Lightweight and Anonymous Authentication based on PUF Without CRP leakage for Industrial Internet of Things

Fengqun Wang[1,2], Jie Cui[1,2], Wuquan Wen[1,2], and Ke Hu[1,2]
[1]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China
[2]Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China

Physical unclonable function (PUF) is a critical hardware primitive that provides unique identities for authenticating a large number of devices in the Industrial Internet of Things (IIoT). Most existing PUF-based schemes face challenge-response pair (CRP) leakage during machine-learning attack. Some studies that use hardware or time-consuming cryptographic operations to protect the PUF responses are expensive and unsuitable for existing IIoT devices. To address these issues, a lightweight and anonymous PUF-based authentication scheme is proposed for resource-constrained IIoTs. Using elliptic curve cryptography and zero-knowledge proof, a lightweight blinding mechanism is designed in the proposed scheme that prevents explicit CRP leakage and ensures anonymity. In addition, the authenticated keys are random with forward and backward secrecy. Moreover, the security of the proposed scheme is demonstrated using a random oracle model. Experimental results demonstrate that the proposed scheme is notably more efficient and practical for resource-constrained devices compared to other related schemes.

Index Terms—**Industrial Internet of Things (IIoT), lightweight authentication, CRP leakage, physical unclonable function (PUF), anonymous authentication.**

I. Introduction

With the advancement of modern industry, the Industrial Internet of Things (IIoT) [1]–[3] has rapidly developed. In an IIoT system, numerous devices communicate over a network, generating, processing, and transmitting large amounts of data. The openness of a channel is likely to lead to data leakage or tampering, rendering the IIoT system highly vulnerable to attacks. Dishonest device nodes can steal or leak sensitive information, raising several security concerns, including identification and privacy issues [4], [5]. Verifying the identity of a connected IIoT device is crucial for the entire system before it accesses IIoT services [6]. In addition, the privacy of IIoT devices is also important because external attackers can learn trade secrets by analyzing the access patterns of devices. Therefore, it is necessary to design an anonymous device authentication scheme to protect device privacy and ensure the security of IIoT [7]–[9].

Recently, the physical unclonable function (PUF) has become a potential hardware security primitive and is typically utilized to design authentication schemes [10]–[12], which have reliable and resilient security features [13], [14]. A PUF can be integrated into IoT devices and each PUF generates unique outputs in response to the same challenge. In addition, it reduces the implementation cost of authentication applications while enabling strong security [15], [16]. Existing schemes are mainly based on the security features of PUFs, such as tampering and no storage requirements [10], and the device-side verifier needs to store the challenge-response pairs (CRPs) of the PUF. However, when devices in IIoT are maliciously attacked, there is a risk of CRP leakage [17]. By contrast, an attacker can obtain a few number of PUF CRP relationships by listening to an open channel. Then, the attacker can then analyze the CRPs and use machine learning to infer the response to any challenge with high accuracy [18], [19].

Some schemes have been proposed to prevent such machine learning attacks by obfuscating the CRP relationship of the PUF. However, some schemes have issues such as inefficiency and excessive resource consumption [20], [21], and for resource-constrained IIoT devices, these schemes are unsuitable [22], [23]. On the one hand, some schemes solve the CRP leakage problem from a hardware perspective. Different hardware components were used to design a PUF structure to prevent CRP leakage from the PUF. However, the hardware costs of resource-constrained devices are extremely high. In addition, the primary issue with these schemes is that the deployed IIoT devices cannot update the hardware. On the other hand, some schemes also use complex cryptographic operations [24], [25] such as bilinear mapping. Most IIoT devices are terminal devices with weak communication and computing capabilities and limited resources. The resource consumption of these schemes is high, making them unaffordable for resource-constrained devices. In addition, anonymity is not considered in most schemes, which is also important for the IIoT, as previously mentioned.

Therefore, to address the above issues regarding CRP leakage and authentication, we aim to design a lightweight and anonymous mutual authentication protocol based on PUF.

These schemes have significantly contributed to previous PUF-based authentication methods. However, some issues remain. For example, existing PUF-based authentication systems require a high overall overhead that does not satisfy the lightweight requirements of resource-constrained devices, and they also face CRP leakage problems. To address the above issues, the major contributions of this study are as follows:

1) To protect the privacy of IIoT devices, a lightweight and anonymous mutual authentication scheme is proposed.

Manuscript received June 22, 2025; revised August 27, 2025. Corresponding author: Jie Cui (email: cuijie@mail.ustc.edu.cn).

The PUF is utilized to realize a trusted authentication between smart devices and servers. The proposed scheme imposes minimal computing overhead on the device side. In addition, random keys are established with forward and backward secrecy.

2) To solve the CRP leakage problem in the PUF-based scheme field, a lightweight CRP blinding scheme based on zero-knowledge proof is designed, which also makes the proposed scheme anonymous to external attackers. In addition, our scheme reduces the CRP storage and query overhead on the server side, as well as the communication overhead.
3) We performed a formal security analysis to confirm the proposed scheme's security. The performance analysis demonstrates that the proposed scheme effectively balances security and efficiency.

The remainder of this paper is organized as follows: The related studies are reviewed in Section II. Section III describes the preliminaries, including the PUF and ECC. The system model, threat model, and security threats and goals are presented in Section IV. The details of the proposed scheme are described in Section V. The security of the proposed scheme is analyzed in Section VI. Section VII introduces the experimental configuration and results. In the end, the conclusion of the study is in Section VIII.

II. Related Work

In this section, we present a systematic overview of existing security schemes based on PUF and discuss their limitations.

In an IIoT system, the deployment of IIoT typically involve numerous smart devices operating in unsupervised environments. These devices are often low-cost and resource-constrained. However, many current PUF-based authentication schemes continue to face issues related to CRP leakage. Recently, some hardware-based schemes for PUF authentication schemes are proposed. Ye et al. [26] proposed a PUF design method based on obfuscated logic. A new Boolean obfuscation module is proposed, which can be used to obfuscate the challenge bits in the arbitrated PUF. This increases PUF uncertainty and reduces the risk of CRP leaks and modeling attacks. Konigsmark et al. [27] introduced the initial PUF architecture featuring deliberate uncertainty, termed PolyPUF. This structure enables the PUF to operate dynamically, switching between multiple modes, thereby increasing the uncertainty and unpredictability of its challenge-response behavior. Ye et al. [28] employed a random number generator to enhance the randomization of CRP datasets. The input challenge undergoes primary randomization facilitated by these generators, ensuring that the resultant dataset poses significant difficulty for attackers attempting modeling attacks. Gu et al. [29] proposed a mutual authentication method designed to withstand modeling attacks. The scheme generates a training set of false or invalid responses and tricks the opponent into using this training set, thereby preventing the opponent from correctly predicting the response to an unknown challenge. However, the devices must package some hardware components in advance, with a certain hardware cost. These schemes all use hardware methods to prevent PUF CRP leakage problem. Although the purpose of avoiding leakage can be achieved to a certain extent, the hardware cost is too high, and the overall scheme is too complicated. Therefore, these schemes pose limitations for resource-constrained devices in the IIoT, failing to fully meet requirements.

In addition, several schemes aim to solve the CRP leakage problem at the protocol level. Chatterjee et al. [17] introduced an authentication and key exchange protocol that integrates PUF, Identity-Based Encryption, and key hash functions. This design aims to create an authentication protocol that minimizes database storage overhead. However, it faces challenges due to high memory costs on devices, which are deemed unacceptable. Majzoobi et al. [30] presented a slender PUF scheme. This secure scheme reliably validates responses generated by a strong PUF. The scheme limits exposure of comprehensive CRP information and ensures opponents cannot obtain the complete dataset. Verifiers in the scheme only provide a partial subset of the response during authentication. Chen et al. [22] proposed a mutual authentication protocol based on a strong PUF model. The scheme employs Shamir's secret sharing to modify the storage approach. by not storing the response on the server side while preventing exposure to adversaries. These protocols attempt to address the issue of CRP leakage at the protocol level, but they all suffer from excessive overhead and cannot meet the requirements of resource-constrained devices.

