




## Research Article

# LAAP: Lightweight Anonymous Authentication Protocol for IoT Edge Devices Based on Elliptic Curve

Xinghui Zhu,<sup>1,2</sup> Zhong Ren,<sup>1</sup> Ji He ,<sup>1,2,3,4</sup> Baoquan Ren,<sup>4</sup> Shuangrui Zhao ,<sup>1,2</sup> and Pinchang Zhang <sup>5</sup>

<sup>1</sup>School of Computer Science and Technology, Xidian University, Xi'an 710071, China

<sup>2</sup>Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an 710071, China

<sup>3</sup>Guangzhou Institute of Technology, Xidian University, 510555 Guangzhou, China

<sup>4</sup>Institute of Systems General, Academy of Systems Engineering, Academy of Military Sciences, Beijing 100101, China

<sup>5</sup>School of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China

Correspondence should be addressed to Ji He; [garyhej1991@gmail.com](mailto:garyhej1991@gmail.com)

Received 24 June 2022; Revised 17 August 2022; Accepted 5 September 2022; Published 22 September 2022

Academic Editor: Kechen Zheng

Copyright © 2022 Xinghui Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The massive heterogeneous devices and open channels of the Internet of Things (IoT) lead to low efficiency and privacy leakage in the authentication process, which brings great challenges to identity authentication. This paper focuses on the anonymous authentication between the IoT edge device and the cloud server. In this work, we first propose a novel lightweight anonymous authentication protocol (LAAP) to meet security and efficiency requirements. Especially, the proposed protocol uses dynamic pseudonyms to prevent the traceable attacks caused by fixed identity identification and also uses symmetric encryption to optimize the server's search for anonymous device information, and the time complexity is reduced from  $O(n)$  to  $O(1)$ . Then, the formal security analysis and informal security analysis are provided to prove the security of the proposed protocol. Finally, extensive numerical results indicate that the proposed LAAP protocol is superior to the benchmarks in terms of computing overhead and communication overhead, while the storage overhead is consistent with the lowest level among other protocols.

## 1. Introduction

The Internet of Things (IoT) aims to connect massive sensing devices through wireless networks to realize information interaction between the physical world and the virtual world. With the wide application of IoT, it has been involved in all walks of life, such as the Internet of Vehicles, Internet of Medical Things, and Smart City. According to GSMA forecast, the number of IoT devices worldwide will reach about 23.3 billion in 2025 [1]. Due to the limited storage, computing, communication, and power capabilities of IoT sensing devices, combining edge-embedded devices with cloud computing creates a new paradigm called CloudIoT [2]. Under this paradigm, embedded devices can rely on the processing power of cloud computing and use various services provided by cloud computing. However, when an embedded device

establishes a communication connection with a cloud server, security is the primary concern.

In recent years, privacy and data security issues caused by IoT terminal devices have frequently occurred. On June 8, 2020, security experts disclosed a new UPnP vulnerability named "Call Stranger" [3], which affects the security of billions of devices, including the TV and network equipment of ASUS, Belkin, Dell, Samsung, TP-Link, and other companies. The vulnerability could be exploited by a remote, unauthenticated attacker. In September 2021, researchers discovered a high-level security vulnerability CVE-2021-36260 in Hikvision IP camera/NVR device firmware. The attack can fully control the device through the shell and obtain any information of the owner and further laterally attack the internal network without leaving any daily login information [4]. The report released by the Unit 42 team

[5] shows that 98% of IoT devices leak user privacy due to unencrypted traffic, 57% of devices are vulnerable to moderate or severe attacks, and devices have become the preferred target for attackers.

Authentication can guarantee the identity legitimacy of communication parties in the IoT and is a key technology to solve security problems. The authentication process usually involves two parts, i.e., identity authentication and key negotiation. Identity authentication is to ensure the legitimacy of the identities of both communication parties. Key negotiation is used to establish a session key for subsequent security access and secure data transmission. Note that security authentication protocols need to consider the following principles: (1) for lightweight, most IoT devices cannot support complex authentication protocols because of their limited computing resource [6]; (2) for privacy protection, during the interaction of the device, the advanced techniques (e.g., anonymity and blockchain) need to be adopted to prevent malicious attackers from obtaining the private information of the devices and users [7].

*1.1. Related Work.* To implement the security authentication of IoT devices, various protocols and methods were studied. Kalra and Sood in [8] proposed a two-way authentication scheme to realize mutual authentication and meet essential security requirements. Considering the security defects and structural problems of the protocol [8], the improved protocols in [9, 10] were proposed to defend against server emulation attacks. Rostampour et al. then proposed a privacy-preserving anonymous authentication protocol named ECC-BAP in [11]; the results indicated that the proposed protocol can achieve the untraceable purpose by traversing the registry. Then, the authors in [12] proposed an authentication protocol based on bilinear pairing to solve the problems of privacy protection and authentication table theft. Subsequently, the enhanced IoT mutual authentication protocol and improved ECC-based authentication protocol were proposed in [13, 14], respectively. To defend against more types of attacks including known temporary information attacks, DoS attacks, Panda and Chattopadhyay proposed an anonymous authentication scheme integrating a password validator in [15]. Further, Bhuarya et al. in [16] proposed an enhanced authentication scheme to defend known session-specific temporary information attack, where the hypertext transfer protocol (HTTP) cookies were used to authenticate clients. However, the large exponential powers employed by these protocols leads to a large amount of computation.

The dynamic pseudonym is an effective method to solve traceable problems. A pseudonym ID is used to conduct authentication between client and server and is dynamically updated after completion of authentication. Das et al. first proposed an authentication protocol [17] where dynamic ID technology was used to avoid the risk of ID theft. However, the protocol suffers from smart card theft attacks. Then, Jiang and Das et al. devoted to solving the problem in [18, 19], respectively. Notice when the above schemes are attacked asynchronously, i.e., the attacker blocks the exchange messages of the authentication protocol, and the interaction between the authentication parties is out of sync,

such that the protocol cannot work. Thus, Gope et al. in [20–23] studied the authentication scheme based on emergency ID and secret key technology to solve the problem of asynchronous attack. In such schemes, clients and servers share a set of emergency IDs and keys in addition to dynamic pseudonyms. Once the dynamic pseudonyms are out of sync, emergency IDs and keys are used to interact. However, the emergency IDs and keys occupy a large amount of storage space, and once the emergency ID and emergency key are used up, the device must be reregistered. Recently, some researchers devoted to designing grant-free access scheme for M2M communications [24] and used the advanced methods to realize the authentication, e.g., deep learning [25, 26] and blockchain [27, 28]. However, they are not suitable for IoT devices with limited computing overhead.

*1.2. Motivation and Contribution.* To sum up, it can be found that the mentioned authentication protocols based on identity and pseudonym do not consider the privacy protection and cannot resist traceable attacks. For example, if the long-term key is leaked, the attacker can simulate the session key negotiation between the terminal and the server and occupy the position of the legitimate device. As a result, the legitimate device cannot carry out normal session key negotiation. Furthermore, during authentication process, the server needs to traverse the password verifier table to find the relevant registration information, and the search time increases linearly with the number of devices. The protocols with privacy protection cannot take into account both authentication efficiency and security while realizing anonymity. Dynamic pseudonym schemes are vulnerable to asynchronous attacks. Therefore, this paper focuses on the authentication between the edge server and device in the IoT and designs a new authentication protocol to realize the privacy protection and improve the efficiency of authentication. Our main contributions can be summarized as follows:

- (i) We propose a lightweight anonymous authentication protocol (LAAP) based on elliptic curve cryptography (ECC) to implement security authentication between the servers and devices. Dynamic pseudonym is used to defend against traceable attacks caused by fixed identity identification. Besides, symmetric encryption is used to optimize the server's search for anonymous device information, and the time complexity is reduced from  $O(n)$  to  $O(1)$
- (ii) We provide the formal analysis and informal analysis to validate the security of the proposed protocol. The analysis shows that the proposed protocol can satisfy the anonymity and defend against asynchronous attacks. We also perform random oracle models and AVISPA Tool to prove the security of the certification process
- (iii) We provide extensive simulation results to testify the authentication efficiency of the proposed

protocol. The results indicate that the proposed scheme outperforms the benchmarks in terms of computation overhead, communication overhead, and storage overhead

Organization of this paper is as follows: In Section 2, we introduce the preliminaries. In Section 3, we present details of our proposed authentication protocol. Security analysis and performance evaluation are given in Sections 4 and 5, respectively. At last, Section 6 offers our conclusions and potential future works.

## 2. Preliminaries

In this section, we first introduce the system model and then introduce the related elliptic curve cryptography and the corresponding mathematical problems.

**2.1. System Model.** In this work, we focus on the authentication between cloud server ( $S$ ) and embedded devices ( $D$ ) in the IoT shown as Figure 1. The embedded device can be small devices (e.g., environmental sensors, cameras, and smart meters) or large-scale devices (e.g., intelligent vehicles and smart charging piles). The cloud server has powerful computing resources and storage resources, so that it can provide various services for embedded devices. For example, in mobile edge computing networks [29], the device first uploads data to the cloud server and then uses the computing resources to process its data. The cloud server also provides a bootstrap program for the system, enabling authentication to be performed smoothly. Before providing these services, they authenticate between the device and server to ensure legitimate access via wireless channels.

