A Featherweight Authentication and Key Agreement Scheme for Internet of Drones Applications

Cong Pu
Department of Computer Science, Oklahoma State University, Stillwater, OK, United States
cong.pu@ieee.org

Abstract*—The Internet of Drones (IoD) will have revolutionized civil and commercial applications, in much the similar way that the Internet of Things (IoT) transformed the way information is exchanged with other devices and systems over the Internet. The drones are generally considered to have constrained resources, which make them less compatible with complicated algorithms and more prone to attacks. Moreover, the IoD applications are facing information security and privacy challenges in the cyber-threat environment, where the adversary could intercept communicating messages and compromise their confidentiality or integrity. Consequently, the security protocols which are designed for IoD applications should not only provide desirable security guarantees, but also be resource-efficient. Existing authenticated key exchange protocols can authenticate the identities of communication parties and realize the exchange of session key, however, they either incur high communication overhead, suffer from non-negligible computational cost, or have inherent security design flaws. Thus, these approaches are not suitable for resource-constrained drones involved in critical IoD applications. To address the above challenges, this paper presents a featherweight authentication and key agreement scheme (hereafter referred to as *fwAKA*) for IoD applications based on elliptic curve cryptography, physical unclonable function, hash function, and XOR operation. The *fwAKA* only requires two handshakes to achieve authenticated key agreement. We prove that the *fwAKA* is perfectly secure in the adversarial setting through the security verification using the AVISPA. We set up a simulation environment, implement the *fwAKA* and its counterparts, and conduct performance evaluation in terms of communication overhead and running time. Experimental results indicate that not only is the *fwAKA* robust against well-known attacks but also it is more resource-efficient than its opponents.

***Index Terms*—Security and Privacy, Mutual Authentication, Session Key Agreement, Featherweight, Internet of Drones**

I. Introduction

The drone market has left the nascent stage and broken into the mainstream. No one can deny the fact that the drone industry has seen a spike in market growth in the last couple of years, and the global drone market is estimated to surpass USD 63.5 billion by 2025 [1]. In order to enable heterogeneous drones to autonomously connect over the Internet, the Internet of Drones (IoD) paradigm [2] has been created to generate a network of interconnections among drones as well as ground base stations. The IoD paradigm is widely envisioned as the enabling framework that supports various emerging and potential applications such as target tracking [3], parcel delivery [4], precision agriculture [5], etc. In the era of pervasive intelligence, we envision that the IoD technology will have the potential to transform every aspect of our lives for good within a very short period [6].

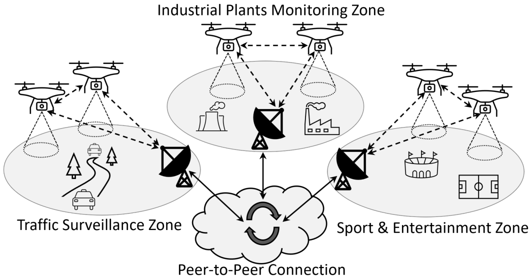

Industrial Plants Monitoring Zone
Traffic Surveillance Zone
Sport & Entertainment Zone
Peer-to-Peer Connection

Fig. 1. IoD architecture and example applications: traffic surveillance zone; sport & entertainment zone; and industrial plants monitoring zone.

In the IoD paradigm, by equipping with sensing devices, processing module, and storage system, each drone can be turned into an aerial smart object which is able to communicate with nearby drones or ground base station through wireless communication [7]. As depicted in Fig. 1, the geographic area along with its airspace is partitioned into different task zones. One or more ground base stations are deployed in each task zone, where the communication can be realized between drones and ground base stations to enable information gathering, sharing, and processing. The adoption of IoD framework in commercial and civil applications has fueled the global research and development activities on integrating drones into existing communication infrastructures and networks. In addition, the Third Generation Partnership Project (3GPP) and The Institute of Electrical and Electronics Engineers (IEEE) have been actively devoted to the establishment of aerial communication standards. For instance, the IEEE standards committees have started efforts to provide standards for self-organized aerial networks as well as the traffic management and application of low-altitude drones [8].

