5, MAC_{45}] to ZSP Z_s.

4) ZSP Z_s first retrieves $N_i^{t+1'}$ and $R_i^{t+1'}$ from M_4 and M_5 through $S^{-1}(M_4)$ and $S^{-1}(M_5)$, respectively. Then, it calculates $\mathrm{MAC}_{45}^{'}$ as follows:

$$\mathrm{MAC}_{45}^{'} = C\left(M_4 \| M_5 \| N_i^{t+1'} \| R_i^{t+1'}\right).$$

If $\mathrm{MAC}_{45}^{'} = \mathrm{MAC}_{45}$, the message verification succeeds. Otherwise, it discards the message. After that, it calculates the following:

$$M_6 = S\left(\mathrm{PID}_j^t \| Z_s \| N_{j,s}^{t+1}\right)_{\left(C_j^t, R_j^t\right)}$$
$$M_7 = S\left(\mathrm{PID}_j^t \| Z_s \| N_{j,s}^{t+1} \| N_i^{t+1'}\right)_{\left(C_j^t, R_j^t\right)}$$
$$M_8 = S\left(\mathrm{PID}_j^t \| Z_s \| N_{j,s}^{t+1} \| N_i^{t+1'} \| \mathrm{PID}_i^t\right)_{\left(C_j^t, R_j^t\right)}$$
$$\mathrm{MAC}_{678} = C\left(M_6 \| M_7 \| M_8 \| N_{j,s}^{t+1} \| N_i^{t+1'} \| \mathrm{PID}_i^t\right).$$

Finally, it computes drone ID_i's new challenge C_i^{i+1} and pseudonym $\mathrm{PID}_i^{t+1'}$ as shown below, updates the corresponding entry, and sends message [M_6, M_7, M_8, MAC_{678}] to drone ID_j

$$C_i^{i+1} = S\left(N_{i,s}^{t+1} \| N_i^{t+1'}\right)_{(C_i^t, R_i^t)}$$
$$\mathrm{PID}_i^{t+1'} = H\left(\mathrm{ID}_i \| R_i^{t+1'}\right).$$

5) Drone ID_j first retrieves $N_{j,s}^{t+1'}$, $N_i^{t+1''}$, and $\mathrm{PID}_i^{t'}$ from M_6, M_7, and M_8 through $S^{-1}(M_6)$, $S^{-1}(M_7)$, and $S^{-1}(M_8)$, respectively. Then, it calculates $\mathrm{MAC}_{678}^{'}$ as follows:

$$\mathrm{MAC}_{678}^{'} = C\left(M_6 \| M_7 \| M_8 \| N_{j,s}^{t+1'} \| N_i^{t+1''} \| \mathrm{PID}_i^{t'}\right).$$

If $\mathrm{MAC}_{678}^{'} = \mathrm{MAC}_{678}$, the message verification succeeds. Otherwise, it discards the message. Next, it generates a random number N_j^{t+1} and computes its new CRP (C_j^{i+1}, R_j^{t+1}) in the following:

$$C_j^{i+1} = S\left(N_{j,s}^{t+1'} \| N_j^{t+1}\right)_{\left(C_j^t, R_j^t\right)}$$
$$R_j^{t+1} = P\left(C_j^{i+1}\right).$$

After that, it calculates the following:

$$M_9 = S\left(\mathrm{PID}_j^t \| Z_s \| \mathrm{PID}_i^{t'} \| N_{j,s}^{t+1'} \| N_j^{t+1}\right)_{\left(C_j^t, R_j^t\right)}$$
$$M_{10} = S\left(\mathrm{PID}_j^t \| Z_s \| \mathrm{PID}_i^{t'} \| N_{j,s}^{t+1'} \| N_j^{t+1} \| R_j^{t+1}\right)_{\left(C_j^t, R_j^t\right)}$$
$$\mathrm{MAC}_{910} = C\left(M_9 \| M_{10} \| N_j^{t+1} \| R_j^{t+1}\right).$$

Finally, it sends message [M_9, M_{10}, MAC_{910}] to ZSP Z_s, updates its CRP (C_j^{t+1}, R_j^{t+1}), and calculates the secret session key as follows:

$$SK_{j,i} = H\left(N_j^{t+1}\right) \oplus H\left(N_i^{t+1''}\right).$$

6) ZSP Z_s first retrieves $N_j^{t+1'}$ and $R_j^{t+1'}$ from M_9 and M_{10} through $S^{-1}(M_9)$ and $S^{-1}(M_{10})$, respectively. Then, it calculates $\mathrm{MAC}_{910}^{'}$ as follows:

$$\mathrm{MAC}_{910}^{'} = C\left(M_9 \| M_{10} \| N_j^{t+1'} \| R_j^{t+1'}\right).$$

If $\mathrm{MAC}_{910}^{'} = \mathrm{MAC}_{910}$, the message verification succeeds. Otherwise, it discards the message. Next, it computes drone ID_j's new challenge C_j^{i+1} and pseudonym $\mathrm{PID}_j^{t+1'}$ as shown below, and updates the corresponding entry

$$C_j^{i+1} = S\left(N_{j,s}^{t+1} \| N_j^{t+1'}\right)_{\left(C_j^t, R_j^t\right)}.$$

$$\mathrm{PID}_j^{t+1'} = H\left(\mathrm{ID}_j \| R_j^{t+1'}\right).$$

Finally, it calculates encrypted message M_{11} and MAC MAC_{11} in the following, and sends message $[M_{11}, \mathrm{MAC}_{11}]$ to drone ID_i

$$M_{11} = S\left(\mathrm{PID}_i^t \| Z_s \| \mathrm{PID}_j^t \| N_i^{t+1'} \| N_j^{t+1'}\right)_{\left(C_i^t, R_i^t\right)}.$$

$$\mathrm{MAC}_{11} = C\left(M_{11} \| N_i^{t+1'} \| N_j^{t+1'}\right).$$

7) Drone ID_i first retrieves $N_j^{t+1'}$ from M_{11} through $S^{-1}(M_{11})$ and calculates $\mathrm{MAC}_{11}^{'}$ as follows:

$$\mathrm{MAC}_{11}^{'} = C\left(M_{11} \| N_i^{t+1} \| N_j^{t+1'}\right).$$

If $\mathrm{MAC}_{11}^{'} = \mathrm{MAC}_{11}$, the message verification succeeds. Otherwise, it discards the message. Then, it updates the entry $[\mathrm{ID}_i, \mathrm{PID}_i^{t+1}, (C_i^{t+1}, R_i^{t+1})]$ with previously calculated values. Finally, it calculates the secret session key in the following:

$$SK_{i,j} = H\left(N_i^{t+1}\right) \oplus H\left(N_j^{t+1'}\right).$$

By this time, the mutual authentication between drone ID_i and drone ID_j is finally succeeded and the secret session key $SK_{i,j}$ has been securely established for the subsequent communications.