**2.2. Elliptic Curve Cryptography.** The security properties of ECC are mainly based on the intractable problem of discrete logarithms in elliptic curves. Given a prime field  $\mathbb{F}_p$ , the elliptic curve point is set  $E_p(a, b)$  on the finite field can be expressed as

$$E_p(a, b): \{(a, b) | y^2 = x^3 + ax + b \pmod{p}, x, y \in \mathbb{F}_p, 4a^3 + 27b^2 \pmod{p} \neq 0\} \cup \{O\}, \quad (1)$$

where  $a, b \in \mathbb{F}_p$ , the prime number  $p (p > 3)$  is the order of the finite field, and  $O$  represents the infinity point. The Ellipse Curve Discrete Logarithm Problem (ECDLP) can be described as follows:

**Definition 1.** ECDLP: let  $\mathbb{G}$  denote the cyclic group generated by the base point  $G$  and the operation rules of Abelian groups on the elliptic curve  $E_p(a, b)$ . For a given  $P, Q \in \mathbb{G}$ , if  $Q = kP$ , where  $k \in \mathbb{Z}_p^*$ ,  $k$  cannot be solved in polynomial time, which is usually used as the private key.

Based on the ellipse curve and the ECDLP, the security of the ECC can be described as the Ellipse Curve Computation Diffie-Hellman Problem (ECCDHP), which is defined as follows.

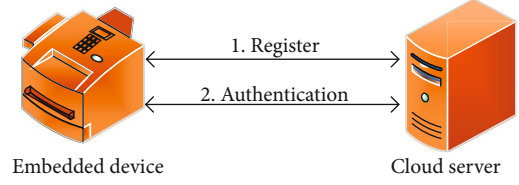


FIGURE 1: System model.

**Definition 2.** ECCDHP: let  $\mathbb{G}$  be the cyclic group generated by the base point  $G$  and the operation rules of Abelian groups on the elliptic curve  $E_p(a, b)$ . For  $P, Q, R \in \mathbb{G}$ , if  $Q = xP$  and  $R = yP$ , where  $x, y \in \mathbb{Z}$ , we have that computing  $xyP$  in polynomial time is a hard problem.

**2.3. Random Oracle Model.** Random oracle model (ROM) is proposed by Bellare and Rogaway [30], which made the provable security methodology that was purely theoretical research in the past make significant progress in practical applications. When applying the ROM, the necessary work is to establish a security model that treats different subjects as random oracles (RO). The RO has the following three characteristics: (1) consistency: for the same query, RO will always return the same output; (2) computability: for different queries, RO can obtain results and return them in polynomial time; and (3) uniform distribution: for different queries, the output of RO is evenly distributed in the value space without collision that the output obtained by different queries is always different.

To prove the security of the model, it is necessary to establish an attacker  $\mathcal{A}$  for the model and to provide the attacker with a simulated environment indistinguishable from the actual environment. For  $\mathcal{A}$ , the complexity and safety of the model boil down to mathematical computational difficulties (e.g., large factorization, ECDLP, and ECCDHP). In ROM, the convention judgement appears as

- (1) Formally define the security of the scheme, assuming that the attacker can destroy the security of the protocol with a nonnegligible probability in polynomial time
- (2) The attacker simulates the real environment by querying different random oracles
- (3) The way of attacking the attacker and the result boils down to solving a mathematical problem

Although the ROM methodology cannot be used as absolute proof that the actual solution is safe, it can still be a necessary basic safety test. Thus, this paper adopts it to validate the security of the proposed protocol.

## 3. Design of the Authentication Protocol

In this section, we introduce the proposed LAAP protocol including three phases, i.e., initialization phase, registration phase, and authentication phase. A summary of the notations used in this article is provided in Table 1.

TABLE 1: List of the related notations.

Notation	Description
$D_i, ID_i$	The $i$ device and its device $ID$
$S$	Cloud server
$x$	Cloud server symmetric key
$PK_S, K$	Server public key, private key
$NID$	Device pseudonym identification
$NID'$	Updated pseudonym identification
$DID$	Device real identity
$DID'$	Device updated real identity
$Sync$	Server sync value
$N_1, N_2, R_i$	Random number
$\mathbb{G}$	Cyclic additive group of order $q$
$S_i$	Device-side hash chain value
$S_{ii}$	Server-side hash chain value
$h()$	One-way hash function
$\parallel$	Connect operation
$SK$	Session key

**3.1. Initialization Phase.** Before authentication, the server needs to perform necessary parameter initialization operations. Initialization parameters are divided into public parameters and private parameters. The server selects an elliptic curve  $E$  based on the finite prime field  $\mathbb{F}_p$  and selects the additive group of curve  $E$  of the order  $q$ . Then, the public key of the server  $PK_S$  can be calculated as  $PK_S = K \times G$ , where  $K(K \in \mathbb{Z}_q^*)$  is the private key and  $G$  is the generator of the group  $\mathbb{G}$ . The server also needs to select a suitable one-way hash function  $h : \{0, 1\}^* \rightarrow \{0, 1\}^{l_h}$ , where the input is any length binary string and the output is a binary string of fixed length  $l_h$ . The server generates a random key  $x$  as a symmetric encryption key and selects an appropriate symmetric encryption algorithm as the basic algorithm for device identification update. The server publishes the parameters  $\langle \mathbb{G}, PK_S, G, h \rangle$  as public parameters and stores  $\langle x, K \rangle$  as private parameters.

**3.2. Registration Phase.** Before starting key negotiation, the device first needs to complete the registration. The LAAP protocol can ensure that the registration is completed by the public channel, and check whether the message responded by the server is legal. The registration process can be divided into three steps described as Figure 2. In the following, we will detail the three steps.

*Step 1.* The device  $D$  first selects a unique  $ID$ , which is only known by the device. Then, the device generates a random number  $N_1$  to randomize its  $ID$  and calculates  $PID, Z_1, Z_2$  and  $PPID$  as  $PID = h(ID \parallel N_1), Z_1 = N_1 \times G, Z_2 = N_1 \times PK_S$  and  $PPID = PID \oplus Z_2$ , respectively. Finally, the device sends the message  $\langle Z_1, PPID \rangle$  to the server through the public channel.

*Step 2.* After receiving the registration information  $\langle Z_1, PPID \rangle$  from  $D$ , server  $S$  uses the key  $K$  to restore the data, where  $Z_2^*$  and  $PID$  are calculated according to  $Z_2^* = Z_1 \times K, PID = PPID \oplus Z_2^*$ , respectively. After restoring  $PID$ ,  $T_i$  is generated based on the device registration identity as  $T_i = h(R_i \parallel PID \parallel h(K))$ , which is used to calculate the identity information for each authentication of the device. Otherwise, the server also generates a new identity  $PID_{new} = h(PID \parallel R_i)$  for the device as the identity of the authentication stage. The server will save two IDs for each device, i.e.,  $PID_{new}$  and  $PID_{old}$ , where  $PID_{new}$  is the new device ID and  $PID_{old}$  is the old ID of the previous session. Then, the hash value of  $PID$  is used as the initial value of the hash chain, which will be updated during each session key negotiation process.

Subsequently, the server uses the hash value of the key  $K$  to encrypt  $T_i$  and  $S_i$  and stores the results in its database. After that, the server calculates the message of the response device as  $Z_3 = (NID \parallel PID) \oplus S_i$  and  $Z_4 = (T_i \parallel PID) \oplus S_i$ . Finally, the server sends  $\langle Z_3, Z_4 \rangle$  to the device through the public channel and stores  $\langle PID_{new}, PID_{old}, U_i, X_i, Sync \rangle$  as the registration information corresponding to the device.

*Step 3.* After receiving the message  $\langle Z_3, Z_4 \rangle$  from the server, the device uses the initial value of the hash chain to restore and verify the data as  $(NID \parallel PID') = Z_3 \oplus S_i, (T_i \parallel PID') = Z_4 \oplus S_i$ . By splitting the data, the device verifies whether the decrypted  $PID'$  is the same as the  $PID$  saved by itself. If they are the same, the device confirms that the message was sent by the server and stores  $\langle NID, T_i \rangle$  as the registration information for subsequent identity authentication and key negotiation.

**3.3. Authentication Phase.** When the device wants to upload data or access the server, the device and server need to complete identity authentication and key negotiation. The authentication process can be divided into four steps described as Figure 3. In the following, we will introduce the four steps for details.

*Step 1.* Device  $D$  first generates a random number  $N_1$  and calculates  $P_1 = N_1 \times G, CK_i' = h(T_i \parallel S_i) \times G$ , and  $A_i = CK_i' \times N_1$ , where  $CK_i'$  and  $A_i$  are temporary values generated by the synchronization hash chain and the identity information in the registration phase. Then, the verification message  $P_2$  is generated as  $P_2 = h(A_i \parallel S_i \parallel P_1)$ . On the one hand, it can prevent message tampering, and on the other hand, it can verify whether the identity is legitimate. Finally, the device sends the message  $\langle P_1, P_2, NID \rangle$  to the server for authentication.