Although the IoD applications have significant potentials to bring economic and social benefits to the citizens, businesses, and governments, the security of the communication between the drone and the ground base station remains to be tackled before the wide adoption of IoD paradigm. The authentication between the drone and the ground base station along with the establishment of session key are essential security measures for protecting the follow-up communications. However, designing a secure, effective, and efficient authentication and key agreement protocol remains a challenging issue. First, the authentication and key agreement protocol should be secure against well-known cyber attacks and does not have any

978-1-6654-6483-3/23/$31.00 © 2023 IEEE

security design flaws. Unfortunately, the existing scheme has inherent vulnerabilities which can be exploited by the adversary to compromise the communication [9]. Second, the drones are generally constrained in terms of storage space, battery lifetime, and computing power, as a result, the authentication and key agreement protocol should be designed with resource-friendly operations. Nonetheless, some approaches just adopt the opposite design strategy. For example, FourQ and Boyko-Peinado-Venkatesan pre-calculation techniques are chosen to realize the authentication among the communication entities in the IoD [10], where the pre-calculation algorithm requires auxiliary storage space. Third, the whole process of authenticated key exchange should be fast and communicationally efficient. But, some emerging approaches require the communication entity to query the blockchain server to retrieve the authentication parameters, which incurs arbitrarily long delays in communication [11]. In addition, many authentication and key agreement protocols demand three [12], [13] or four [14] handshakes between the drone and the ground base station, which causes a high communication overhead.

From the above discussion and analysis, it is clear that the state-of-the-art authenticated key exchange protocols fail to guarantee security as well as efficiency. As a result, the full potential of IoD paradigm cannot be fully exploited in the cyber-threat environment if the defective and inefficient security protocols are adopted. Thus, what has been lacking in the current theory is a secure and featherweight security protocol that adopts resource-friendly computing operations to achieve the efficiency, security, and privacy requirements of IoD communications. The realization of such a novel security protocol would be unprecedented because the similar technique is not currently available in the IoD community, and the proposed work will fill this research gap. In this paper, we present a novel authentication and key agreement protocol to address the abovementioned challenging issues in the IoD environment. In summary, our major contribution is briefly summarized in the following:

- We propose a featherweight authentication and key agreement scheme (hereafter referred to as *fwAKA*) for IoD applications based on elliptic curve cryptography, physical unclonable function, hash function, and XOR operation. The *fwAKA* only requires two handshakes to achieve authenticated key agreement.
- We verify and prove the security of *fwAKA* through the security verification using the AVISPA [15]. The *fwAKA* is safe from well-known cyber attacks and can guarantee the secrecy of exchanged critical information.
- We build a simulation environment for experimental study. For performance comparison, we select *modAKA* [12] and *SecAuth* [16], implement them along with the *fwAKA*, and conduct experiments in terms of communication overhead and running time. The *fwAKA* outperforms its counterparts.

II. Related Work

The authors in [17] adopt the blockchain technique and develop a blockchain-assisted authentication and key agreement protocol for Internet of Drones (IoD) systems. In their approach, the drone and the user of drone are regarded as one entity in the system, which is called the remote user. First, the remote user goes through the registration process with the ground station and obtains the secret information which will be stored in its storage unit. Then, the remote user authenticates itself with the ground station through submitting an authentication request. During the following mutual authentication phase, the pre-negotiated secret information will be used for the remote user and the ground station to verify each other's identities and establish initial trust. However, the major drawback in their approach is that the vital secret information is being directly stored in the storage unit of remote user (e.g., drone), which significantly increases the risks of cryptographic security parameters being compromised. This is because an adversary may physically capture the drone and attempt to probe its storage unit to extract secret information. In [12], there are three distinct IoD entities, control server, user, and drone. First, the control server produces the master key and publishes a set of system parameters for the use of other entities. Second, the user and the drone register themselves with the control server and obtain their pseudonym and secret value through a secure channel. Finally, the control server serves as an intermediary for the drone and the user to establish the session key for the following-up communication. Actually, the above security protocol is mainly designed based on the existing work [18] with minor extension, where a large number of hash and XOR operations are needed for the authentication and key establishment purposes. Another problem is that the drone and the user have to explicitly store their pseudonym and secret value in the memory, which makes the security protocol prone to physical attack.