Based on the issue that the current scheme for the CRP leak problem is not lightweight enough, we can also refer to the traditional lightweight certification scheme. At present, as the focus on limited device resources grows, many authors have also designed some lightweight authentication schemes. Zhou et al. [31] proposed an efficient identity privacy authentication scheme by combining IoT architecture with cloud servers. However, it cannot defend against cloning attacks. Li et al. [32] designed a three-factor user authentication scheme for industrial wireless sensor networks. The experiments show that the scheme is robust and has low computational cost. However, this scheme cannot meet the needs of user privacy protection, and the scope of application of the scheme is greatly reduced. Lee et al. [33] proposed two lightweight cloud-based RFID authentication and key protocol protocols for electronic healthcare systems using BS-PUF. The proposed protocol uses BS-PUF with exchange characteristics for key exchange. Li et al. [34] introduced an IoT mutual authentication and key exchange protocol. This protocol integrates PUF with certificateless public key encryption on an elliptic curve, achieving "three handshakes" authentication without real-time server involvement. Esfahani et al. [35] introduced a lightweight authentication method for M2M communication in an IIoT system using only hash and exclusive-OR operations. However, Aghili et al. [36] showed that it is risky and malicious wireless sensors can obtain pre-shared and session keys. Although these schemes address the need for lightweight to some extent, they do not consider the leakage of CRPs.

In conclusion, existing schemes still do not solve the CRP leakage problem well and meet the needs of resource-constrained devices. Therefore, we aim to design a lightweight authentication scheme that solves the CRP leakage problem

to meet the device authentication requirements in the IIoT system.

III. Preliminaries

This section will briefly discuss the preliminaries used in this study, including the physical unclonable functions and an elliptic curve cryptosystem.

A. Physical Unclonable Function

The PUF is a special chip that works as follows: when receiving specific different inputs, it will produce different outputs due to the particularity of its internal structure, equivalent to a digital fingerprint of a chip [37]. The output response is not stored in digital memory. The properties of PUFs depend on the random physical factors introduced during their manufacture and such random physical factors cause each PUF to have a different microstructure. The presence of these stochastic factors renders the output of PUF challenging to anticipate and reproduce. It also makes the structure of PUF difficult to clone. The PUF has the following characteristics:

- *Unique*: Each PUF chip is randomly distributed and unique.
- *Anti-cloning*: The chip itself is extracted during the chip manufacturing process due to uncertainties, and it is impossible to reproduce the same PUF value.
- *Unpredictability*: Due to the nature of the chip circuitry, there is no way to predict the operating mode of the PUF before it is manufactured at the factory.
- *Tamper-proof*: The value of the CRP of the PUF cannot be locally modified.
- *No need for storage*: Each time the PUF is called, it only needs to be extracted in the circuit structure, and no storage elements are required to store it.

B. Elliptic Curve Cryptography

Consider the finite field $\mathbb{F}_p$, where p is a prime number. Let E denote a set of points on an elliptic curve defined over $\mathbb{F}_p$ by the equation $y^2 = x^3 + ax + b\,(mod\,p)$, where $a, b \in \mathbb{F}_p$. The key properties of this elliptic curve are summarized below [38].

- Point addition: Take any two points P and Q on the elliptic curve (if P and Q coincide, make the tangent to P) and make a straight line intersecting the elliptic curve at another point $R^{'}$. The point $R^{'}$ has symmetry R about the x-axis. Finally, $P + Q = R$.
- Scalar point multiplication: The scalar multiplication of E is defined as $mP = P + P + ... + P\,(m\,times)$, where $m \in Z_q^*$, and $m > 0$.
- Elliptic curve discrete logarithm problem (ECDLP): Given a point P and another point Q on an elliptic curve, ECDLP is to find the integer s such that $sP = Q$. Here s is the discrete logarithm. Given s and P, it is easy to compute Q, but given P and Q, it is very difficult to compute s from the known tuple $\{P, Q = sP \in G\}$.

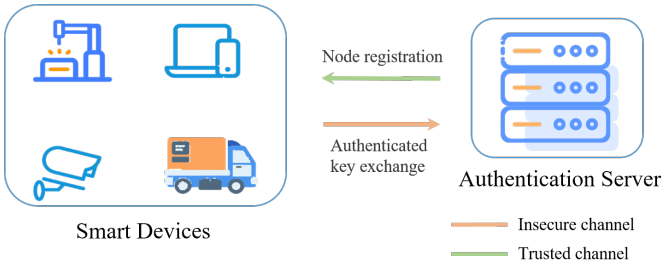


Fig. 1. System model.

IV. Models and Security Goals

This section begins with an exposition of the system model for the proposed scheme, followed by a concise overview of the threat model and security goals.

A. System Model

The system has two major types of entities, as shown in Fig. 1: the IIoT device and the server.

- **Device ($\mathbf{D_i}$)**: The D_i is a terminal smart device in our model, with relatively limited computational and storage resources. When the device joins the system, the device initiates an authentication request to the server. Upon successful authentication, the device connects with the server to access services. The PUF is embedded in the device, and the device can perform cryptographic hash functions, exclusive-OR operations, scalar multiplication operations, etc.
- **Server (S)**: The S The server, acting as a semi-honest entity in the proposed scheme, supplies resources to certify devices effectively, equipped with ample capacity to manage complex operations. When a device joins the system, the server authenticates its identity. Upon successful authentication, the server grants service access permissions to the device and provides the required services. The server has a local data storage database and can perform cryptographic hash functions and exclusive-OR operations, scalar multiplication operations, etc.

B. Threat Model

We outline the threat model that the proposed scheme satisfies, focusing on the following aspects:

- In the proposed scheme, adversaries, including the conventional Dolev-Yao model, are positioned within the network architecture, clandestinely intercepting messages exchanged between devices and servers. They aim to access sensitive information regarding IIoT systems, such as production decisions.
- The adversary can also modify and block certain information sent by both entities. After intercepting a message, an attacker can alter and manipulate segments of the message before transmitting the modified content to the intended recipient. Additionally, an adversary may forge messages by posing as a valid user.

- An adversary may also collect CRP information from the PUF by tapping the channel. When a certain amount of CRP is collected, the adversary can model the PUF to impersonate a legitimate user.

C. Security Threats and Goals

In this subsection, we outline the security threats and goals that the proposed scheme must achieve, focusing on the following aspects:

- **Semi-trusted S**: In our scheme, we consider the server as a semi-trusted entity, i.e., it is honest but curious. Specifically, it honestly performs granting and authentication protocols, but it is curious about all the access logs it maintains and wants to know additional information about the devices, especially the identity of the authenticated devices.
- **Mutual authentication**: To ensure that the identities of the entities in the system are legitimate and to avoid leaking private information, the system must verify the identities of both parties before sending sensitive information.
- **Confidentiality**: To ensure that adversaries do not capture important information in the system, the confidentiality of the information must be guaranteed.
- **Anonymity**: Ensures the device's true identity remains protected. When the smart device sends a message, only the server can discern its actual identity. The adversary cannot ascertain the device's true identity even if intercepted.
- **Unlinkability**: Because of the presence of random numbers, even if an attacker obtains two pseudonyms associated with the same device, they cannot ascertain the link between the two pseudonyms.
- **Forward and backward secrecy**: To protect previously used data from attacks, data that will be used in the future will not be affected. Our scheme supports both forward and backward security.
- **Resistance to common attacks**: Our scheme effectively defends against common threats like replay attacks, man-in-the-middle attacks, and modification attacks, ensuring robust communication security within the system.

V. Proposed Scheme

This section begins with an overview of the proposed scheme, followed by detailed explanations. The scheme is structured into four phases: setup, registration, authentication, and update. Furthermore, Table I summarizes all the notations used throughout this study.

A. Overview of the Proposed Scheme

Firstly, during the setup phase, the server initializes certain public parameters. Secondly, in the registration phase, the device embedded with PUF must register with the server. The server stores the blinded response sent by the device in the local database and generates anonymous and temporary identities for the device. During authentication, both the server and the device mutually authenticate each other. Once this mutual authentication succeeds, the session key is negotiated for subsequent communication. At the end of the session, a dynamic update phase is performed to ensure that the same CRP is not reused.

TABLE I
Notations and Definitions Used

Notations	Definition
D_i	The ith smart device
S	The server
ID_i	The real identity of the device D_i
PID_i	The pseudonym of the device D_i
T_D, T_S	The current timestamp
$H_i(i = 1, 2, 3)$	Hash function
SK_i	Session key negotiated by D_i and S
M_i	The message sent by D_i or S
PUF	The physical unclonable function
$CRP(C_i, R_i)$	Challenge-response pair for the ith round
ECC	Elliptic curve cryptosystem
ΔT	The validity period of the M_i
$\oplus$	The exclusive-OR operation
$\|\|$	The concatenation operation
Enc/Dec	Symmetric encryption/decryption

B. Setup Phase

The system is initialized in the setup phase, and system public parameters are generated for subsequent sessions. The server S selects the parameters $(\mathbb{G}, q)$ of the elliptic curve as the foundation for generating system parameters. Then the server S randomly chooses two generators k and h of $\mathbb{G}$. In addition, the server S chooses three one-way hash functions: $H_i : \{0,1\}^* \rightarrow Z_q^*$. Finally, the public parameter of the system is $Paras = \{q, \mathbb{G}, h, k, H_1, H_2, H_3\}$.