VI. Security Verification and Analysis

In this section, we first verify PMAP using AVISPAs tool [11]. Then, we provide a formal security analysis of PMAP based on the formal security protocol analysis approach in [12]. The automated security verification and formal security analysis prove that any adversary cannot obtain or alter critical communication information. Third, we informally analyze the resilience and immunity of PMAP against various types of attacks. Finally, we compare PMAP with AKA [15] and IBE-Lite [16] in terms of various security requirements.

A. Security Verification Using AVISPA

In this section, we demonstrate how we use AVISPAs tool [11] to verify whether PMAP is secure against replay and man-in-the-middle attacks. AVISPA is invented as a push-button tool with the integration of a modular and expressive formal language that can be used to design security protocols and their properties. In addition, AVISPA provides different back-ends that implement various modern automated analysis techniques. Taking advantage of AVISPA's HLPSL, users can easily define a security problem that the security protocol is going to solve. HLPSL is an expressive, modular, role based, formal language that is used to specify control-flow patterns, data structures, alternative intruder models, complex security properties, as well as different cryptographic primitives and their algebraic properties.

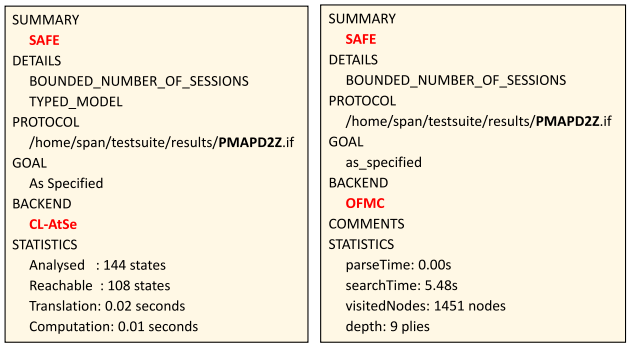


Fig. 5. Security verification results of PMAP^{D2Z} using CL-AtSe and OFMC back-ends in AVISPA.

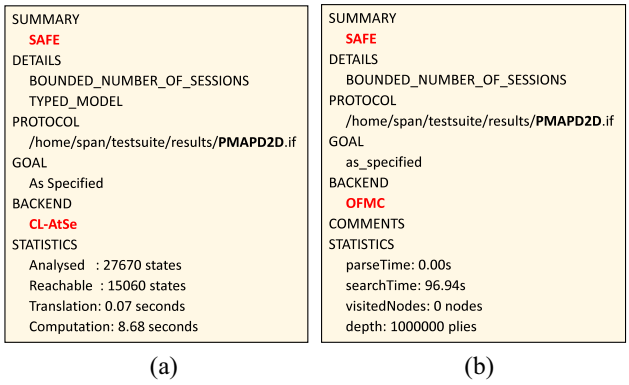


Fig. 6. Security verification results of PMAP^{D2D} using CL-AtSe and OFMC back-ends in AVISPA.

We first implement PMAP^{D2Z} and PMAP^{D2D} using HLPSL, and then choose CL-AtSe and OFMC back-ends [11] to evaluate their security performance. The CL-AtSe provides a translation from any security protocol specification written as transition relation in the intermediate format (IF) into a set of constraints that can be effectively used to find attacks on protocols. The OFMC can be employed not only for falsification of protocols (i.e., fast detection of attacks), but also for protocol verification (i.e., proving the protocol correct for a bounded number of sessions). In our implementation, there are two basic roles: 1) drone and 2) ZSP. In addition to these two basic roles, the other four mandatory roles, such as session, goal, environment, and intruder roles, are also implemented for the security analysis of PMAP^{D2Z} and PMAP^{D2D} in CL-AtSe and OFMC back-ends. Finally, we set up a complete and fully functional SPAN + AVISPA [29] on Ubuntu 10.04 which is running in Virtual Box [30]. The security verification results of PMAP^{D2Z} and PMAP^{D2D} are shown in Figs. 5 and 6, respectively. As we can see that PMAP is secure against replay attack and man-in-the-middle attack. The HLPSL security verification program of CL-AtSe and

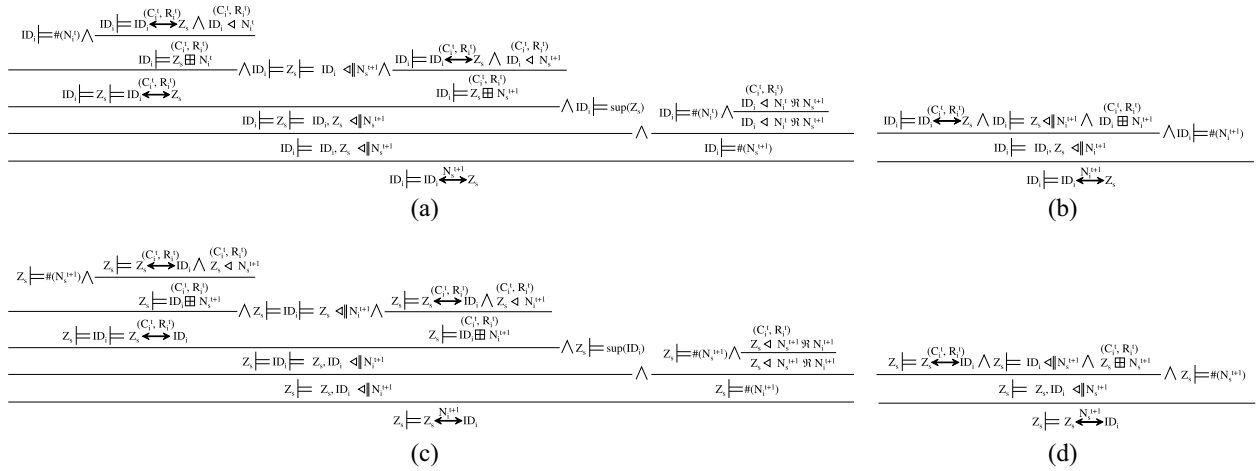


Fig. 7. Formal security analysis of PMAP^{D2Z} for the communication between drone ID$_i$ and ZSP Z_s. (a) Proof that drone ID$_i$ believes N_s^{t+1} is a good shared secret between drone ID$_i$ and ZSP Z_s. (b) Proof that drone ID$_i$ believes N_i^{t+1} is a good shared secret between drone ID$_i$ and ZSP Z_s. (c) Proof that ZSP Z_s believes N_i^{t+1} is a good shared secret between ZSP Z_s and drone ID$_i$. (d) Proof that ZSP Z_s believes N_s^{t+1} is a good shared secret between ZSP Z_s and drone ID$_i$.

OFMC for both PMAP^{D2Z} and PMAP^{D2D} can be found at the https://github.com/congpu/PMAP.