The authors in [14] investigate aerial networks and propose an authentication and key agreement scheme with the assistance of cloud computing and blockchain techniques. In their approach, the drone and the user will authenticate each other through a sequence of hash and XOR operations. Moreover, the ground station is connected to the cloud server and is assumed to have infinite computing resources, thus, it is responsible for generating public key cryptographic parameters to interact with the blockchain and use them as communication credentials. Nonetheless, the frequent communications between the ground station and the cloud server/blockchain will incur arbitrarily long communication delays, which is not suitable for time-sensitive IoD applications. A symmetric key authentication protocol is designed for industrial Internet of Things in [19], where the authors use hash function and XOR operation to realize mutual authentication, key agreement, and message integrity. In order to establish a secure session key between the user and the industrial IoT device, however, three messages which are built through a number of hash and XOR operations are required. In [20], the authors study the security and privacy issues in the vehicular ad hoc networks and propose a physical unclonable function (PUF) based authentication protocol. The rationale behind the usage of PUF is to protect the secrets of vehicles from being physically extracted by the adversary. The authors also mention that their approach is different from the state-of-the-art because the trusted authority is not required in the process of mutual authentication between the vehicles and the road side units. In the harsh environment, however, the reliability of PUF is in doubt because the PUF might not re-generate the same response with the same challenge. Thus, the pre-negotiated secret information based on the PUF response cannot be restored and the entire authentication process will fail.

III. System Model, Adversarial Model, and Security Requirements

A. System Model

In the system, there are two major participants: the drone and the ground base station, as shown in Fig. 2. The drone is regarded as an untrusted participant and deployed to collect information in the task zone. Since the drone might be flying in low altitude, there is a chance for the adversary to capture it with the specific equipment and extract critical information cached in the storage unit. Thus, in order to eliminate the risk of physical probing attack, the integrated circuit of drone is built with physical unclonable functions (PUF) primitive [21], where the vital cryptographic information (e.g., pseudonym and secret value) is dynamically calculated with PUF. The ground base station is considered as a trusted participant, and its major responsibilities include registering drones with the system and collect drones' observational data.

B. Adversarial Model

The adversary is assumed to have the abilities specified in the well-known Dolev-Yao threat model [22], where the wireless channel is regarded as unsecure and the adversary is able to eavesdrop the communication between the drone and the ground base station. In order to impersonate a legitimate drone, the adversary might try to capture a drone and extract its legal identity and/or cryptographic information. However, the adversary's attempt might not succeed because the drone does not store the identity and cryptographic information directly in the storage unit, but dynamically calculating those information. In addition, the physical probing attack will change or even destroy the PUF mapping which causes the same response not being re-generated with the same challenge. Thus, it is reasonably to assume that the internal adversary does not exist in the system. In short, the primary goal of the adversary is to obtain the secret key and compromise the drones' observational data. Other attacks such as denial-of-service or jamming attacks can also be launched by the adversary, however, they are outside the scope of this paper.

C. Security and Performance Requirements

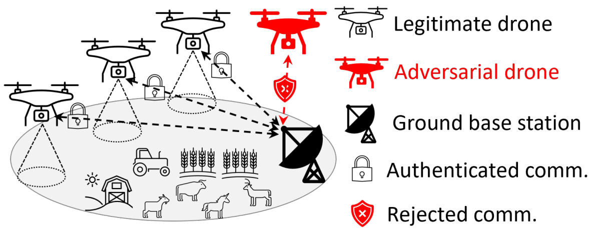


Fig. 2. System model.

In this paper, we require the proposed protocol to meet the following security and performance objectives: (i) Authentication: The drone and the ground base station will verify each other's identities before establishing the session key; (ii) Confidentiality: The messages being exchanged between the drone and the ground base station is not intelligible to the adversary; (iii) Integrity: The drone and the ground base station can verify whether the received message has been altered; (iv) Anonymity: The drone will hide its real identity and use the pseudonym for the communication with the ground base station; (v) Session Key Establishment: A secret session key should be established between the drone and the ground base station after mutual authentication; and (vi) Computational Complexity: The computational complexity of the proposed protocol is lower than existing schemes.

IV. The Proposed Protocol

A. Physical Unclonable Function

Physical unclonable functions, or PUFs, have been widely used as one of effective hardware-specific security primitives because of its non-clonability. In the experimental study, the PUF is usually simulated as one-way function, which is provided with an input, termed *challenge*, and generates an output, named *response*. The challenge together with its corresponding response is called challenge-response pair, or just CRP. For the same PUF, when we provide the identical challenge, the same response can be expected. However, if a minor change is made in the challenge, we can expect a totally distinct response generated by the same PUF. As mentioned before, the PUF can be represented as a mathematical function [23]. Thus, we denote the PUF as $res = F_{puf}(che)$, where *che* and *res* indicate PUF's challenge and response, respectively.