C. Registration Phase

In the proposed scheme, all devices that want to join the system need to ensure that the PUF embedding is performed by the PUF generator. The whole registration process is done in a secure channel. Then, the device can initiate a registration request to the server S.

Step 1 : Device $\rightarrow$ Service: $M_1 = \{ID_i\}$

The device D_i sends its ID_i to the server S to start the registration process.

Step 2 : Server $\rightarrow$ Device: $M_2 = \{C_i, PID_i\}$

(1) After receiving the device ID_i, the server selects a challenge C_i randomly ($C_i \in \{0,1\}^*$) and sends them to D_i.

(2) Also a random number $b \in Z_q^*$ is selected for generating the pseudonym of D_i and sent to D_i. The identity calculation formula is as follows:

$$PID_i = H_1(C_i||ID_i\|b) \tag{1}$$

The server S stores the pseudonym in the database. Then it sends message $M_2 = \{C_i, PID_i\}$ to device.

Step 3 : Device $\rightarrow$ Server: $M_3 = \{A_i\}$

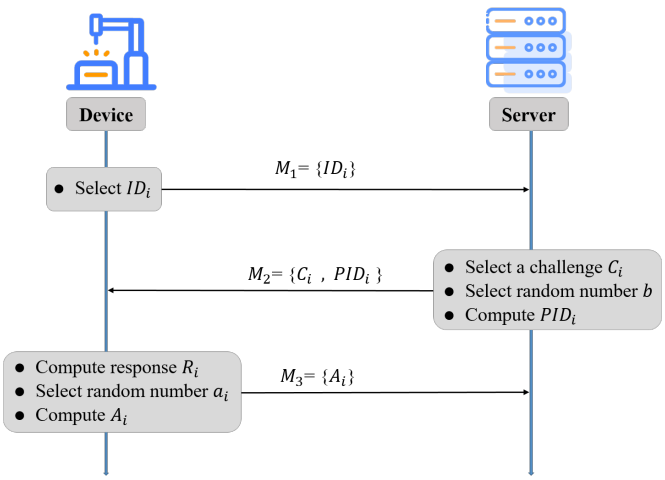


Fig. 2. Registration phase.

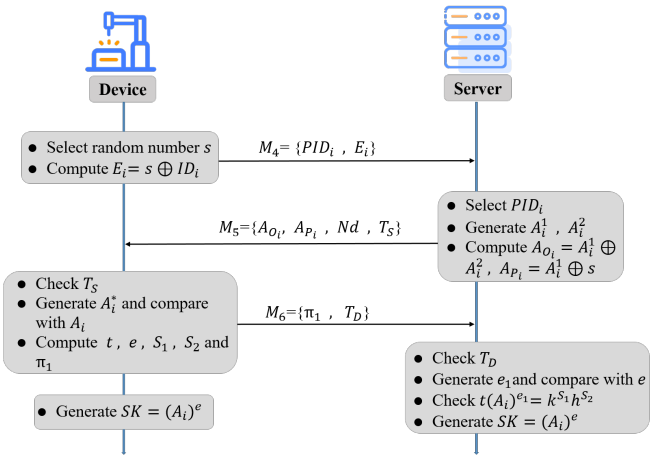


Fig. 3. Login and authentication phase.

When the device D_i receives the challenge from the server S, it generates the corresponding responses R_i using PUF and selects a random number a_i for generating the blinded response with the following formula:

$$A_i = k^{R_i} h^{a_i} \tag{2}$$

Where A_i is the blinded response for the subsequent normal authentication process. After generating the response, the device D_i stores the ID_i in the database. The data format stored by D_i is $\{a_i, C_i, ID_i, PID_i\}$. Then D_i sends the message $M_3 = \{A_i\}$ to the server S.

Step 4 :

When the server S receives the message M_3, it parses the blinded response and stores it in the database. Then the server S stores the data in the format $\{PID_i, ID_i, A_i\}$.

D. Login and Authentication Phase

Step 1 : Device $\rightarrow$ Server: $M_4 = \{PID_i, E_i\}$

Firstly, the device D_i selects random number $s \in Z_q^*$, calculates $E_i = s \oplus ID_i$, and sends pseudonyms PID_i and E_i to the server S for authentication through an open channel, i.e., it sends message $M_4 = \{PID_i, E_i\}$ to the server S.

Step 2 : Server $\rightarrow$ Device: $M_5 = \{A_{O_i}, A_{P_i}, Nd, T_S\}$

(1) After After receiving the message from the device D_i, the server S authenticates the request information sent by D_i.

(2) The server S first records the current time as T_S. Then, the server S will query the database for the corresponding ID using the PID_i sent by the device D_i and extract the random value s from the E_i transmitted by the device. After the extracting is complete, the server S selects the relevant blind response A_i from the database for the next authentication.

(3) The server S divides the blinded response A_i into two characters of equal length, A_i^1 and A_i^2, where $A_i = A_i^1 || A_i^2$. Then the server S computes A_{O_i} and A_{P_i} from the parsed random number s with the following formula:

$$A_{O_i} = A_i^1 \oplus A_i^2 \tag{3}$$

$$A_{P_i} = A_i^1 \oplus s \tag{4}$$

Then a random number $Nd \in Z_q^*$ is selected by the server S and it sends the message $M_5 = \{A_{O_i}, A_{P_i}, Nd, T_S\}$ to the device D_i.

Step 3 : Device $\rightarrow$ Server: $M_6 = \{\pi_1, T_D\}$

(1) When the message M_5 is received by the device D_i, it first verifies the timestamp's freshness to ensure the transmission delay falls within the permissible time interval ΔT, i.e., $|T_D^* - T_S| < \Delta T$. ΔT is the average time threshold (empirical value) for the server S and D_i to complete authentication several times to prevent the message from being replayed. D_i records the time value T_D at that moment.

(2) The device D_i recovers the blinded response A_i^* from the received message. D_i reads the corresponding challenge C_i and the random number a_i from the database. Then, D_i inputs the challenge C_i into the PUF to get the response R_i. Then, the device D_i generates the blinded response A_i with the response R_i and compares the computed value A_i with the A_i^*. If they match, the server S successfully authenticates D_i. Otherwise, the authentication fails.

(3) After the identity of the server is confirmed, the device D_i selects two random numbers $r_1, r_1 \in Z_q^*$, then calculates the following equation:

$$t = k^{r_1} h^{r_2} \tag{5}$$

$$e = H_2(t||Nd||A_i) \tag{6}$$

$$S_1 = r_1 + R_i e \tag{7}$$

$$S_2 = r_2 + a_i e \tag{8}$$

Finally Device generates the proof $\pi_1 = (t||S_1||S_2)$ for authentication. The device D_i sends the message $M_6 = \{\pi_1, T_D\}$ to the server S.

Step 4 :

(1) After the message M_6 is received by the server S, it first verifies the timestamp's freshness to ensure the

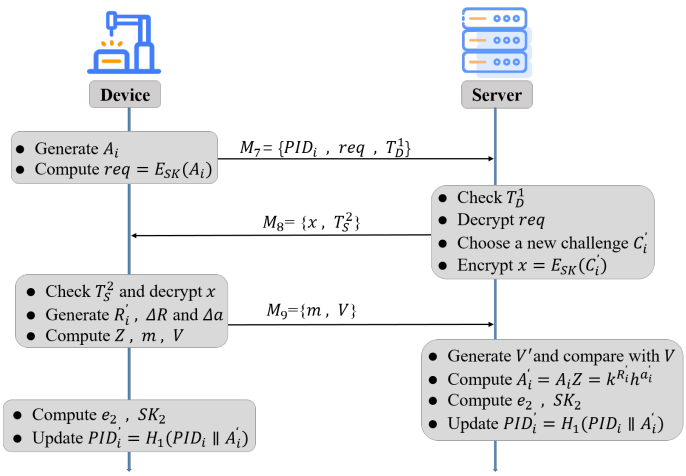


Fig. 4. Update Phase.

transmission delay falls within the permissible time interval T_D is within the allowed time interval ΔT, i.e., $|T_S^* - T_D| < \Delta T$.