B. Formal Security Analysis of PMAP^{D2Z}

In this section, we present a formal security analysis for the mutual authentication and key agreement between drone ID$_i$ and ZSP Z_s, where the secrets N_s^{t+1} and N_i^{t+1} are shown to be good shared secrets between drone ID$_i$ and ZSP Z_s. In other words, the secrets N_s^{t+1} and N_i^{t+1} cannot be obtained or altered by any adversary. To establish the security analysis of shared secrets N_s^{t+1} and N_i^{t+1}, Mao and Boyd [12] adopted a set of inference rules. In addition, the initial beliefs or assumptions of security analysis for drone ID$_i$ and ZSP Z_s mutual authentication and key agreement are given in the following.

1) $ID_i \models ID_i \overset{(C_i^t, R_i^t)}{\longleftrightarrow} Z_s$ *and* $Z_s \models Z_s \overset{(C_i^t, R_i^t)}{\longleftrightarrow}$ ID$_i$*:* The initial CRP (C_i^t, R_i^t) of drone ID$_i$ is securely shared between drone ID$_i$ and ZSP Z_s.
2) $ID_i \models Z_s \triangleleft \| ID_i$*:* The real identify of drone ID$_i$ is known by ZSP Z_s.
3) $ID_i \models ID_i \overset{PID_i^t}{\longleftrightarrow} Z_s$ *and* $Z_s \models Z_s \overset{\text{PID}_i^t}{\longleftrightarrow}$ ID$_i$*:* ZSP Z_s saves the pseudonym of drone ID$_i$ in its database, while drone ID$_i$ can compute its PID$_i^t$ using its real identify and CRP (C_i^t, R_i^t).
4) $Z_s \models$ ID$_i \triangleleft \| N_s^{t+1}$ *and* ID$_i \models Z_s \models \{ID_i\} \triangleleft \| N_s^{t+1}$*:* ZSP Z_s generates a new N_s^{t+1} each time.
5) $ID_i \models Z_s \triangleleft \| N_i^{t+1}$ *and* $Z_s \models$ ID$_i \models \{Z_s\} \triangleleft \| N_i^{t+1}$*:* Drone ID$_i$ generates a new N_i^{t+1} each time.
6) $ID_i \models sup(Z_s)$*:* ZSP Z_s is the super principal to drone ID$_i$.
7) $Z_s \models sup($ID$_i)$*:* Drone ID$_i$ is the super principal to ZSP Z_s.
8) $ID_i \models \#(N_i^{t+1})$*:* Drone ID$_i$ generates a fresh N_i^{t+1} each time.
9) $ID_i \models \#(N_i^t)$*:* Drone ID$_i$ generates a fresh N_i^t each time.
10) $Z_s \models \#(N_s^{t+1})$*:* ZSP Z_s generates a fresh N_s^{t+1} each time.
11) $ID_i \overset{(C_i^t, R_i^t)}{\boxplus} N_i^t$*:* Drone ID$_i$ encrypts the message M_1 piggybacked with N_i^t using its CRP (C_i^t, R_i^t).
12) $Z_s \overset{(C_i^t, R_i^t)}{\triangleleft} N_i^t$*:* ZSP Z_s decrypts the encrypted message M_1 using drone ID$_i$'s CRP (C_i^t, R_i^t).
13) $Z_s \overset{(C_i^t, R_i^t)}{\boxplus} N_s^{t+1}$*:* ZSP Z_s encrypts the message M_2 piggybacked with N_s^{t+1} using drone ID$_i$'s CRP (C_i^t, R_i^t).
14) $ID_i \overset{(C_i^t, R_i^t)}{\triangleleft} N_i^t \Re N_s^{t+1}$*:* Drone ID$_i$ decrypts the encrypted message M_2 using its CRP (C_i^t, R_i^t).
15) $ID_i \overset{(C_i^t, R_i^t)}{\boxplus} N_i^{t+1}$*:* Drone ID$_i$ encrypts the message M_3 piggybacked with N_i^{t+1} using its CRP (C_i^t, R_i^t).
16) $ID_i \overset{(C_i^t, R_i^t)}{\boxplus} R_i^{t+1}$*:* Drone ID$_i$ encrypts the message M_4 piggybacked with R_i^{t+1} using its CRP (C_i^t, R_i^t).
17) $Z_s \overset{(C_i^t, R_i^t)}{\triangleleft} N_s^{t+1} \Re N_i^{t+1}$ *and* $Z_s \overset{(C_i^t, R_i^t)}{\triangleleft} R_s^{t+1}$*:* ZSP Z_s decrypts the encrypted message M_3 and M_4 using drone ID$_i$'s CRP (C_i^t, R_i^t), respectively.
18) $ID_i \models \#((C_i^{t+1}, R_i^{t+1}))$, $ID_i \models Z_s{}^c \triangleleft \| (C_i^{t+1}, R_i^{t+1})$, *and* $Z_s \models ID_i \models \{Z_s\}^c \triangleleft \| (C_i^{t+1}, R_i^{t+1})$*:* Drone ID$_i$ computes a new CRP (C_i^{t+1}, R_i^{t+1}) each time using its PUF.

Fig. 7 shows the formal security analysis of the mutual authentication and key agreement between drone ID$_i$ and ZSP Z_s, where the security claim that the secrets N_s^{t+1} and N_i^{t+1} are good shared secrets between drone ID$_i$ and ZSP Z_s is proved. For example, Fig. 7(b) presents the proof of security claim that drone ID$_i$ believes N_i^{t+1} is a good shared secret between drone ID$_i$ and ZSP Z_s. To establish this security claim, the statement, ID$_i \models$ ID$_i \overset{N_i^{t+1}}{\longleftrightarrow} Z_s$, is first created and placed at the bottom of the logical construct. Next, we apply the Good Key rule from [12] to the statement ID$_i \models$ ID$_i \overset{N_i^{t+1}}{\longleftrightarrow} Z_s$. The Good Key rule from [12] states that if ID$_i$ believes that N_i^{t+1} is only available to ID$_i$ and Z_s (ID$_i \models \{$ID$_i, Z_s\} \triangleleft \| N_i^{t+1}$), and ID$_i$ knows that N_i^{t+1} is fresh (ID$_i \models \#(N_i^{t+1})$), then ID$_i$ believes that