In the current state-of-the-art of research, the PUF becomes a popular technique to generate cryptography-related information. However, the PUF itself becomes very unstable in the hash environments, where there is no guarantee that the same challenge will make the PUF output the same response. It is widely considered as one disadvantage of the PUF. In order to make the PUF stably work in the severe circumstance, we develop an error correction code and a fuzzy extractor to integrate with the PUF. First, an algorithm, called *rGen*, is created to generate the response. The *rGen* algorithm, as shown in Algorithm 1, is designed to produce a set $\{$*res*, $S\}$. Here, *res* is the CRP response, which is the value to be regenerated by the PUF. S is a helper string which is fed into the PUF to regenerate the CRP response *res*. The error correction code [24] is adopted to eliminate up to x bit errors in the CRP response *res*. Second, we design an algorithm $rRes$ to restore the same response, where the major operations are shown in Algorithm 2. With the $rRes$ algorithm, the PUF is able to regenerate *res* with the helper string S and the error decoding algorithm D_{er}, even if the PUF produces an output $O^{'}$ that differs from the original output O by at most x bits.

Algorithm 1: Response Generation Algorithm *rGen*

```
Input: Modulus n; Challenge che
Function rGen(n, che):
    /* $\xleftarrow{\circledast}$ denotes sampling */
    /* $\oplus$ denotes exclusive OR function */
    /* $\mathbb{Z}_n$ denotes the set of remainders in
       arithmetic modulo n */
    $O = F_{puf}(che)$;
    $res \xleftarrow{\circledast} \mathbb{Z}_n$;
    $S = O \oplus ECC(res)$;
    return $\{res, S\}$;
```

Algorithm 2: Response Restore Algorithm *rRes*

```
Input: Challenge che; Helper string S
Function rRes(che, S):
    $O' = F_{puf}(che)$;
    $res = D_{er}(S \oplus O')$;
    return res;
```

B. The Proposed fwAKA Protocol

We assume that a drone ID_i is deployed to collect data in the task zone, and then submits the observational data to a nearby ground base station B_m. Since the observational data is transmitted over public wireless communication medium, thus, the drone ID_i and the ground base station B_m need to verify each other's identities and establish a secure session key for the encryption of observational data. In summary, the *fwAKA* is composed of three phases: (i) system initialization; (ii) drone registration; and (iii) authentication and key establishment.

1) System Initialization: In this phase, the ground base station B_m initializes the system by generating and publishing a set of system parameters:

1) B_m selects a large prime number p as well as a non-singular elliptic curve $E(p)$.
2) B_m chooses a cyclic additive group $\mathbb{G}$ of order q with an arbitrary generator n.
3) B_m specifies two one-way hash functions H_a and H_b, where H_a:{0,1}*→$\mathbb{Z}_q^*$ and H_b:{0,1}*→{0,1}^{t}, where t indicates the number of bits.
4) B_m publishes all system parameters as $\{p, q, n, H_a, H_b\}$.

2) Drone Registration: In this phase, the drone ID_i registers with the ground base station B_m in the following steps:

1) ID_i chooses its real identity RID_i and initial PUF challenge $che_i^{t_x}$ arbitrarily, and calculates the corresponding response $res_i^{t_x} = F_{puf}(che_i^{t_x})$. Here, t_x is the timestamp.
2) ID_i selects a random number $r_i^{t_x}$ and uses its PUF to compute a secret value $s_i^{t_x} = F_{puf}(r_i^{t_x})$.
3) ID_i calculates its pseudonym $PID_i^{t_x} = H_b(RID_i \parallel res_i^{t_x} \parallel s_i^{t_x})$.
4) ID_i computes two public dynamic parameters $R_{i,a}^{t_x} = res_i^{t_x} \cdot P$ and $R_{i,b}^{t_x} = s_i^{t_x} \cdot P$.
5) ID_i shares (RID_i, $PID_i^{t_x}$, $res_i^{t_x}$, $s_i^{t_x}$, $R_{i,a}^{t_x}$, $R_{i,b}^{t_x}$) with B_m via a secure channel.
6) ID_i stores its RID_i, $che_i^{t_x}$, and $r_i^{t_x}$ but $res_i^{t_x}$, $s_i^{t_x}$, $PID_i^{t_x}$, $R_{i,a}^{t_x}$, and $R_{i,b}^{t_x}$ in the memory. For security reasons, ID_i does not store the critical cryptographic value such as $res_i^{t_x}$ and $s_i^{t_x}$ directly. In order to save the storage space, ID_i can dynamically re-calculate $PID_i^{t_x}$, instead of storing. If the storage space is not a concern, ID_i can choose to cache $PID_i^{t_x}$ directly.