*Step 2.* After receiving the message  $\langle P_1, P_2, NID \rangle$ ,  $S$  restores the identity information  $DID$  based on the symmetric key  $x$ . If the recovery is successful,  $S$  can judge the recovery based on the matching information of the database; otherwise, terminate the session. When  $DID = PID_{new}$ , the server updates the synchronization hash value as  $S_{ii} = h(S_{ii} \parallel Sync)$  and reconstructs the authentication temporary value for this session as  $CK_i = h(T_i \parallel S_{ii}), A_i' = CK_i \times P_1$ . Then,  $S$  calculates the verification message  $P_2'$  as  $P_2' = h(A_i' \parallel S_{ii} \parallel P_1)$ . If  $P_2 \neq P_2'$ , it

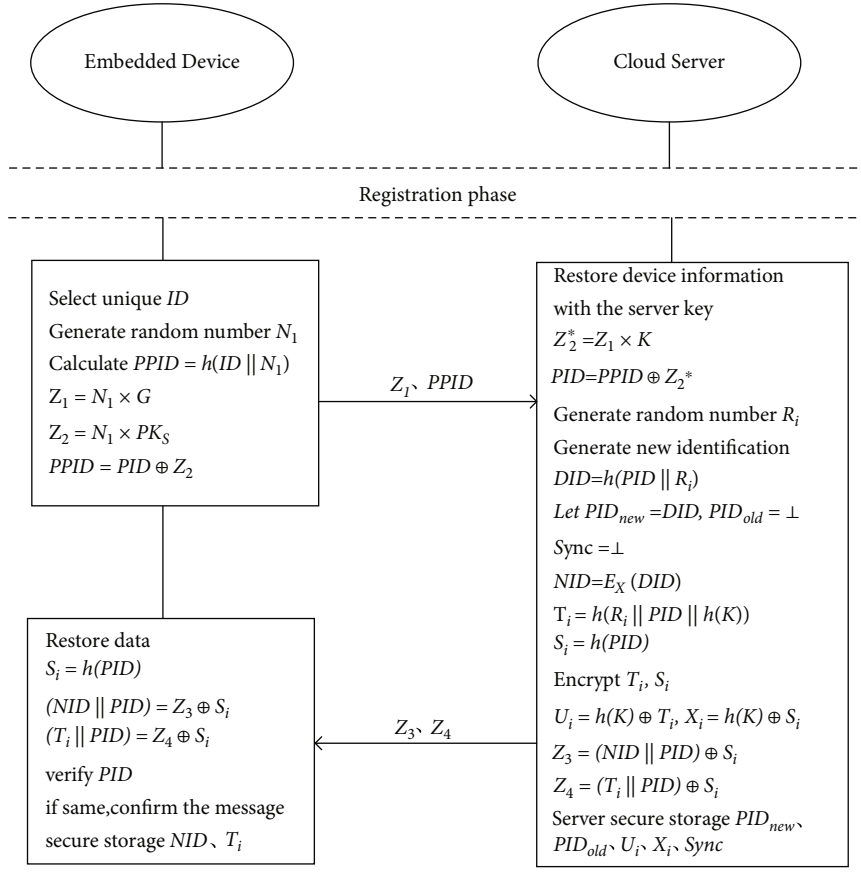


FIGURE 2: Illustration of registration phase of LAAP.

means that the message has been tampered with and the session is terminated. Otherwise,  $S$  generates a random number  $N_2$  and a new identity for  $D$  as  $DID' = h(DID || N_2)$ . Finally,  $S$  calculates the verification messages as  $P_3 = NID' \oplus S_{ii}$ ,  $P_4 = h(NID' || S_{ii} || A_i')$  and sends them to  $D$ .

*Step 3.* After receiving the message  $\langle P_3, P_4 \rangle$ ,  $D$  decrypts and verifies the new ID  $P_4' = h(NID' || S_{ii} || A_i')$  where  $NID' = S_i \oplus P_3$ . If  $P_4 \neq P_4'$ , the message verification fails, and the session is terminated; otherwise,  $D$  calculates the negotiated session key and the verification message as  $SK = h(A_i || NID' || NID)$ ,  $P_5 = h(SK || NID' || NID)$ . Finally, the device sends the message  $\langle P_5 \rangle$  to the server.

*Step 4.* After receiving the message  $\langle P_5 \rangle$ ,  $S$  can calculate the negotiated session key and verify the message as  $SK = h(A_i' || PID_{new} || PID_{old})$  and  $P_5' = h(SK || PID_{new} || PID_{old})$ , respectively. If  $P_5 \neq P_5'$ , the server terminate the session, otherwise, the session key negotiation is successful.

#### 4. Security Analysis

In this section, we provide the formal analysis and informal analysis to validate the security of the proposed protocol.

*4.1. Formal Security Proof.* For the formal analysis, we first adopts the widely accepted ROM [30] to verify the security of the proposed protocol.

*4.1.1. Formal Security Proof with ROM.* The extended RO is described as follows:  $TestID(D_i, NID_i, OID_i)$  is used to query the real identity information of the device, where  $NI D_i$  and  $OID_i$  represent the identity after session key negotiation and the identity before session key negotiation, respectively.  $Corrupt(D_i, a)$  is used to query the secret information of the device and simulate the attack of the device being stolen.

There are two participants  $\Pi$  in this work, i.e., server  $S$  and the embedded device  $D$ . Each participant has multiple instances (i.e., ROs). Let  $D_i$  and  $S_i$  represent the  $i$ -th instance of them, respectively;  $DID_i$  and  $SID_i$  represent the identity of  $D_i$  and  $S_i$  that are used to negotiate the session key, respectively;  $NID_i$  and  $OID_i$  indicate the updated ID and the pre-updated ID of  $D$ , respectively;  $H_D^i$  and  $H_S^i$ , respectively, represent the hash chain status of  $D$  and  $S$ ;  $SK_j^i$  represents the negotiated key for the  $j$ -th time.

If the instance  $S_i$  receives all the expected messages according to the predetermined steps, the instance enters the accepting state denoted as  $Acc_{\Pi}^i = 1$ . In this protocol, the parties negotiating the secure session key should meet the following conditions: (1) both  $S$  and  $D$  enter the

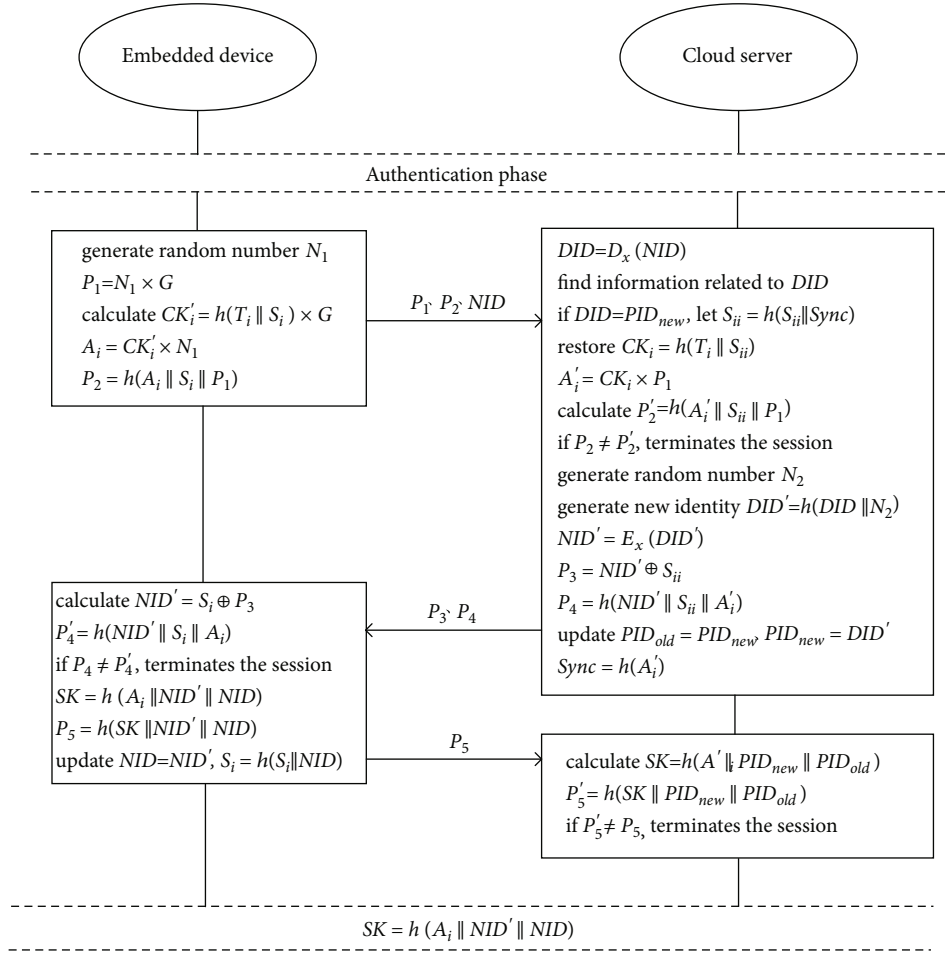


FIGURE 3: Illustration of authentication phase of LAAP.

receiving state, i.e.,  $Acc_D^j = Acc_S^j = 1$ ; (2)  $D$  updates the identity,  $DID_i = NID_i$ ; (3) the identities of  $S$  and  $D$  are not empty, i.e.,  $DID_i \neq null$ ,  $SID_i \neq null$

In the ROM, an attacker can simulate the attack by querying the RO. The included query is defined as follows:

Passive attack  $Execute(D_i, S_i)$ : the attacker can query the oracle to obtain the messages exchanged between  $D_i$  and  $S_i$ , giving the attacker the ability to eavesdrop on the channel.