(2) Then it reads the corresponding blinded response A_i of the device D_i from the database, and then parses t, S_1 and S_2 from π_1. Then it calculates $e_1 = H_2(t||Nd||A_i)$ and compares the calculated value e_1 with the e sent by D_i. If it is equal, it proceeds to the next authentication operation. If it is unequal, the server S will deny further authentication.

(3) The server S uses the calculated e_1 for the next authentication step and calculates whether the following formula holds: $t(A_i)^{e_1} = k^{S_1} h^{S_2}$. If the above formula holds, the server's authentication of the device D_i succeeds. Otherwise, the server's authentication of D_i fails.

(4) When the device D_i completes the authentication, the server S and D_i carry out the session key negotiation. They generate the session key as $SK = (A_i)^e$ and then directly use it for communication.

E. Update Phase

To achieve dynamic device management and ensure the entire communication process is secure, it is necessary to update some of the variables used, including CRP, pseudonyms, and session keys.

Step 1 : Device $\rightarrow$ Server: $M_7 = \{PID_i, req, T_D^1\}$

The device D_i produces a blinded response A_i using the challenge C_i and the random number a_i stored in the database and generates an update request $req = E_{SK}(A_i)$ with A_i, records the current time as T_D^1, and then sends the message $M_7 = \{PID_i, req, T_D^1\}$ to the server S.

Step 2 : Server $\rightarrow$ Device: $M_8 = \{x, T_S^2\}$

(1) When the message M_7 is received by the server S, it first verifies the timestamp's freshness to ensure the transmission delay falls within the permissible time interval T_D is within the allowed time interval ΔT, i.e., $|T_S^1 - T_D^1| < \Delta T$. If it is within the time interval, the server S records the current time as T_S^2.

(2) Then the server S finds the corresponding blinded response A_i in the database based on the pseudonym of the device D_i and decrypts the update request sent by D_i with the negotiated session key to get A_i^*, compares A_i^* and A_i to see if they are equal. It agrees to the update request if they are equal.

(3) The server S selects a new challenge $C_i^{'}$, encrypts it with the session key, and gets the ciphertext x, which is calculated as $x = E_{SK}(C_i^{'})$. After the computation, the server S sends the message $M_8 = \{x, T_S^2\}$ to the device D_i.

Step 3 : Device $\rightarrow$ Server: $M_9 = \{m, V\}$

(1) When the device D_i receives the message from the server S, it first checks whether the transmission delay T_S^2 is within the allowable time interval ΔT, i.e., $|T_D^2 - T_S^2| < \Delta T$. If it is within the time interval, the server S records the current time as T_D^3.

(2) The device D_i then decrypts the message x from the server S with the negotiated session key to obtain a new challenge $C_i^{'}$, D_i inputs the new challenge $C_i^{'}$ into the PUF, generating a fresh response $R_i^{'}$. Subsequently, D_i obtains the update increment of the response $\Delta R = R_i^{'} - R_i$. At the same time, the device D_i generates a random number $a_i^{'} \in Z_q^*$ and calculates the increment of the random number as $\Delta a = a_i^{'} - a_i$.

(3) The device D_i generates Z with the update increment and random number increment, and then generates the encrypted ciphertext m with Z and the new challenge $C_i^{'}$, calculated by the following formula:

$$Z = k^{\Delta R} h^{\Delta a} \tag{9}$$

$$m = E_{SK}(Z \oplus C_i^{'}) \tag{10}$$

Finally, device generates $V = H_3(SK||m||C_i^{'})$ with this data, and then the device D_i sends the message $M_9 = \{m, V\}$ to the server S.

Step 4 :

(1) After receiving the message from the device D_i, the server S generates $V^{'}$ with the received ciphertext m, the new challenge $C_i^{'}$, and the session key, and compares the received V with the generated $V^{'}$ to check whether they are equal. If they are not, the update fails.

(2) If it is equal, the server S recovers Z from the ciphertext and then calculates a new blind response $A_i^{'}$, calculated as $A_i^{'} = A_i Z = k^{R_i^{'}} h^{a_i^{'}}$, and generates a new session key SK_2 with the new blind response, calculated as follows:

$$e_2 = H_2(A_i^{'}||C_i^{'}) \tag{11}$$

$$SK_2 = (A_i^{'})^{e_2} \tag{12}$$

(3) Finally, the used device pseudonym needs to be updated, and the new device pseudonym is set as $PID_i^{'} = H_1(PID_i||A_i^{'})$, and the data format stored by the server S is $\{PID_i^{'}, A_i^{'}, SK_2\}$.

(4) The device D_i also updates the used alias to PID_{i+1} and calculates the new session key SK_2 in the same way as the server S, storing data in the format $\{PID_i^{'}, C_i^{'}, SK_2\}$.

VI. Security Proof And Analysis

In this section, we analyze the security of the proposed scheme using a random oracle model.

A. Security Proof

Based on the proposed scheme, we initially establish a security model delineating the attacker's capabilities and simulating interactions with them. The proposed scheme involves two types of participants, i.e., a device D_i and a server S. Participants are denoted by the symbol Π^i_Λ, where Λ signifies the participant type and i indicates their index. As described below, the adversary and the challenger interact under predefined rules.

(1) Execute (Π^i_D, Π^j_S). This rule represents the communication between instances Π^i_D, Π^j_S, which a passive attacker can eavesdrop on to intercept messages transmitted during the process.
(2) Send(Π^i_D, m). This rule indicates situations where an adversary could alter or falsify a message. Upon receiving the adversary's query, the challenger will promptly respond.
(3) Reveal(Π^i_D). This rule will send the session key of Π^i_D to the adversary after receiving the query from the adversary. This is done by the challenger checking whether the two instances Π^i_D and Π^i_S are accepted. If yes, the challenger returns the relevant key to Π^i_D. Otherwise, the challenger rejects the query and outputs $\perp$.
(4) Corrupt(Π^i_D). Oracle uses this to simulate perfect forward secrecy. Upon receipt of this query, a long-term session key of Π^i_D is returned to the adversary $\mathcal{A}$.
(5) Test(Π^i_D, Π^j_S). This rule allows the adversary to query only once. Upon receiving a query from the adversary $\mathcal{A}$, a random bit $b \in \{0,1\}$ is selected by the challenger. The challenger verifies whether the value of b equals 1. If it does, the challenger transmits the session key to the opponent. Otherwise, the challenger sends a random key to the opponent.

Definition 1 (Partnership). This definition means that if two instances, Π^i_Λ and Π^i_Λ, can establish a communication channel with each other and negotiate a public session key, then the two instances are a partnership.

Definition 2 (Correctness). This definition means that in the process that two instances Π^i_Λ and Π^j_Λ with partnership are communicating, i.e., these two instances should generate an identical session key that is not empty. All instances should compute the session key correctly.

Definition 3 (Freshness). An instance Π^i_Λ is deemed fresh if neither Π^j_Λ nor its partner has not failed to query the Reveal(Π^i_Λ) oracle.

Definition 4 (Semantic Security). At the end of the communication, adversary $\mathcal{A}$ is prompted to query the (Π^i_Λ) oracle and transmit a $b^{'}$ ($b^{'} \in \{0,1\}$) signifying a correct guess. Assuming $\mathcal{A}$ guesses the value of b as W, the advantage of adversary $\mathcal{A}$ in compromising the semantic security of AKA protocol P can be quantified as $Adv^A_P = 2|Pr[W] - 1/2|$. If If the probability Adv^A_P is is extremely low for the adversary $\mathcal{A}$, then semantic security is upheld.

Theorem 1. If ECDLP is unsolvable for polynomial-time bounded probability adversary $\mathcal{B}$, then we can say that the proposed scheme P is secure for polynomial-time bounded probability adversary $\mathcal{A}$ under the security model defined above.

Proof. we use the game-jumping technique, which consists of a series of games with interactions between adversaries and challengers. Denote the event where $\mathcal{A}$ wins game i as G_i. These games can be structured in the following manner.

Game G_0**.** In G_0, $\mathcal{A}$ A has permission to query $\mathcal{B}$ as if consulting an oracle, where $\mathcal{B}$ is acting as a semantically secure game challenger to $\mathcal{A}$'s session key. Then $\mathcal{B}$ will return the response. To respond to $\mathcal{A}$, $\mathcal{B}$ need to establish initial parameters. Specifically, $\mathcal{B}$ sends the public parameter $Paras = \{q, \mathbb{G}, h, k, H_1, H_2, H_3\}$ to $\mathcal{A}$.

