Bilinear Pairing and PUF Based Lightweight Authentication Protocol for IoD Environment

Cong Pu[†] Andrew Wall[‡] Kim-Kwang Raymond Choo[¶]
[†]Dept. of Computer Science, Oklahoma State University, Stillwater, OK, United States
[‡]Dept. of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, United States
[¶]Dept. of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX, United States
cong.pu@ieee.org wall48@marshall.edu raymond.choo@fulbrightmail.org

***Abstract*—As the popularity growth of drones is witnessed in various fields, people start attaching importance to the Internet of Drones (IoD) paradigm. In the IoD, the regional aviation administration (i.e., Zone Service Providers (ZSPs)) regulates the usage of vast yet limited airspace and provides necessary services (i.e., supplemental data services) for various drone applications. In order to create a secure environment for communications, authentication and key agreement protocols have an important role to play in the IoD. A few conventional security protocols specifically designed for traditional communication networks cannot be directly exercised in the IoD environment because of their non-negligible computational overhead and the distinctive characteristics of IoD (i.e., insufficient resources of drones). In this paper, we propose a bilinear pairing and physical unclonable function based lightweight authentication protocol (hereafter referred to as *liteCrypto*) for the IoD environment. In *liteCrypto*, a drone and the ZSP mutually authenticate each other and establish a secure session key based on bilinear pairing and physical unclonable function before sharing any critical information over an insecure wireless channel. In terms of performance evaluation, we first implement *liteCrypto* in High-Level Protocol Specification Language (HLPSL) and verify its security performance in the Automated Validation of Internet Security Protocols and Applications (AVISPA) environment, and then present a security analysis of *liteCrypto*. In addition, we develop a real-world testbed, implement *liteCrypto* and its two counterparts (i.e., ECCAuth and RAMP-IoD), conduct extensive experiments, and provide an in-depth performance analysis. Our performance evaluation shows that not only is *liteCrypto* a secure communication protocol, but also outperforms its counterparts in terms of computational overhead, energy consumption, as well as communication cost.**

***Index Terms*—Bilinear Pairing, Physical Unclonable Function, Security, Lightweight Authentication, Internet of Drones.**

I. Introduction

Although originally developed as a radio-controlled aerial missile deployer by United Kingdom and United States during the First World War, drones, officially called unmanned aerial vehicles, have seen a noticeable change in its role in the 21st century [1]. In the present day, drones have been witnessed in many non-military places such as lab sample pick-up and delivery during the global COVID-19 pandemic, drone light shows, etc. As a reflection of the continuing heavy investments, the drone market is estimated to be worth about $43 billion in 2025, which doubles the value ($23 billion) in 2020 [2]. In order to further explore the potential of drones, a novel aerial-ground communication paradigm, Internet of Drones [3], has been proposed and is considered to be a promising communication architecture to drive further growth and success of drone technology.

In the IoD, a set of stationary Zone Service Providers (ZSPs) are distributed in an area of interest, where each ZSP regulates and administers the corresponding airspace and serves as a central connection point for drones to communicate. As the main player, drones freely fly in the mission area, and collect and/or deliver data to nearby ZSP via wireless channel for follow-up analyses. A telling example is the recent revelation that a fleet of surveillance drones is deployed to observe crowds and deliver survey data to ZSPs for modeling and forecasting the spread of coronavirus disease [4]. In the age of Internet of Everything, with the assistance of other advanced technologies (i.e., fifth-generation (5G) telecommunications and artificial intelligence (AI)), we anticipate that the IoD will certainly yield unusually brilliant results in the near future.

When people are completely enchanted by various neoteric IoD applications, the growing threat of security attacks from today's progressively adverse environment has given people a wake-up call. First, the inherent vulnerabilities of wireless medium naturally make drones an easy target; data being transmitted via insure wireless channel can be easily captured and then further manipulated to attack drones [5]. A research group from Johns Hopkins University (Maryland, United States) has discovered three different ways to use mobile device to issue malicious commands to drones, which can intervene in the normal operations of drones and cause them to crash [6]. Second, the insufficient resources (i.e., limited processor capability, memory size, and battery power) of drones make people wonder whether conventional security protocols (i.e., AES, RSA, etc. [7]) can be directly utilized. In [8], it was proven experimentally that the resource-constrained drones are not compatible with the ready-made cryptographic protocols and standard primitives with regard to the consumption of time and energy. Third, the recently developed security protocols for IoD environment only consider a few security primitives; most importantly, they have some inherent vulnerabilities. For instance, an adversary might capture a drone and probe its integrated circuit to retrieve some critical data [9].

In this paper, we propose a cryptographic protocol to protect the communications between drones and the ZSP in the IoD environment. In addition, we verify the proposed security protocol using a security verification tool and analyze its

2155-6814/22/$31.00 ©2022 IEEE
DOI 10.1109/MASS56207.2022.00023

security resilience against various cyber attacks. Finally, we conduct performance evaluation on a real-world testbed, and measure and analyze the performance trade-off of the proposed security protocol. In brief, our contribution is summarized in the following:

- We propose a bilinear pairing and physical unclonable function (PUF) based lightweight authentication protocol (hereafter referred to as *liteCrypto*) for the IoD environment. In *liteCrypto*, a drone and the ZSP mutually authenticate each other and establish a secure session key based on bilinear pairing and PUF before sharing any critical information.
- We implement *liteCrypto* in High-Level Protocol Specification Language (HLPSL) [10] and verify its security performance in the Automated Validation of Internet Security Protocols and Applications (AVISPA) [11] environment. In addition, we present a security analysis to show *liteCrypto* is secure against various security attacks.
- We develop a real-world testbed and conduct extensive experiments to evaluate the performance of *liteCrypto*. We also choose two benchmark schemes, ECCAuth [12] and RAMP-IoD [13], and implement them to work on the testbed for performance comparison and analysis.