Active attack  $Send(II, m)$ : the attacker can interact with any participant by querying the oracle machine, and the oracle machine processes the message. If the message is valid, the oracle machine returns the processing result of message  $m$ ; if the message is invalid, the oracle machine ignores the message.

$Reveal(II)$ : the attacker can obtain the session key of any participant by querying the oracle, and this query will only return the held key if the participant actually holds the session key. When an attacker queries the random oracle, the correct session key will be returned only if  $II$  is accepted; otherwise, a random element in the state space will be returned.

$Corrupt(D_i, a)$ : when  $a = 1$ , the hash chain value of  $D_i$  is fed back during the query. If the hash chain value is invalid, the random element in the state space is returned. When  $a$

$= 2$ , the query information is the registration information  $T_i$  of  $D_i$ . If the registration information  $T_i$  is invalid, the random element in the state space is returned.

$TestID(D_i, NID_i, OID_i)$ : the attacker can obtain the real identity of the device by querying the oracle. If the sent message  $D_i$  is accepted, it will return the real identity of the device; otherwise, it will return a random element in the state space. This oracle is used to test the anonymity of the protocol.

$TestSK(II)$ : when the attacker queries the oracle, the RO throws an unbiased coin  $b$ , and the result is used to determine whether the query returns the correct result. If  $b = 0$ , it returns a random element in the state space; if  $b = 1$ , and the participant holds the session key, return the correct session key; otherwise return it. The oracle tests the security of the negotiated session key, where the query can only be executed once.

Semantic security for session keys. In the defined ROM, attacker  $\mathcal{A}$  can query the session key through  $Reveal(II)$  or  $TestSK(II)$ , and random elements in the state space will be returned during the query process; query through  $TestID(D_i, NID_i, OID_i)$  The real identity of the device.  $\mathcal{A}$  needs to distinguish between random elements and real information. The goal of  $\mathcal{A}$  is to guess the real information. At the end of

the experiment, the attacker returns a guess bit  $c^*$ . If  $c^* = c$ , then  $\mathcal{A}$  wins the game event, which destroys the security of the protocol.  $Succ_i$  denotes that  $\mathcal{A}$  wins the  $i$ th experiment, and  $P$  denotes the constructed LAAP protocol. More precisely, the advantage of  $\mathcal{A}$  overcoming the semantic security of the protocol is  $Adv_p = |2 \cdot \Pr[Succ_0]| - 1$ , if the experiment ends, the probability of obtaining  $\mathcal{A}$  attack success is negligible, indicating that the protocol is semantically secure.

Based on the above definitions, we have the following theorem which proves the security of the proposed protocol.

**Theorem 3.** *Let  $Adv_p$  denote the advantage of an adversary  $\mathcal{A}$  to break through the semantic security of the proposed protocol  $P$ , and let  $Adv_{E_x}^{SE}$  denote the advantage of  $\mathcal{A}$  in cracking the ciphertext symmetric encrypted with the server key pair within a probability polynomial, and let  $Adv_{E_p}^{ECDLP}$  denote the advantage of solving the ECDLP problem of  $E_p$  in any polynomial time*

$$Adv_p \leq \frac{2(q_s + q_e)^2 + 2q_t^2}{2^{2h}} + 2Adv_{E_p}^{ECDLP}(t) + 2Adv_{E_x}^{SE}(t), \quad (2)$$

where  $E_p$  and  $E_x$  are the elliptic curve group and the symmetric encryption algorithm, respectively, and  $q_s$ ,  $q_e$ , and  $q_t$  denote the times that attacker  $\mathcal{A}$  executes the queries  $Send(\Pi, m)$ ,  $Execute(D_i, S_i)$ , and  $TestID(D_i, a)$ , respectively

*Proof.* Let  $\Pr[Succ_i]$  denote the probability that  $\mathcal{A}$  wins in the  $i$ -th experiment. The contribution of the  $(i+1)$ -th experiment to the probability of  $\mathcal{A}$  winning can be expressed  $|\Pr[Succ_i] - \Pr[Succ_{i+1}]|$ . The proof process can be described as the following five different experiments.

Experiment 1. This experiment corresponds to a real attack in the ROM. When  $\mathcal{A}$  implements a real attack on the protocol  $P$  under the ROM model, we have

$$Adv_p = |2 \cdot \Pr[Succ_0]| - 1. \quad (3)$$

Experiment 2. This experiment is used to simulate an eavesdropping attack of an adversary  $\mathcal{A}$ . We know  $SK = h(A_i || NID' || NID)$  in the protocol, where  $A_i$  is calculated by the embedded device through  $CK_i$  and random number  $N_1$ , and  $NID'$  is encrypted by the server with the secret key  $x$ . Even if intercepting all parameters transmitted during the authentication phase,  $\mathcal{A}$  still cannot get it any information. Therefore implementing an eavesdropping attack cannot increase the probability of  $\mathcal{A}$  winning, and we can obtain

$$\Pr[Succ_0] = \Pr[Succ_1]. \quad (4)$$

Experiment 3. This experiment is used to simulate all possible hash collisions in the authentication phase based on Experiment 2.  $\mathcal{A}$  tries to find hash collisions, and if the same output is produced for different inputs, the game ends. According to the birthday paradox (the number of collision tests for a hash table of  $N$  bit length is not  $2N$  but only  $N$ ),

we have

$$|\Pr[Succ_2] - \Pr[Succ_1]| \leq \frac{(q_s + q_e)^2}{2^{2h}}, \quad (5)$$

where  $q_s$  and  $q_e$  represent the times that attacker  $\mathcal{A}$  queries  $Send(\Pi, m)$  and  $Execute(D_i, S_i)$ , respectively.

Experiment 4. Based on Experiment 3, attacker  $\mathcal{A}$  queries the device's secret information  $T_i$  and hash chain value  $S_i$  by adding  $TestID(D_i, NID_i, OID_i)$ . If  $\mathcal{A}$  successfully obtains the information, the probability that  $\mathcal{A}$  wins the experiment is

$$|\Pr[Succ_2] - \Pr[Succ_1]| \leq \frac{q_t^2}{2^{2h}}, \quad (6)$$

where  $q_t$  represents the times that the attacker  $\mathcal{A}$  queries  $TestID(D_i, a)$ .

Experiment 5. Based on Experiment 4, the experiment adds that  $\mathcal{A}$  can tamper with the authentication information and make legitimate participants believe the tampered message, i.e.,  $\mathcal{A}$  can eavesdrop on the message and can make Hash collision. The following two cases will occur: (1)  $\mathcal{A}$  tampers with message  $P_1$ , and (2)  $\mathcal{A}$  tampers with message  $P_3$ .

Case 1. In this case, after tampering with  $P_1$ , the adversary needs to solve how to correspond to the verification message  $P_2$ . For this reason, the adversary needs to solve the ECDLP problem, and guess  $N_1$  and  $CK_i$ , (i.e.,  $2Adv_{E_p}^{ECDLP}(t)$ ), so that it can guess  $CK_i$ . Besides,  $\mathcal{A}$  still needs to solve a symmetric key problem to generate legal  $P_5$ , i.e.,  $Adv_{E_x}^{SE}(t)$ . Overall, we have

$$\Pr[Succ_4 | \text{Case1}] \leq 2Adv_{E_p}^{ECDLP}(t) + Adv_{E_x}^{SE}(t). \quad (7)$$

Case 2. In this case,  $\mathcal{A}$  tampers with  $P_3$  to impersonate the server. Assuming that  $\mathcal{A}$  has obtained the synchronization value  $S_i$  shared by the device and the server,  $\mathcal{A}$  still needs to solve the symmetric key decryption problem, i.e.,  $Adv_{E_x}^{SE}(t)$ . Similarly, if  $\mathcal{A}$  has decrypted the current symmetric key problem,  $\mathcal{A}$  still needs to solve the ECDLP problem to obtain the legal  $A_i$ , i.e.,  $Adv_{E_p}^{ECDLP}(t)$ . Thus, we have

$$\Pr[Succ_4 | \text{Case2}] \leq Adv_{E_p}^{ECDLP}(t) + Adv_{E_x}^{SE}(t). \quad (8)$$