- send(Π^i_Λ, m). Depending on the specific type of request, this oracle can be segmented into sub-oracles as follows:

(1) send(Π^i_D, req). After this query is received, a random number s is selected by $\mathcal{B}$, $\mathcal{B}$ computes $E_i = s \oplus ID_i$, and sends message tuple $M_4 = \{PID_i, E_i\}$ to $\mathcal{A}$.
(2) Send (Π^i_S, M_4). After this query is received, $\mathcal{B}$ parses the random value s from E_i transmitted by the device D_i, records the current time as T_S, calculates the computed $A_{O_i} = A^1_i \oplus A^{'}_i$ and $A_{P_i} = A^1_i \oplus s$ and sends the message $M_5 = \{A_{O_i}, A_{P_i}, T_S\}$ to $\mathcal{A}$.
(3) Send(Π^i_D, M_5). After this query is received, $\mathcal{B}$ verifies the accuracy of A_{O_i}, A_{P_i}. If verification fails, the query will be rejected by $\mathcal{B}$ and $\mathcal{B}$ sends $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ obtains the current timestamp T_D, selects two random numbers r_1 and r_1, and computes the $t = k^{r_1}h^{r_2}, e = H_2(t||Nd||A_i), S_1 = r_1 + R_i e$ and $S_2 = r_2 + a_i e$ and generates the proof $\pi_1 = (t||S_1||S_2)$ for authentication. The device D_i sends the message $M_6 = \{\pi_1, T_D\}$ to $\mathcal{A}$.
(4) Send (Π^i_D, M_6). After $\mathcal{B}$ receives the query, $\mathcal{B}$ verifies the proof in message M_6. If verification fails, the query will be rejected by $\mathcal{B}$ and $\mathcal{B}$ sends $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ accepts the instance Π^i_D, and M_4, M_5, M_6 will be added by $\mathcal{B}$ to the message list ML.

- Execute (Π^i_D, Π^i_S). After this query is received by $\mathcal{B}$, it extracts the corresponding message tuple $\{M_4, M_5, M_6\}$ from ML and sends the tuple to $\mathcal{A}$.
- Reveal(Π^i_Λ). After receiving the query, $\mathcal{B}$ verifies whether the instance Π^i_Λ has already been reviewed. If so, $\mathcal{B}$ promptly transmits the corresponding session key to $\mathcal{A}$. Alternatively, $\mathcal{B}$ rejects the query if the instance has not been reviewed and sends $\perp$ to $\mathcal{B}$.
- Test (Π^i_Λ). At the end of the game, $\mathcal{A}$ A needs to execute this query once against $\mathcal{B}$. After receiving this query, Upon receiving a query from the adversary, the challenger chooses a random bit $b \in \{0,1\}$. The challenger verifies whether the value of b equals 1. If it does, the challenger transmits the session key to the opponent. Otherwise, the challenger sends a random key to the opponent. Game G_0 simulates the original attack. Therefore, the advantage of $\mathcal{A}$ in breaking G_0 satisfies:

$$Adv^A_P = 2\big|Pr[W_0] - 1/2\big| \tag{13}$$

Game G_1**.** G_1 is the same as G_0, except that the hash prophecy machine is simulated in G_1. Specifically, G_1 is the same as G_0, except that it simulates hash prediction. In particular, $\mathcal{B}$ establishes three key-value structures L_1, L_2 and L_3. Upon receiving the hash query m_i for $h_i (1 \leq or \leq i \leq or \leq 3)$ from $\mathcal{A}$, $\mathcal{B}$ verifies if m_i exists as a key in L_i. If so, $\mathcal{B}$ sends $L_i < m_i >$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ selects a random number $x_i \in Z_q^*$, assigns $L_i < m_i >= x_i$, and and transmits x_i to $\mathcal{A}$. In $\mathcal{A}$, $G_1$is indistinguishable from G_0. Therefore,

$$Pr[W_0] = Pr[W_1] \tag{14}$$

Game G_2**.** G_2 is the same as $G1$, except that there are no collisions. In the game $G2$, h_1 is modeled as a random predictor. According to the birthday paradox formula, the collision probability for h_i is at most $q_h^2/2q$. In message tuples $\{M_4, M_5, M_6\}$, the random numbers s, r_1 and r_2 are modeled, which implies that the probability of collision of message tuples is at most $(q_s + q_e)^2/(2q))$. Thereby,

$$\left|Pr[W_2] - Pr[W_1]\right| \leq or \leq ((q_s + q_e)^2 + \sum_{i=1}^{3} q_{h_i}^2)/(2q) \tag{15}$$

Lemma 2. In the proposed scheme, adversary $\mathcal{A}$ cannot obtain the CRP of the PUF.

Proof. In the authentication process of the proposed scheme, the CRP generated by the PUF is the key information in the secure authentication process. The device side does not generate the starting incentive C_i for scheme authentication. Still, it is randomly selected by the service and sent to the device side, preventing adversary $\mathcal{A}$ from calling the Corrupt() instruction to read the incentive C_i. To prevent the CRP of the PUF from being modeled, the server side does not store the challenge of the used PUF. The device side also does not store the response but sends the generated blinded response to the server for storage. the blinded response is not directly transmit by the the server S when performing the verification but parses it into two parts by bit, plus a random number operation before transmitting it, to ensure that the blinded response will not be intercepted in the transmission process. Moreover, even if the database on the server side is leaked, only the blinded response is stored, and the probability of getting the response by cracking the blinded response. Thereby, getting the response is not greater than the probability of random guessing, and the probability of cracking it is equivalent to solving the ECDLP problem successfully, which is extremely difficult, i.e., the CRP of the PUF will not be accessible by the adversary.

B. Security Analysis

In this subsection, we analyze the security features that the proposed scheme satisfies.

(1) Mutual Authentication: Interaction messages related to the server S are protected using ECDLP issues. This includes registration messages sent by the device and pseudonymized messages sent by the server S. Therefore, these messages are computationally difficult to modify. AKA messages sent from devices are essentially protected by zero-knowledge proof mechanism. To be specific, the requester device D_i must register its blinded response before entering an AKA session. Once the device registration by the server S is successful, no one can know the response of the PUF and cannot get the blinded response through computation, greatly protecting the challenge response pair correspondence mechanism of the PUF. From the perspective of the device D_i, the server S will responds to the authentication request D_i after accepting it from D_i. Only the server S can accurately send the authentication message due to the confidentiality of A_i. So, the server S can prove the legitimate identity to D_i. Through a single interaction round, the device D_i and the server S can mutually authenticate because the session key negotiated between the device D_i and the server S remains unknown to the adversary, leveraging the unidirectional nature of the hash function. If the ECDLP problem is intractable, an adversary cannot create valid messages. Therefore, as long as the following equations $t(A_i)^{e_1} = k^{S_1} h^{S_2}$ are satisfied, the scheme can guarantee mutual authentication.

(2) Confidentiality: The key negotiated during the authentication phase is $SK = (A_i)^e$, which is known only to the device D_i and the server S. All other entities must compute A_i and e to obtain SK. However, the element that computes A_i is known only to device D_i and server S. Even if an adversary computes e, it cannot know A_i. Thereby, no entity other than device D_i and server S can find the correct value using a better method than random guessing. So, the final established session key is kept secret.

(3) Untraceability: The message from the device D_i contains a pseudonym, a hash value, an elliptic curve point, and a timestamp. The pseudonym generated by D_i is obtained based on the hash function. To ascertain the real identity of D_i, the adversary must be able to compute $PID_i = H_1(C_i||ID_i||b)$. However, the real identity ID_i of the device is difficult to obtain for the adversary after processing the one-way hash function. After the update phase, pseudonyms are refreshed for each round, with each round being unique, and the adversary can not obtain the random number a_i, the adversary is incapable of inferring the real identity of the device from the previous round of pseudonyms. Therefore, the proposed scheme

TABLE II
COMPARISON OF SECURITY PROPERTIES

Security Properties	[39]	[17]	[40]	ours
Mutual Authentication	✓	✓	✓	✓
No Third Party Required	×	×	✓	✓
No Explicit CRP	×	✓	×	✓
No NVM Required	✓	✓	×	✓
Session Key Agreement	✓	×	✓	✓
Resist Replay Attack	✓	✓	✓	✓
Forward Secrecy	✓	✓	✓	✓
Backward Secrecy	×	✓	×	✓
Untraceability	×	✓	✓	✓

✓: The security property is satisfied
×: The security property is not satisfied.

achieves untraceability.

(4) Forward and backward secrecy: The session key of the whole system process is $SK = (A_i)^e$, in which the calculation of A_i uses the random number a_i, the calculation of e also uses the random number, as well as the keyed hash function, and the calculation results are one-time and random, which results in the final output of the session key of the different sessions are independent. In other words, the keys SKs generated during separate sessions are distinct and not directly correlated, and they will not leak and affect each other. Therefore, the scheme can satisfy both forward secrecy and backward secrecy.