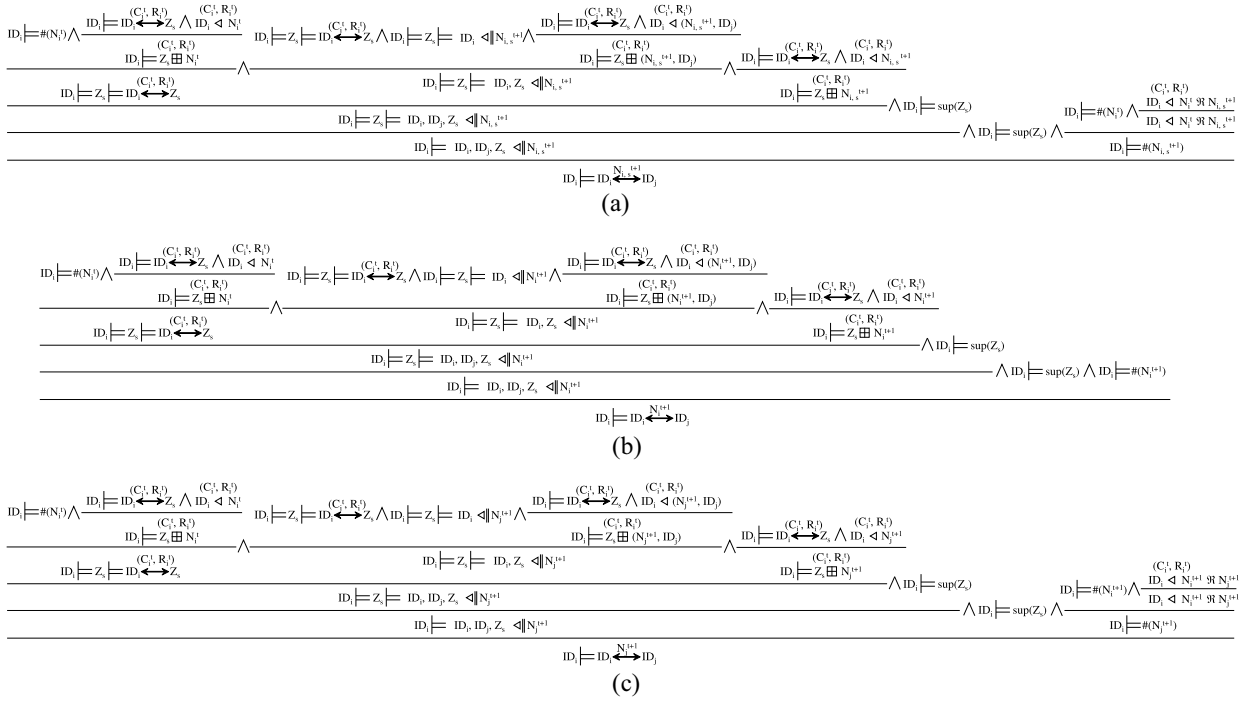


Fig. 8. Formal security analysis of PMAP^{D2D} for the communication between drone ID$_i$ and drone ID$_j$. (a) Proof that drone ID$_i$ believes $N_{i,s}^{t+1}$ is a good shared secret between drone ID$_i$ and drone ID$_j$. (b) Proof that drone ID$_i$ believes N_i^{t+1} is a good shared secret between drone ID$_i$ and drone ID$_j$. (c) Proof that drone ID$_i$ believes N_j^{t+1} is a good shared secret between drone ID$_j$ and drone ID$_i$.

N_i^{t+1} is a good shared secret key between ID$_i$ and Z_s. Then, the Confidentiality rule from [12] can be adopted to prove ID$_i \models \{\text{ID}_i, Z_s\} \triangleleft \| N_i^{t+1}$, which requires to show that (C_i^t, R_i^t) is a shared secret key between ID$_i$ and Z_s (ID$_i \models$ ID$_i \overset{(C_i^t, R_i^t)}{\longleftrightarrow} Z_s$), ID$_i$ encrypts it (ID$_i \overset{(C_i^t, R_i^t)}{\boxplus} N_i^{t+1}$), and ID$_i$ sends it to Z_s without sharing with anyone else (ID$_i \models Z_s \triangleleft \| N_i^{t+1}$). Since these statements can be easily found in the initial beliefs above, we can easily prove the truth of the security claim, ID$_i \models$ ID$_i \overset{N_i^{t+1}}{\longleftrightarrow} Z_s$. By following the similar idea, Fig. 7(a) proves the security claim that ID$_i$ believes N_s^{t+1} is a good shared secret key between ID$_i$ and Z_s. Similar security analysis for N_s^{t+1} and N_i^{t+1} on the side of ZSP Z_s is shown in Fig. 7(c) and (d), respectively.

In summary, as the formal security analysis shown in Fig. 7, without the knowledge of the initial CRP (C_i^t, R_i^t), it is impossible for any adversary to correctly decrypt messages and obtain the secrets N_s^{t+1} and N_i^{t+1}. In addition, even if an adversary can physically capture drone ID$_i$, it still cannot retrieve a valid CRP (C_i^t, R_i^t) because drone ID$_i$ does not store its CRP in the memory. Last but not least, an adversary may attempt to probe or alter the circuit of captured drone ID$_i$, however, this attempt will change the PUF challenge-response mapping, or even destroy the PUF. Therefore, the mutual authentication and key agreement between drone ID$_i$ and ZSP Z_s are resilient and secure.

C. Formal Security Analysis of PMAP^{D2D}

In this section, we present a formal security analysis for the mutual authentication and key agreement between drone ID$_i$ and drone ID$_j$, where the secrets $N_{i,s}^{t+1}$, N_i^{t+1}, and N_j^{t+1} are shown to be good shared secrets between drone ID$_i$ and drone ID$_j$. Fig. 8 shows the proof of above security claim on the side of drone ID$_i$. The similar idea can be applied to obtain the formal security analysis on the side of drone ID$_j$ for the secrets $N_{j,s}^{t+1}$, N_i^{t+1}, and N_j^{t+1}. It is obvious that any adversary cannot compromise PMAP^{D2D} for the communication between drone ID$_i$ and drone ID$_j$ without knowing valid CRP (C_i^t, R_i^t). As a result, it is not possible to obtain the secrets $N_{i,s}^{t+1}$, N_i^{t+1}, and N_j^{t+1} to establish the valid secret session key. Therefore, the mutual authentication and key agreement between drone ID$_i$ and drone ID$_j$ are resilient and secure.

D. Resilience and Immunity Analysis to Various Attacks

In this section, we informally exhibit that PMAP is resilient and immune to drone impersonation attack, ZSP spoofing attack, message modification attack, drone capture attack, replay attack, known session key attack, and man-in-the-middle attack. For simplicity, we consider mutual authentication and key agreement between drone ID$_i$ and ZSP Z_s shown in Fig. 3 for analysis.

1) Drone Impersonation Attack: Suppose that an adversary $\mathcal{A}$ wants to impersonate a legitimate drone ID$_i$ in order to establish authentication with ZSP Z_s to cause some financial and strategic damages. In order to send a valid authentication request to ZSP Z_s, e.g., $[M_1, \text{MAC}_1]$ on behalf of legitimate drone ID$_i$, $\mathcal{A}$ obtains the identifier of ZSP Z_s and then generates a random number N_i^t. However, without having the valid CRP (C_i^t, R_i^t) of legitimate drone ID$_i$, it is a difficult task for $\mathcal{A}$ to shuffle and calculate M_1 which can be correctly decoded

by ZSP Z_s through reshuffling. As a result, $\mathcal{A}$ can not generate a valid authentication request on behalf of legitimate drone ID_i. Therefore, PMAP is resilient and immune to drone impersonation attack.