In summary, the probability that the adversary  $\mathcal{A}$  wins in Experiment 5 is

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq Adv_{E_p}^{ECDLP}(t) + Adv_{E_x}^{SE}(t). \quad (9)$$

All random predictions are simulated in the above four experiments. The results indicate that  $\mathcal{A}$  has no advantage in guessing the bit  $c$ , and the only way to pass the test is to

perform  $TestSK(II)$  query guessing, i.e.,

$$\Pr[\text{Succ}_4] = \frac{1}{2}. \quad (10)$$

Using the triangle inequality, we can obtain

$$\frac{1}{2} \text{Adv}_P = |\Pr[\text{Succ}_0] - \Pr[\text{Succ}_4]|. \quad (11)$$

Based on (3)–(8), we have

$$\begin{aligned} |\Pr[\text{Succ}_0] - \Pr[\text{Succ}_4]| \leq & \frac{(q_s + q_e)^2}{2^{l_h}} + \frac{q_t^2}{2^{l_h}} \\ & + \text{Adv}_P^{\text{ECDLP}}(t) + \text{Adv}_{E_x}^{\text{SE}}(t). \end{aligned} \quad (12)$$

Submitting (12) into (11), we can obtain (2).  $\square$

*Remark 4.* This result indicates that the adversary  $\mathcal{A}$  has no extra advantage to win the experiment and the proposed scheme is secure.

*4.1.2. Formal Security Proof with AVISPA Tool.* In this part, the AVISPA verification tool is used to verify the security of the LAAP protocol. The experimental environment is Oracle VM VirtualBox, SPAN-Ubuntu10.10-light. The HPSL language description of the protocol is divided into the following five dimensions.

*Role attributes:*  $D$  and  $S$  are two agents, Hash and Mutli are two hash functions,  $Kab$  is a symmetric key, and Snd and RCV are the communication channels between the client and the outside world. The local variables defined are the same as the protocol description, as shown in Figure 4(a). The modeling of the server is similar to that of the client, as shown in Figure 4(b).

*Role conversion process:* the conversion process of  $D$  in LAAP is divided into three stages: register1 means that  $D$  starts to register and sends registration information to  $S$ ; register2 means that  $D$  receives the response from  $S$ , conducts authentication calculation, and initiates an authentication request; authentication1 indicates that  $D$  receives the response from  $S$  and completes the final authentication process, as shown in Figure 5(a). Similarly, the conversion process of  $S$  is also divided into three stages: register indicates that  $S$  requests the registration information of  $D$ ; authentication1 means that  $S$  receives the authentication request of  $D$  and performs verification and response; authentication2 means that  $S$  receives the response of  $D$  and completes the final authentication process, as shown in Figure 5(b).

*Session attributes:* the modeling of LAAP session attributes defines the rules that the communicating entities follow. The definition of basic attributes includes the role agents  $D$  and  $S$ , hash functions Hash and Multi, symmetric key  $Q_i$ , and communication channels SND and RCV as shown in Figure 5(c).

*Environmental attributes:* the definition content of LAAP environment includes the communication channel of the communication entity, communication entity (includ-

ing  $d$ ,  $s$ , and  $i$ ), security target constant, and session combination, as shown in Figure 5(d).

*Safety goals:* the security goal describes the secret information “secrecy\_of” and the authentication quantity “authentication\_o” of the communication entity defined in the protocol shown in Figure 6.

The OFMC simulation results are shown in Figure 6, and the ATSE simulation results are shown in Figure 6. From Figure 7, we can see that the LAAP realizes two-way authentication while resisting man-in-the-middle attacks and replay attacks, which proves the security of the protocol.

*4.2. Informal Security Analysis.* Informal security analysis mainly consists of two parts, i.e., basic function security and common attack defense. Basic functional security includes mutual authentication and device anonymity. Common attacks resistance mainly includes traceable attack defense, asynchronous attack defense, DoS attack defense, replay attack defense, and simulation attack defense.

*Mutual authentication.* In the LAAP protocol, server  $S$  authenticates the legal identity of device  $D$  based on the message  $\langle P_1, P_2, NID \rangle$  and then restores the identity of the device with  $NID$ . This process is performed through symmetric encryption. If  $S$  gets a string of garbled characters after decrypting  $NID$  or cannot find matching information in the verification table,  $S$  will discard the authentication message. After successfully decrypting  $NID$  and obtaining the device’s identity  $DID$ ,  $S$  verifies the authenticity of the device with  $CK_i = h(T_i \| S_{ii})$ ,  $P'_2 = h(A'_i \| S_{ii} \| P_1)$ . If  $P_2 = P'_2$ , the verification is passed. In the response message  $\langle P_3, P_4 \rangle$ ,  $P_3$  contains the new identity of the device, which is encrypted by the hash chain value synchronized by both parties, and  $P_4$  contains the authentication information  $A_i$ . Notice that only valid  $S$  can calculate  $A'_i$ . Thus, if the  $P'_4$  calculated by the device is the same as the received  $P_4$ ,  $D$  can confirm the legal identity of  $S$ .

*Device anonymity.* Device anonymity means that attacker  $\mathcal{A}$  cannot obtain any identifying information about the participants by listening to the messages in the channel. In the LAAP, the method of dynamic pseudonym and synchronous hash chain are used to solve the anonymity of the device. Notice that the identity identification  $NID$  is dynamically updated in the second phase of device authentication so that the identity identifications are different at different session stages. Under the Deolv-Yao attack model [31],  $\mathcal{A}$  is completely unable to distinguish the attribution of different sessions. Therefore, the proposed protocol satisfies the anonymity requirement of the device.

*Traceable attack defense.* Traceable attack means that attacker  $\mathcal{A}$  can identify the belonging of messages by listening to the information in the channel, so as to carry out specific analysis to undermine the security of the protocol. Recall that the LAAP protocol used the dynamic pseudonym. Thus, after each successful session key negotiation, the device identity is updated. That is, the  $NID^i$  sent in the  $i$ -th session is completely different from the  $NID^{i+1}$  sent in the  $(i + 1)$ -th session. Therefore,  $\mathcal{A}$  cannot determine which communication entity the session message belongs to and



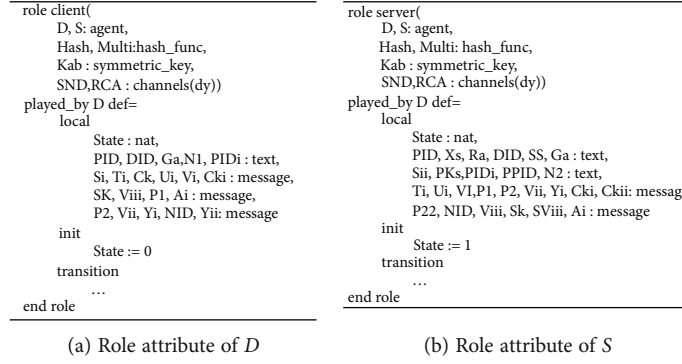
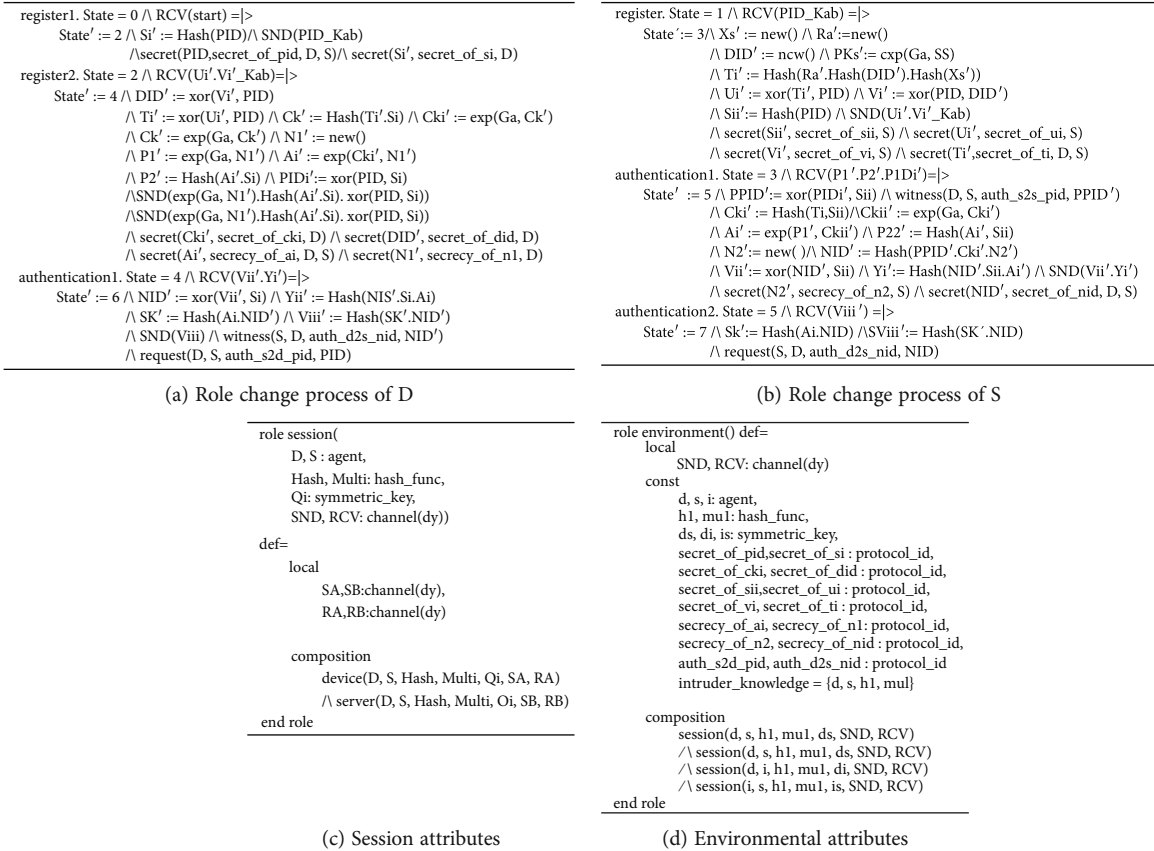
FIGURE 4: Role attributes of the client  $D$  and the serve  $S$ .