According to experimental results and analysis, we conclude that our approach *liteCrypto* provides superior performance than its counterparts in terms of computational overhead, energy consumption, as well as communication cost. To increase creative and innovative work in the realm of security protocols within the IoD community, we open source at the *https://github.com/congpu/liteCrypto*[1].

II. Related Work

The authors in [14] adopt the cryptographic techniques such as FourQ and Boyko-Peinado-Venkatesan (BPV) precalculation to protect the communications among drones, users, and ground control stations. After three major phases such as system initialization, registration, and login and authentication, the drone and the user can verify each other's identities and achieve an agreement on a safe session key for subsequent communications. However, the BPV algorithm has an inherent limitation, which is the increased size of private key (i.e., 64 KB or even more). As for mini drones with limited storage capability, this additional storage requirement is a huge burden. In addition, storing credentials in drone's memory is not a smart move, which makes drones vulnerable to probing attacks on integrated circuit. In [15], an identity based security protocol is proposed for the IoD environment. In the initial phase, critical numerical values are loaded into each communication entity (e.g., sensor, drone, access point, and server) in the IoD. In the registration phase, each entity accepts its requisite system parameters for the following authentication phase, where the mutual authentication is achieved between the sensor and the drone, between the drone and the access point, as well as between the access point and the server, respectively. A major drawback of the proposed protocol is that the single server can easily become a single point of failure for the IoD system. In addition, the role of access point is very vague, because most of computations and communications are performed by sensor, drone, and server.

The authors in [16] propose a blockchain-based data management framework for the IoD environment, where drones and ground stations can establish secure communications through access control mechanism and secure session key. Additionally, they also design a consensus algorithm for the competition of adding blocks in the private blockchain. However, according to the analysis provided by [17], the above-mentioned blockchain-based data management framework has several serious vulnerabilities that make the IoD system very fragile when suffering from impersonation attack, man-in-the-middle attack, and replay attack. In [18], the critical issues of centralized security approach such as single point of failure and the infeasibility of cross-domain validation are raised. To resolve those existing issues, the authors propose a cross-domain security system based on the techniques of blockchain and 5G for the IoD environment. The threshold cryptography is adopted to achieve the federated identity across multiple domains. In addition, they design a smart contract so that the authentication between drones coming from diverse domains can be realized. The authors in [12] propose an authentication scheme based on elliptic curve cryptography so that the user and the drone can establish a secure communication in the designated airspace. In [13], elliptic curve cryptography and hash function are selected to design an authentication scheme for the IoD applications, where the user's identity is validated and a secure session key is created for both user and drone so that they can communicate securely. Based on the performance evaluation, the proposed authentication mechanism meets the pre-determined security requirements and provides competitive performance. However, the major drawback is that it does not support dynamic privacy preservation.

Ever since the IoD came into the spotlight, many researchers have proposed various security protocols to protect the communications in the IoD environment. Unfortunately, it appears that no effort has been spared to design a bilinear pairing and PUF based lightweight authentication protocol for the IoD environment, where a drone and the ZSP mutually authenticate each other and establish a secure session key before sharing any critical information. In addition, the difficulties to build a real-world experimental testbed have confined the implementation and evaluation of security protocols to network simulation. In this paper, we develop a real-world testbed for performance evaluation. Most importantly, we open source *liteCrypto* to drive other scholars to contribute to the development of security protocols in the IoD community.

III. Preliminary Background

A. Bilinear Pairing

Suppose that $\mathbb{G} = \langle \mathrm{P} \rangle$ is a cyclic additive group, where the order of $\mathbb{G}$ is n and P is an arbitrary generator of $\mathbb{G}$. Let $\mathbb{G}_T$ be a multiplicatively-written cyclic group of the same order n.

[1] *liteCrypto* source codes and its security verification programs are publicly available at the *https://github.com/congpu/liteCrypto*.

n is usually set to a very large prime (i.e., at least 1024-bit). A bilinear pairing map on ($\mathbb{G}$, $\mathbb{G}_T$) is defined as $\hat{e}: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, which has the following properties:

1) $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$ and $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$, where P, Q, and $R \in \mathbb{G}$.
2) $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, where a and $b \in \mathbb{Z}$.
3) $\hat{e}(P, P) \neq 1$. Here 1 is the identity element of $\mathbb{G}_T$.
4) $\hat{e}(P, Q) = \hat{e}(Q, P)$.
5) $\hat{e}$ can be efficiently calculated.

The security of bilinear pairing map is built upon the intractability of computational Diffie-Hellman problem such that there is no efficient algorithm to compute $abP \in \mathbb{G}$ (or compute $\hat{e}(P, P)^{abc}$) within polynomial time, given P, aP, bP, and cP. However, the decisional Diffie-Hellman problem can be easily solved; it is easy to decide whether $cP = abP$ (or $ab = c \bmod n$) through checking $\hat{e}(aP, bP) \overset{?}{=} \hat{e}(P, cP)$.