2) ZSP Spoofing Attack: Suppose that $\mathcal{A}$ wants to pretend itself as a trusted ZSP Z_s and sends a message $[M_2, \mathrm{MAC}_2]$ to legitimate drone ID_i. $\mathcal{A}$ generates a random number N_s^{t+1} and computes M_2 with N_i^t. Since $\mathcal{A}$ does not have the message $[M_1, \mathrm{MAC}_1]$, it has to randomly generate a number as N_i^t. On receiving the message $[M_2, \mathrm{MAC}_2]$, drone ID_i retrieves N_i^t with its CRP (C_i^t, R_i^t), computes $MAC_2^{'}$, and checks whether MAC_2 is equal to $MAC_2^{'}$. However, since $\mathcal{A}$ could not either get valid N_i^t through reshuffling M_1 or shuffle M_2 with drone ID_i's valid CRP (C_i^t, R_i^t), drone ID_i can easily find out that ZSP Z_s is vicious. Therefore, PMAP is resilient and immune to ZSP spoofing attack.

3) Message Modification Attack: As shown in Fig. 3, the transmitted messages M_2 and M_3 are composed of random numbers for establishing a secret session key. The legitimate communicating entities can easily estimate whether the message is modified by $\mathcal{A}$ through checking the equation of $\mathrm{MAC}_2 = \mathrm{MAC}_2^{'}$ or $\mathrm{MAC}_{34} = \mathrm{MAC}_{34}^{'}$. Therefore, PMAP is resilient and immune to message modification attack.

4) Drone Capture Attack: Assume that $\mathcal{A}$ has physically captured legitimate drone ID_i who is currently communicating with ZSP Z_s. $\mathcal{A}$ can obtain drone ID_i's stored valuable communication information, such as ID_i and $SK_{i,s}$, with the help of power analysis attacks. The CRP (C_i^t, R_i^t) is embedded in the PUF and $\mathcal{A}$ may try to probe or alter the integrated circuit of captured drone ID_i to retrieve CRP (C_i^t, R_i^t). However, this attempt will irreversible modify the slight physical variations in the integrated circuit and in turn destroy the PUF. Thus, even though $\mathcal{A}$ gets ID_i and $SK_{i,s}$, it can not retrieve the valid CRP (C_i^t, R_i^t). Note that $\mathcal{A}$ can only compromise the current communication session between drone ID_i and ZSP Z_s because different secret session keys are used for communications between other drones and ZSP Z_s. Thus, the secret session keys between other noncaptured drones and ZSP Z_s can not be compromised by $\mathcal{A}$. As a result, PMAP is resilient and immune to drone capture attack.

5) Replay Attack: As shown in Fig. 3, both drone ID_i and ZSP Z_s choose random numbers $(N_i^t, N_s^{t+1}$, and $N_i^{t+1})$ and calculate authentication request and response messages (M_1, M_2, and M_3). Because of the freshness of random numbers, drone ID_i and ZSP Z_s can easily distinguish the replayed message from previously received messages through message validation. Therefore, PMAP is resilient and immune to replay attack.

6) Known Session Key Attack: Assume that $\mathcal{A}$ knows the secret session key $SK_{i,s}$ for a particular communication session between drone ID_i and ZSP Z_s. Since the secret session key $SK_{i,s}$ is calculated as the bitwise XOR operations between two hash values of random numbers, $\mathcal{A}$ can not calculate random numbers (e.g., N_s^{t+1} and N_i^{t+1}) from $SK_{i,s}$ due to the collision-resistant feature of a secure one-way hash function. Therefore, PMAP is resilient and immune to known session key attack.

7) Man-in-the-Middle Attack: As shown in Fig. 3, it is clear that drone ID_i can be authenticated by ZSP Z_s through its CRP (C_i^t, R_i^t), and ZSP Z_s can be authenticated by drone ID_i because ZSP Z_s knows ID_i's CRP (C_i^t, R_i^t). As a result, two communicating entities, drone ID_i and ZSP Z_s, can authenticate each other and establish a secret session key. Therefore, PMAP is resilient and immune to man-in-the-middle attack.

TABLE III
COMPARISON OF SECURITY REQUIREMENTS

Security Requirement	AKA	IBE-Lite	PMAP
Auth. Between Drone and User*	Yes	Yes	**Yes**
Auth. Between Drone and Drone	No	No	**Yes**
Integrity	No	No	**Yes**
Anonymity	Yes	No	**Yes**
Session Key Agreement	Yes	No	**Yes**
Drone Impersonation Attack	Yes	No	**Yes**
ZSP Spoofing Attack	Yes	-	**Yes**
Message Modification Attack	Yes	No	**Yes**
Drone Capture Attack	Yes	Yes	**Yes**
Replay Attack	Yes	-	**Yes**
Known Session Key Attack	Yes	-	**Yes**
Man-In-The-Middle Attack	Yes	Yes	**Yes**

∗: In *PMAP*, ZSPs are equivalent to users in both AKA and IBE-Lite.

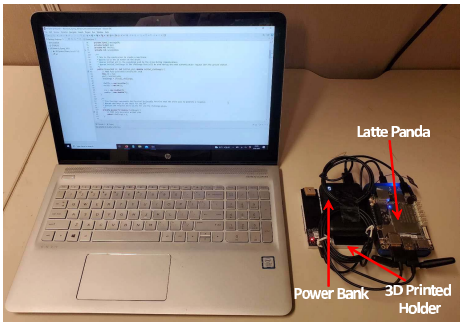


Fig. 9. Real-world testbed with one HP ENVY notebook laptop, one Latte Panda development board, and one power bank bounded together with the 3D-printed plastic holder.

E. Security Comparison

The comparison of security requirements between PMAP and two prior security protocols, i.e., AKA [15] and IBE-Lite [16], is provided in Table III. In summary, PMAP can satisfy all security requirements mentioned in Section IV, and provide better security performance than its opponents.

VII. Performance Evaluation

A. Experimental Testbed and Benchmarks

As shown in Fig. 9, we build a real-world testbed consisting of one HP ENVY Notebook laptop [13], one Latte Panda development board [14], and one power bank. In terms of testbed specifications, the HP ENVY Notebook laptop is running a 64-bit Windows 10 Pro operating system, and its central processing unit (CPU) is the seventh Generation Intel Core i7-7500U processor, 4M Cache, up to 3.5 GHz. The Latte Panda development board runs a full version of Windows 10, and has an Intel Cherry Trail Z8350 Quad Core processor, 2M cache, up to 1.92 GHz, and 4-GB random-access memory (RAM). The power bank is capable enough to support the Latte Panda development board for two hours. To bind the Latte Panda

development board and power bank together, we build a specific plastic holder using 3-D printer. In the developed testbed, the laptop and the Latte Panda development board are used to simulate ZSP and drone, respectively. PMAP and benchmark schemes have been implemented in Eclipse for Java environment [31] which was set up in Latte Panda as well as a laptop. Like most other works in [15], [16], and [19], we assume an ideal wireless medium between communication entities. Thus, PMAP and benchmark schemes are executed directly in the developed testbed, and there is no wireless communication (i.e., message exchanges) between communication entities in the experiment.