FIGURE 5: Role change process, session attributes, and environmental attributes.

```

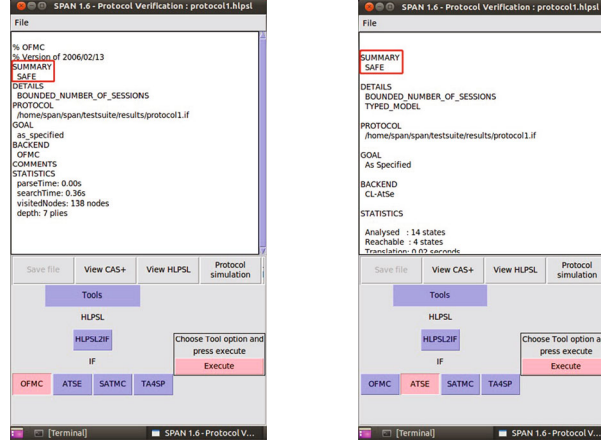
goal
  % device register
  secrecy_of secret_of_pid, secret_of_si
  secrecy_of secret_of_cki, secret_of_did
  % device register
  secrecy_of secret_of_sii, secret_of_ui
  secrecy_of secret_of_vi, secret_of_ti
  secrecy_of secret_of_n2, secret_of_nid
  secrecy_of secret_of_ai, secret_of_n1
  authentication_on auth_s2d_pid
  authentication_on auth_d2s_nid
end goal

```

FIGURE 6: Description of the safety goals.

also cannot track the session information of specific equipment.

**Asynchronous attack defense.** An asynchronous attack intercepts message transmission to make protocol participants lose synchronization. As a result, the protocol cannot be executed correctly, thus destroying the protocol. According to the LAAP, the messages transmitted on the public channel includes  $\langle P_1, P_2, NID \rangle$ ,  $\langle P_3, P_4 \rangle$ , and  $\langle P_5 \rangle$ . There are two messages related to synchronization information, i.e., the initial authentication message  $\langle P_1, P_2, NID \rangle$  and the response message  $\langle P_3, P_4 \rangle$  sent by  $S$  and  $D$ , respectively. For attacker  $\mathcal{A}$ , intercepting message  $\langle P_1, P_2, NID \rangle$  has no



(a) OFMC backend simulation results (b) ATSE backend simulation results

FIGURE 7: Verification results of security with AVISPA.

effect on the synchronization of the protocol. Thus, we only consider the following two cases.

*Case 1* ( $\mathcal{A}$  intercepts the message).  $S$  has updated the device ID as  $PID_{new} = DID^{i+1}$ ,  $PID_{old} = DID^i$  because  $S$  has already processed the message  $i$ . At this time, due to the information interception, the identity identifier  $DID^i$  of  $D$  is the corresponding  $NID^i$ . When the timer of  $D$  expires,  $D$  will regenerate the random number to reauthenticate and send the message  $\langle P_1^i, P_2^i, NID^i \rangle$  to  $S$ . We can see that  $PID_{old} = DID^i$ , and  $S$  will determine that  $D$  is out of synchronization, and the current hash value is directly used for authentication.

*Case 2* ( $\mathcal{A}$  intercepts the message  $\langle P_5 \rangle$ ). The system status is that  $S$  updated the device ID, and  $D$  updated the device ID and hash chain value. After  $\langle P_5 \rangle$  is intercepted, the hash chain value of the protocol participant  $S$  is synchronously behind  $D$ . When the timer of device  $D$  expires,  $D$  resends the authentication message  $\langle P_1^{i+1}, P_2^{i+1}, NID^{i+1} \rangle$ , and  $PID_{new} = DID^{i+1}$ . the hash chain value will be updated, and  $Sync$  will be used in the update process. Therefore,  $D$  and  $S$  will resume synchronization.

**DoS attacks defense.** DoS attack means that attacker  $\mathcal{A}$  sends a large amount of invalid authentication information to the server, which consumes the computing resources of the server and makes the server unable to provide services normally. In the protocols in [11, 15], there is a way to find information about related devices by traversing the password check table or the local registry. Thus, their time complexity is  $O(n)$ . When there are enough registrations, even if most of the devices are offline, the server will go through all the devices during the authentication process. The proposed LAAP protocol combine the dynamic pseudonym with symmetric encryption, and the time complexity is reduced from  $O(n)$  to  $O(1)$ . So, the proposed protocol has a high authentication efficiency and can resist DoS attacks.

**Replay attacks defense.** Replay attack means that the attacker resends the message sent in the history negotiation stage to the server, thus achieving the purpose of spoofing. In LAAP, the messages transmitted by the public channel consist  $\langle P_1, P_2, NID \rangle, \langle P_3, P_4 \rangle$ , and  $\langle P_5 \rangle$ . Let the message sent by the device in the  $i$ -th session be  $\langle P_1^i, P_2^i, NID^i \rangle$ , the message sent by the server be  $\langle P_3^i, P_4^i \rangle$ , and the response message from the device be  $\langle P_5^i \rangle$ . Thus, there would be the following three cases.

*Case 1* ( $\mathcal{A}$  replays the message  $\langle P_1^i, P_2^i, NID^i \rangle$ ). We will analyze it from two subcases. In subcase 1, the attacker launches a replay attack in the middle of the  $i$ -th and  $(i+1)$ -th key negotiation at the device side. Note that the device has not performed the  $(i+1)$ -th key negotiation. Because the server will store the  $i$ th device identity, the server will find  $PID_{old} = D_x(NID^i)$  in the authentication table and will consider that device is out of asynchrony. So in the next step of message verification, the server will calculate  $Sync = h(A_i^i)$  and classify the message as a replay attack and discard the session. In subcase 2,  $\mathcal{A}$  uses the device history negotiation information  $\langle P_1^{i-n}, P_2^{i-n}, NID^{i-n} \rangle$ , where  $n \in [0, i-1)$ : After receiving  $\langle P_1^{i-n}, P_2^{i-n}, NID^{i-n} \rangle$ , the server uses key symmetric decryption to get the  $(i-n-1)$ -th device identity based on  $NID^{i-n}$ . But the server cannot find the relevant information in the database, and it will discard the session.

*Case 2* ( $\mathcal{A}$  replay the message  $\langle P_3^i, P_4^i \rangle$ ). Similarly, we will analyze it from two subcases. Subcase 1 is similar to the above subcase. After receiving  $\langle P_3^i, P_4^i \rangle$ , the device will decrypt  $P_3^i$  to obtain the new device identifier  $NID^i$ . At this time, the hash chain value at the device has been updated, and  $P_4^i$  computed by the device is different from the received  $P_4$ . So, the device will terminate the session. For subcase 2, after receiving the history information  $\langle P_3^{i-n}, P_4^{i-n} \rangle$  (where  $n \in [0, i-1)$ ), the device will decrypt  $P_3^{i-n}$  to get the identity  $NID^{i-n+1}$ . However, because the hash chain value has been updated several times,  $NID^{i-n+1}$  obtained by decrypting

$P_4^{i-n'}$  is different from  $P_4^{i-n}$ . Thus, the device will terminate the session.

Case 3 ( $\mathcal{A}$  replay the historical message  $\langle P_5^i \rangle$ ). The server will use the new authentication information and message to calculate as follows:

$$\begin{aligned} SK &= h\left(A'_{i\_new} \| NID'_{new} \| NID_{new}\right), \\ P'_{5\_new} &= h(SK_{new} \| SK_{new}). \end{aligned} \quad (13)$$

We can find that  $P'_{5\_new}$  is different from  $P_5^i$ . Thus, the session key negotiation cannot be successful, and the server will discard the session.

Based on the analysis of the above three cases, we proof that the proposed protocol can resist the replay attacks.