3) Authentication and Key Establishment: In this phase, the drone ID_i and the ground base station B_m authenticate each other and negotiate a session key for the submission of observational data.

1) ID_i computes its old response $res_i^{t_x} = F_{puf}(che_i^{t_x})$ and old secret value $s_i^{t_x} = F_{puf}(r_i^{t_x})$.
2) ID_i calculates its old pseudonym $PID_i^{t_x} = H_b(RID_i \parallel res_i^{t_x} \parallel s_i^{t_x})$.
3) ID_i chooses a new PUF challenge $che_i^{t_y}$ randomly and computes a new PUF response $res_i^{t_y} = F_{puf}(che_i^{t_y})$.
4) ID_i selects a new random number $r_i^{t_y}$ and calculates a new secret value $s_i^{t_y} = F_{puf}(r_i^{t_y})$.
5) ID_i calculates $msg_{i,a}^{t_y} = res_i^{t_y} \oplus H_b(RID_i \parallel B_m \parallel PID_i^{t_x} \parallel res_i^{t_x})$.
6) ID_i calculates $msg_{i,b}^{t_y} = s_i^{t_y} \oplus H_b(RID_i \parallel B_m \parallel PID_i^{t_x} \parallel res_i^{t_y} \parallel s_i^{t_x})$.
7) ID_i computes $sig_i^{t_y} = res_i^{t_x} + H_a(RID_i \parallel PID_i^{t_x}) \cdot s_i^{t_x}$.
8) ID_i sends the authentication request $req_{i,m} = \{PID_i^{t_x}$, $msg_{i,a}^{t_y}$, $msg_{i,b}^{t_y}$, $sig_i^{t_y}\}$ to B_m over wireless channel.
9) ID_i calculates the secure session key as $SK_{i,m} = H(res_i^{t_x} \oplus s_i^{t_x}) \oplus H(res_i^{t_y} \oplus s_i^{t_y})$.
10) B_m checks whether $PID_i^{t_x}$ is valid (i.e., an entry exists in the database). If not, the authentication request is rejected. Otherwise, B_m retrieves ID_i's authentication information.
11) B_m calculates $res_i^{t'_y} = msg_{i,a}^{t'_y} \oplus H_b(RID'_i \parallel B_m \parallel PID_i^{t'_x} \parallel res_i^{t'_x})$.
12) B_m computes $s_i^{t'_y} = msg_{i,b}^{t'_y} \oplus H_b(RID'_i \parallel B_m \parallel PID_i^{t'_x} \parallel res_i^{t'_y} \parallel s_i^{t'_x})$.
13) B_m verifies $sig_i^{t'_y}$ as follows
$$\begin{aligned} sig_i^{t'_y} \cdot P &= (res_i^{t'_x} + H_b(RID'_i \parallel PID_i^{t'_x}) \cdot s_i^{t'_x}) \cdot P \\ &= res_i^{t'_x} \cdot P + H_b(RID'_i \parallel PID_i^{t'_x}) \cdot s_i^{t'_x} \cdot P \\ &= R_{i,a}^{t'_x} + H_b(RID'_i \parallel PID_i^{t'_x}) \cdot R_{i,b}^{t'_x}. \end{aligned}$$
If the verification fails, B_m rejects the authentication request. Otherwise, B_m calculates the secure session key $SK_{m,i} = H(res_i^{t'_x} \oplus s_i^{t'_x}) \oplus H(res_i^{t'_y} \oplus s_i^{t'_y})$.
14) B_m updates ID_i's authentication information (RID_i, $PID_i^{t_y}$, $res_i^{t_y}$, $s_i^{t_y}$, $R_{i,a}^{t_y}$, $R_{i,b}^{t_y}$) in the database. Here, the updated $PID_i^{t_y}$, $R_{i,a}^{t_y}$, and $R_{i,b}^{t_y}$ can be directly calculated by B_m.
15) B_m sends the authentication response $rep_{m,i} = H_b(RID_i \parallel B_m \parallel res_i^{t'_y} \parallel s_i^{t'_y})$ to ID_i.
16) ID_i compares $rep_{m,i}$ with its calculated value. If the validation succeeds, ID_i believes B_m is legitimate. Oth-

