IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 16, 2021, accepted March 25, 2021, date of publication April 2, 2021, date of current version April 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070683

A Lightweight Authentication Protocol for UAV Networks Based on Security and Computational Resource Optimization

YUAN LEI[1], LINING ZENG[2], YAN-XING LI[1], MEI-XIA WANG[1], AND HAISHENG QIN[1]
[1]School of Information and Engineering, Guangxi University of Foreign Languages, Nanning 530000, China
[2]School of Business Administration, Hunan University of Finance and Economics, Changsha 410000, China

Corresponding author: Lining Zeng (will120120@126.com)

This work was supported by the Young and middle-aged teachers' basic ability improvement of Guangxi colleges in 2021 under Grant 2021KY1781, and by the Scientific Research Project of Hunan Provincial Department of Education under Grant 19C0324.

ABSTRACT The widespread use of Unmanned Aerial Vehicles (UAV) has made the security and computing resource application efficiency of UAV a hot topic in the security field of the Internet of Things. In this paper, an optimized lightweight identity security authentication protocol, Optimized Identity Authentication Protocol (ODIAP) is proposed for Internet of Drones (IoD) networks. The protocol is targeted to the security risks faced by IoD networks, and proposes the security authentication mechanism consisting of 3 phases and 7 authentication processes, which enables the protocol has both forward and backward security, and can resist mainstream network attacks. Meanwhile, this paper fully considers the computational load and proposes the identity information generation and verification method based on the Chinese residual theorem, which reduces the computational load of resource-constrained nodes and shifts the complex computational process to server nodes with abundant computational resources. Moreover, after security protocol analysis and tool verification based on the automated security verification tool Proverif, the protocol in this paper has complete security. At the same time, the performance analysis and comparison with other mainstream protocols shows that this protocol effectively optimizes the use of computing resources without compromising security.

INDEX TERMS UAV, Internet of Drones, lightweight authentication, Proverif, security.

I. INTRODUCTION

A. BACKGROUND

With the development of network and embedded system, Internet of things (IoT) has become the most important and widely used concepts in modern society. The Internet of Things is an overall system consisting of a series of smart devices that interact through the network, where the smart devices have strong processing and communication capabilities and have locatable Internet Protocol addresses (IP addresses) [1]. Furthermore, the Internet of Things assumes the function of directly integrating computing systems with the physical world by sensing, analyzing, and transmitting information about the physical environment. The Internet of Drones (IoD) is the typical mobile IoT system [2]. In recent years, UAV has become an important application method for remote access boards due to its advantages in terms of coverage, exploration capabilities, and intelligence level. Besides, there has been a strong demand for consumer-grade UAV around the world, and the demand for UAV among the general public has gradually increased. The use of UAV is becoming more widespread. For example, in the fields of aerial photography, agriculture, miniature selfie, film and television shooting to name a few, which have further expanded the use of UAV themselves. It has been reported that the UAV industry is expected to grow at a rate of upwards of 29.9% per year in the coming years, reaching an industry size of $4.5 billion by 2026 [3]. Drones are steadily moving forward in the Internet of Things and have great potential to lead our people into the era of drones.

However, communication security is the key issue that needs to be urgently improved in the application of IoD. For examples, in January 2016, Mexican drug traffickers used satellite navigation signal spoofing technology to send fake

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

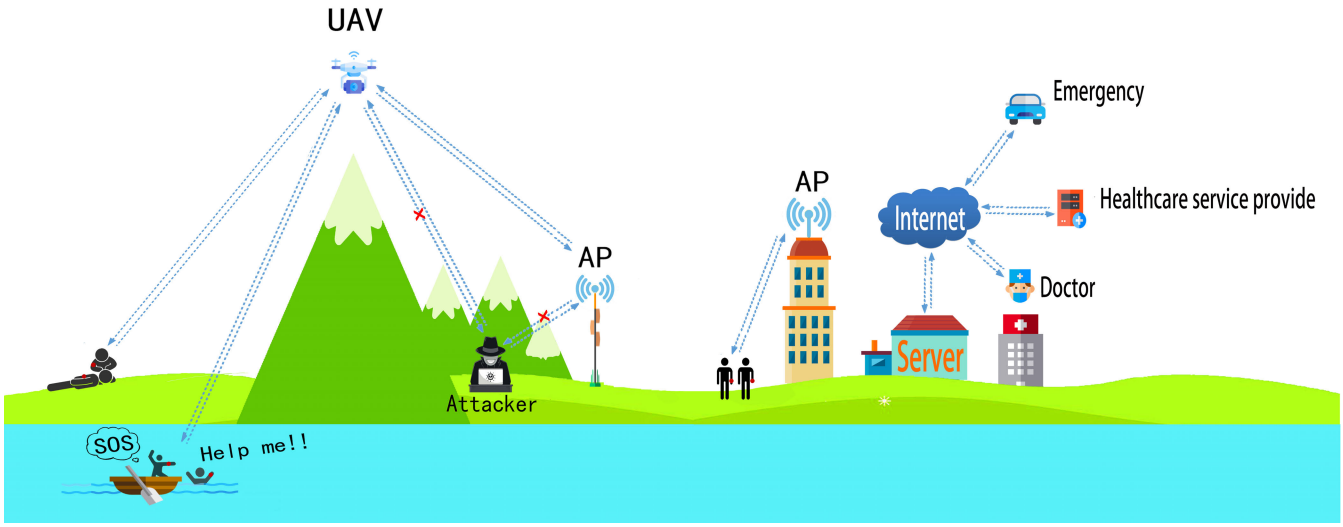

FIGURE 1. Application scenarios of UAV.

GPS signals to attack U.S. border patrol aircraft to achieve illegal border crossing. In July 2016, IBM security expert Nils Rodday at the Black Hat Security Asia Summit, clearly stated that drones loaded with unencrypted chips are vulnerable to hijack, which to a certain extent can make the rescue extremely difficult. In June 2019, the U.S. MQ-9 drone was held hostage by the Iranian military, causing the serious information leak. Therefore, it is urgent to address the communicative security of UAV during patrols and rescues [4].

As a typical IoT smart device, UAV has two important characteristics, variable environment and resource-constrained. The UAV network always changes with the movement of the UAV and sensor carriers and their connected state with different network facilities (e.g., AP, servers, etc.). Thus, the identity of each node in the network needs to be constantly authenticated in multiple rounds. In addition, mobile nodes (UAV, sensors, etc.), as an embedded device, are subject to large limitations on their computing power and resources, and the too frequent and complex authentication mechanism will inevitably affect the execution of their own functions and the endurance of the device. Therefore, it is an important issue in the field of IoD security to ensure the security of the network environment, avoid the loss of people and properties due to network attacks and information leakage, and reduce the consumption of security mechanism operation as much as possible, so as to improve the operational efficiency of smart devices.

B. RELATED WORKS

In this paper, we propose an Optimized IoD Identity Authentication Protocol (ODIAP) for IoD networks. The protocol fully considers the security of the system, especially satisfying both forward security and backward security. Meanwhile, through the application of the Chinese residual theorem, the protocol in this paper achieves the transfer of computational load from mobile nodes to servers, reduces the requirement of computational resources for mobile nodes, and optimizes the range and application effect of mobile nodes.

A sensing network for UAV communication consists of sensors and drones in the air, which communicates through wireless technology [5]. Figure 1 shows a typical scenic IoD network architecture, in which several smart devices are installed in the scenario, such as sensors, servers, and drones, etc. AP usually denotes access points built in scenic areas or base station devices of operators. The sensors need to communicate with the servers through the access point AP. Due to construction costs and physical constraints, AP usually do not have the ability to cover 100% of the area. For the reason, the sensors on the visitors are connected to the Internet through their nearby drones as signal relays. This network architecture enables the server to access real-time data from multiple visitor devices through the drones and grant user-authorized access for them [6].

Figure 1 shows attacker, UAV, APs and the server. Two tourists standing in the center are within the coverage of AP, while other tourists in the lower left corner and in the lake are not within the safe coverage. Since the scenic area is large, full signal coverage is not possible, so there are cases where tourists are out of the coverage area. For example, in the bottom left corner, there are tourists who fainted with sudden sick and those who unfortunately fell into the water. These tourists left the coverage area of AP without knowing it. When their companions encounter dangerous situations, such as sickness, fainting, or accidentally falling into the water, they cannot send the distress signal to the server in time. Therefore, in order to ensure that the tourists are in a safe range of activities and the tourists' distress signals can be sent out, the UAV will patrol the places where the AP coverage

signal is weak on time. The UAV patrol plays a key role in extending the coverage. When the UAV receives the distress signal outside the coverage area, it immediately transmits the signal to the server, so that the rescue team can accurately and timely carry out the rescue task. Meanwhile,this network architecture can also be effectively applied to other scenarios, such as nursing homes, kindergartens, urban and forest firefighters and so on. It can effectively and accurately detect human physiological data and geographic location in real time, and the monitoring organization can also remotely understand their specific location and health status. When they have abnormal physiological data and geographical location, the monitoring agency can find and rescue them in time.

As mentioned above, although the visitor's sensors can collect various data from the human body, such as temperature, geographic location, etc.

However, these sensors have very few resources at their disposal. For example, the visitor on the left in Figure 1 is in an area where the access point AP signal is weak or even no signal. Therefore, periodic patrols by UAV are needed to act as repeaters between the sensors and the AP of the access points for the purpose of increasing the coverage of the whole scheme. The security department and medical department of the scenic area can access the server through the Internet to remotely monitor the health status of visitors inside the entire scenic area. Once a visitor's physiological and geographical data is found to be abnormal (such as rapid heartbeat, high or low temperature, abnormal geographical location), rescue can be carried out in time and more first aid time can be obtained. The above solution greatly reduces the health risk of visitors in the scenic area and improves the efficiency of the security department and medical sector.

From the above, it is clear that there is a large amount of important private information in the sensing network, so it is necessary to secure the communication of this sensing network. However, most of the existing schemes are based on asymmetric encryption, for example, Yao et al. [7], [7] proposed another scheme using elliptic curve cryptography (ECC). The sensors used by tourists are resource-constrained devices and the computational resources cannot afford the asymmetric cryptography operations. Thus, Püllen *et al.* [8] further proposed to use a lightweight authentication scheme, however, due to the exponential time complexity of some operations, the burden of sensor operations is high, coupled with the inability to effectively resist various known attacks. In order to provide better security, Liang *et al.* [9] proposed an anonymous lightweight user authentication mechanism scheme and claimed that it has the ability to resist various attacks known so far. In contrast, the protocol does not have a complete security proof process, and the communication bit overhead problem in this protocol is not solved.