(5) Resistance to replay attack: After receiving a new message, the receiver needs to verifies the timestamp's freshness to ensure the transmission delay falls within the permissible time $|T_D^* - T_S| < \Delta T$. The message will be considered expired by the receiver if it exceeds this duration. In addition, the random number Nd used by the server in different rounds is not the same. Even if the adversary intercepted π_1, the π_1 used random numbers. Thereby, the adversary could not pass the previous round of π_1 to pass the next authentication round. These ensure that the scheme is resistant to replay attacks.

(6) Resistance to man-in-the-middle attacks: Active adversaries or passive adversaries based on PPT cannot forge the message content in the scheme. Moreover, in this scheme, even if the attacker obtains the message content through the public channel, The probability of the attacker obtaining valid information is zero. Because the pseudonyms and random numbers used throughout the protocol are protected using a one-way hash function, the secret message is known only to the device and the server. So, even if an attacker intercepts a message, it cannot be decrypted and used. Moreover, since The session key SK remains entirely confidential from the attacker, any modification of the message content will not pass the checking process performed by the receiver, and if the attacker modifies the message, it will be discovered by the authenticated parties, and the communication will be rejected.

(7) Resistance to modification attack: We can ascertain any alteration to the message $M_6 = \{\pi_1, T_D\}$ by verifying the validity of the formula $t(A_i)^{e_1} = k^{S_1} h^{S_2}$. If the computation results are unequal, it means that the message is modified by the adversary, and the receiver of the message will think that it is not a legitimate message and refuse authentication. Therefore, the authentication scheme proposed in this study can be used to defend against modification attacks.

C. Security and Privacy Comparisons

In this subsection, we compare our proposed scheme with other relevant schemes such as Chuang et al. [39], Chatterjee et al. [17], Gope et al. [40] based on their fulfillment of security and privacy requirements. Table II summarizes this comparison, emphasizing that our scheme satisfies a wider array of security and privacy requirements.

VII. Performance Analysis

In this section, we first describe the experimental configuration for the experiments in this study and then compare and evaluate our scheme with the comparison schemes [17], [39], [40]. The proposed scheme is implemented on a single mainframe computer with Windows as the experimental environment and a 2.5GHz Intel Core i7-11700 CPU and 16GB of RAM. For the software implementation, we use the BLS12381 curve of Miracle Core [41] in the proposed scheme. The functionality of PUF is simulated by Python's pypuf library.

A. Computation Cost Analysis and Comparison

The computational overhead of the proposed scheme and some comparison schemes is analyzed in this section. Some notations for the computation time are defined as shown in the table III.

We give a comparison of the specific number of operations, computational cost of the schemes of Chuang et al. [39], Chatterjee et al. [17], Gope et al. [40] and our scheme in Table IV, summarised in Fig. 5. During the comparison, we do not consider the setup phase of scheme [17], [39], [40] and our scheme, only the main authentication phase. Because the cost of the setup phase is generally small. Therefore, we mainly compare the computational elapsed time of the AKA phase. We have omitted some operations with negligible computational cost, such as the exclusive-OR operation.

In the comparison of Table IV and Fig. 5, we can see that the computation of Chuang et al.'s scheme [39]. On the verifier side involves one bilinear pairing, two scalar multiplication, one dot-add, and six hash operations, which in total require a computational cost of $1T_{bp}+2T_{sm}+1T_{add}+6T_h = 1.984\,ms$. The device is required to perform the same operations on the device side. i.e., one bilinear pairing, two dot multiplication, one dot addition, and six hash operations. Therefore, the computational cost on the device side is the same as that on the verifier side, which is $1T_{bp}+2T_{sm}+1T_{add}+6T_h = 1.984\,ms$. Moreover, in Chuang et al.'s scheme [39], a trusted third party is required to conduct the authentication process, and an independent data provider is employed to carry out a certain amount of data storage. These both increase the storage cost and computational overhead to some extent.

In Chatterjee et al.'s scheme [17], the computation on the verifier side of the authentication process involves one bilinear pairing, two scalar multiplication, five dot-add, and seven hash operations, with a total required computational cost of $1T_{bp} + 2T_{sm} + 5T_{add} + 7T_h = 1.993\,ms$. On the device side of the process, the major computation involves one bilinear pairing, two scalar multiplication, ten hash, and one PUF computation operations. So the total computation time required is $1T_{bp}+2T_{sm}+1T_{PUF}+10T_h = 3.607\,ms$. In Chatterjee et al.'s scheme [17], a trusted third party is required to generate the security credentials, and a security association provider is necessary to associate the PUFs with the devices. These add computational overhead and some exposure risk.

In Gope et al.'s scheme [40], the computation on the verifier side involves one FHD and six hash operations during the authentication process. So, the total computational cost

TABLE III
COMPUTATION TIME

Operation	Definition	Symbolic description	Execution time (ms)
T_{sm}	The scalar multiplication operation	$xP, i.e., P+P+...+P$	0.217
T_{add}	The point addition operation	$P+Q$	0.002
T_{PUF}	The PUF operation	$R=PUF(C)$	1.621
T_h	The hash operation.	$H(x)$	0.001
T_{FHD}	The FHD operation	$FHD(R,R^*)$	0.002
T_{bp}	The bilinear pairing operation	$e(P,Q)$	1.542

TABLE IV
COMPARISONS OF COMPUTATION OVERHEAD

Schemes	Device	Verifier
Chuang *et al.* [39]	$1T_{bp}+2T_{sm}+1T_{add}+6T_h=1.984\,ms$	$1T_{bp}+2T_{sm}+1T_{add}+6T_h=1.984\,ms$
Chatterjee *et al.* [17]	$1T_{bp}+2T_{sm}+1T_{PUF}+10T_h=3.607\,ms$	$1T_{bp}+2T_{sm}+5T_{add}+7T_h=1.993\,ms$
Gope *et al.* [40]	$1T_{FHD}+1T_{PUF}+6T_h=1.806\,ms$	$1T_{FHD}+6T_h=0.185\,ms$
Ours	$2T_{sm}+1T_{add}+1T_h=0.437\,ms$	$3T_{sm}+1T_{add}+1T_h=0.654\,ms$

required is $1T_{FHD}+6T_h=1.806\,ms$. Whereas on the device side, it mainly involves one FHD, six hash, and two PUF computation operations, and the total computational time required is $1T_{FHD}+2T_{PUF}+6T_h=0.185\,ms$. It can be seen that on the device side, two PUF calculations are required, which has a high computational cost. The PUF challenge will be transmitted during the authentication process, which will risk being captured by eavesdroppers and conducting man-in-the-middle attacks.

In the proposed scheme, our computation on the verifier side involves one hash, three scalar multiplication, and two dot-add operations, with a total required computation cost of $3T_{sm}+1T_{add}+1T_h=0.654\,ms$. On the device side, it mainly involves two scalar multiplication, one dot-add, and one hash operations, with a total required computation time of $2T_{sm}+1T_{add}+1T_h=0.437\,ms$. Combined with Fig. 5, it is evident that our scheme's overhead is less than the comparison scheme at both the device and the verifier sides. The proposed scheme does not require additional trusted third parties and service providers, reducing the storage overhead and computation overhead. For comparing scheme [40], although the proposed scheme has a slightly higher overhead on the verifier side, our overhead on the device side is much lower than scheme [40]. This means that the proposed scheme shifts computational burdens from the device side to the verifier side, enhancing overall computational efficiency significantly. In terms of feature comparison, scheme [40] meets less security than the proposed scheme, which offers greater security and meets additional security features.

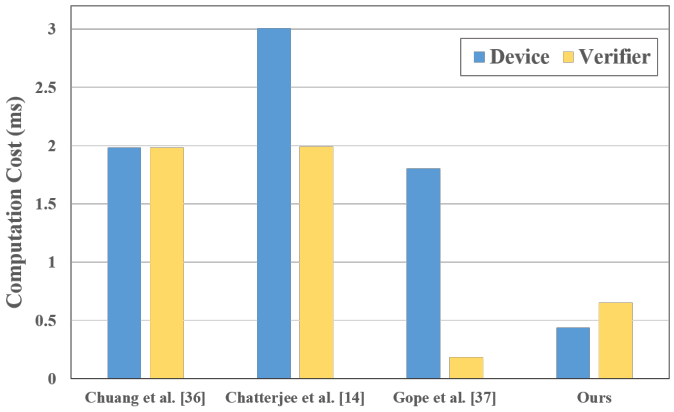


Fig. 5. The comparison of computation cost.