Algorithm 3: Drone ID_i Authentication Algorithm

```
Function DroneAuth():
    $res_i^{t_x} = F_{puf}(che_i^{t_x})$; $s_i^{t_x} = F_{puf}(r_i^{t_x})$;
    $PID_i^{t_x} = H_b(RID_i \| res_i^{t_x} \| s_i^{t_x})$;
    $che_i^{t_y} \leftarrow RandNum()$; $res_i^{t_y} = F_{puf}(che_i^{t_y})$;
    $r_i^{t_y} \leftarrow RandNum()$; $s_i^{t_y} = F_{puf}(r_i^{t_y})$;
    $msg_{i,a}^{t_y} = res_i^{t_y} \oplus H_b(RID_i \| B_m \| PID_i^{t_x} \| res_i^{t_x})$;
    $msg_{i,b}^{t_y} = s_i^{t_y} \oplus H_b(RID_i \| B_m \| PID_i^{t_x} \| res_i^{t_y} \| s_i^{t_x})$;
    $sig_i^{t_y} = res_i^{t_x} + H_a(RID_i \| PID_i^{t_x}) \cdot s_i^{t_x}$;
    $SK_{i,m} = H(res_i^{t_x} \oplus s_i^{t_x}) \oplus H(res_i^{t_y} \oplus s_i^{t_y})$;
    $Send(B_m, PID_i^{t_x}, msg_{i,a}^{t_y}, msg_{i,b}^{t_y}, sig_i^{t_y})$;
Function StationAuth():
    if $PID_i^{t_x}$ is not valid then
        reject;
    else
        $res_i^{t'_y} = msg_{i,a}^{t'_y} \oplus H_b(RID'_i \| B_m \| PID_i^{t'_x} \| res_i^{t'_x})$;
        $s_i^{t'_y} = msg_{i,b}^{t'_y} \oplus H_b(RID'_i \| B_m \| PID_i^{t'_x} \| res_i^{t'_y} \| s_i^{t'_x})$;
        $sig^{tmp} = R_{i,a}^{t'_x} + H_b(RID'_i \| PID_i^{t'_x}) \cdot R_{i,b}^{t'_x}$;
        if $sig_i^{t'_y} \cdot P \neq sig^{tmp}$ then
            reject;
        else
            $SK_{m,i} = H(res_i^{t'_x} \oplus s_i^{t'_x}) \oplus H(res_i^{t'_y} \oplus s_i^{t'_y})$;
            $Update(RID_i, PID_i^{t_y}, res_i^{t_y}, s_i^{t_y}, R_{i,a}^{t_y}, R_{i,b}^{t_y}\})$;
            $rep_{m,i} = H_b(RID_i \| B_m \| res_i^{t'_y} \| s_i^{t'_y})$;
            $Send(ID_i, rep_{m,i})$;
        end
    end
```

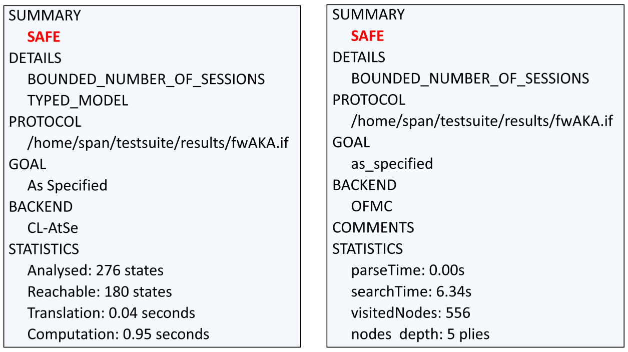
erwise, ID_i aborts the authentication process.

At this moment, the process of authentication and key agreement has been completed between the drone ID_i and the ground base station B_m, and they can securely communicate with the session key.

V. Experimental Study

A. Security Verification

We use AVISPA [25] to evaluate the security and logic of *fwAKA* to see whether the protocol design has any potential security flaw or vulnerability. To be specific, we use two backends in AVISPA, On-the-fly Model-Checker (OFMC) and Constraint-Logic-based Attack Searcher (CL-AtSe), to analyze the behaviors of *fwAKA* and validate its security features. We download a fully-functional SPAN+AVISPA [15] virtual machine image and conduct security verification in Virtual Box [26]. First, the *fwAKA* is implemented in HLPSL [15] which is a AVISPA-specific programming language. Then, the HLPSL is executed in AVISPA, which will evaluate the *fwAKA* under masquerading attacks, replay attacks, and other unknown attacks in the adversary setting. If the *fwAKA* is vulnerable to a specific attack, AVISPA will output a sequence diagram showing the vulnerable scenario. Otherwise, the *fwAKA* is marked as ‘‘safe’’ by AVISPA. The outputs of OFMC and CL-AtSe are shown in Fig. 3, where we can easily observe that the *fwAKA* is identified as a safe security protocol. Meanwhile, we also conclude that the *fwAKA* does not have any design flaws or vulnerabilities that could be exploited by well-known cyber attacks.



```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/testsuite/results/fwAKA.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed: 276 states
  Reachable: 180 states
  Translation: 0.04 seconds
  Computation: 0.95 seconds
```