B. Physical Unclonable Function

A physical unclonable function (PUF) is defined as a function that maps an input query to a specific output based on the mulishly complicated physical randomness of integrated circuit. The rationale behind the usage of PUF is that each integrated circuit experiences a slight variation in the manufacturing process, which can be regarded as the unique identity characteristics. Here, the input query is called challenge while the specific output is named as response. A challenge along with its associated response is widely known as a challenge-response pair (CRP). Generally, a secure one-way function, denoted by F_{puf}, is adopted to simulate the PUF, $res = F_{puf}(che)$. Here, res represents the response and che indicates the challenge. One consequence of the PUF property is that the same res is generated if the PUF receives the same che every time. Nonetheless, the PUF will generate totally distinct res with different che.

In hash and unstable environments, the PUF could be influenced by external factors and it is possible that the identical responses cannot be re-generated with the same challenge. As a result, the cryptographic value might not be restored or feasible with the same secret input. In the past few years, PUFs with noise immunity feature have been designed and are in widespread use in noisy environments [19], where zero-bit-error can be guaranteed. Thus, we assume that an ideal and noise-resistant PUF [20] is deployed in drones in this paper.

C. System Model

Fig. 1 demonstrates the system model which consists of two major components: drones and ZSP. For example, drone D_i is deployed in a designated area to collect the data of crowd and send them to nearby ZSP Z_s for further analyzing the spread of coronavirus disease. We assume that a PUF is implemented in the integrated circuit of drone D_i. Due to the limitation of size and weight, drone D_i is assumed to be resource-constrained. In addition, ZSP Z_s is regarded as a fully trusted component and is not concerned with resources.

Drone D_i might unwittingly fly into an adverse environment, and there is some chances to get caught and then compromised by an adversary [21]. However, the adversary is incapable of probing the integrated circuit of drone D_i for cryptographic information. This is because any probing attempt will inevitably change the physical environment of the integrated circuit, resulting in the damage of PUF. Furthermore, drone D_i and ZSP Z_s are communicating via an insecure and open wireless medium, thus they are implicitly assumed to be untrustworthy [7]. In addition, if the plaintext data is being transmitted over insecure wireless channel, it can be easily eavesdropped, captured, manipulated, and then replayed. Thus, before drone D_i and ZSP Z_s perform any critical information exchange, they need to mutually authenticate each other and establish a secure session key.

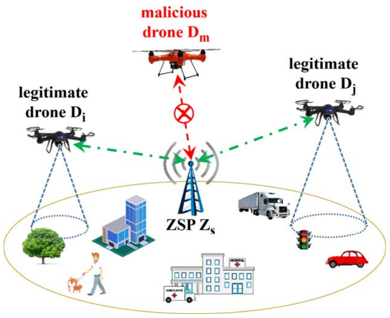


Fig. 1. System model where a set of drones are deployed to collect and/or deliver data to nearby ZSP over an insecure wireless medium.

D. Security Requirements of liteCrypto

We outline the following security requirements to be met by *liteCrypto* according to [7], [22].

- Authentication: *liteCrypto* shall ensure that the identity of drones and ZSP can be verified and the adversary cannot masquerade as any legitimate entity.
- Integrity: *liteCrypto* shall guarantee that the content of messages can be validated by the receiver. And the adversary cannot manipulate messages.
- Confidentiality: *liteCrypto* shall assure that sensitive data are transmitted in the encrypted format after the session key is established.
- Anonymity: *liteCrypto* should guarantee that the pseudonym of drone, rather than the real identity, is transmitted in the message.
- Session Key Agreement: *liteCrypto* shall assure that a secure session key can be established between a drone and the ZSP for subsequent communications after mutual authentication.
- Secure Against Various Attacks: *liteCrypto* should be secure against diverse security attacks including ZSP spoofing attack, drone impersonation attack, drone capture attack, message modification attack, replay attack, and man-in-the-middle attack.

IV. *liteCrypto*: Lightweight Authentication Protocol

Considering the scenario that drone D_i collects the data of interest and delivers them to ZSP Z_s. Due to insecure wireless channels, drone D_i and ZSP Z_s first have to achieve

TABLE I
NOTATIONS

Notation	Meaning
Z_s	ZSP Z_s
ZID_s	ZSP Z_s's identity
D_i	Drone D_i
RID_i	Drone D_i's real identity
PID_i	Drone D_i's pseudonym
ts	Timestamp
$F_{puf}(\cdot)$	PUF
che_i	Drone D_i's PUF challenge
res_i	Drone D_i's PUF response
(che_i, res_i)	Drone D_i's PUF CRP
$F_{mac}(\cdot)$	Message authentication code (MAC) function
$H(\cdot)$	Secure hash function, H:{0,1}^{m} and $m \in \mathbb{Z}$
$H_a(\cdot)$	Secure hash function, H_a:{0,1}*→$\mathbb{G}$
$H_b(\cdot)$	Secure hash function, H_b:{0,1}*→$\mathbb{Z}$
$\|$	Concatenation operation
M_i	Message i
MAC_i	MAC of M_i
Sig_i	Signature of M_i
$\mathbb{G}$	Cyclic additive group
P	An arbitrary generator of $\mathbb{G}$
$\mathbb{G}_T$	Cyclic multiplicative group
n	The order of $\mathbb{G}$ and $\mathbb{G}_T$
$\hat{e}$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$	A bilinear pairing map on ($\mathbb{G}$, $\mathbb{G}_T$)
s	A secret random number
SAC	Successful authentication code
$Key_{i,s}$ or $Key_{s,i}$	Session key between drone D_i and ZSP Z_s

mutual authentication, and then establish a secure communication session through *liteCrypto*. *liteCrypto* consists of three phases: system initialization phase, drone registration phase, and mutual authentication and key agreement phase. Table I lists all notations used in this paper.