Shepard *et al.* (2012) [10] proposed the lightweight authentication based on elliptic curve cryptosystem (ECC). Although the ECC authentication used in this scheme achieves the condition of lightweight authentication, it is vulnerable to password guessing attacks leading to session-specific information leakage. Therefore, Yan *et al.* (2016) [11] proposed an SDN-based mutual authentication security protocol for multi-drone networks. However, we found no detailed security comparison and lacking of detailed performance analysis process in that, and there is a risk of session key violation. And then, Li *et al.* (2019) [12] proposed an anonymous lightweight user authentication mechanism scheme, which claims to have the ability to resist various attacks known so far. Pu and Li (2020) [13] also proposed the mutual authentication protocol asPCAP based on physical unclonable function (PUF), however, we found that the protocol proposed in this paper does not have complete forward security.

In the meantime, we found that detailed authentication for both forward and backward security of the protocols is rarely found in any of the current proposed protocols in the current study. The protocol proposed by Amin *et al.* (2016) [14], on the other hand, provides more detailed authentication for forward security and designs AKA schemes based on passwords, smart cards, and biometrics for securing the communication process. Jiang et al.(2017) [15] pointed out that the scheme of Amin *et al.* (2016) [14] has no ability to fight back against the loss of smart card and offline password guessing attacks, and further proposed the signature-based ak scheme. Challa *et al.* (2017) [16] proposed a new signature-based aka scheme. Furthermore, it increases security, and it also greatly increases communication and computing costs. On the basis of the predecessors, Zhang *et al.* (2020) [17] proposed a lightweight AKA scheme based on the previous work with the characteristics of UAV. However, the performance of this protocol is poor and only a small number of protocols are compared. Chamola *et al.* (2020) [1] proposed the framework for UAS that collects information independently, and the lightweight authentication protocol was designed for this framework. Although this protocol does not use any complex functional operations and is superior in terms of performance, it is relatively poor in security and does not resist most known attacks. Barman *et al.* (2019) [18] proposed a two-way authentication and key negotiation scheme in a multi-server environment. However, Ali *et al.* (2020) [20] pointed out that the protocol is vulnerable to attacks such as server emulation, session key leakage, user emulation, secret temporary parameter leakage and other attacks, and lack of user anonymity. In addition, their scheme suffers from scalability issues.

Unmanned Aerial Vehicle (UAV) is a new industry, which lacks communication security. The communication security between users and UAV needs more attention [21], [22]. Although many security protocols have been designed in recent years, they are not sufficient to prevent common attacks [4]. A large number of solutions have been proposed to enable secure communication between users and UAV, but all of them have some shortcomings. Wazid *et al.* (2018) [23] proposed a novel lightweight user authentication scheme for UAV distributed networks. However, we found that the scheme did not have perfect forward security.

Meanwhile, Srinivas *et al.* (2019) [24] proposed an anonymous lightweight user authentication mechanism in IoD environment based on temporal credentials. However, the protocol was pointed out by Ali *et al.* (2020) [19] as not being resistant to traceability and stolen authentication attacks.

According to the related research results, we found that most of the protocols did not pay much attention to the reasonableness of the consumption of communication entities assigned to the various parts of the protocol involved in the communication. For example, the communication entities consumed by each part of the protocol proposed in these researchers [24]–[26] are extremely unreasonable and present great difficulties for practical applications.

C. CONTRIBUTION

In response to the issues raised above, this paper designs the lightweight authentication protocol, Optimized IoD Identity Authentication Protocol (ODIAP) for IoD network environment, which achieves optimization of security and computational efficiency.

1) The protocol fully considers various network threats in IoD networks, and develops the targeted authentication scheme for four factors (sensors, drones, AP and servers) through the design of the protocol flow. In particular, the protocol in this paper has complete forward and backward security, even if a session key is leaked, it will not affect other information.
2) In the protocol, we design the generation and authentication of mobile node identity information based on the Chinese residual theorem, which optimizes the resource utilization of mobile nodes by significantly reducing the computation of mobile nodes without affecting the security, and the complex decoding and verification work are transferred to the server nodes with abundant resources.
3) The security of the protocols in this paper is demonstrated through security protocol analysis and verification by Proverif-based tools. Meanwhile, the advantages of this paper's protocol in terms of resource utilization are verified by system performance analysis.

The remaining content of this paper is organized as follows. Section 2 mainly introduces the network model, the overall authentication process, the security threats and the security requirements. In Section 3, we describe the authentication process of the proposed protocol in detail, including the steps of the initialization phase, the registration phase and the authentication phase. In Section 4, we perform the functional analysis of the protocol, along with the comprehensive security verification using Proverif code. In Section 5, we select eight representative protocols for more comprehensive comparison. Finally, we conclude the paper and provide an outlook on future work in Section 6.

II. PROBLEM DESCRIPTION

A. NETWORK MODEL

The object of study in this paper is the UAV sensing network. There are four types of nodes in the network.

- Server. The server node is the core processing node of the network and has the most powerful computing power and abundant computing resources, and can undertake any level of computing work in the network.
- AP. The AP has strong computing power and abundant computing resources, which can run some complex operations.
- UAV. The UAV has computing power and resources, but cannot run complex functions.
- Sensor. The sensors have significantly limited computing power and resources such as processors and energy consumption, and can only complete simple calculations.

A hierarchical structure exists among Server, UAV and N considering the logic of network formation. Similar to the Ding *et al.* [27], the network model in this paper is based on the two-hop centralized architecture. However, for the specific application environment of IoD, as shown in Figure 2, this paper extends the network model with several hierarchical applications.

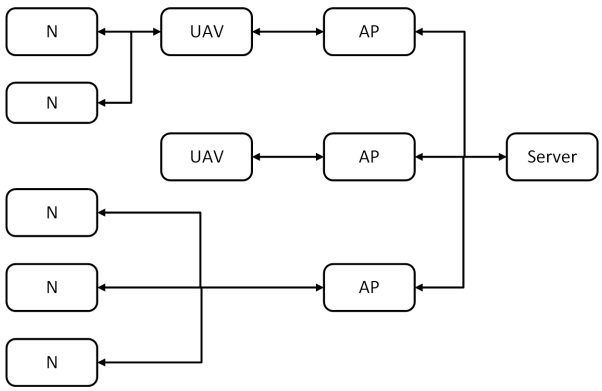

FIGURE 2. Network model.

Among them, this paper assumes that each application in the IoD has only one server corresponding to it. Moreover, with the server as the core, the applications in each IoD network are deployed in disjoint clusters of network nodes, so that the network security study with the deployment of either application is feasible.

In order to ensure that the model meets the diversity of applications, in the model of this paper, communication does not need to strictly adhere to the sensor-UAV-AP-server hierarchy, but rather take the three forms as follows. (1) The sensing node N in the specific cluster sends its sensed information to its own UAV, and then the UAV sends the information to the AP, which forwards the information to the server. (2) Without sensor nodes, the UAV passes the information to the AP, then transmits the information to the server. (3) The sensing node N in the specific cluster, passing

through the UAV node level, sends its sensed information to its own AP, and finally the AP sends the information to the server. Three communication methods can needed the security protocol requirements for any kind of application in IoD. It is worth noting that the communication between the sensing node N and its associated UAV, as well as the communication between the UAV and the server, takes place over the wireless channel.

B. OVERALL AUTHENTICATION PROCESS

In the IoD network, the authentication protocol is the most essential element to ensure network security. In an established IoD network, authentication mainly consists of the following processes.

1) Broadcast: The UAV continuously broadcast authentication requests during patrols, attempting to communicate with the human sensors (N) within communication range.
2) Response: When the human sensor (N) is in the communication range of the UAV and receives the broadcast request from the UAV, it responds to the authentication message.
3) Authentication: The UAV receives the response message, performs preliminary calculations, and sends the authentication message to the server.
4) Verification: The server verifies the legitimate identity of the UAV and the human sensor, updates the session key and sends it to the UAV and the sensor.
5) Communication: The UAV and the human sensor (N) obtain the new key and communicate with the new key.

Based on the above, certification must ensure the following points. (1) The identity is true; (2) The information is trusted; (3) The authentication itself is reliable. It is worth noting that authentication in an IoD network is not a one-off act. Furthermore, during the operation of the IoD network, authentication is a recurrence process, and each node in the network will continuously repeat the authentication process with the certain frequency to ensure system security. In addition, when the network environment changes, such as UAV flying in and out of the AP's range, sensor nodes joining and withdrawing, etc., it may bring a new authentication process.

C. SECURITY THREATS AND SECURITY NEEDS

In order to evaluate the security of the protocol, this paper uses the dolev Yao (DY) threat model which is widely used in the field of wireless sensor network security [14] as the threat model.

In the DY model, the attacker is considered to have the ability to control the channel of the whole network. Thus, in the context of the IoD network studied in this paper, the capabilities possessed by the attacker and the security threats it may lead to include the following four aspects.

1) Two parties can communicate freely over the public channel, but all channels cannot be guaranteed to be secure. Therefore, in the process of transmitting data over the channel, the attacker may have complete control over the communication channel, obtain all the data exchanged in the channel, and can perform the following operations on the contents of the channel: read, modify, delete, and inject messages, thus forming the replacement or the replay attacks for posing the security threat.
2) The attackers have data processing capabilities. When the attacker has enough information to be able to manipulate data through computation. For example: calculating an element in a tuple or decrypting the message directly with the necessary key in hand.
3) All mobile nodes in the IoD network, including AP, sensors N and UAV, are constrained by the resource environment and may be subject to malicious physical damage by attackers, resulting in the leakage of the data stored in their media. Therefore, both sensors and UAV can be defined as nodes that cannot be fully trusted. It is worth stating that since the server is the gateway for all communication nodes to communicate with the outside, but its security does not belong to the security category of wireless sensor networks, we assume that the server is absolutely secure when discussing the security of infinite sensing networks.
4) If the enemy is the Narrowa-Strong attacker, the attacker can continuously listen to the protocol when it does not miss the key update process, causing the computation process of the protocol itself to be identified.

To address these security threats, this paper studies lightweight authentication mechanisms. In the mechanism, two aspects need to be addressed.

(1) Security. Since all the channels are not trusted under the DY model and most of the nodes are not trusted in IoD. Furthermore, the system needs a complete security mechanism in order to ensure the security of the system. The mechanism should be able to guarantee the following three aspects.