According to [25], we implement the PUF as a 256-bit hash function [26]. In addition, the random shuffling function is implemented as follows. First, the to-be-shuffled message is represented as an array. Second, the CRP pair $(C_i^t,\ R_i^t)$ is used as the initial condition of the Henon map (1) to generate a sequence of points. Third, starting from the first point in the sequence, the coordinates of a point are converted into a unique integer, indicating the new location where the first element of an array is to be put in the output array. Now considering the array element from the second to the last, the abovementioned process is repeated till the last array element is shuffled. Finally, the output array contains the shuffled message. Please note that the random shuffling function is executed differently in every communication session. This is because the drone will compute a new CRP pair during the process of mutual authentication and session key establishment. Since the CRP pair is used as the initial condition of the Henon map and a minor change of the initial condition in the Henon map will cause the generation of a distinct sequence of points (see more details about Henon map's features in Section III), the random shuffling operation is performed differently in every communication session.

We revisit recent security protocols, AKA [15] and IBE-Lite [16], and implement them to work in the testbed for performance comparison and analysis. The original idea of these two benchmark schemes is briefly discussed in the following.

1) *AKA [15]:* The basic idea of AKA is that drones and users mutually authenticate each other using a secure one-way hash function and bitwise XOR operations. AKA consists of three phases: a) setup; b) registration; and c) mutual authentication. In the setup phase, the control server generates its master private key and other public system parameters. In the registration phase, drones and users register with the control server and get their secret key via a secure channel. In the mutual authentication phase, drones and users communicate with each other and establish a session key.
2) *IBE-Lite [16]:* IBE-Lite consists of three phases: a) initialization; b) data encryption; and c) data decryption. In the initialization phase, the user loads public parameters to the drone and registers the master secret key with the certificate authority (CA). After collecting the required data, the drone creates a string according to a preagreed syntax. Using this string, the drone can derive a public key by the public parameters, encrypt the collected data, and sends the ciphertext to the cloud for storage and processing. When a different user wishes to access the collected data in the cloud, he/she needs to query the CA for permission and uses the derived secret key to decrypt the collected data which was encrypted by the drone.

TABLE IV
COMPARISON OF COMMUNICATION COST

Metrics	$PMAP^{D2Z}$	AKA	IBE-Lite
Number of Messages	3	7	8
Size of Messages (byte)	447	1,603	1,176
Energy Consumption (joule)	3.38×10^{-4}	7.88×10^{-4}	9.01×10^{-4}

Please note that AKA and IBE-Lite only provide mutual authentication and session key agreement between drone and user. According to the basic idea of the drone-to-user scheme in AKA and IBE-Lite, we also implement the drone-to-drone scheme of AKA and IBE-Lite for performance comparison. Our approach PMAP consists of two schemes: 1) PMAP^{D2Z}, which mutually authenticates drone and ZSP and establishes a secure session key and 2) PMAP^{D2D}, which mutually authenticates drone and drone and establishes a secure session key.

We measure the performance of PMAP, AKA, and IBE-Lite in terms of communication cost, running time, CPU time, and energy consumption by changing the number of executed algorithms. Communication cost is measured with regard to the number of messages, the size of messages, and energy consumption of communication. We directly count the number of exchanged messages and calculate the size of messages for PMAP, AKA, and IBE-Lite. The energy consumption of communication is calculated based on the number of sent and received messages [32]. Running time is the elapsed time from when the algorithm starts running to when the algorithm finishes running. CPU time (or processing time) is the amount of time for which the CPU is used for processing instructions of the algorithm.[2] Energy consumption of the algorithm is measured as the amount of electronic power consumed during running the algorithm. Moreover, running time, CPU time, and energy consumption of algorithms are measured through VisualVM [33]. VisualVM is a tool that provides a visual interface for viewing detailed information about Java applications/algorithms while they are running on a Java virtual machine.

B. Experimental Results and Analysis

First, we measure the communication cost in terms of the number of messages, size of messages, and energy consumption of communications in Table IV. Since AKA and IBE-Lite only provide mutual authentication and key agreement service between drone and user, the number of messages is measured for PMAP^{D2Z}. To mutually authenticate drone and user and establish a session key, AKA requires seven messages to be

[2]As opposed to running time, CPU time does not include waiting for input/output (I/O) operations or entering low-power (idle) mode.

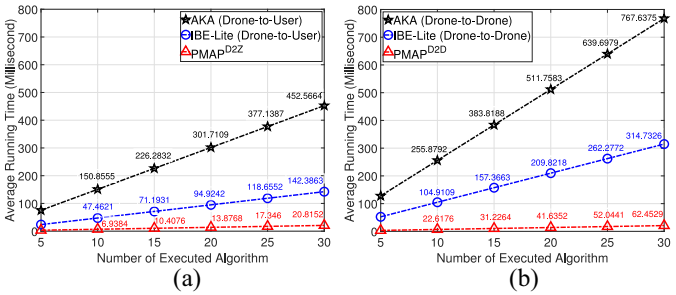


Fig. 10. Performance of average running time against the number of the executed algorithm.

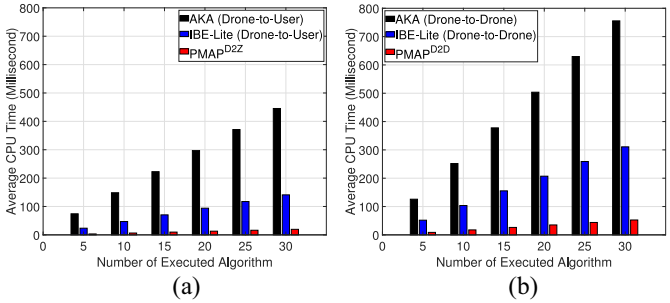


Fig. 11. Performance of average CPU time against the number of executed algorithm.