A. System Initialization Phase

In this phase, ZSP Z_s initializes the system through generating pubic parameters in the following steps:

1) Z_s chooses a cyclic additive group $\mathbb{G}$ of order n with an arbitrary generator P.
2) Z_s chooses a cyclic multiplicative group $\mathbb{G}_T$ of the same order n.
3) Z_s generates a bilinear pairing map on ($\mathbb{G}$, $\mathbb{G}_T$), $\hat{e}$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.
4) Z_s chooses two one-way secure hash functions H_a and H_b, where H_a:{0,1}*→$\mathbb{G}$ and H_b:{0,1}*→$\mathbb{Z}$.
5) Z_s advertises all public system parameters as $\{\mathbb{G}, \mathbb{G}_T, n, P, H_a, H_b\}$.

B. Drone Registration Phase

In this phase, drone D_i registers itself to ZSP Z_s according to the following steps:

1) D_i randomly selects its real identity RID_i and feeds its PUF challenge che_i into $F_{puf}(\cdot)$ to compute the corresponding response res_i = $F_{puf}(che_i)$.
2) D_i generates a secret random number s using its response res_i, s = $H_b(res_i)$, and computes sP.
3) D_i calculates its pseudonym PID_i = $H(RID_i \parallel sP)$, where H:{0,1}^{m} is a set of fixed length (saying m bits) strings.

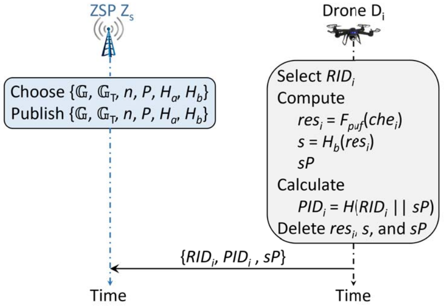


Fig. 2. System initialization phase and drone registration phase.

4) D_i shares (RID_i, PID_i, sP) with Z_s via a secure channel.
5) D_i stores its RID_i, PID_i, and che_i but res_i, s, and sP in the memory.

The steps of system initialization phase and drone registration phase are shown in Fig. 2.

C. Mutual Authentication and Key Agreement Phase

In this phase, drone D_i and ZSP Z_s authenticate each other and establish a secure session key for subsequent communications in a secure manner. First, drone D_i initiates the process of mutual authentication and key agreement by following the steps specified below:

1) D_i computes its response res_i = $F_{puf}(che_i)$ and generates the secret random number s = $H_b(res_i)$.
2) D_i calculates $H_a(PID_i)$ and $sH_a(PID_i)$ as its public key and private key, respectively.
3) D_i creates the message M_1 = $\{PID_i \parallel ZID_s \parallel ts\}$ and calculates the corresponding MAC as MAC_1 = $F_{mac}(M_1)$.
4) With message M_1 and private key $sH_a(PID_i)$, D_i computes the digital signature Sig_1 = $H_b(M_1) \cdot sH_a(PID_i)$.
5) D_i sends $\{M_1, MAC_1, Sig_1\}$ to Z_s.
6) D_i calculates the session key $Key_{i,s}$ = $\hat{e}(sH_a(PID_i), P)$.

Upon receipt of $\{M_1, MAC_1, Sig_1\}$, ZSP Z_s performs the following operations:

1) Z_s calculates $MAC_1^{'}$ = $F_{mac}(M_1)$ and verifies whether $MAC_1^{'} \overset{?}{=} MAC_1$. If $MAC_1^{'}$ = MAC_1, the message verification succeeds and Z_s proceeds to the next step. Otherwise, Z_s rejects the authentication request.
2) Z_s checks whether the signature Sig_1 is valid through verifying $\hat{e}(Sig_1, P) \overset{?}{=} \hat{e}(H_b(M_1) \cdot H_a(PID_i), sP)$.
$$\begin{aligned}\hat{e}(Sig_1, P) &= \hat{e}(H_b(M_1) \cdot sH_a(PID_i), P)\\ &= \hat{e}(H_b(M_1) \cdot H_a(PID_i), sP).\end{aligned}$$
If Sig_1 passes the above verification, Z_s proceeds to the next step. Otherwise, Z_s rejects the authentication request.
3) Z_s calculates the session key $Key_{s,i}$ = $\hat{e}(H_a(PID_i), sP)$. Please note that $Key_{s,i}$ is equal to $Key_{i,s}$ because
$$\begin{aligned}\hat{e}(H_a(PID_i), sP) &= \hat{e}(H_a(PID_i), P)^s\\ &= \hat{e}(sH_a(PID_i), P).\end{aligned}$$