- The authentication protocol itself has the complete logic.
- The protocol not only defends against remote attacks, such as impersonation attacks, asynchronous attacks, and replay attacks, but also against threats resulting from the compromise of physical devices.
- The protocol has both forward and backward security, and the compromise of one round of messages will not cause the next security problems.

(2) Performance. The IoD network is the typical wireless sensor network. Its core feature is that all nodes except the server has constraints in terms of computing power and energy consumption, which is the reason for the development of lightweight authentication technology. Therefore, for the authentication mechanism of IoD networks, in addition to security assurance, the system needs to meet the following requirements.?

- Lightweight. Mobile nodes, such as sensors and UAV, cannot take on high-complexity computations. Where possible, it is best to move calculation to the server with higher processing power.
- Real-time. Due to the variability of the IoD network, the system requires that the computation must be completed within the limited time, otherwise the efficiency of the authentication will be greatly affected.
- Streamlined communication. The frequency and amount of data required for the system to implement secure authentication should be as low as possible.

III. PROTOCOL AUTHENTICATION PROCESS

The process of authentication is as follows.

(1) The initialization phase generates core security parameters for each node in IoD.

(2) The registration phase passes the necessary security parameters between the nodes to build the initial data architecture of the authentication protocol.

(3) During the authentication phase, continuous authentication is performed during the operation of IoD, and the identity of each node in the network is lightweight authenticated in real time.

Table 1 and Table 2 shows the key symbols in the protocol, where the subjects involved in the authentication are shown in Table 1, including one server node, several sensors, UAV and AP nodes.Table 2 shows the core parameters required in the protocol process. This section describes the computation and authentication process during these phases.

TABLE 1. Notations.

Symbols	Notes
N_i	The i-th sensor node, i∈N
U_i	The i-th UAV node,i∈N
AP_i	The i-th access point,i∈N
Server	Server
IDN_i	IDNi i-th sensor identifier,i∈N
IDU_i	The i-th UAV identifier,i∈N
$IDAP_i$	The i-th access point identifier,i∈N
$PRNG()$	Random number generation algorithm
$PUF()$	Physical unclonable function
$Hash()$	Hash function
$\oplus$	XOR operation

TABLE 2. Parameter notations.

Symbol	Notes
r_n	Random number
$R_{i,j}$	The index value of the i-th round of node N_j,i≥ 1
T_{cur}	Timestamp
S	Shared key of the server, the sensor node and the UAV
K_i	The i-th round session key,i∈N
Y_N,Y_U	Sensor node, UAV authentication message
q_i	4 prime private keys stored by the server,$i = \{1, 2, 3, 4\}$
$\{P_1 P_2 P_3 \ldots P_n\}$	4 prime private keys stored by the server
PKN	Public key stored by the sensor
PKU	Public key stored by the UAV

A. INITIALIZATION PHASE

As shown in Figure 3 below, during the initialization phase of the protocol, the system configures the necessary parameters for all nodes. Specifically, the server first generates four large prime numbers q_1, q_2, q_3, q_4 as private keys. Then, the following parameters are generated.

1) The administrator generates the shared key S, random large prime q_i, where $i = \{1, 2, 3, 4\}$.
2) The administrator stores the shared key S in the memory of N, UAV, and Server.
3) The administrator stores the random large prime number q_i in the memory of Server.

During the initialization phase, all security parameters are not transmitted in the network and are stored locally by the Server.

B. REGISTRATION PHASE

In the registration phase, all sensors and UAV nodes complete registration with the server to achieve initial identification, and construct the information mapping and store.

Specifically, the registration phase contains the following steps, and the system administrator registers N, UAV and AP as follows.

1) For each N, the server generates a unique identity IDN, the initial claim value R_i, the shared key S and the session key K_i. Moreover, the server generates a unique identity IDU_i, for each UAV and a unique identity $IDAP_i$ for each AP, where i is the serial number of the node.
2) The sensor public key PKN and UAV public key PKU are calculated by using the four large prime numbers q_1, q_2, q_3, q_4, generated in the initialization phase, where $PKN = q_3 \times q_4$, $PKU = q_1 \times q_2$.
3) The parameter tuple $(IDN_i, R_{i,j}, S, K_i, PKN)$ is generated for any sensor node N_i and stored in the local memory of the sensor node N_i.
4) Generates the parameter tuple (IDU_i, S, PKU), for any UAV node UAV_i which is stored in the local memory of UAV node UAV_i.
5) Generates a tuple $(IDN_i, IDU_i, S, K_i, q_i)$ for the server and storing it in the local memory of the server.
6) The $IDAP_i$ of any AP node is stored in the local memory of the server.

During the above process, the server key K_i is dynamical. In the registration phase, K_0 is generated as the initial server key. As the authentication process advances, after the successful authentication in round $i-1$, the server will regenerate the server key K_i for round i on the next round of authentication. In addition, for any sensor node N_j, the index value $R_{i,j}$ of round i is determined by the server key of the current round with the identification of the node.

$$R_{i,j} = IDN_j \oplus K_i \tag{1}$$

C. AUTHENTICATION PHASE

When all nodes complete registration, the IoD network begins continuous authentication, the main process of which is shown in Section 2 of this paper. The steps of authentication are described in details as follows.

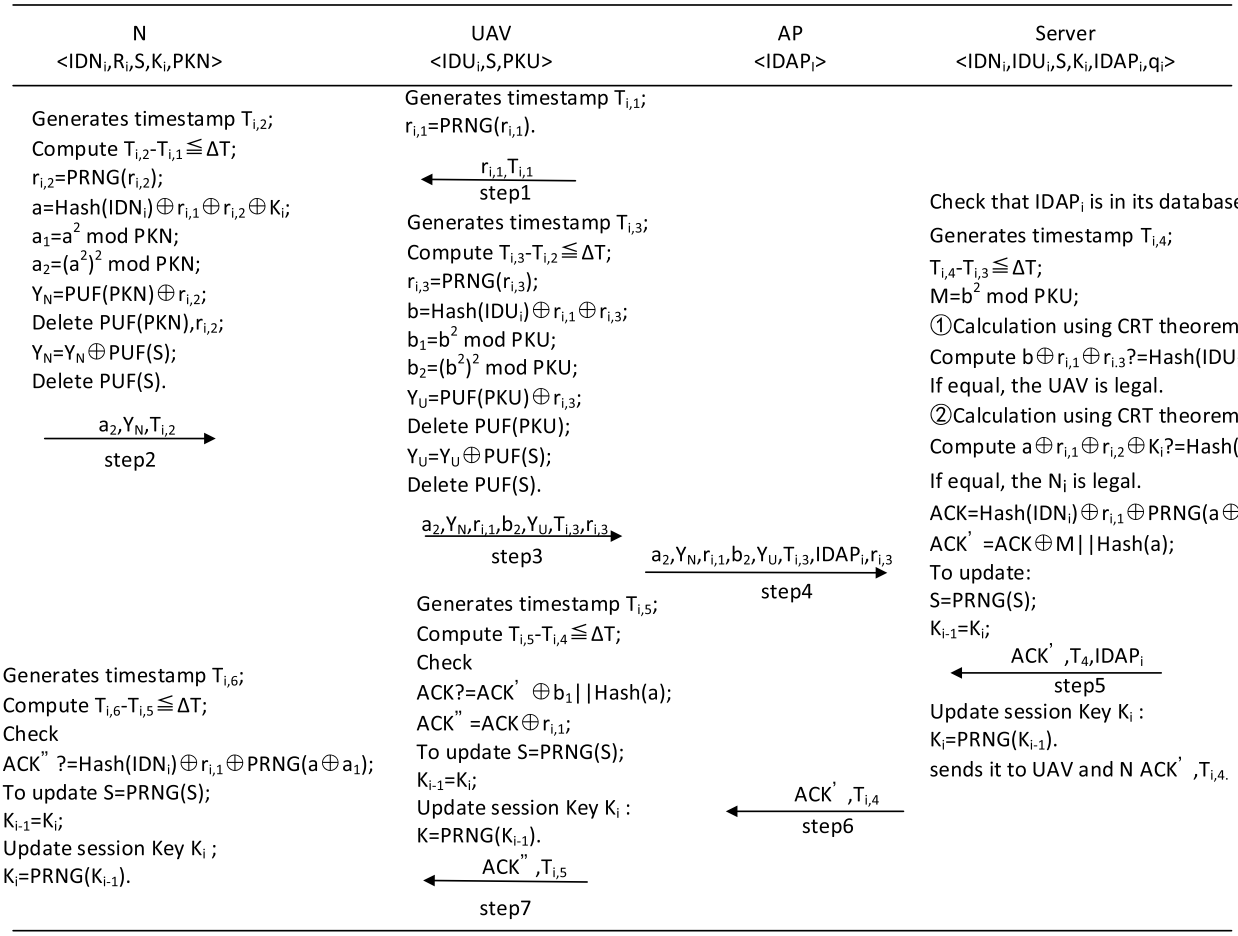

FIGURE 3. Flowchart of the agreement.

1) STEP 1

In round i of the authentication process, the authentication is initiated by the UAV, which generates the random number $r_{i,1}$ and the time stamp $T_{i,1}$ and then broadcasts $r_{i,1}$, $T_{i,1}$ to its own range.

2) STEP 2

When the sensor nodes within the range of the UAV receive the broadcast message from the UAV, they respond to the message.

In response, the sensor first generates the timestamp $T_{i,1}$, and compares the magnitude of the time difference with ΔT, where ΔT is the maximum transmission time difference allowed by the system, and the sensor considers the broadcast message from the UAV as valid when and only when $T_{i,2} - T_{i,1} \leq \Delta$T.

Subsequently, the sensor needs to generate its own authentication information. Most lightweight authentication protocols generally use the method of generating random numbers and performing multiple hash calculations to generate authentication information. However, in practical applications, hash computation requires high computational resources, which is a burden for sensor nodes and UAV nodes with limited resources. Therefore, this paper proposes a method to generate authentication information in sensor and UAV nodes using the Chinese residual theorem.

The Chinese residual theorem is the method for solving a system of one-element congruence equations, which is widely used in cryptography applications. In IoD networks, the Chinese residual theorem can be used to verify the authentication information of mobile nodes by generating parameters on resource-limited mobile nodes (sensors, UAV) and solving them on resource-rich server nodes, thus achieving the transfer of computational load and leaving the complex solving process to the server, while the mobile nodes only need to perform simple computations.