exchanged among drone, control server, and user. To be specific, a drone and an user first exchange two messages with the control server during the registration phase, respectively. Then, the user sends an authentication request message to the control server through a public channel. After receiving the authentication request message from the user, the control server verifies the message, and sends the message to drone via a public channel. Finally, the drone checks the validation of message and sends a message to user if the verification successes. In IBE-Lite, the CA and the cloud storage are used for generating secret key and data outsourcing. The user who wishes to store the shared data in the cloud needs to communicate with the CA to set up the master key and public parameters. The user loads the parameters to the drone and registers the master secret key together with additional instructions with the CA. Then, the user who wishes to access the data in the cloud needs to query the CA for permission. Therefore, more messages are required in order to exchange the critical information in IBE-Lite. In our approach, as shown in Fig. 3, PMAP^{D2Z} only requires three messages to achieve mutual authentication and secure session key agreement. First of all, a drone sends an authentication request message piggybacked with its pseudonym, PUF response, and random number to ZSP. Then, ZSP verifies the received message and replies a message with one random number. Lastly, the drone forwards a message with the updated CRP to ZSP. By this time, the mutual authentication is completed and the secure session key has been established between drone and ZSP. As shown in Fig. 4, PMAP^{D2D} will require six messages to achieve mutual authentication between drone and drone and establish the secure session key. However, AKA and IBE-Lite require nine messages and ten messages, respectively. In addition, the average size of messages for AKA, IBE-Lite, and PMAP^{D2Z} is 1603, 1176, and 447 bytes, respectively. And the energy consumption of communication for AKA, IBE-Lite, and PMAP^{D2Z} are 7.88×10^{-4}, 9.01×10^{-4}, and 3.38×10^{-4} J, respectively. It is clear that our approach PMAP^{D2Z} has a lower communication overhead compared to other two schemes. PMAP^{D2D} requires six messages to achieve the goal of mutual authentication and secure session key agreement between drone and drone. However, AKA and IBE-Lite require nine messages and ten messages, respectively.

Second, we measure the performance of average running time against the number of the executed algorithm in Fig. 10.

TABLE V
COMPARISON OF ALGORITHM ENERGY CONSUMPTION (JOULE)

Execution Times	5	10	15	20	25	30
AKA	0.62	1.24	1.86	2.48	3.11	3.73
IBE-Lite	0.23	0.47	0.7	0.93	1.16	1.41
$PMAP$	0.04	0.07	0.11	0.15	0.19	0.22

As shown in Fig. 10(a), PMAP^{D2Z} achieves the lowest average running time compared to AKA and IBE-Lite. This is because PMAP^{D2Z} employs lightweight cryptographic operations, such as a one-way hash function and random shuffling, which consumes less amount of energy. Most importantly, these cryptographic operations are executed less number of times. In addition, AKA adopts a one-way hash function and bitwise XOR operations. However, it repeatedly performs a hash function and bitwise XOR operations on various secret values, which takes more time. For example, in PMAP^{D2Z}, the one-way hash function is used nine times to generate secret information. However, in AKA, the one-way hash function is used 21 times. IBE-Lite adopts the identity-based encryption, which is a type of public-key encryption. In the identity-based encryption, a user generates a public key from a known unique identifier, and a trusted third-party server calculates the corresponding private key from the public key. IBE-Lite applies a lightweight function to generate a public key using an arbitrary string. Thus, the energy consumption of IBE-Lite is lower compared to that of AKA. In Fig. 10(b), we measure the performance of average running time with a varying number of an executed algorithm for the drone-to-drone authentication scenario. It is clearly shown that PMAP^{D2D} still outperforms AKA and IBE-Lite.

Third, as shown in Fig. 11, the average CPU time of AKA, IBE-Lite, and PMAP increases as the number of the executed algorithm increases. AKA still shows the highest CPU time because hashing and bitwise XOR operations are repeatedly executed. With 30 times algorithm execution, our approach PMAP still has very competitive performance compared to AKA and IBE-Lite. This is because PMAP adopts a lightweight hashing function and applies random shuffling to generate credentials. Finally, the measurement of energy consumption with a varying number of executed algorithm is shown in Table V. Overall, the lowest energy consumption is obtained by PMAP. This is because fewer operations are required to achieve mutual authentication and key

agreement. As a result, less amount of energy is consumed by PMAP.

VIII. Discussion

In this section, we first investigate PMAP's design features, and then discuss future research direction.

First, PMAP is designed with the lightweight operations, e.g., PUF, chaotic system, random shuffling, hash function, and bitwise XOR, to achieve mutual authentication and session key establishment for communication entities in the IoD system. As a result, the computation overhead can be significantly reduced. Second, unlike existing schemes, where ZSPs or ground stations can only authenticate with drones, our approach PMAP can provide mutual authentication for the communications between drones and ZSP as well as between drones and drones. Third, PMAP supports conditional privacy-preserving so that the genuine identity of drones can only be revealed by trusted ZSPs. In addition, the drone will use different pseudonyms for each communication session. Fourth, we have evaluated the security performance of PMAP through formal and informal security analysis and security verification. Our security analysis and verification results have proved that PMAP can defend against various well-known security attacks. Finally, we release PMAP's source code and security verification programs to promote the broad adoption and drive creative advancement.

As a future work, we plan to integrate PMAP with the blockchain technique and develop a secure data collection and storage mechanism for IoD environments, where ZSPs will pack the collected data into blocks and compete to add its blocks into the blockchain.

IX. Conclusion

In this article, we proposed a lightweight and privacy-preserving mutual authentication and key agreement protocol (PMAP) based on PUF and a chaotic system to achieve mutual authentication and establish a secure session key between communication entities in the IoD. PMAP consists of schemes for two different scenarios in the IoD: 1) when drone and ZSP want to mutually authenticate with each other and establish a secure session key and 2) when two drones want to mutually authenticate with each other and establish a secure session key. The security verification as well as formal and informal security analyses provide strong evidence that any adversary cannot obtain or alter critical communication information, and PMAP is resilient and immune against various types of security attacks. In addition, we developed a real-world testbed and compared PMAP with recent benchmark schemes for performance evaluation and comparison. The experimental results show that PMAP has better performance in terms of computation cost, energy consumption, and communication overhead, indicating a viable and competitive approach for securing communications in the IoD. In summary, this article makes the following contributions to the IoD community. First, we proposed a lightweight and privacy-preserving mutual authentication and key agreement protocol based on a chaotic system and PUF. The proposed security scheme will have important implications for other security mechanisms in the IoD networks, and will provide design considerations to the broader IoD community seeking new cryptographic research directions. Second, in order to promote the broad adoption and drive creative advancement in the realm of security protocols within the IoD community, we release PMAP source codes and its security verification programs at the https://github.com/congpu/PMAP.

References

[1] N. Ngoenriang, S. J. Turner, D. Niyato, and S. Supittayapornpong, "Joint UAV-placement and data delivery in aerial inspection under uncertainties," *IEEE Internet Things J.*, early access, Sep. 20, 2021, doi: 10.1109/JIOT.2021.3113713.

[2] K. Caswell. "The Post-Pandemic Future of the Drone Industry." 2020. [Online]. Available: https://www.dronegenuity.com/post-covid-future-of-the-drone-industry

[3] Y. K. Teoh, S. S. Gill, and A. K. Parlikad, "IoT and fog computing based predictive maintenance model for effective asset management in industry 4.0 using machine learning," *IEEE Internet Things J.*, early access, Jan. 11, 2021, doi: 10.1109/JIOT.2021.3050441.

[4] C. Pu and L. Carpenter, "Psched: A priority-based service scheduling scheme for the Internet of Drones," *IEEE Syst. J.*, vol. 15, no. 3, pp. 4230–4239, Sep. 2021.

[5] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," *IEEE Access*, vol. 8, pp. 90225–90265, 2020.