(a)

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/testsuite/results/fwAKA.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 6.34s
  visitedNodes: 556
  nodes  depth: 5 plies
```

(b)

Fig. 3. Security verification results using AVISPA’s CL-AtSe and OFMC.

TABLE I
Communication Overhead

Scheme	No. of Transmitted Msg	Communication Energy Cost
*fwAKA**	2	2.2524×10^{-3}
modAKA$^\diamond$	3	3.3786×10^{-3}
SecAuth‡	3	3.3786×10^{-3}

*: The *fwAKA* requires the drone to send one (1) authentication request message and the ground base station to reply one (1) authentication response message.

$^\diamond$: The *modAKA* exchanges one (1) message between the user and the control server, one (1) message between the control server and the drone, and one (1) message between the drone and the user.

‡: The *SecAuth* first needs the drone to send one (1) message to initiate the authentication process with the ground station. After that, the drone and the ground station exchange two (2) messages to complete the authentication and session key agreement process.

B. Performance Evaluation

We build an Eclipse platform to conduct simulation-based experiments and observe the performance of *fwAKA* in terms of communication overhead and execution time. To be specific, we install Eclipse IDE for Java Developers on a Windows PC, select two benchmark schemes, *modAKA* [12] and *SecAuth* [16], and implement them in Java programming language. The experimental desktop PC has the following specifications: Windows 10 Pro 64-bit operating system and the 4th Generation Intel(R) Core(TM) i5-4690K CPU (6M Cache, up to 3.90 GHz). In the *modAKA*, the control server first generates a master key and makes a group of system parameters available to the user and the drone. In order to be involved in the IoD system, the user and the drone are required to go through the registration process at the control server and obtain their cryptographic information over a secure channel. Finally, the drone and the user authenticate each other and set up the session key with the help of the control server. In the *SecAuth*, the authentication process consists of two steps: drone registration and drone-to-ground station authentication. The drone and the ground station will negotiate cryptographic information during the drone registration phase. In drone-to-ground station authentication phase, mutual authentication is achieved between the drone and the ground station after performing various computations and exchanging messages.

We present the communication overhead of *fwAKA*, *modAKA*, and *SecAuth* in Table. I, where the number of

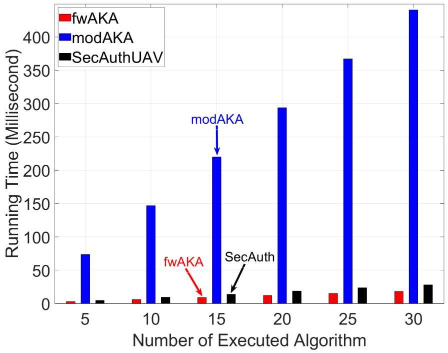


Fig. 4. Running time against the number of algorithm executions.

exchanged messages and the energy consumption of communication are obtained. In the *fwAKA*, two (2) messages are required for the entire process of authentication and key agreement, and the energy consumption of *fwAKA* is 2.2524×10^{-3} joule. However, for both *modAKA* and *SecAuth*, they will need to exchange three (3) messages to finally establish a secure session key. Since the energy consumption of communication is measured based on the number of exchanged messages [27], thus, *modAKA* and *SecAuth* consume the same amount of energy, which is 3.3786×10^{-3} joule.

We also measure the running time of *fwAKA*, *modAKA*, and *SecAuth*, and the results are presented in Fig. 4. Overall, the *fwAKA* shows the lowest running time among three schemes. This is because the *fwAKA* adopt resource-friendly operations as wells as requires less number of operations for the authentication key agreement. The highest running time belongs to the *modAKA* because the *modAKA* is designed based on a large number of hash and XOR operations. The *SecAuth* is also designed with the PUF, however, a higher running time is observed than that of the *fwAKA*. This is because the *SecAuth* requires more computations than the *fwAKA*.

VI. Conclusion