Specifically, taking the sensor node as an example, the sensor generates the random numbers $r_{i,2}$, and calculates the values a, a_1, a_2 after completing the verification of the timestamp.

$$a = Hash\,(\mathrm{IDN_i}) \oplus \mathrm{r}_{i,1} \oplus \mathrm{r}_{i,2} \oplus \mathrm{K_i} \tag{2}$$

$$\mathrm{a_1} = \left(\mathrm{a}^2\right) \bmod \mathrm{PKN} \tag{3}$$

$$\mathrm{a_1}modsqt = (\mathrm{a_{11}}, \mathrm{a_{12}}, \mathrm{a_{13}}, \mathrm{a_{14}}) \tag{4}$$

$$\mathrm{a_2} = \left(\mathrm{a}^2\right)^2 \bmod \mathrm{PKN} \tag{5}$$

Based on the equations above, if the server solves for a_1 by the residue theorem, it will get four modulo square root solutions such as (a_{11}, a_{12}, a_{13}, a_{14}), etc. Therefore, in order to ensure that the server can obtain the authentication information accurately, we use a_2 as the authentication information so that the server can solve for the unique modulo square root solution a.

In this process, the sensor node performs only one hash, and the rest of the computations are simple computations such as modulo and multiplication. Compared with the traditional method of generating information with multiple hashes, the method used in this paper has significant reduction effect on reducing the computational consumption of the mobile node.

And then, the sensor node computes its own session key Y_N, computes PUF(PKN) through PUF circuit and computes $\mathrm{Y_N} = \mathrm{PUF(PKN)} \oplus \mathrm{r}_{i,2}$. Furthermore, the sensor removes the process data $r_{i,2}$ and $PUF(PKN)$ from its own memory, making it impossible for an attacker to obtain the complete key even if he has compromised the sensor, thus truncating the key association between different rounds, achieving both forward and backward security. Finally, as the same form, the sensor performs the second PUF to compute PUF(S) and updates $\mathrm{Y_N} = \mathrm{Y_N} \oplus \mathrm{PUF(S)}$ then removes PUF(S).

The protocol to here is marked as event(SensorToUAV).

The sensor protects the sensor random number $r_{i,2}$ by Y_N,and protects the sensor identifier H(IDN_i) by a_2, which achieves the anonymity and untraceability of the sensor and achieves the purpose of protecting the privacy of the sensor location. Finally, the sensor sends (a_2, Y_N) as interaction information to the UAV through the RF antenna.

3) STEP 3

Similar to Step 2, the UAV follows the following process.

1) Generates the timestamp $T_{i,3}$ and verifies $T_{i,3} - T_{i,2} \leq \Delta\mathrm{T}$?
2) Generates $r_{i,3}$ and computes the values b, $\mathrm{b_1}$, $\mathrm{b_2}$ according to the method of Chinese residual theorem.

$$b = Hash\,(\mathrm{IDN_i}) \oplus \mathrm{r}_{i,2} \oplus \mathrm{r}_{i,3} \oplus \mathrm{K_i} \quad (6)$$

$$\mathrm{b_1} = \left(\mathrm{b}^2\right) \bmod \mathrm{PKN} \quad (7)$$

$$\mathrm{b_1}modsqt = (\mathrm{b_{11}}, \mathrm{b_{12}}, \mathrm{b_{13}}, \mathrm{b_{14}}) \quad (8)$$

$$\mathrm{b_2} = \left(\mathrm{b}^2\right)^2 \bmod \mathrm{PKN} \quad (9)$$

3) Computes its own session key Y_U using PUF method and removes the process parameters $\mathrm{r}_{i,3}$, PUF(PKN), PUF(S)?

$$\mathrm{Y_U} = \mathrm{PUF(PKN)} \oplus \mathrm{r}_{i,3} \quad (10)$$

$$\mathrm{Y_U} = \mathrm{Y_U} \oplus \mathrm{PUF(S)} \quad (11)$$

4) The UAV sends (a_2, Y_N, $r_{i,1}$, $r_{i,3}$, b_2, Y_U, $T_{i,3}$) to the AP.

4) STEP 4

After receiving the authentication information from the UAV, the AP node adds its own ID information and sends (a_2, Y_N, $r_{i,1}$, $r_{i,3}$, b_2, Y_U, $T_{i,3}$, $IDAP_i$) to the server.

5) STEP 5

The server receives the message from the AP, then generates the timestamp $\mathrm{T}_{i,4}$ and verifies $T_{i,3} - T_{i,2} \leq \Delta\mathrm{T}$.

If the verification passes, the server starts to verify the identity of the sensor and the UAV.

First, the server reduces b by b_2 in the message, using a system of linear congruence equations solved by the Chinese residue theorem, as shown in equation 12.

$$\begin{cases} b \equiv b_{11} \,(\mathrm{mod} q_1) \\ b \equiv b_{12} \,(\mathrm{mod} q_2) \\ b \equiv b_{13} \,(\mathrm{mod} q_3) \\ b \equiv b_{14} \,(\mathrm{mod} q_4) \end{cases} \quad (12)$$

The exact calculation is as follows.

1) Computes the product of all moduli $Q = q1 \times q2 \times q3 \times q4$;
2) For the i-th equation:
   Computes $m_i = \frac{Q}{q_i}$;
   Computes the inverse of m_i in the sense of modulo q_i m_i^{-1};
   Computes $c_i = m_i m_i^{-1}$.
3) The unique solution of the equations is $b = \sum_{i=1}^{k} b_{1i} c_i (\mathrm{mod} Q)$.

After restoring b, the server compute $\mathrm{b}_2' = \left(\mathrm{b}^2\right)^2 \bmod \mathrm{PKU} \| \left(\mathrm{b}^2\right)^2 = \mathrm{b_2} \bmod \mathrm{PKU}$ and compares b_2' with $\mathrm{b_2}$ in elimination. if they are equal, the server confirms the identity of the UAV.

The protocol to here is marked as event(UAVToServer)

Subsequently, the server uses a similar approach to compute the value of a based on the received a_2 and verifies the identity of the sensor by calculating a_2' and comparing a_2.

The protocol to this step is marked as event(ServerToUAV)

After all the UAV and sensors are authenticated and legal, they enter the key update phase for the protocol authentication related parameters.

Computes

$$\mathrm{ACK} = \mathrm{Hash}\,(\mathrm{IDN_i}) \oplus \mathrm{r}_{i,1} \oplus \mathrm{PRNG}\,(\mathrm{a} \oplus \mathrm{a_1}) \quad (13)$$

$$\mathrm{ACK}' = \mathrm{ACK} \oplus \mathrm{M} || Hash(\mathrm{a}) \quad (14)$$

Among them, ACK' contains the authentication information of the server in order for the UAV and the sensor to verify the legitimate identity of the server and achieve a more secure two-way authentication. In the end, the server sends the messages ACK', T_4, $IDAP_i$ to the UAV.

6) STEP 6

After receiving the message from the server, the AP verifies the $IDAP_i$ and sends (ACK', $\mathrm{T_4}$) to the correspond UAV.

7) STEP 7

After receiving the message from the AP, the UAV generates the timestamp $T_{i,5}$ and verifies $T_{i,5} - T_{i,4} \leq \Delta T$.

The UAV computes $ACK' \oplus b_1||Hash(a)$ and compares the result with ACK, if equal, the authentication process is considered valid and then updates S=PRNG(S), $K_{i-1} = K$, and computes $ACK'' = ACK \oplus r_{i,1}$.

The protocol to here is marked as event(UpdateSessionKey).

The UAV sends (ACK'', T_5) to the corresponding sensor.

After the sensor receives the message from the UAV, it generates the timestamp $T_{i,6}$ and verifies $T_{i,6} - T_{i,5} \leq \Delta T$.

The sensor computes $Hash\,(IDN_i) \oplus r_1 \oplus PRNG\,(a \oplus a_1)$ and compares the result with the received ACK'' if equal, updates S=PRNG(S), K_{i-1}=K_i and session key K_i=PRNG(K_{i-1}).

The protocol to here is marked as event(end).

The server authenticates successfully and computes K_i=PRNG(K_{i-1}) for the shared key update of the sensor.

IV. SECURITY ANALYSIS

A. PROTOCOL FUNCTIONAL ANALYSIS

This section establishes the security model of the protocol operating environment and attackers with different capabilities based on the Vaudenay model to demonstrate the security and privacy of the protocol in this paper. The following sections analyze that this protocol satisfies forward security and backward security, and is able to resist common attack types such as impersonation attack, denial of service, replay attack, traceability attack and brute force attack.

1) FORWARD SECURITY (FS)

In the calculation of the protocol in this paper, when a complete authentication is performed, the protocol will update the parameters used in the authentication.

However, when the information used in the current round is leaked, the protocol can ensure that the authentication in the next round or the previous round is secure relying on forward security and backward security, which is an important element of security assurance.

Based on Narrow-Strong's attacker capability model, we first consider forward security. Suppose that the attacker gets the internal information and session log of the ith session of the sensor by intruding and listening to the ith session of the sensor, and then the attacker continuously listens to the i+1 session log of the sensor, forward security will analyze whether the attacker can infer the internal information and output information of the ith+1st session of the sensor based on this. The flow of the algorithm for verifying forward untraceability is shown in Figure 4, which can be divided into four aspects.

(1)Attackers obtain the sensor output information and key of the $i+1$ round, and acquires the a^{i+1} value of the sensor in the $i+1$ session through the calculation of equation 15.

$$a^{i+1} = H\,(N_{IDi}) \oplus r_{i,1}^{i+1} \oplus r_{i,2}^{i+1} \oplus K_{i+1} \tag{15}$$

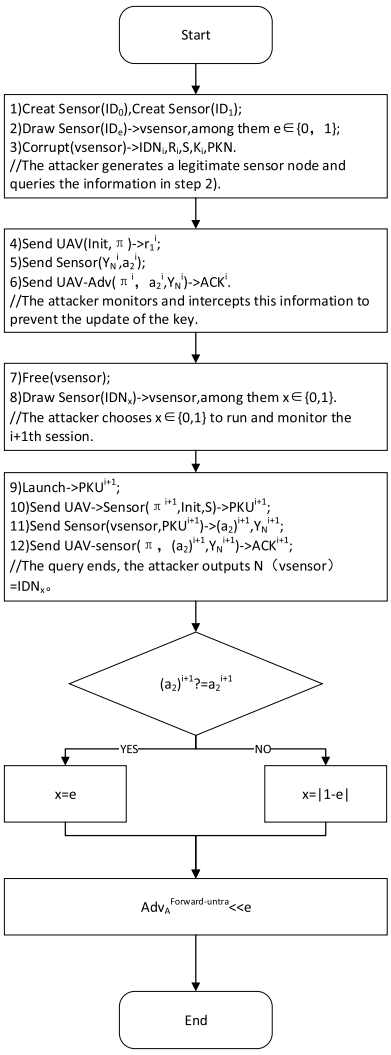

FIGURE 4. Forward security certification.