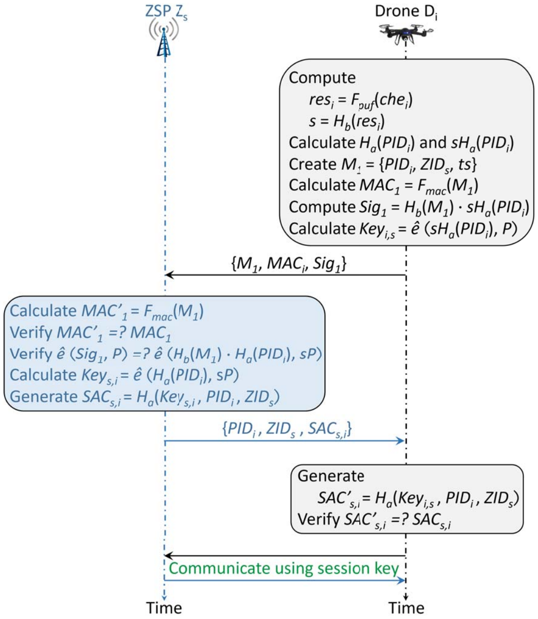

ZSP Z_s
Drone D_i
Compute
$res_i = F_{puf}(che_i)$
$s = H_b(res_i)$
Calculate $H_a(PID_i)$ and $sH_a(PID_i)$
Create $M_1 = \{PID_i, ZID_s, ts\}$
Calculate $MAC_1 = F_{mac}(M_1)$
Compute $Sig_1 = H_b(M_1) \cdot sH_a(PID_i)$
Calculate $Key_{i,s} = \hat{e}\,(sH_a(PID_i), P)$
$\{M_1, MAC_i, Sig_1\}$
Calculate $MAC'_1 = F_{mac}(M_1)$
Verify $MAC'_1 =?\ MAC_1$
Verify $\hat{e}\,(Sig_1, P) =?\ \hat{e}\,(H_b(M_1) \cdot H_a(PID_i), sP)$
Calculate $Key_{s,i} = \hat{e}\,(H_a(PID_i), sP)$
Generate $SAC_{s,i} = H_a(Key_{s,i}, PID_i, ZID_s)$
$\{PID_i, ZID_s, SAC_{s,i}\}$
Generate
$SAC'_{s,i} = H_a(Key_{i,s}, PID_i, ZID_s)$
Verify $SAC'_{s,i} =?\ SAC_{s,i}$
Communicate using session key
Time
Time

Fig. 3. Mutual authentication and key agreement phase.

4) Z_s generates a successful authentication code $SAC_{s,i} = H_b(Key_{s,i} \parallel PID_i \parallel ZID_s)$ and sends $\{PID_i, ZID_s, SAC_{s,i}\}$ to D_i.

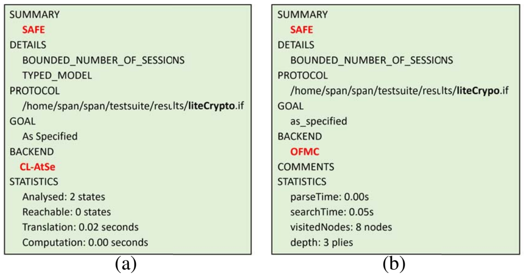
After receiving $\{PID_i, ZID_s, SAC_{s,i}\}$ from ZSP Z_s, drone D_i generates its own successful verification code $SAC_{i,s}$ using $Key_{i,s}$, PID_i, and ZID_s, and then checks whether $SAC_{i,s} \overset{?}{=} SAC_{s,i}$. If $SAC_{i,s}$ and $SAC_{s,i}$ match, ZSP Z_s is believed to be legitimate and drone D_i can use the session key $Key_{i,s}$ to communicate with ZSP Z_s confidently. Otherwise, drone D_i discards the message and refuses any further communications with ZSP Z_s. The steps of mutual authentication and key agreement phase are shown in Fig. 3.

V. Security Verification and Analysis

A. Security Verification Using AVISPA

We verify *liteCrypto* using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [11]. AVISPA is a security verification tool, where the security protocol and its properties can be easily defined and implemented as a security problem using a modular and expressive formal language, which is known as High-Level Protocol Specification Language (HLPSL).

In AVISPA, users are provided with two back-ends, Constraint-Logic-based Attack Searcher (CL-AtSe) and On-the-fly Model-Checker (OFMC), for security verification. CL-AtSe is used to uncover potential vulnerabilities of replay attack and man-in-the-middle attack in the protocol through evaluating a set of constraints. OFMC can demonstrate whether the protocol is secure via a bounded number of sessions. We first install Ubuntu 10.04 in Virtual Box [23], and then set up and configure SPAN + AVISPA [24] environment. In Fig. 4, we present the results of security verification of *liteCrypto* in both CL-AtSe and OFMC back-ends. The results of security verification have demonstrated that *liteCrypto* is secure against replay attack and man-in-the-middle attack.



```
SUMMARY
   SAFE
DETAILS
   BOUNDED_NUMBER_OF_SESSIONS
   TYPED_MODEL
PROTOCOL
   /home/span/span/testsuite/results/liteCrypto.if
GOAL
   As Specified
BACKEND
  CL-AtSe
STATISTICS
   Analysed: 2 states
   Reachable: 0 states
   Translation: 0.02 seconds
   Computation: 0.00 seconds
```

(a)

```
SUMMARY
   SAFE
DETAILS
   BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
   /home/span/span/testsuite/results/liteCrypo.if
GOAL
   as_specified
BACKEND
   OFMC
COMMENTS
STATISTICS
   parseTime: 0.00s
   searchTime: 0.05s
   visitedNodes: 8 nodes
   depth: 3 plies
```

(b)

Fig. 4. Security verification results using CL-AtSe and OFMC in AVISPA.