In this paper, we proposed a featherweight authentication and key agreement scheme (also called *fwAKA*) for IoD applications. The major advantage is that the *fwAKA* only requires two handshakes to achieve the authenticated key agreement. The *fwAKA* was implemented in the security-sensitive protocol modeling language and evaluated using the AVISPA framework. We also implemented the *fwAKA* and two benchmark schemes, and conducted experimental simulation to evaluate their performance. Experimental results indicate that the *fwAKA* not only is a secure protocol without any design flaw, but also provides superior communication and computation performance. As a future work, we plan to integrate the *fwAKA* with blockchain technology to realize the cross-domain authentication for IoD systems.

References

[1] S. Chinthi-Reddy, S. Lim, G. Choi, J. Chae, and C. Pu, "DarkSky: Privacy-preserving target tracking strategies using a flying drone," *Vehicular Communications*, vol. 35, no. 4, p. 100459, 2022.

[2] 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.

[3] H. Jagarlapudi, S. Lim, J. Chae, G. Choi, and C. Pu, "Drone Helps Privacy: Sky Caching Assisted k-Anonymity in Spatial Querying," *IEEE Systems Journal*, vol. 16, no. 4, pp. 6360–6370, 2022.

[4] C. Pu and P. Zhu, "Defending against Flooding Attacks in the Internet of Drones Environment," in *Proc. IEEE GLOBECOM*, 2021, pp. 1–6.

[5] P. Goodrich, O. Betancourt, A. Arias, and T. Zohdi, "Placement and drone flight path mapping of agricultural soil sensors using machine learning," *Computers and Electronics in Agriculture*, vol. 205, p. 107591, 2023.

[6] C. Pu, A. Wall, and K. Choo, "Bilinear Pairing and PUF Based Lightweight Authentication Protocol for IoD Environment," in *Proc. IEEE MASS*, 2022, pp. 115–121.

[7] 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.

[8] *IEEE Approved Draft Trial-Use Standard for Aerial Ad Hoc Networks*, Last accessed: Feb 24, 2023, https://standards.ieee.org/ieee/1920.1/10352/.

[9] M. Zhang, C. Xu, S. Li, and C. Jiang, "On the Security of an ECC-Based Authentication Scheme for Internet of Drones," *IEEE Systems Journal*, vol. 16, no. 4, pp. 6425–6428, 2022.

[10] 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, pp. 3319–3332, 2021.

[11] 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.

[12] D. Chaudhary, T. Soni, K. Vasudev, and K. Saleem, "A modified lightweight authenticated key agreement protocol for Internet of Drones," *Internet of Things*, vol. 21, p. 100669, 2023.

[13] 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.

[14] S. Yu, J. Lee, A. Sutrala, A. Das, and Y. Park, "LAKA-UAV: Lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad-hoc networks," *Computer Networks*, vol. 224, p. 109612, 2023.

[15] *SPAN*, Last accessed: Feb 27, 2023, http://www.avispa-project.org/.

[16] T. Alladi, N. Naren, G. Bansal, V. Chamola, and M. Guizani, "SecAuthUAV: A Novel Authentication Scheme for UAV-Base Station Scenario," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 068–15 077, 2020.

[17] M. Akram, H. Ahmad, A. Mian, A. Jurcut, and S. Kumari, "Blockchain-based privacy-preserving authentication protocol for UAV networks," *Computer Networks*, vol. 224, p. 109638, 2023.

[18] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for internet of drones," *Computer Communications*, vol. 154, pp. 455–464, 2020.

[19] Y. Zhang, D. He, P. Vijayakumar, M. Luo, and X. Huang, "SAPFS: An Efficient Symmetric-Key Authentication Key Agreement Scheme with Perfect Forward Secrecy for Industrial Internet of Things," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[20] Y. Liang, E. Luo, and Y. Liu, "Physically Secure and Conditional-Privacy Authenticated Key Agreement for VANETs," *IEEE Transactions on Vehicular Technology*, pp. 1–12, 2023.

[21] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. ACM/IEEE DAC*, 2007, pp. 9–14.

[22] D. Dolev and A. Yao, "On the Security of Public Key Protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[23] J. Wallrabenstein, "Practical and Secure IoT Device Authentication Using Physical Unclonable Functions," in *Proc. IEEE FiCloud*, 2016, pp. 99–106.

[24] F. MacWilliams and N. Sloane, *The Theory of Error Correcting Codes*. Elsevier, 1977.

[25] *Automated Validation of Internet Security Protocols and Applications*, http://www.avispa-project.org.

[26] *VirtualBox*, Last accessed: Sep 15, 2022, https://www.virtualbox.org/.

[27] 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.