Meanwhile, the attacker obtains the complete set of information stored in the sensor (N_{IDi}, R_{NID}, S, K, K^{i-1}, m) in the ith round of authentication, and obtains $r_{i,1}^{i+1}$ by listening to the ith+1st sensor session. Since $r_{i,2}^{i+1}$, K^{i+1} is all 128-bit random numbers and K^{i+1} is updated after the i-th authentication by a pseudo-random function such that $K^{i+1} \neq K^i$.We can compute the probability that the attacker successfully simulates $r_{i,2}^{i+1}$, K^{i+1} as follows.

$$\begin{aligned} Adv_A^{Forward-untra}\left(a^{i+1}\right) &= Pr\,(K_{i+1})\,Pr\left(r_{i,2}^{i+1}\right) \\ &= 1/2^{128} \times 1/2^{128} = 1/2^{256} \ll \varepsilon \end{aligned} \tag{16}$$

Thus, the forward security of the protocol can be considered to be satisfied at this point.

(2)Attackers use the i+1st sensor output message $\left(a_2^{i+1}\right)$ to compute

$$(a_2)^{i+1} = \left(\left(a^{i+1}\right)^2\right)^2 \bmod n \tag{17}$$

The value of a^{i+1} is obtained by solving in the reverse direction.

From formula 17, attackers compromise the ith session sensor to obtain internal information n. The security of the protocol depends on the prime decomposition problem based on large integers N. Attackers cannot decompose N in the absence of p_1, p_2 private keys, so attackers cannot solve $(a_1)^{i+1}$ and a^{i+1}.

(3) Attackers listens to the sensor session record, and computes

$$\mathrm{ACK}^{\mathrm{i+1}} = \mathrm{H}(\mathrm{IDN_i}) \oplus \mathrm{r}_{i,1}^{\mathrm{i+1}} \oplus \mathrm{PRNG}(\mathrm{a})^{\mathrm{i+1}} \oplus (\mathrm{a_1})^{\mathrm{i+1}} \quad (18)$$

Solve the i + 1st time to confirm the character ACK^{i+1}.

From equation 18, it can be seen that the attacker compromises the sensor to obtain the internal information $\mathrm{H}(IDN_i)$ and listens to get the i+1st session record $r_{i,1}^{i+1}$. The attacker solving ACK^{i+1} needs to compute.

$$ACK^{i+1} = \mathrm{PRNG}\left(\mathrm{a}^{\mathrm{i+1}} \oplus \mathrm{a_1}\right)^{\mathrm{i+1}} \quad (19)$$

Furthermore, from equation 18, attackers cannot solve for a^{i+1} and (a_1^{i+1}), so attackers cannot trace the confirmation character ACK^{i+1} of the sensor.

(4)Attackers solve the session key Y_N^{i+1} of the i+1st sensor. Specifically, attackers intrude the sensor at the i-th session and the session key Y_N^{i+1} of the sensor in the protocol is generated by the two-step PUF function and the running parameters in the memory are deleted after each step of PUF computation. Attacker solving Y_N^{i+1} requires two physical intrusions into the sensor to obtain the PUF function output values P(PKN) and P(S), respectively. Due to the tamper-proof nature of the PUF function, the attacker's physical intrusion will destroy the structure of the PUF and thus cannot generate the PUF again, and the attacker can only obtain PUF(PKN) or PUF(S). Furthermore, attackers mathematically simulate the PUF circuit output and r_2^{i+1} values to compute the probability of Adv obtaining the sensor output Y_N^{i+1}.

$$\begin{aligned}\mathrm{AdV}_A^{Forvarduntra}\left(Y_N^{i+1}\right) &= \Pr(P(m)) \cap \Pr\left(r_{i,2}^{i+1}\right) \\ &= 1/2^{128} \times 1/2^{128} = 1/2^{256} \ll \varepsilon\end{aligned} \quad (20)$$

Thus, the forward security of the protocol can be considered to be satisfied at this point.

In summary, attackers cannot solve the a^{i+1} of the i+1th session of the sensor and the output information of the sensor $\left((a_2^{i+1}), Y_N^{i+1}, ACK^{i+1}\right)$. Therefore, attackers cannot distinguish the specific sensor in the forward session, and the attacker's advantage of forward tracking $A\ \mathrm{dv}_{\mathrm{A}}^{\mathrm{Forward\text{-}untra}} \ll \varepsilon$, which proves the protocol in this paper satisfies forward untraceability.

2) BACKWARD SECURITY

Narrow-Strong's attacker capability model, this protocol satisfies backward untraceability. The attacker of narrow strong

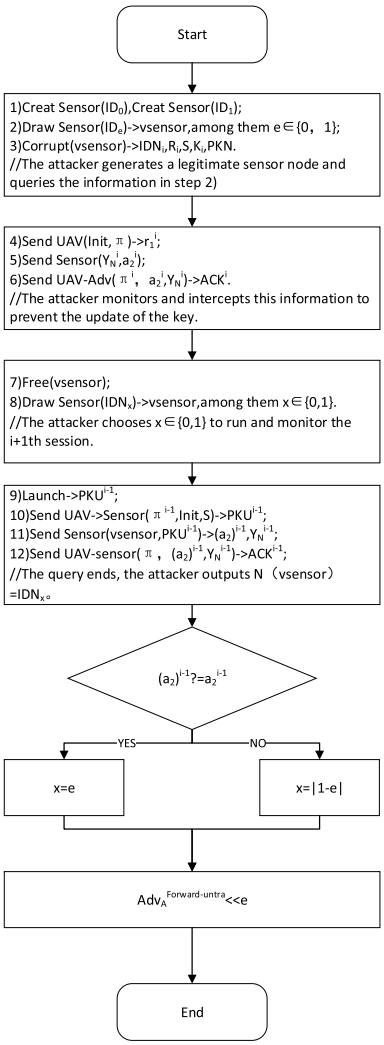

FIGURE 5. Backward security certification.

invades the sensor in the i-th session of the sensor and obtains the sensor internal information $(IDN_i, R_i, S, K_i, K_{i-1}, PKN)$ and the session record. As shown in Figure 5 of the backward untraceability proof flow, the attacker cannot infer the i − 1th sensor key a^{i-1} and the output information $\left((a2^{i-1}), Y_N^{i-1}, ACK^{i-1}\right)$, and thus cannot distinguish specific sensors in the backward session. The attacker's backward tracing advantage is $A\ \mathrm{dv}_{\mathrm{A}}^{\mathrm{Forward\text{-}untra}} \ll \varepsilon$, and the protocol satisfies backward untraceability.

3) IMPERSONATION ATTACK

According to Narrow-Destructive model, the attacker has the ability of impersonation attack.

(1) Narrow-Destructive response of attacker's impersonation label (a_2, K_N), the server does not recognize the attacker as a legitimate label and thus resists the attacker's impersonation attack on the label.

$$a^2 = \left(a^2\right)^2 \bmod PKN \quad (21)$$

$$Y_N = \mathrm{PUF}(PKN) \oplus r_{i,2} \quad (22)$$

In this process, since the attacker is unable to compute the a-value positively and thus a_2. The possibility of the attacker to successfully compute Y_N is analyzed according to the following two aspects.

Aspect 1. The attacker compromises the sensor and obtains the internal information of the sensor.

According to the protocol process, the attacker needs to simulate PUF (PKN) and PUF (S) for compute Y_NPUF function based on the circuit in the manufacturing process exists deviations, and each PUF circuit produces a response sequence with uniqueness and irreproducibility. Therefore, the attacker cannot simulate PUF (PKN) and PUF (S) through mathematical operations attacker's physical intrusion into the tag will be on the tag's PUF circuit cause irreversible damage to the PUF circuit of the tag, resulting in the deactivation of the PUF circuit and $\mathrm{PUF}(\mathrm{a}_1) \neq \mathrm{PUF(a)}$. Furthermore, the attacker's intrusion into the tag cannot obtain all the parameters of the Y_N so that the attacker cannot successfully compute the Y_N.

Aspect 2. The attacker listens to the session records of the wireless channel to impersonate the tag.

The attacker queries the session record between the tag and the reader, and uses the real response of the tag to impersonate the legal tag to interact with the server. Since the tag updates the tag key K_i at the end of each authentication, making $\mathrm{K_{i+1}} \neq \mathrm{K_i}$. Therefore, the attacker cannot use the historical response of the sensor to complete the authentication.

In summary, the protocol can resist the impersonation attack on the sensor by the attacker of Narrow-Destructive.

2) Impersonation attacks of other nodes

The attacker of Narrow-Destructive impersonates the UAV response $\left(\mathrm{a}_2,\ \mathrm{Y_N}, \mathrm{r}_{i,1}, \mathrm{a}_2\ \mathrm{Y_U}\right)$ the server does not identify the attacker as a legitimate UAV, thus resisting the attacker's impersonation attack on the UAV.

The session information between the UAV and the server is not transmitted on the wireless RF channel, and the attacker cannot eavesdrop on the UAV's response and can only impersonate the UAV by hacking the UAV. The attacker can obtain the UAV information (IDU_i, S, PKU) by hacking the reader and writer. In addition, the attacker needs to compute the UAV session key Y_U in order to impersonate the UAV, and requires to compute $\mathrm{Y_U} = \mathrm{PUF(PKU)} \oplus \mathrm{r}_{i,3} \oplus \mathrm{PUF(S)}$.

From the above, it is clear that the attacker cannot simulate the PUF to output PUF(PKU) and PUF(S) through mathematical operations and thus cannot compute Y_U. Therefore, the protocol proposed in this paper can resist the impersonation attacks on UAV by Narrow-Destructive attackers.

Attackers who want to obtain communication data by impersonating an UAV node must be able to compute the specific encryption used in the protocol, but it is almost impossible to calculate the correct value if the attacker does not have access to the node's stored information. Even if the counterfeit is successful, the node will not be able to respond correctly to the node authentication message and cause the protocol to terminate. In this protocol, the random number and the key are encrypted, and the attacker cannot crack the key and the random value even if the impersonation is successful, so it cannot authenticate successfully with the server and the sensor node.