B. Analysis of Security Requirements

First of all, we exhibit that *liteCrypto* satisfies the basic security objectives specified in Section III.D. *liteCrypto* can achieve mutual authentication between communication entities in the IoD environment. This is because the drone and the ZSP verify each other's identity. Therefore, *liteCrypto* can achieve mutual authentication. *liteCrypto* can achieve integrity so that the source of messages and the content of messages can be verified by the receiving entity. This is because a message authentication code (MAC) is generated using the one-way hash function for each communication message. As a result, integrity can be achieved by *liteCrypto*. Since each critical communication message is encrypted using the established secure session key between the drone and the ZSP, thus, *liteCrypto* can guarantee confidentiality. *liteCrypto* can support anonymous communication in the IoD environment. This is because the real identity of drone is not transmitted directly in plaintext, but in the pseudonym format. Therefore, *liteCrypto* can achieve anonymity. *liteCrypto* can achieve session key agreement between communication entities in the IoD environment. This is because the drone and the ZSP will verify each other's identity and then compute the secure session key and use it for future communications. Therefore, *liteCrypto* can achieve session key agreement.

In the following, *liteCrypto* is analyzed to show that it can defend against various security attacks. First, *liteCrypto* is secure against physical capture attack. The adversary might obtain information stored in drone's memory such as RID_i, PID_i, and che_i through power analysis. However, they cannot retrieve critical information, i.e., res_i, s, and sP, from drone's integrated circuit. This is because any probing attempt to the integrated circuit of drone will destroy drone's PUF, which is unable to reproduce the same res_i. Second, *liteCrypto* is immune to reply attack because the timestamp ts is piggybacked in the message. The ZSP can easily check ts and discard any replayed messages. Third, *liteCrypto* is resilient to drone impersonation attack. Since each PUF is unique and will produce totally different responses with the same challenge, it is impossible for an adversary to generate a valid CRP (che_i, res_i) of legitimate drone. As a result, an adversary cannot illegally impersonate any legitimate drone. Fourth, message modification attack does not pose any threat to *liteCrypto*, because each communication entity can easily

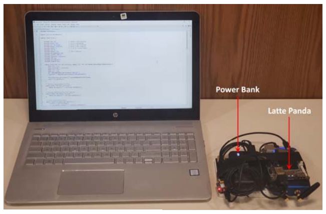


Fig. 5. Real-world testbed: HP ENVY Notebook laptop and Latte Panda development board.

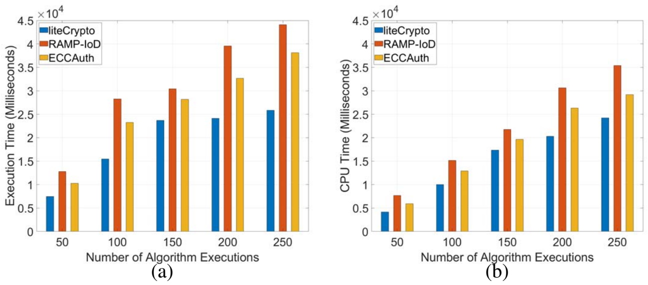
detect any message modification through the verification of MAC and SAC. Finally, *liteCrypto* is secure against man-in-the-middle attack. Since a drone and the ZSP can mutually authenticate each other and establish a secure session key using timestamp ts and secret information sP, thus the adversary cannot secretly relay and possibly alter the messages being transmitted between the drone and the ZSP. In summary, *liteCrypto* can satisfy all required security requirements and is immune against physical capture attack, reply attack, drone impersonation attack, message modification attack, as well as man-in-the-middle attack.

VI. Performance Evaluation

A. Experimental Testbed and Benchmark Schemes

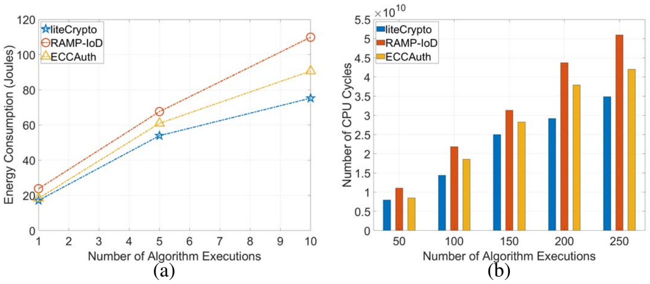
We conduct extensive experiments on a real-world testbed, which is composed of one Latte Panda development board [25] and one HP ENVY Notebook laptop [26]. The operations of Latte Panda development board is supported by the attached power bank. Moreover, Windows 10 operating system (OS) is installed on Latte Panda development board where the central processing unit (CPU) is Intel Cherry Trail Z8350 (2M cache, 1.92 GHz) and the size of random-access memory (RAM) is 4GB. The HP ENVY Notebook laptop has Windows 10 Pro OS (64-bit) and 7th Generation Intel Core i7-7500U CPU (4M Cache, 3.5 GHz). We show the developed real-world testbed in Fig. 5, where the ZSP and the drone are simulated by the laptop and the Latte Panda development board, respectively. Finally, we set up an Eclipse environment [27] on the testbed, where we implement *liteCrypto* and benchmark schemes.