[6] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100218.

[7] "Poor Security Puts Millions at Risk as 13 Year Old Demonstrates Drone Hack." 2019. [Online]. Available: https://www.redrobot.org/events/poor-security-puts-millions-at-risk-as-13-year-old-demonstrates-drone-hack/

[8] B. Nassi, R. Bitton, R. Masuoka, A. Shabtai, and Y. Elovici, "SoK: Security and privacy in the age of commercial drones," in *Proc. IEEE Symp. Security Privacy*, 2021, pp. 73–90.

[9] C. Pu and Y. Li, "Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system," in *Proc. IEEE LANMAN*, 2020, pp. 1–6.

[10] M. Hénon, "A two-dimensional mapping with a strange attractor," *Commun. Math. Phys.*, vol. 50, pp. 69–77, Feb. 1976.

[11] "Automated Validation of Internet Security Protocols and Applications." [Online]. Available: http://www.avispa-project.org/ (Accessed: Jul. 22, 2021).

[12] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *Proc. Comput. Security Found. Workshop VI*, 1993, pp. 147–158.

[13] "Laptop Computers." [Online]. Available: https://www.hp.com/us-en/shop/cat/laptops (Accessed: Jul. 22, 2021).

[14] "Latte Panda." [Online]. Available: https://www.lattepanda.com/ (Accessed: Jul. 22, 2021).

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

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

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

[18] Y. Tian, J. Yuan, and H. Song, "Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones," *J. Inf. Security Appl.*, vol. 48, no. 7, 2019, Art. no. 102354.

[19] T. Alladi, N. Naren, G. Bansal, V. Chamola, and M. Guizani, "SecAuthUAV: A novel authentication scheme for UAV-ground station and UAV–UAV communication," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15068–15077, Dec. 2020.

[20] M. Tanveer, A. U. Khan, N. Kumar, and M. M. Hassan, "RAMP-IoD: A robust authenticated key management protocol for the Internet of Drones," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1339–1353, Jan. 2022.
[21] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K.-K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of Drones," *IEEE Internet Things J.*, early access, Sep. 17, 2021, doi: 10.1109/JIOT.2021.3113321.
[22] K.-H. Chuang, E. Bury, R. Degraeve, B. Kaczer, D. Linten, and I. Verbauwhede, "A physically unclonable function using soft oxide breakdown featuring% native BER and 51.8 fJ/bit in 40-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2765–2776, Oct. 2019.
[23] T. Alladi, V. Chamola, and Naren, "HARCI: A two-way authentication protocol for three entity healthcare IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 361–369, Feb. 2021.
[24] Y. Li and C. Pu, "Lightweight digital signature solution to defend micro aerial vehicles against man-in-the-middle attack," in *Proc. IEEE CSE*, 2020, pp. 92–97.
[25] J. R. Wallrabenstein, "Practical and secure IoT device authentication using physical unclonable functions," in *Proc. IEEE FiCloud*, 2016, pp. 99–106.
[26] "Java Security Standard Algorithm Names." [Online]. Available: https://docs.oracle.com/en/java/javase/12/docs/specs/security/standard-names.html (Accessed: Oct. 30, 2021).
[27] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Upper Saddle River, NJ, USA: Pearson, 2016.
[28] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-based one-time password algorithm," IETF, RFC 6238, 2011.
[29] "SPAN." [Online]. Available: http://www.avispa-project.org/ (Accessed: Jul. 22, 2021).
[30] "VirtualBox." [Online]. Available: https://www.virtualbox.org/ (Accessed: Jul. 22, 2021).
[31] "Eclipse." [Online]. Available: https://www.eclipse.org/downloads/ (Accessed: Jul. 22, 2021).
[32] C. Pu and S. Lim, "A light-weight countermeasure to forwarding misbehavior in wireless sensor networks: Design, analysis, and evaluation," *IEEE Syst. J.*, vol. 12, no. 1, pp. 834–842, Mar. 2018.
[33] "VisualVM." [Online]. Available: https://visualvm.github.io/ (Accessed: Jul. 22, 2021).

Cong Pu (Member, IEEE) received the B.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2009, and the M.S. and Ph.D. degrees in computer science from Texas Tech University, Lubbock, TX, USA, in 2013 and 2016, respectively.

From 2014 to 2016, he was an Instructor with the Department of Computer Science, Texas Tech University. He is currently an Assistant Professor with the Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, USA. His primary research interests include cryptography, network security, wireless networks, mobile computing, and information-centric networking.

Andrew Wall is currently pursuing the Undergraduate degree with the Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, USA.

His research interests include network security and Internet of Drones.

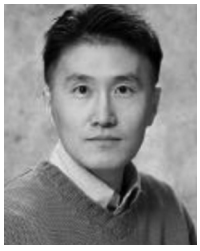
Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from Queensland University of Technology, Brisbane, QLD, Australia, in 2006.

He currently holds the Cloud Technology Endowed Professorship with The University of Texas at San Antonio, San Antonio, TX, USA.

Dr. Choo was a recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the Outstanding Associate Editor of 2018 for IEEE ACCESS, the British Computer Society's 2019 Wilkes Award Runner-up, the 2019 EURASIP *Journal on Wireless Communications and Networking* Best Paper Award, the Korea Information Processing Society's *Journal of Information Processing Systems* Survey Paper Award (Gold) 2019, the IEEE Blockchain 2019 Outstanding Paper Award, the Inscrypt 2019 Best Student Paper Award, the IEEE TrustCom 2018 Best Paper Award, the ESORICS 2015 Best Research Paper Award, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. In 2016, he was named the Cybersecurity Educator of the Year—APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn) and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen–Nuremberg. He is also a Fellow of the Australian Computer Society and the Co-Chair of IEEE Multimedia Communications Technical Committee's Digital Rights Management for Multimedia Interest Group.

Imtiaz Ahmed (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada.

He is an Assistant Professor with the Department of Electrical Engineering and Computer Science, Howard University, Washington, DC, USA. He works in the areas of wireless communications, signal processing, and computer networks. He worked with Intel Corporation, San Diego, CA, USA, as a Wireless Systems Engineer and Marshall University, Huntington, WV, USA, as an Assistant Professor. He is currently working on artificial intelligence aided physical layer design, integration of aerial and terrestrial communication networks, and communication with energy harvesting nodes.



Sunho Lim (Senior Member, IEEE) received the B.S. degree (*summa cum laude*) in computer science and the M.S. degree in computer engineering from Korea Aerospace University, Goyang, South Korea, in 1996 and 1998, respectively, and the Ph.D. degree in computer science and engineering from The Pennsylvania State University, University Park, PA, USA, in 2005.

He was an Assistant Professor with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA, from 2005 to 2009. He is currently an Assistant Professor with the Department of Computer Science, Texas Tech University, Lubbock, TX, USA. His research interests include areas of cybersecurity, mobile data management and privacy, and wireless networks and mobile computing.