In summary, the protocol can defend against impersonation attacks on servers and UAV by Narrow-Destructive attackers.

4) DENIAL OF SERVICE (DoS)

Under the Narrow-Destructive attacker capability model, the proposed protocol in this paper can resist denial of service from attackers. Attackers cannot cause the shared secret key S between the server and the sensor to be unsynchronized by blocking the channel or forging the acknowledgement character ACK. Asynchronous attacks are considered from two aspects.

(1) The attacker blocks the channel between the UAV and the sensor.

The sensor fails to update the shared secret key S because the sensor does not receive the confirmation character ACK from the UAV. The sensor still uses the previously not updated key $\mathrm{S_{-1}}$ in the next authentication process. However, the protocol in this paper, where the server stores both S and $\mathrm{S_{-1}}$, still enables authentication of the legitimate tag.

(2) The attacker forges an acknowledgement ACK and sends it to the tag.

The attacker forges the ACK in such a way that the tag updates the shared key $\mathrm{S_2}$, which causes the server to permanently reject the tag's authentication request. Moreover, the attacker cannot solve the a and $\mathrm{a_1}$ needed to forge the ACK, so the tag computes $\mathrm{ACK} = \mathrm{Hash}\left(\mathrm{IDN}_i\right) \oplus \mathrm{r}_{i,1} \oplus \mathrm{PRNG}\left(\mathrm{a} \oplus \mathrm{a}_1\right)$ which cannot be verified to pass, and thus the tag does not update the key S.

The previous RFID authentication scheme required sharing and synchronizing data, resulting in the protocol that was vulnerable to denial-of-service attacks. Attackers are able to modify the data transmitted in the protocol so that the two parties communicating in the protocol are out of sync but not detected. However, in the protocol proposed in this paper, the integrity and confidentiality of the random numbers are guaranteed. Moreover, as evidenced above, attackers cannot change the values without obtaining the key. Likewise, attackers cannot update the data when the receiver of the protocol communication is unable to receive the data sent by the sender. However, the protocol proposed in this paper does not suffer from this problem, since this protocol has two data (an old value and an updated value) for both parties when authentication is performed, in this case the protocol sender can still authenticate the other party using the old value.

In our scheme, during the registration phase and authentication phase, if an incorrect IDU_i or IDN_i of the user or UAV is entered in the legitimate user interface, local verification is performed via the check bar. Only after successful verification, the user interface's login request is sent to the server. In addition, new session key updates occurred only after successful verification of the old session key in the authentication phase.Our scheme is secure against denial-of-service attack.

In summary, this protocol can defend against denial of service by attackers.

5) REPLAY ATTACK (RA)

Attackers can obtain all the information transmitted over the channel by eavesdropping on the wireless channel and later use this information to spoof the other party of the authentication by masquerading as a node in the protocol. However, in the protocol proposed in this paper, it is difficult for attackers to forge a valid node to pass the authentication. The reason is that a new random number will be generated in the initial stage of the protocol, and in order to protect the random number, the random number is cryptographically protected in the protocol, which is difficult for an attacker to crack. When an attacker disguise himself as either of the sensor nodes or UAV nodes, the replayed information will not affect the other party of the communication because the updated information is the same as before and does not leak the key, which has strong security.

In summary, this protocol can resist replay attacks by attackers.

6) TRACEABILITY ATTACK

During each authentication process, since all transmitted messages are competing for random numbers, it is known from Theorem 3 that the sensor does not reveal its ID or key. In addition, the pseudonym ID and key is updated after each successful authentication. Even if the attacker knows the internal state of the node, he cannot obtain the previous information of the node because the unavailable values are also encrypted in the protocol using the hash function.

In summary, this protocol can resist traceability attack by attackers.

7) BRUTE FORCE ATTACK

In the protocol, the transmitted information is encrypted and cannot be directly accessed by the attacker. Even if the attacker intercepts the communication data during the communication process, the attacker still cannot obtain the complete random number and crack the complete key because the protocol encrypts the data. Meanwhile, the attacker does not have access to the specific encryption method, so he cannot crack the key to obtaining the key. In these authentication calculations, the protocol achieves three-way bi-directional authentication. To ensure the integrity and stability of the message, the protocol uses one-way hash functions and shift combinations to encrypt the authentication message so that the attacker cannot track the message with random numbers and keys, thus ensuring the integrity and stability of the message. Furthermore, the protocol is also resistant to a variety of attacks, such as impersonation attacks, replay attacks, denial of service, and brute-force attacks, and it has completed forward security. Because lightweight authentication is constrained by low cost, it is theoretically impossible for the attacker to crack under two-way authentication conditions, and the protocol has met the security requirements.

In summary, this protocol can resist brute-force cracking attacks by attackers.

8) MAN-IN-THE-MIDDLE ATTACK

During the registration and authentication phase, the attacker can try to capture and adjust the transmitted messages in Step1, Step2 and Step3 to make other participants believe that the message is true. But in order to perform this task, the attacker is impossible to get the parameters $\{r_{i,1}\}$ for Step1, $\{a, PKN\}$ for Step2 and $\{IDU_i, PKU\}$ for Step3. Therefore, the scheme is able to resist man-in-the-middle attack.

9) USER ANONYMITY

Our scheme uses random $r_{i,n}$ and current timestamp in the registration phase and authentication phase, and in various exchange messages such as step1, Step2 and Step3. For this reason, messages in Step1, Step2, and Step3 etc. is different for each session. Therefore, the attacker cannot track users, servers, and UAV. Moreover, these messages do not directly involve the identification or pseudo-identification of IDN_i, IDU_i and $IDAP_i$, and these are embedded in the conflict-resistant cryptographic one-way hash function. Thus, our scheme provides user anonymity.

B. TOOL VERIFICATION

In addition to the security verification analyzed in the previous section, this paper also uses the tool verification method to verify the security of the protocol. In this paper, Proverif is selected as the verification tool. Although the above analysis shows that our protocol is resistant to common attacks, it does not include all types of attacks. In addition, we can take the advantage of this tool to validate the correspondence assertions, observational equivalences, and reachability properties [28]. ProVerif is an industry-renowned automated analysis tool for authentication algorithms that analyzes the security of authentication protocols and verifies the reliability of encryption. It can handle many different branches of cryptography, such as hash functions, MAC, digital signatures, etc., all of which are within the scope of Proverif's capabilities. Meanwhile, Proverif also has an infinite message space that can generate an infinite number of sessions for a protocol to process, provide false attacks on the protocol to be processed, and automatically perform the security analysis of the protocol. When the ProVerif tool is used to verify a cryptographic protocol, it gives a sequence of attacks if the protocol is vulnerable. As a result, Proverif has been widely used in recent years for protocol verification, and the results are considered to be true and valid.

The variables of the Proverif code of our scheme are defined as shown in the previous Figure 6, and the tripartite agreement of sensor nodes, UAV nodes and servers are converted into the Proverif code as shown in Figures 7 and 8. The events are defined as shown in Figure 9.

In the authentication phase, the sequence of events is an extremely important security objective. For example, suppose

TABLE 3. Security comparison with other protocols.

	[19]	[23]	[24]	[25]	[29]	[26]	[30]	[16]	ours
Anti-Impersonation attack	✓	✓	✓	✓	✓	✓	×	✓	✓
Anti-Replay attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
Anti-Asynchronous attack	×	✓	✓	✓	×	✓	✓	✓	✓
Anti-Brute Force attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
Anti-Traceability attack	✓	✓	✓	×	✓	×	✓	✓	✓
Denial-of-service attack	✓	✓	✓	—	✓	×	×	—	✓
Man-in-the-middle attack	✓	✓	✓	—	—	×	×	×	✓
User anonymity	✓	✓	✓	✓	✓	✓	✓	✓	✓
Forward security	✓	×	×	×	✓	×	×	—	✓
Backward security	—	×	—	×	—	×	×	—	✓

```
(* channel *)
free c1:channel.
free c2:channel.

(* constants *)
free NID:bitstring [private].
free UID:bitstring [private].
free S:bitstring [private].
free K:bitstring [private].
free Kn:bitstring [private].
free Kx:bitstring [private].
free Ku:bitstring [private].
free a:bitstring [private].
free a1:bitstring [private].
free a2:bitstring [private].
free b:bitstring [private].
free b1:bitstring [private].
free b2:bitstring [private].
free r1:bitstring [private].
free r2:bitstring [private].
free r3:bitstring [private].
free m:bitstring [private].
free n:bitstring [private].
free M:bitstring [private].
free Ack:bitstring [private].
free Ackx:bitstring [private].

(* functions, reductions, and equations *)
fun XOR(bitstring, bitstring): bitstring. (* XOR operation *)
equation forall x:bitstring, y:bitstring; XOR(XOR(x,y),y) = x.
fun Hash(bitstring): bitstring. (* hash operation *)
fun PUF(bitstring): bitstring. (* PUF operation *)
fun PRNG(bitstring): bitstring. (* PUF operation *)
fun OR(bitstring,bitstring): bitstring. (* OR operation *)
fun Sqrt(bitstring): bitstring. (* sqrt operation *)
fun CRT(bitstring,bitstring): bitstring. (* CRT operation *)
```

FIGURE 6. Declaration of channels, functions and events.

```
(* process of the UAV *)
let UAV(UID:bitstring,S:bitstring,n:bitstring,Ku:bitstring)=
    let r1 = PRNG(r1) in
 out(c1,(r1));
 in(c1,(a2:bitstring,Kn:bitstring));
    let r3 = PRNG(r3) in
 let b = XOR(XOR(Hash(UID),r1),r3)  in
 let b1 = CRT(Sqrt(b),n) in
    let b2 = CRT(Sqrt(Sqrt(b)),n) in
    let Ku = XOR(PUF(n),r3) in
    let Ku = XOR(PUF(S),Ku) in
 out(c2,(a2,Kn,r1,b2,Ku));
 in(c2,(Ackx:bitstring));
 if(Ack = OR(XOR(Ackx,b1),Hash(a))) then
 let Ack = XOR(Ack,r1) in
 out(c1,(Ack));
  event UpdateSessionKey().

(* process of the Sensor *)
let Sensor(NID:bitstring,S:bitstring,K:bitstring,Kx:bitstring,m:bitstring)=
 in(c1,(r1:bitstring));
 let r2 = PRNG(r2)in
 let a = XOR(XOR(XOR(Hash(NID),r1),r2),K) in
 let a1 = CRT(Sqrt(a),m) in
    let a2 = CRT(Sqrt(Sqrt(a)),m) in
    let Kn = OR(PUF(m),r2) in
    let Kn = XOR(PUF(S),Kn) in
 out(c1,(a2,Kn));
 event SensorToUAV();
 in(c1,(Ack:bitstring));
    if(Ack = XOR(XOR(r1,Hash(NID)),PRNG(XOR(a1,a1)))) then event end().
```