TABLE V
SIZE OF DIFFERENT OPERATIONS

Operation	Definition	Size (bytes)
$\|Z_q^*\|$	random number	48
$\|G\|$	point on $\mathbb{G}$	128
$\|PUF\|$	PUF input or output	48
$\|T\|$	Timestamp	4
$\|H\|$	hash output	48
$\|ID\|$	Identity or pseudonym	48

B. Communication Overhead Analysis and Comparison

In this subsection, we examine the communication overhead of the proposed scheme and the scheme of Chuang et al. [39] , Chatterjee et al.'s scheme [17] and Gope et al.'s scheme [40]. In the subsequent comparison, we will only focus on analyzing the authentication phase costs of each scheme. The lengths of all operations are shown in Table V. So that the comparison is fair enough, Table VI shows the final comparison results.

In the scheme of Chuang et al. [39], the device needs to send $\{ID_A\}$ twice and $msg_{A\_B}=\{N_A,v_A\}$, where $N_A\in\mathbb{G}$. So the communication overhead is $2|ID|+1|H|+1|G|=272\,bytes$. And on the verifier side, it needs to send $\{ID_B\}$ twice and $msg_{B\_A}=\{N_B,v_B\}$, where $N_B\in\mathbb{G}$. So the communication overhead is $2|ID|+1|H|+1|G|=272\,bytes$. In addition, the data provider sends two messages

TABLE VI
COMMUNICATION COST

Schemes	Device	Verifier
Chuang *et al.* [39]	$2\|ID\| + 1\|H\| + 1\|G\| = 272\,bytes$	$2\|ID\| + 1\|H\| + 1\|G\| = 272\,bytes$
Chatterjee *et al.* [17]	$2\|ID\| + 2\|H\| + 3\|G\| = 576\,bytes$	$1\|ID\| + 1\|H\| + 4\|G\| + 2\|Z_q^*\| = 704\,bytes$
Gope *et al.* [40]	$1\|ID\| + 3\|Z_q^*\| + 2\|H\| = 288\,bytes$	$3\|Z_q^*\| + 1\|H\| = 192\,bytes$
Ours	$1\|ID\| + 1\|Z_q^*\| + 1\|G\| + 1\|T\| = 228\,bytes$	$1\|Z_q^*\| + 1\|T\| + 1\|G\| = 180\,bytes$

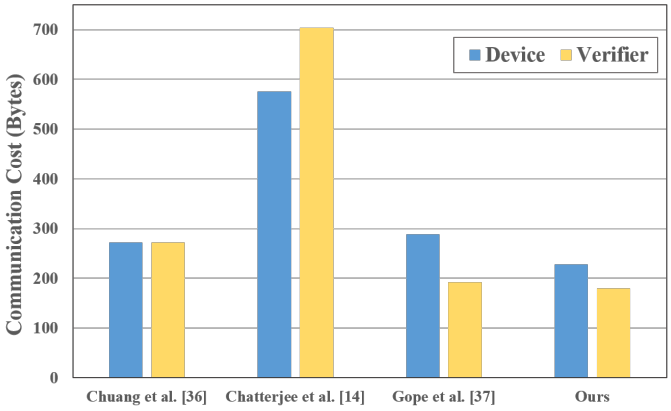


Fig. 6. The comparison of communication cost.

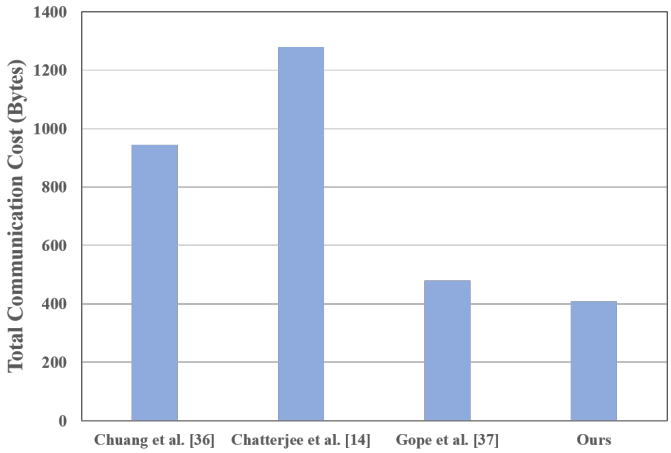


Fig. 7. The comparison of total communication cost.

$msg_{(DP\_A)} = \{ID_A, Data_A, MPK\}$ and $msg_{(DP\_B)} = \{ID_B, Data_B, MPK\}$. So, it requires an additional communication overhead of $2|ID| + 1|H| + 2|G| = 400\,bytes$. Thereby, the total communication cost is $944\,bytes$.

In the AKA phase of Chatterjee et al.'s scheme [17], the device needs to send $\{ID_A, ID_B\}$ and $\{V_A', KA_{PUB}, Y_A, H3(P_S' + KA_{PUB})||H3(Y_A)\}$, where $\{V_A', KA_{PUB}, Y_A\} \in \mathbb{G}$. So the communication overhead is $2|ID| + 2|H| + 3|G| = 576\,bytes$. On the verifier side, it needs to send $\{ID_B, A_i, HLP_A, Q_A\}$ and $\{KB_{PUB}, Q_B, Y_B, H3(H1(P_A)||H1(KB_{PUB})||H1(Q_B)||H1(Y_B))\}$, where $\{Q_A, Q_B, KB_{PUB}, Y_B\} \in \mathbb{G}$. So the communication overhead is $1|ID| + 1|H| + 4|G| + 2|Z_q^*| = 704\,bytes$. Thereby, the total communication cost is $1280\,bytes$.

In the AKA phase of Gope et al.'s scheme [40], the device needs to send $M_1 = \{OID_T^i, N_t^*, V_0\}$ and $M_3 = \{R_{i+1}^*, V_2, X\}$. So the communication overhead is $1|ID| + 3|Z_q^*| + 2|H| = 288\,bytes$. On the verifier side, the verifier needs to send $M_2 = \{C_i, R_i^{1^*}, V_1, N_s^*\}$. So the communication overhead is $3|Z_q^*| + 1|H| = 192\,bytes$. Thereby, the total communication cost is $480\,bytes$.

In the proposed scheme, on the device side, the device needs to send $M_4 = \{PID_i, E_i\}$ and $M_6 = \{\pi_1, T_D\}$. So the communication overhead is $1|ID| + 1|Z_q^*| + 1|G| + 1|T| = 228\,bytes$. On the verifier side, the device needs to send $M_5 = \{C_{O_i}, C_{P_i}, Nd, T_S\}$. Then the communication overhead is $1|Z_q^*| + 1|T| + 1|G| = 180\,bytes$. Thereby, the total communication cost is $408\,bytes$.

A comparison of our communication overheads is shown in Fig. 6, where our scheme has lower communication overheads than all the other compared schemes. Although our scheme is not much lower than Gope et al.'s scheme [40], scheme [40] requires the use of different PUF hardware, which is burdensome for resource-constrained devices.

As shown in Fig. 7, our scheme and Gope et al.'s scheme [40] have the lowest communication cost. Although we have a slightly higher overhead on the server side, the proposed scheme imposes significantly less computational overhead on devices compared to Gope et al.'s scheme [40], making it more suitable for resource-constrained devices. Overall, our proposal is still more advantageous.

VIII. CONCLUSION

In the proposed scheme, lightweight and anonymous PUF-based authentication is designed to address security and privacy issues, particularly CRP leakage, in resource-constrained IIoT devices. The proposed scheme uses lightweight blinded responses based on zero-knowledge proof and elliptic curve cryptography to prevent CRP leakage and achieve anonymity for external attackers, which also reduces query and storage overheads. The security analysis proves that our scheme is secure and meets the essential security requirements. The scheme can achieve anonymity and untraceability and can resist common attacks. Finally, experimental results show that this scheme has certain advantages over other related schemes, in particular, the device-side cost is significantly lower than that of the other schemes. This demonstrates the suitability of the proposed scheme in resource-limited devices.

IX. Acknowledgment

The work was supported in part by the National Natural Science Foundation of China under Grant 62272002, Grant U24A20243, Grant 62372002, and Grant 62202005, in part by the Natural Science Foundation of Anhui Province, China under Grant 2508085QF243, and in part by the China Postdoctoral Science Foundation under Grant 2025M771549. The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

References

[1] Y. Liao, E. de Freitas Rocha Loures, and F. Deschamps, "Industrial internet of things: A systematic literature review and insights," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4515–4525, 2018.