Simulation attack defense. An emulation attack means that the attacker tampers authentication information to establish session keys on the simulated device or server. An attacker  $\mathcal{A}$  can tamper with or send historical authentication messages to spoof the device or the server based on the intercepted authentication messages.

For messages  $\langle P_1, P_2, NID \rangle$ ,  $\mathcal{A}$  has no access to the registration information  $T_i$  and the synchronization hash value of the device, so it cannot obtain the legitimate  $P_2$  to spoof  $S$ . If  $\mathcal{A}$  replays the historical messages, see the above analysis for details. For messages  $\langle P_3, P_4 \rangle$ ,  $\mathcal{A}$  cannot know symmetric encryption key  $x$  of  $S$  and the authentication message  $A'_i$  generated in this session, so it cannot compute the legitimate  $P_3$  and  $P_4$ . If  $\mathcal{A}$  replays the historical message, the session is terminated due to authentication failure. For message  $\langle P_5 \rangle$ ,  $\mathcal{A}$  cannot know the authentication message  $A'_i$  of this session, so that it cannot compute  $P_5$ . If  $\mathcal{A}$  replays the history message, the authentication will fail, and  $S$  terminates the session. In summary, the proposed protocol can resist the simulation attacks.

We provide Table 2 to show the comparison of security performance between LAAP and the benchmarks. The dimension of comparison is based on the basic functional security and common attack resistance described in the above. In Table 2, where “Yes” (resp. “No”) indicate that the protocol can (cannot) support the security feature, and “—” indicates that the protocol does not involve this security feature. From Table 2, we can see that the proposed LAAP protocol can resist more security attacks.

## 5. Performance Evaluation

In this section, we provide the performance comparisons between the proposed LAAP protocol with several related protocols [9–11, 15] in terms of computational overhead, storage overhead, and communication overhead. For a fair comparison, all experiments use the <http://golang.org/x/crypto/bn256> curve and the hash function SHA-256.

**5.1. Computational Cost Analysis.** We first provide the comparison of a computational overhead between the proposed protocol and the benchmarks. The computational cost is

TABLE 2: Comparison of security performance.

Index Bench	S1	S2	S3	S4	S5	S6	S7
Cws	No	No	No	No	Yes	No	—
Wang	Yes	No	Yes	No	Yes	No	—
Panda	Yes	Yes	Yes	Yes	Yes	No	—
Rostampour	Yes	Yes	Yes	Yes	No	Yes	—
Bhuarya	Yes	Yes	Yes	Yes	Yes	No	—
LAAP	Yes	Yes	Yes	Yes	Yes	Yes	Yes

S1-S7 are the index of the mutual authentication, device anonymity, traceable attack defense, asynchronous attack defense, DoS attacks defense, replay attacks defense, and simulation attack defense, respectively.

divided into the time-consuming of the device in the registration phase and the authentication phase. The calculation overhead is shown in Table 3. From Table 3, we can see that in the registration stage, the proposed protocol increases the computational overhead of the device but reduces the overhead of the server. We also can observe that in the authentication stage, we reduce the computational overhead of both the device and server. Finally, the results of total overhead indicate that the proposed scheme outperforms the benchmarks.

In order to show the total computation cost under different numbers of devices, we plot Figure 8. We can observe that compared with the benchmarks, the proposed protocol can bring a lower computation cost. Furthermore, as the number of devices increases, the performance improvement of our protocol becomes more obvious. Therefore, the proposed protocol is more suitable for deployment in the IoT with a large number of devices.

**5.2. Storage Cost Analysis.** Here, we provide the comparison of the store overhead between the proposed protocol and the benchmarks. The store cost consists the space-consuming of the device in the registration phase and the authentication phase. In the analysis process, SHA-256 and bn256 are used as the hash function and elliptic curve, respectively.

According to Figure 2, the storage information at the device includes  $NID_i$ ,  $T_i$ , and  $S_i$ . So the storage space required at the device side is  $256 + 256 + 256 = 768$  bits in the registration phase. As Figure 3 shown, the storage space required by the server is  $256 + 256 + 256 + 256 + 256 = 1280$  bits in the authentication phase. The storage overhead of the benchmarks is calculated in the same way, and the detailed data is shown in Table 4. To show the comparison results more visually, we provide Figure 9. We can observe that the storage space required by the LAAP protocol at the device is consistent with the minimum storage required by the benchmarks, and the store overhead at the server only be higher than the protocol in [11]. Thus, our protocol requires higher storage overhead than the protocol in [11] but lower than other protocols in [9, 10, 15].

**5.3. Communication Cost Analysis.** Finally, we provide the comparison of the communication cost between the proposed protocol and the benchmarks. Similarly, the

TABLE 3: The comparison of calculation overhead (ms).

		Regist.	Authen.	Total
[9]	Device	0	71.0163	206.4185
	Server	53.4174	81.9848	
[10]	Device	0	66.0148	183.1098
	Server	54.8872	62.2078	
[11]	Device	9.8241	93.9545	318.3577
	Server	52.4631	162.116	
[15]	Device	0.5751	94.8175	328.901
	Server	12.0247	221.4837	
LAAP	Device	45.1705	46.3724	149.3163
	Server	12.998	44.7754	

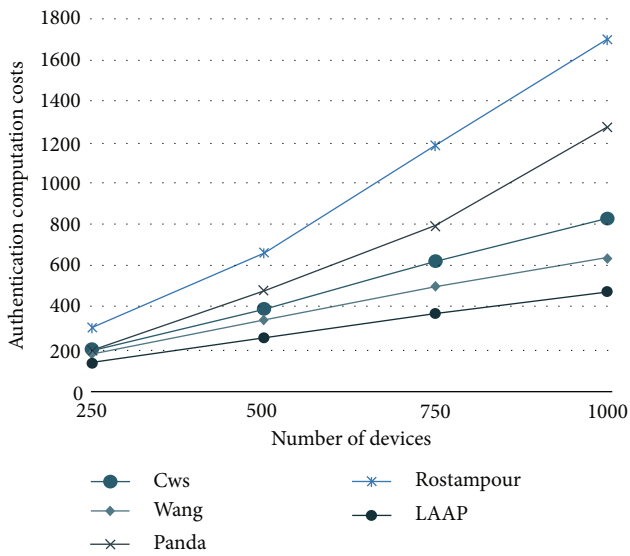


FIGURE 8: Total computation cost vs. various number of devices.

TABLE 4: The comparison of store overhead (bits).

	Registration	Authentication	Total
[9]	1024	1280	2304
[10]	768	1280	2048
[11]	768	768	1536
[15]	1536	1793	3329
LAAP	768	1280	2048

communication cost consists the consumption of the device in the registration phase and the authentication phase.

According to Figure 2, the information transmitted by the device includes  $Z_1$ ,  $PPID_i$ ,  $Z_3$ , and  $P_4$ . So the communication cost in the registration phase is  $256 + 512 + 256 + 256 = 1280$  bits. As Figure 3 shown for the proposed protocol, the communication overhead at the server is  $512 + 256 + 256 + 256 + 256 + 256 = 1792$  bits in the authentication phase. The communication overhead of the benchmarks is calculated in the same way and the detailed data is shown in Table 5..

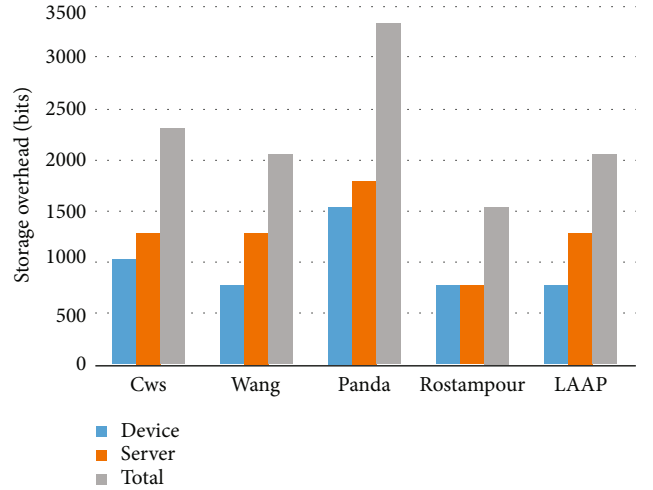


FIGURE 9: Storage cost comparison.

We further provide Figure 10 to show the detailed comparison. We can see from Figure 10 that the communication overhead of the proposed protocol in the registration phase is slightly higher than the benchmarks. That is due to the fact that the proposed protocol takes necessary encryption measures to ensure public channel registration. However, the proposed protocol takes the lowest communication overhead. Considering that in practical applications, the number of registration stages is much less than the number of authentication stages, it is acceptable to increase the overhead of registration stage slightly. We can also find that the proposed protocol has the lowest total communication overhead and an average of 12.73% reduction in terms of communication overhead compared to other protocols.

*5.4. Performance Analysis under Different Number of Devices.* To verify the performance of the proposed protocol under a large number of devices, the stress testing tool GOWRK and custom scripts are used in this subsection to analyze the performance of the device registration module, server registration module, and identity authentication module of the system.