FIGURE 7. UAV and sensor process.

```
(* process of the Server *)
let Server(UID:bitstring,NID:bitstring,S:bitstring,K:bitstring,Kx:bitstring)=
 in(c2,(a2:bitstring,Kn:bitstring,r1:bitstring,b2:bitstring,Ku:bitstring));
    let M = CRT(Sqrt(b),n) in
 if(XOR(XOR(b1,r1),r3) = Hash(UID)) then
 if(XOR(XOR(XOR(a,r1),r2),K) = Hash(NID)) then event UAVToServer();
    let Ack = XOR(XOR(Hash(NID),r1),PRNG(XOR(a,a1))) in
    let Ackx =  OR(XOR(Ack,M),Hash(a)) in
 out(c2,(Ackx));
 let S = PRNG(S) in
 let K = PRNG(K) in
 event ServerToUAV().
```

FIGURE 8. Server process.

that the UAV sends a new session key to the sensor first, and then the server sends a new session key to the UAV, which is clearly illogical. The new session key should be generated first by the server. In other words, the UAV should not get the session key earlier than the server, otherwise it is likely to indicate that the UAV has been compromised. The correct sequence of events should be event (UAV sends new session key) ==> (server generates new session key), where the symbol B ==> A means that event B occurs after event A.

We can summarize the sequence of events as follows, and also check the protocol to find that the sequence of events is fully satisfied.

1) First the sensor responds to the authentication request to the UAV, then the UAV sends the authentication request to the server, expressed as event(UAVToServer) ==> event(SensorToUAV).

```
(* events *)
event SensorToUAV().
event UAVToServer().
event ServerToUAV().
event UpdateSessionKey().
event end().

(* queries *)
query attacker(S).
query attacker(K).
query attacker(Kn).
query attacker(Ku).
query attacker(a2).
query attacker(b2).
query attacker(Ack).

query inj-event(UAVToServer()) ==> inj-event(SensorToUAV()).
query inj-event(ServerToUAV()) ==> inj-event(UAVToServer()).
query inj-event(UpdateSessionKey()) ==> inj-event(ServerToUAV()).
query inj-event(end()) ==> inj-event(UpdateSessionKey()).

process
 ((!UAV(UID,S,n,Ku)) | (!Sensor(NID,S,K,Kx,m)) | (!Server(UID,NID,S,K,Kx)))
```

FIGURE 9. Queries and main process.

```
-- Query not attacker(S[])
Completing...
Starting query not attacker(S[])
RESULT not attacker(S[]) is true.
-- Query not attacker(K[])
Completing...
Starting query not attacker(K[])
RESULT not attacker(K[]) is true.
-- Query not attacker(Kn[])
Completing...
Starting query not attacker(Kn[])
RESULT not attacker(Kn[]) is true.
-- Query not attacker(Ku[])
Completing...
Starting query not attacker(Ku[])
RESULT not attacker(Ku[]) is true.
-- Query not attacker(a2[])
Completing...
Starting query not attacker(a2[])
RESULT not attacker(a2[]) is true.
-- Query not attacker(b2[])
Completing...
Starting query not attacker(b2[])
RESULT not attacker(b2[]) is true.
-- Query not attacker(Ack[])
Completing...
Starting query not attacker(Ack[])
RESULT not attacker(Ack[]) is true.
-- Query inj-event(UAVToServer) ==> inj-event(SensorToUAV)
Completing...
Starting query inj-event(UAVToServer) ==> inj-event(SensorToUAV)
RESULT inj-event(UAVToServer) ==> inj-event(SensorToUAV) is true.
-- Query inj-event(ServerToUAV) ==> inj-event(UAVToServer)
Completing...
Starting query inj-event(ServerToUAV) ==> inj-event(UAVToServer)
RESULT inj-event(ServerToUAV) ==> inj-event(UAVToServer) is true.
-- Query inj-event(UpdateSessionKey) ==> inj-event(ServerToUAV)
Completing...
Starting query inj-event(UpdateSessionKey) ==> inj-event(ServerToUAV)
RESULT inj-event(UpdateSessionKey) ==> inj-event(ServerToUAV) is true.
-- Query inj-event(end) ==> inj-event(UpdateSessionKey)
Completing...
Starting query inj-event(end) ==> inj-event(UpdateSessionKey)
RESULT inj-event(end) ==> inj-event(UpdateSessionKey) is true.
PS D:\proverif2.00>
```

FIGURE 10. Proverif output for the protocol.

2) First the UAV sends an authentication request to the server, and then the server updates the session key, expressed as
event(ServerToUAV) ==> event(UAVToServer).
3) First the server updates the session key, then the UAV updates the key, expressed as
event(UpdateSessionKey) ==> event(ServerToUAV).
4) First the UAV updates the key, then the end of the authentication phase, expressed as
event(end) ==> event(UpdateSessionKey).

The results of the execution of the ProVerif code are shown in Figure 10, where the protocol is simulated for three processes executed in parallel, implementing 17 queries. Three parallel execution processes are successfully started and terminated. Based on the results, it can be proved that the attacker cannot obtain sensitive information such as N_{IDi}, U_{IDi}, K, Setc. Meanwhile, Figure 10 shows that the sequence of events is normal. Therefore, the protocol proposed in this paper can fully satisfy the security requirements of lightweight authentication protocols by satisfying the sequence of events and achieving complete forward and backward security.

C. COMPARISON OF SECURITY WITH OTHER PROTOCOLS

To further compare the safety of the protocols, we selected Ali *et al.* [19], Wazid *et al.* [23], Srinivas *et al.* [24], Dammak *et al.* [25], Das [29], He *et al.* [26], Turkanović *et al.* [30] and Challa *et al.* [16] and made comparisons as shown in Table 3.

In Table 3 below we compared with the earlier proposed working schemes (e.g., those of [19]*et al.*). The comparison is based on several safe and functional properties with Ali *et al.* [19], Wazid *et al.* [23], Srinivas *et al.* [24], Dammak *et al.* [25], Das [29], He *et al.* [26], Turkanović *et al.* [30]and Challa *et al.* [16]. It is clear from the Table 3 that the protocol proposed in this paper has more functional properties and provides better security features than the other schemes.

V. PERFORMANCE ANALYSIS

A. TIME COST

Due to the minimal resources available for human sensors, we have to reduce the cost of computation and storage as much as possible to reduce the cost of the protocol, which is conducive to the popularization and application of the protocol.

To ensure the protocol's security, the confidentiality of the input messages of this protocol are extremely high, and the connections of the transmitted messages in the protocol are controlled. First, the communication messages in the protocol are encrypted. Second, each transmitted message should be as irrelevant as possible in terms of external performance. As a result, in some methods, the adversary cannot decrypt the message through the association among each message.

Regarding the computational cost, the sensor encryption computation in this protocol involves four operations: XOR, AND, the hash function, and physical unclonable function (PUF). The first two operations used in the protocol are low-cost and all the encryption methods used in the protocol are easy to implement on the sensors. It is clear that the operations performed in this protocol are lightweight and can be easily implemented on a low-cost sensor with limited resources. In the following, the performance of the proposed protocol and several common lightweight protocols will be compared.

The operating system of the experimental platform is windows 10 64-bit, the processor of intel core i7-9700F@3.00GHz octa-core, and internal memory of Kingston DDR4 2666 16GB*2=32GB. The time required

to perform the corresponding operations on our experimental platform is set in the database 6×103 tags, and the simulation experiments of search time consumption are performed for the 1st×103, 2×103, 3×103, and 6×103 specific tags to test the time from the server receiving the authentication request for the specific tag to the successful identification in different protocols [31]. Due to the small differences in each run of the computer, the method of testing 10 times to obtain the average value was used as the comparison result. It takes 0.0026ms and 0.0017ms for the sensor and server to compute hash function, and 2.374ms and 2.045ms for the elliptic curve multiplication, respectively.

The experimental results given in Table 4 were used to compute the estimated computational cost of this protocol and other related schemes as shown in Table 7. Ali *et al.* [19], Wazid *et al.* [23], Srinivas *et al.* [24], Dammak *et al.* [25], Das [29], Challa *et al.* [16] requires about 2.4301, 2.4474, 2.4439, 2.4769, 2.4345, and 34.3225 ms, respectively. However, the computational cost required for the UAV is very high because it requires 2Th ≈ 0.0052 ms, which is only a few milliseconds. This is better than the cost Ali *et al.* [19], Wazid *et al.* [23], Srinivas *et al.* [24], Dammak *et al.* [25], Das [29], Challa *et al.* [16]. However, the communication time is larger compared to the He *et al.* [26], Turkanović *et al.* [30]. Furthermore, it is also known from Table 3 that although the scheme of He et al., Turkanović *et al.* requires a lower overall computational effort than the present protocol, the present protocol scheme is more secure than the scheme of He *et al.*, Turkanović *et al.*

TABLE 4. Time cost.

Protocol	(user) Sensor	(Drone) UAV	Server	Total cost
Ali et al. [19]	10Th+Tfe	7Th	7Th	2.4301ms
Wazid et al. [23]	16Th+Tfe	7Th	8Th	2.4474ms
Srinivas et al. [24]	14Th+1Tm	7Th	9Th	2.4439ms
Dammak et al. [25]	16Th	19Th+Tfe	7Th	2.4769ms
Das [29]	Tfe+9Th	11Th	5Th	2.4345ms
He et al. [26]	6Th	10Th	7Th	0.0535ms
Turkanović et al. [30]	7Th	5Th	7Th	0.0431ms
Challa et al. [16]	Tfe+5Tm+5Th	5Tm+4Th	4Tm+3Th	34.3225ms
ours	2Th	2Th	4Th+Tfe	2.0622ms

[1] Th(Hash function):Sensor&UAV=0.0026ms;Server=0.0017ms.
[2] Tm(Multiplication):Sensor&UAV=2.374ms;Server=2.045ms
[3] Tfe(Analog Extractor)≈Tm:Sensor&UAV=2.374ms;Server=2.045ms

B. COMMUNICATION COST

The results of communication bit overhead calculation for each participating part of the protocol is given in Table 5, from which we can know the communication bit overhead to be consumed by each part of the protocol. To demonstrate the effectiveness of this protocol compared to existing schemes, we compare the communication overhead of different participants during the login and authentication phases, where messages are transmitted by the participants. We consider the bit sizes of various parameters such as random number, identity, timestamp, elliptic curve point and the hash output (if we set SHA-1 to h(x)) as 160, 160, 32, (160+160)=320 and 160 bits, respectively. Moreover, the 80-bit key size of symmetric key encryption algorithms (e.g., Double Data Encryption Standard (2DES)) provides the same security as 1024-bit RSA and 160-bit ECC.