We choose two benchmark schemes, i.e., ECCAuth [12] and RAMP-IoD [13], and implement them to work on the testbed for performance comparison and analysis. We measure the performance in terms of execution time, CPU time, CPU cycles, energy consumption, as well as communication cost for *liteCrypto*, ECCAuth, as well as RAMP-IoD. Execution time is the amount of time which is measured from when the algorithm starts running to when the algorithm stops running. We measure CPU time as a time period required by CPU to finish all instructions of the algorithm. Energy consumption is the amount of energy consumed to finish the running of all algorithm's operations. The number of messages and the energy consumption of communication are selected to represent communication cost. The number of exchanged messages is counted directly for *liteCrypto*, ECCAuth, and RAMP-IoD. The energy consumption of communication is calculated based on the number of exchanged messages [28]. Moreover, execution time, CPU time, CPU cycles, and energy consumption are directly measured through VisualVM [29].



(a) (b)

Fig. 6. The performance of execution time and CPU time against the number of algorithm executions.



(a) (b)

Fig. 7. The performance of energy consumption and CPU cycles against the number of algorithm executions.

B. Experimental Results and Analysis

First, the execution time and CPU time of *liteCrypto*, ECCAuth, and RAMP-IoD are measured and the corresponding results are presented in Fig. 6. Please note that the major difference between execution time and CPU time is that the execution time includes the time elapsed during waiting for I/O operations or entering idle mode. As shown in Fig. 6(a), the execution time of all three schemes increase linearly as the number of algorithm executions increases. This is because when the algorithm is repeatedly executed, a longer execution time will be observed. No matter how many times (from 50 to 250) the algorithm is executed, our approach *liteCrypto* always outperforms ECCAuth and RAMP-IoD. The rationale is that our approach *liteCrypto* adopts lightweight cryptographic operations such as bilinear pairing, physical unclonable function, as well as hash function. As a result, *liteCrypto* can run fast and the lowest execution time is obtained. RAMP-IoD shows the highest execution time because compute-intensive techniques such as AEGIS and fuzzy extractor are being used to achieve mutual authentication and session key agreement. In Fig. 6(b), the overall CPU time of *liteCrypto*, ECCAuth, and RAMP-IoD increase steadily as the number of algorithm executions increases from 50 to 250. However, *liteCrypto* still shows the best performance.

Second, we measure energy consumption and CPU cycles with varying number of algorithm executions and present the experimental results in Fig. 7. In Fig. 7(a), we observe the energy consumption of running *liteCrypto*, ECCAuth, and RAMP-IoD. The more complex the algorithm is, the more energy it will consume. Since RAMP-IoD adopts the most complex techniques and includes password/biometric change, revocation, and dynamic drone deployment phases, RAMP-IoD is the largest consumer of energy power. In our approach

TABLE II
COMPARISON OF COMMUNICATION OVERHEAD

Metrics	*liteCrypto*	ECCAuth	RAMP-IoD
Number of Messages	3	3	6
Energy Consumption (Joule)	3.38×10^{-4}	3.38×10^{-4}	6.76×10^{-4}

liteCrypto, relatively lightweight techniques such as bilinear pairing, physical unclonable function, as well as hash function are utilized to realize the goal of authentication and key agreement. Thus, a smaller energy consumption is observed by *liteCrypto*. As shown in Fig. 7(b), when the number of algorithm executions increases, the CPU cycles of *liteCrypto*, ECCAuth, and RAMP-IoD also increase. This is because a larger number of algorithm executions will require more operations to run each algorithm repeatedly. As a consequence, more CPU cycles will be required for the execution of algorithm. However, our approach *liteCrypto* still delivers the least number of CPU cycles. Compared to ECCAuth and RAMP-IoD, *liteCrypto* requires a less number of operations to achieve mutual authentication and session key agreement, thus, the least number of CPU cycles belongs to *liteCrypto*.

Third, we observe the communication overhead of *liteCrypto*, ECCAuth, and RAMP-IoD in Table. II. Our approach *liteCrypto* and ECCAuth require the same number of messages to be exchanged to achieve mutual authentication and key agreement. However, RAMP-IoD will need to exchange six messages to complete all seven phases. Moreover, the energy consumption of communication for *liteCrypto*, ECCAuth, and RAMP-IoD is 3.38×10^{-4}, 3.38×10^{-4}, 6.76×10^{-4}, respectively, indicating *liteCrypto* is an energy efficient algorithm.