[2] J. Koo, G. Kang, and Y.-G. Kim, "Access control framework for cross-platform interoperability in the industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 1, pp. 801–810, 2025.

[3] F. Wang, J. Cui, Q. Zhang, D. He, and H. Zhong, "Blockchain-based secure cross-domain data sharing for edge-assisted industrial internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3892–3905, 2024.

[4] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer networks*, vol. 76, pp. 146–164, 2015.

[5] F. Wang, J. Cui, Q. Zhang, D. He, C. Gu, and H. Zhong, "Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial internet of things," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–18, 2023.

[6] Q. Zhang, Y. Fu, J. Cui, D. He, and H. Zhong, "Efficient fine-grained data sharing based on proxy re-encryption in iiot," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–13, 2024.

[7] Y. Chen and J. Chen, "An efficient mutual authentication and key agreement scheme without password for wireless sensor networks," *The Journal of Supercomputing*, vol. 77, no. 12, pp. 13653–13675, 2021.

[8] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1338–1351, 2022.

[9] W. Liu, H. Liu, Y. Wan, H. Kong, and H. Ning, "The yoking-proof-based authentication protocol for cloud-assisted wearable devices," *Personal and Ubiquitous Computing*, vol. 20, pp. 469–479, 2016.

[10] Q. Zhang, J. Wu, H. Zhong, D. He, and J. Cui, "Efficient anonymous authentication based on physically unclonable function in industrial internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 233–247, 2023.

[11] W. Che, M. Martin, G. Pocklassery, V. K. Kajuluri, F. Saqib, and J. Plusquellic, "A privacy-preserving, mutual puf-based authentication protocol," *Cryptography*, vol. 1, no. 1, p. 3, 2016.

[12] Y. Zheng, W. Liu, C. Gu, and C.-H. Chang, "Puf-based mutual authentication and key exchange protocol for peer-to-peer iot applications," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, pp. 3299–3316, 2023.

[13] C.-H. Chang, Y. Zheng, and L. Zhang, "A retrospective and a look forward: Fifteen years of physical unclonable function advancement," *IEEE Circuits and Systems Magazine*, vol. 17, no. 3, pp. 32–62, 2017.

[14] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[15] M. A. Qureshi and A. Munir, "Puf-rake: A puf-based robust and lightweight authentication and key establishment protocol," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2457–2475, 2022.

[16] A. Rullo, C. Felicetti, M. Vatalaro, R. De Rose, M. Lanuzza, F. Crupi, and D. Saccà, "Puf-based authentication-oriented architecture for identification tags," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 1, pp. 66–83, 2025.

[17] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. M. Prabhu, "Building puf based authentication and key exchange protocol for iot without explicit crps in verifier database," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 424–437, 2019.

[18] M.-D. Yu, M. Hiller, J. Delvaux, R. Sowell, S. Devadas, and I. Verbauwhede, "A lockdown technique to prevent machine learning on pufs for lightweight authentication," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 3, pp. 146–159, 2016.

[19] O. Millwood, F. Hongming, P. Gope, O. Narlı, M. K. Pehlivanoğlu, E. B. Kavun, and B. Sikdar, "A privacy-preserving protocol level approach to prevent machine learning modelling attacks on pufs in the presence of semi-honest verifiers," in *2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 326–336, 2023.

[20] W. Wang, Q. Chen, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, and C. Su, "Blockchain and puf-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8883–8891, 2022.

[21] A. A. Alamr, F. Kausar, J. Kim, and C. Seo, "A secure ecc-based rfid mutual authentication protocol for internet of things," *The Journal of Supercomputing*, no. 9, 2018.

[22] S. Chen, B. Li, Z. Chen, Y. Zhang, C. Wang, and C. Tao, "Novel strong-puf-based authentication protocols leveraging shamir's secret sharing," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14408–14425, 2022.

[23] M. Alkanhal, A. Alali, and M. Younis, "Puf-based authentication protocol with physical layer-based obfuscated challenge-response pair," in *ICC 2023 - IEEE International Conference on Communications*, pp. 5867–5872, 2023.

[24] G. S. Poh, P. Gope, and J. Ning, "Privhome: Privacy-preserving authenticated communication in smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1095–1107, 2021.

[25] N. Xi, W. Li, L. Jing, and J. Ma, "Zama: A zkp-based anonymous mutual authentication scheme for the iov," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22903–22913, 2022.

[26] J. Ye, Y. Hu, and X. Li, "Opuf: Obfuscation logic based physical unclonable function," in *2015 IEEE 21st International On-Line Testing Symposium (IOLTS)*, pp. 156–161, 2015.

[27] S. T. C. Konigsmark, D. Chen, and M. D. F. Wong, "Polypuf: Physically secure self-divergence," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 7, pp. 1053–1066, 2016.

[28] J. Ye, Y. Hu, and X. Li, "Rpuf: Physical unclonable function with randomized challenge to resist modeling attack," in *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*, pp. 1–6, 2016.

[29] C. Gu, C.-H. Chang, W. Liu, S. Yu, Y. Wang, and M. O'Neill, "A modeling attack resistant deception technique for securing lightweight-puf-based authentication," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 6, pp. 1183–1196, 2021.

[30] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, "Slender puf protocol: A lightweight, robust, and secure authentication by substring matching," in *2012 IEEE Symposium on Security and Privacy Workshops*, pp. 33–44, 2012.

[31] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, "Lightweight iot-based authentication scheme in cloud computing circumstance," *Future generation computer systems*, vol. 91, pp. 244–251, 2019.

[32] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K.-K. R. Choo, "A robust and energy efficient authentication protocol for industrial internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1606–1615, 2018.

[33] T.-F. Lee, K.-W. Lin, Y.-P. Hsieh, and K.-C. Lee, "Lightweight cloud computing-based rfid authentication protocols using puf for e-healthcare systems," *IEEE Sensors Journal*, vol. 23, no. 6, pp. 6338–6349, 2023.

[34] S. Li, T. Zhang, B. Yu, and K. He, "A provably secure and practical puf-based end-to-end mutual authentication and key exchange protocol for iot," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 5487–5501, 2021.

[35] A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. G. Tauber, C. Schmittner, and J. Bastos, "A lightweight authentication mechanism for m2m communications in industrial iot environment," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 288–296, 2019.

[36] S. F. Aghili and H. Mala, "Breaking a lightweight m2m authentication protocol for communications in iiot environment," *Cryptology ePrint Archive*, 2018.

[37] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," *Towards Hardware-Intrinsic Security: Foundations and Practice*, pp. 3–37, 2010.

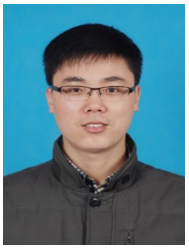
[38] A. J. Menezes and S. A. Vanstone, "Elliptic curve cryptosystems and their implementation," *Journal of cryptology*, vol. 6, pp. 209–224, 1993.

[39] Y.-H. Chuang and C.-L. Lei, "Puf based authenticated key exchange protocol for iot without verifiers and explicit crps," *IEEE Access*, vol. 9, pp. 112733–112743, 2021.

[40] P. Gope, O. Millwood, and B. Sikdar, "A scalable protocol level approach to prevent machine learning attacks on physically unclonable function

based authentication mechanisms for internet of medical things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1971–1980, 2022.
[41] "Miracl core." https://github.com/miracl/core.

Fengqun Wang was born in Anhui Province, China, in 1996. He received his Ph.D. degree in computer science from Anhui University in 2024. He is cur rently a lecture of School of Computer Science and Technology at Anhui University. His research in terests include IoT security, blockchain and applied cryptography. He has multiple scientific publications in reputable journals (e.g. IEEE Transactions on Information Forensics and Security, IEEE Transac tions on Dependable and Secure Computing, IEEE Transactions on Network and Service Management, IEEE Transactions on Industrial Electronics).



Jie Cui (Senior Member, IEEE) was born in Henan Province, China, in 1980. He received his Ph.D.degree in University of Science and Technol ogy of China in 2012. He is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). He has over 150 scientific publications in reputable journals (e.g. IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Secu rity,IEEE Journal on Selected Areas in Communications, IEEE Transactions on Mobile Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers), academic books and international confer ences.

Wuquan Wen is a Ph.D. candidate at the School of Computer Science and Technology, Anhui Uni versity, Hefei, China. His research interests include Unmanned Aerial Vehicle (UAV) security and ap plied cryptography.

Ke Hu is now a research student in the School of Computer Science and Technology, Anhui Uni versity. Her research focuses on the security of the Industrial Internet of Things (IIoT).