TABLE 5. Communication cost.

Protocol	N/SN	UAV/AP	Server	Length(bits)
Ali et al. [19]	512	672	512	1696
Wazid et al. [23]	672	512	512	1696
Srinivas et al. [24]	672	352	512	1536
Dammak et al. [25]	352	832	672	1856
Das [29]	512	384	1088	1984
He et al. [26]	384	704	672	1760
Turkanović et al. [30]	1472	672	576	2720
Challa et al. [16]	512	992	1024	2528
ours	192	352	1024	1568

Communication Bit Cost Table 5 shows the comparative study of the communication cost during login and authentication. During the initialization and authentication phases, this protocol requires four messages, message 1= $\{r_1, T_1\}$, message 2= $\{a_2, K_N, T_2\}$, message 3= $\{a_2, K_N, r_1, b_2, K_U, T_3\}$&$\{T_4, ID_{AP}\}$, of size |MSG1| = (160+32) = 192 bits, |MSG2| = (160+160+32) = 352 bits and |MSG3| = (160+160+160+160+160+160+160+32 +32) = 1024 bits. Then, the total communication cost consumed by this protocol is P3I=1|MSGI| = (192 +352+672+352) = 1568 bits.

Challa *et al.*, Ali *et al.*, Wazid *et al.*, Dammak *et al.*, He *et al.*, Das, Turkanović *et al.* [16], [19], [23], [25], [26], [29], [30] requires communication costs of 1696 bits, 1696 bits, 1600 bits, 1984, 1760, 2720, and 2528 bits, respectively. It is obvious from Table 4 that this protocol requires lower communication cost compared to these schemes. Although Srinivas *et al.* [24] requires lower communication bit cost compared to this protocol, this protocol has lower communication time cost and has higher security.

As we mentioned above, both sensors and UAV in the protocol are resource-constrained devices. Although the UAV can call more resources than the sensor, it is still not suitable for too much computation due to factors such as endurance. Therefore, this protocol assigns most of the heavyweight computations to the server, and resource-constrained drones and human sensing devices only need to perform a small number of computations.

The Ali *et al.*, Wazid *et al.* [19], [23] shows that the computational bits consumed by the three parties of the protocol are relatively balanced. Srinivas *et al.*, Dammak *et al.*, He *et al.* [24]–[26] leaves most of the computation to the UAV and the sensing device, which does not correspond to the resources that can be called by each part of the protocol in real situations, and we believe that these protocols are not applicable in real situations. The communication bit consumption of the individual parts of the protocols in Challa *et al.*, He *et al.* [16], [26] is realistic, but the communication cost required is higher compared to the present protocol.

The computational cost section compares the random numbers, pseudo-random numbers, remainder operations,

hashing, PUF operations, and crossover operations that have large computational costs. The comparison between the proposed protocol and Recent proposed protocols in terms of computational cost and server search cost is shown in Table 5. The tags in the Challa *et al.*, Ali *et al.*, Wazid *et al.*, Srinivas *et al.*, Das [16], [19], [23], [24], [29] require multiple complex encryption of the information and are not applicable to low-cost tags. While the tags in the He et al., Turkanović *et al.* [26], [30] use only hash functions, pseudo-random numbers, remainder and crossover operations. Compared with references [24], [25], this paper proposes a protocol that needs to generate 3 times of pseudo-random number, 2 times of complement and 2 times of PUF operations. However, the PUF function protects the tag key and effectively resists counterfeit attacks, and the protocol achieves higher security.

In summary, the protocol proposed in this paper achieves the transfer of computational load from mobile nodes to servers while ensuring security, thus improving the system sustainability.

VI. CONCLUSION

In this paper, a lightweight authentication protocol is proposed for UAV networks. The protocol optimizes the authentication process while ensuring forward security and backward security, and resists attacks such as impersonation attacks and replay attacks. Meanwhile, by applying the Chinese residual theorem in the protocol, the computational scale of sensor nodes and UAV nodes is reduced, and the transfer of the computational process from mobile nodes to servers are achieved without compromising security. Through the protocol analysis and ProVerif tool, this paper verifies that the protocol has sufficient security. Furthermore, the protocol in this paper significantly optimizes the utilization of computing resources. To sum up, the protocol in this paper can effectively optimize the authentication process of IOD networks.

REFERENCES

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

[2] C. Pu and L. Carpenter, "*Psched*: A priority-based service scheduling scheme for the Internet of drones," *IEEE Syst. J.*, early access, Jun. 11, 2020, doi: 10.1109/JSYST.2020.2998010.

[3] I. Kovalev, A. Voroshilova, and M. Karaseva, "Analysis of the current situation and development trend of the international cargo UAVs market," *J. Phys., Conf. Ser.*, vol. 1399, Dec. 2019, Art. no. 055095.

[4] 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.

[5] M. O. Ozmen and A. Attila Yavuz, "Dronecrypt-an efficient cryptographic framework for small aerial drones," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 1–6.

[6] S. Ponde and S. Lomte, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 21st Annu. Joint Conf. Comput. Commun. Soc.*, Jun. 2020, pp. 1567–1576.

[7] X. Yao, X. Han, and X. Du, "A light-weight certificate-less public key cryptography scheme based on ecc," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Oct. 2014, pp. 1–8, 2014.

[8] D. Püllen, N. A. Anagnostopoulos, T. Arul, and S. Katzenbeisser, "Using implicit certification to efficiently establish authenticated group keys for in-vehicle networks," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2019, pp. 1–8.

[9] W. Liang, S. Xie, J. Long, K.-C. Li, D. Zhang, and K. Li, "A double puf-based rfid identity authentication protocol in service-centric Internet of Things environments," *Inf. Sci.*, vol. 503, pp. 129–147, Dec. 2019.

[10] D. P. Shepard, J. A. Bhatti, T. E. Humphreys, and A. A. Fansler, "Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks," in *Proc. 25th Int. Tech. Meeting Satellite Division Inst. Navigat.*, Sep. 2012, pp. 3591–3605.

[11] Q. Yan, Q. Gong, and F.-A. Deng, "Detection of DDoS attacks against wireless SDN controllers based on the fuzzy synthetic evaluation decision-making model," *Adhoc Sensor Wireless Netw.*, vol. 33, pp. 275–299, Jul. 2016.

[12] T. Li, J. Ma, X. Ma, C. Gao, and J. Zhang, "Lightweight secure communication mechanism towards UAV networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[13] C. Pu and Y. Li, "Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN*, Jul. 2020, pp. 1–6.

[14] R. Amin, S. K. Hafizul Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[15] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[16] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E. J. Yoon, and K. Y. Yoo, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[17] 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, Oct. 2020.

[18] S. Barman, H. P. H. Shum, S. Chattopadhyay, and D. Samanta, "A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme," *IEEE Access*, vol. 7, pp. 12557–12574, 2019.

[19] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–43724, 2020.

[20] Z. Ali, S. Hussain, R. H. U. Rehman, A. Munshi, M. Liaqat, N. Kumar, and S. A. Chaudhry, "Itssaka-ms: An improved three-factor symmetric-key based secure aka scheme for multi-server environments," *IEEE Access*, vol. 8, pp. 107993–108003, 2020.

[21] Son, Yunmok, Noh, Juhwan, Choi, Jaeyeong, Kim, and Yongdae, "Gyros-Finger: Fingerprinting drones for location tracking based on the outputs of MEMS gyroscopes," *Acm Trans. Privacy Secur.*, vol. 21 no. 5, pp. 1–25, 2018.

[22] Y.-M. Kwon, J. Yu, B.-M. Cho, Y. Eun, and K.-J. Park, "Empirical analysis of MAVLink protocol vulnerability for attacking unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 43203–43212, 2018.

[23] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

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

[25] M. Dammak, O. R. M. Boudia, M. A. Messous, S. M. Senouci, and C. Gransart, "Token-based lightweight authentication to secure iot networks," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Mar. 2019, pp. 1–4.

[26] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inf. Sci.*, vol. 321, pp. 263–277, Nov. 2015.

[27] Y. Ding, X. Zhao, D. Zhang, D. Liang, Z. Wang, S. Xi, and S. Du, "Rice lodging area extraction based on ycbcr spatial and texture features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 9228–9231.

[28] D. Abbasinezhad-Mood and M. Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended chebyshev chaotic maps," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4815–4828, Nov. 2018.
[29] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer Netw. Appl.*, vol. 9, no. 1, pp. 223–244, 2016.
[30] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.
[31] F. Armknecht, R. Maes, A. R. Sadeghi, C. Wachsmann, and F. Standaert, "A formalization of the security features of physical functions," in *Proc. IEEE Symp. Secur. Privacy*, May 2011, pp. 397–412.

YUAN LEI received the bachelor's degree from Hunan University, and the master's degree from Guangxi University. He is currently an Associate Professor with the Guangxi University of Foreign Languages. His research interests include information security, data mining, and machine learning.

LINING ZENG received the Ph.D. degree from Hunan University. He is currently working with the Hunan University of Finance and Economics. His research interests include urban logistics, logistics informatization, logistics path planning, and intelligent warehousing.

YAN-XING LI received the bachelor's degree from Dalian Minzu University, and the master's degree from Guangxi University. He is currently an Associate Professor with the Guangxi University of Foreign Languages. His research interests include the Internet of Things and information security.

MEI-XIA WANG is currently pursuing the degree with the Guangxi University of Foreign Languages. Her research interests include machine learning, data mining, and information security.

HAISHENG QIN is currently the Dean of the School of Information Engineering, Guangxi University of Foreign Languages. He has received the second prize of Guangxi Science and Technology Progress Award, the third prize of Guangxi Science and Technology Invention Award, and the first prize of Nanning Science and Technology Progress Award. His main research interests include computer networks, information security, and database application.