VII. CONCLUSION

In this paper, we proposed a lightweight authentication protocol, also called *liteCrypto*, based on bilinear pairing and physical unclonable function for the IoD environment, where a drone and the ZSP mutually authenticate each other and establish a secure session key based on bilinear pairing and physical unclonable function before sharing any critical information over an insecure wireless channel. To prove that *liteCrypto* is a secure protocol and can defend against various security attacks, we first verified *liteCrypto* using AVISPA tool, and then presented a security analysis. According to security verification and analysis results, it had been proved that *liteCrypto* is a secure protocol and can successfully shield IoD systems from various security attacks. Moreover, we conducted extensive experimental evaluation on the real-world testbed, and measured the performance of *liteCrypto* and other two benchmark schemes in terms of computational overhead, energy consumption, as well as communication cost. Based on the experimental results, we concluded that *liteCrypto* provides superior performance than its counterparts while satisfying all security requirements.

REFERENCES

[1] C. Pu and L. Carpenter, "Psched: A Priority-Based Service Scheduling Scheme for the Internet of Drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4230–4239, 2021.

[2] L. Schroth, *The Drone Market Size 2020-2025*, Last accessed: April 07, 2022, https://droneii.com/the-drone-market-size-2020-2025-5-key-takeaways.

[3] C. Pu, A. Wall, I. Ahmed, and K. Choo, "SecureIoD: A Secure Data Collection and Storage Mechanism for Internet of Drones," in *Proc. IEEE MDM*, 2022, pp. 83–92.

[4] A. Çalhan and M. Cicioğlu, "Drone-assisted smart data gathering for pandemic situations," *Elsevier Computers & Electrical Engineering*, vol. 98, p. 107769, 2022.

[5] 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.

[6] *It is Easy to Hack a Drone and Crash it*, Last accessed: April 06, 2022, https://www.itvscience.com/easy-hack-drone-crash/.

[7] W. Stallings, *Cryptography and Network Security: Principles and Practice, 8th Edition*. Pearson, 2020.

[8] M. Ozmen and A. Yavuz, "Dronecrypt - An Efficient Cryptographic Framework for Small Aerial Drones," in *Proc. IEEE MILCOM*, 2018, pp. 1–6.

[9] S. Chaudhry, J. Nebhen, A. Irshad, A. Bashir, R. Kharel, K. Yu, and Y. Zikria, "A Physical Capture Resistant Authentication Scheme for the Internet of Drones," *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 62–67, 2021.

[10] Y. Chevalier, L. Compagna, J. Cuellar, P. Drielsma, J. Mantovani, S. Mödersheim, and L. Vigneron, "A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols," in *Proc. SAPS*, 2004, pp. 1–13.

[11] *Automated Validation of Internet Security Protocols and Applications*, Last accessed: Jan 07, 2022, http://www.avispa-project.org.

[12] S. Hussain, S. Chaudhry, O. Alomari, M. Alsharif, M. Khan, and N. Kumar, "Amassing the Security: An ECC-Based Authentication Scheme for Internet of Drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4431–4438, 2021.

[13] M. Tanveer, A. K. N. Kumar, and M. Hassan, "RAMP-IoD: A Robust Authenticated Key Management Protocol for the Internet of Drones," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1339–1353, 2022.

[14] N. Zhang, Q. Jiang, L. Li, X. Ma, and J. Ma, "An efficient three-factor remote user authentication protocol based on BPV-FourQ for internet of drones," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 3319–3332, 2021.

[15] Y. Lei, L. Zeng, Y. Li, M. Wang, and H. Qin, "A Lightweight Authentication Protocol for UAV Networks Based on Security and Computational Resource Optimization," *IEEE Access*, vol. 9, pp. 53 769–53 785, 2021.

[16] B. Bera, S. Saha, A. 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 Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9097–9111, 2020.

[17] S. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A. Bashir, and Y. Zikria, "GCACS-IoD: A certificate based generic access control scheme for Internet of drones," *Computer Networks*, vol. 191, p. 107999, 2021.

[18] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K. Choo, "Blockchain-based Cross-domain Authentication for Intelligent 5G-enabled Internet of Drones," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 6224–6238, 2022.

[19] K. Chuang, E. Bury, R. Degraeve, B. Kaczer, D. Linten, and I. Verbauwhede, "A Physically Unclonable Function Using Soft Oxide Breakdown Featuring 0% Native BER and 51.8 fJ/bit in 40-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2765–2776, 2019.

[20] C. Pu, H. Zerkle, A. Wall, S. Lim, K. Choo, and I. Ahmed, "A Lightweight and Anonymous Authentication and Key Agreement Protocol for Wireless Body Area Networks," *IEEE Internet Things J.*, pp. 1–1, 2022.

[21] *Dronebuster*, Last accessed: Jan 07, 2022, http://flexforce.us/product/dronebuster/.

[22] C. Pu, A. Wall, K. Choo, I. Ahmed, and S. Lim, "A Lightweight and Privacy-Preserving Mutual Authentication and Key Agreement Protocol for Internet of Drones Environment," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9918–9933, 2022.

[23] *VirtualBox*, Last accessed: Jan 07, 2022, https://www.virtualbox.org/.

[24] *SPAN*, Last accessed: Jan 07, 2022, http://www.avispa-project.org/.

[25] *Latte Panda*, https://www.lattepanda.com/.

[26] *Laptop Computers*, https://www.hp.com/us-en/shop/cat/laptops.

[27] *Eclipse*, https://www.eclipse.org/downloads/.

[28] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal*, vol. 12, no. 1, pp. 834–842, 2018.

[29] *VisualVM*, Last accessed: Jan 07, 2022, https://visualvm.github.io/.