We first analyze the average response time for different numbers of the devices in the registration phase in Table 6. We can see that when the number of registered devices is less than 1400, the server has a relatively fast response rate, and the average response time is 48.57 ms. When the number of registered devices is greater than 1400, the response time increases proportionally as the number of devices increases. It indicates that when the number of registrations is higher than 1400, the system performance is saturated and all resources are fully utilized.

Then, we provide Table 7 to show the average cost under the different number of devices in the certificate phase. As shown in Table 7, under the LAAP protocol, the average time consumption of the devices is 58.15 ms, and as the number of concurrent devices increases, the device time consumption is basically stable. When the server does not reach saturation, the server takes an average of 64.1 ms. After the server reaches saturation, the response time of

TABLE 5: The comparison of communication overhead (bits).

	Registration	Authentication	Total
[9]	1024	2560	3584
[10]	768	2304	3072
[11]	768	3072	3840
[15]	1280	2304	3584
LAAP	1280	1792	3072

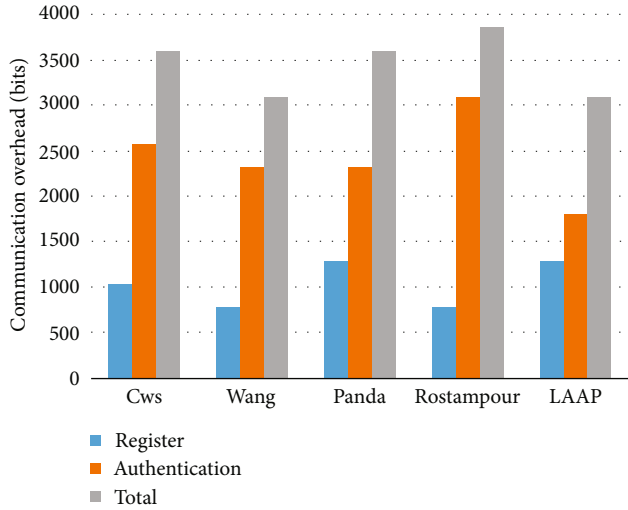


FIGURE 10: Communication cost comparison.

TABLE 6: Average response time of devices in the registration phase (ms).

Num.	Ave-time	Suc. rate	Num.	Ave-time	Suc. rate
200	48.31	100%	1800	93.23	100%
600	47.32	100%	2000	122.76	100%
1000	48.03	100%	2200	157.72	100%
1400	50.61	100%	2400	189.31	100%
1600	60.87	100%	2600	225.12	100%

Num.: number of devices; Ave. time: average response time; Suc. rate: success rate.

TABLE 7: Average cost in the certification phase.

Num.	D-Cost (ms)	S-Cost (ms)	Aut-efficiency
200	58.11	63.21	100%
400	56.98	65.79	100%
600	56.43	64.33	100%
800	57.21	63.31	100%
1000	56.45	62.75	100%
1200	58.97	65.21	100%

Num.: number of devices; D-Cost and S-Cost: average cost at device and serve, respectively.

the server increases with the number of concurrent authentication devices. The saturation threshold of the server is 1200 devices. In the case of a single server, the server can

quickly complete the authentication and key negotiation of 1200 devices.

## 6. Conclusion

In this work, we proposed a lightweight anonymous authentication protocol named LAAP against asynchronous attacks to realize the anonymous authentication between device and server in the IoT. Through informal security analysis and formal security analysis, we found that the proposed protocol has the following advantages: (1) it can solve the problem that the device identification is fixed and easy to be tracked by dynamically updating the identification; (2) the hashing chain value of communication devices can be adaptively synchronized to resist asynchronization attack; (3) the time complexity of finding the registration information of the device through the anonymous identity of the device is  $O(1)$ . Besides, extensive results were provided to indicate that the total overhead is lower than the benchmarks.

Note that this work only considers identity authentication within a single network domain. Therefore, the lightweight anonymous authentication of the IoT across network domains will be the future research direction. In addition, using the intelligent algorithms [32] to optimize our methods and solve the authentication of large-scale heterogeneous devices are also the future research.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Key R&D Program of China (Grant No. 2018YFE0207600), the National Natural Science Foundation of China (NSFC) under Grant 61972308, the Basic and Applied Basic Research Fund of Guangdong Province (Grant No. 2021A1515111017), and the Natural Science Basic Research Program of Shaanxi (Program No. 2019JC-17).

## References

- [1] GSMA, "The mobile economy 2022," 2022, <https://www.gsma.com/mobileeconomy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf>.
- [2] A. R. Biswas and R. Gialfreda, "IoT and cloud convergence: opportunities and challenges," in *2014 IEEE World Forum on Internet of Things*, pp. 375-376, Seoul, Korea, 2014.
- [3] M. R. Simpson, *Assessment of the Impact of Cyberattacks on Power System Stability-Manipulation of Controllable Loads in Smart Homes*, M.S. Thesis, High Voltage Equipment and Grids, Digitalization and Energy Economics, 2021.
- [4] NSFOCUS, "Cybersecurity in the context of building a cyber power," 2022, <http://blog.nsfocus.net/wp-content/uploads/>

- 2022/03/Cybersecurityin-the-Context-of-Building-a-Cyber-Power.pdf.
- [5] Unit 42, *2020 Unit 42 IoT Threat Report*, Technical Representative, 2020.
  - [6] C. Wu, "An overview on the security techniques and challenges of the internet of things," *Journal of Cryptologic Research*, vol. 2, no. 1, pp. 40–53, 2015.
  - [7] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IOT new features on security and privacy: new threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2019.
  - [8] S. Kalra and S. K. Sood, "Secure authentication scheme for IOT and cloud servers," *Pervasive and Mobile Computing*, vol. 24, pp. 210–223, 2015.
  - [9] C.-C. Chang, H.-L. Wu, and C.-Y. Sun, "Notes on "secure authentication scheme for IoT and cloud servers"," *Pervasive and Mobile Computing*, vol. 38, pp. 275–278, 2017.
  - [10] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "A secure authentication scheme for internet of things," *Pervasive and Mobile Computing*, vol. 42, pp. 15–26, 2017.
  - [11] S. Rostampour, M. Saffkhani, Y. Bendavid, and N. Bagheri, "ECCbAP: a secure ECC-based authentication protocol for IoT edge devices," *Pervasive and Mobile Computing*, vol. 67, article 101194, 2020.
  - [12] H.-L. Wu, C.-C. Chang, and L.-S. Chen, "Secure and anonymous authentication scheme for the internet of things with pairing," *Pervasive and Mobile Computing*, vol. 67, article 101177, 2020.
  - [13] S. Bhubaneswari and N. Ananth, "Enhanced mutual authentication scheme for cloud of things," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 15, pp. 1571–1583, 2018.
  - [14] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IOT and cloud servers," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6428–6453, 2018.
  - [15] P. K. Panda and S. Chattopadhyay, "A secure mutual authentication protocol for IoT environment," *Journal of Reliable Intelligent Environments*, vol. 6, no. 2, pp. 79–94, 2020.
  - [16] P. Bhuarya, P. Chandrakar, R. Ali, and A. Sharaff, "An enhanced authentication scheme for internet of things and cloud based on elliptic curve cryptography," *International Journal of Communication Systems*, vol. 34, no. 10, 2021.
  - [17] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.
  - [18] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-to-peer Networking and Applications*, vol. 8, no. 6, pp. 1070–1081, 2015.
  - [19] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-to-peer Networking and Applications*, vol. 9, no. 1, pp. 223–244, 2016.
  - [20] P. Gope, J. Lee, and T. Q. Quek, "Resilience of dos attacks in designing anonymous user authentication protocol for wireless sensor networks," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 498–503, 2017.
  - [21] P. Gope and B. Sikdar, "An efficient data aggregation scheme for privacy-friendly dynamic pricing-based billing and demand-response management in smart grids," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3126–3135, 2018.
  - [22] P. Gope, J. Lee, and T. Q. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.
  - [23] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3953–3962, 2019.
  - [24] H. Han, L. Fang, W. Lu, W. Zhai, Y. Li, and J. Zhao, "A GCICA grant-free random access scheme for M2M communications in crowded massive MIMO systems," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6032–6046, 2022.
  - [25] A. K. Sahu, S. Sharma, and R. Raja, "Deep learning-based continuous authentication for an IoT-enabled healthcare service," *Computers and Electrical Engineering*, vol. 99, article 107817, 2022.
  - [26] S. Zeng, Y. Chen, X. Li, J. Zhu, Y. Shen, and N. Shiratori, "Visibility graph entropy based radiometric feature for physical layer identification," *Ad Hoc Networks*, vol. 127, article 102780, 2022.
  - [27] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.
  - [28] C. Zhang, L. Zhu, and C. Xu, "BPAF: blockchain-enabled reliable and privacy-preserving authentication for fog-based IOT devices," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 88–96, 2022.
  - [29] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.
  - [30] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, Fairfax, Virginia, USA, 1993.
  - [31] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
  - [32] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2687–2700, 2022.