Efficient Blockchain-Based Mutual Authentication and Session Key Agreement for Cross-Domain IIoT

Jie Cui, *Senior Member, IEEE*, Yihu Zhu, Hong Zhong, *Member, IEEE*, Qingyang Zhang, *Member, IEEE*, Chengjie Gu, and Debiao He, *Member, IEEE*

***Abstract*—Several studies have introduced edge computing and blockchain into the Industrial Internet of Things (IIoT) to satisfy the requirements of delay-sensitive applications and support cross-domain authentication. Although there have been many protocols to ensure the security and privacy of devices in the IIoT, existing protocols still suffer from problems. Updating keys and pseudonyms of devices by a trusted third party (e.g., certificate authority) will cause high communication and computation overhead, especially when the number of devices becomes much larger. Furthermore, an increasing number of transactions also cause high-storage overhead on the blockchain. Therefore, we propose a blockchain-based cross-domain authentication protocol. Specifically, we propose a privacy-preserving method based on pseudonyms that offloads the task of generating pseudonyms from a trusted third party to edge servers to ensure the conditional anonymity of the devices. The device is allowed to request pseudonyms in bulk to reduce the number of transactions, thus reducing the storage overhead on the blockchain. Security analysis and experimental results demonstrate that our scheme achieves an efficient tradeoff between security and efficiency.**

***Index Terms*—Blockchain, cross-domain authentication, edge computing, Industrial Internet of Things (IIoT), mutual authentication.**

I. Introduction

THE Internet of Things (IoT) [1], [2] connects people to things and things to things anytime and anywhere through the Internet connected devices embedded with sensors and actuators [3]. Industrial Internet of Things (IIoT) is applying IoT in the industry [4]. The primary objective of the IIoT is to improve product quality and efficiency and reduce production costs by collecting and analyzing a large amount of data from industrial sectors (e.g., factories). Using IIoT, conventional industries will eventually become intelligent.

In IIoT systems, IIoT devices are resource-constrained, and a massive amount of data gathered by the devices is transmitted to the cloud server for storage and computation [5]. However, with the development of IIoT, the system scale is becoming much larger, and conventional cloud-based IIoT platforms cannot satisfy the demands for low-latency and mission-critical tasks. For example, in a smart factory, the data collected by devices can sometimes reach the level of GB per second. If all the data are transmitted to the cloud for processing and analysis, too many bandwidth resources will be consumed, which can cause significant network congestion. This, in turn, can result in latency problems that affect the Quality of Service (QoS) for various production tasks. Thus, the introduction of edge computing to support IIoT has attracted the attention from both industries and academics. As edge computing is near to end users, it can provide real-time services and help devices perform computation-intensive tasks. Although edge computing offers several advantages, several challenges remain unresolved. Edge computing is more vulnerable than cloud computing because of its inherent characteristics, such as mobility and geolocation [6]. A fake edge node can also impersonate a legitimate edge server (ES) and connect to an IIoT device to access its data, threatening its security and privacy. Therefore, designing a practical security solution for edge-computing-based architectures is vital. Mutual authentication is an effective measure to protect communicators' privacy and security before sending sensitive information or requesting services. Several authentication schemes [7], [8], [9] authenticate devices and ESs. However, most of them rely on public key infrastructure (PKI) systems. A trusted authority (TA) called certificate authority (CA) is responsible for managing certificates or updating pseudonyms, which can overburden the authority when the system scales are vast. In addition, the smart device (SD) and server send requests to the CA for identity information (e.g., public keys) before authenticating each other. This may cause an extra delay if there is already a burden on the CA to respond

Manuscript received 18 August 2023; revised 17 October 2023 and 4 December 2023; accepted 5 January 2024. Date of publication 9 January 2024; date of current version 25 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62272002, Grant 62202005, and Grant 62325209; in part by the Excellent Youth Foundation of Anhui Scientific Committee under Grant 2108085J31; in part by the Natural Science Foundation of Anhui Province, China, under Grant 2208085QF198; and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2022-049. *(Corresponding author: Hong Zhong.)*

Jie Cui, Yihu Zhu, Hong Zhong, and Qingyang Zhang are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, and the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn).

Chengjie Gu is with the School of Public Security and Emergency Management, Anhui University of Science and Technology, Hefei 231131, China, and also with the Security Research Institute, New H3C Group, Hefei 230088, China (e-mail: gcj@ustc.edu.cn).

Debiao He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai 201204, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/JIOT.2024.3351892

to the authentication requests of many devices, which could become a bottleneck [10]. Furthermore, most schemes do not consider device mobility. For example, drones that detect potential hazards in power systems can move to different areas. Therefore, it is necessary to design an efficient cross-domain authentication protocol to obtain timely information and provide seamless services when SDs move to another edge network.

The solutions to address cross-domain authentication can be divided into two categories, centralized solutions and the blockchain-based solutions. Centralized schemes rely heavily on trust third party (e.g., CA). Therefore, to achieve cross-domain authentication, it is necessary to verify the authenticity of devices by issuing certificates. However, as the system scales expands, certificate management becomes increasingly complex, potentially burdening CA and leading to efficiency issues. Blockchain acts as a distributed ledger with features, such as transparency and tamper-proofing. Each blockchain node needs to be verified before joining the network and maintains an exact and complete copy of the ledger. The destruction of a single node does not affect the integrity of the entire ledger, significantly improving the data's security and solving the bottleneck problem. These nodes do not trust each other, but they collaborate through a consensus mechanism. Consequently, an increasing number of works have tended to employ blockchain to assist cross-domain authentication [11], [12], [13]. However, several challenges remain to be resolved.

Although these blockchain-based schemes can achieve cross-domain authentication and guarantee the privacy and anonymity of devices, some unresolved problems remain. When the temporary keys and pseudonyms of the devices expire, they need to request new keys and pseudonyms from the CA. According to the International Data Corporation [14], connected IoT devices are expected to reach 41.6 billion by 2025. Therefore, unpredictable pseudonym requests from many devices impose huge computation and communication burdens on the CA. In addition, devices' public keys are recorded on the blockchain in these schemes, which causes enormous storage overhead. Yang et al. [15] designed a two-phase pseudonym service that permits CA to offload the pseudonym distribution task to roadside unit (RSU) proxies to address these issues. This inspired us to let ESs generate pseudonyms instead of the CA. To guarantee the privacy of the device, such as the real identity, we cannot simply adopt the method of generating pseudonyms by the CA because ESs are always deemed semi-trusted entities. Several requirements must be satisfied: 1) the device's privacy must be protected and 2) only one TA can track the real identity of a device.

To address these problems, we propose a blockchain-based cross-domain authentication protocol. Specifically, we build distributed trust among ESs through a blockchain to share cross-domain information. Therefore, we can achieve mutual cross-domain authentication without using a CA. In addition, we design a pseudonym-based privacy-preserving scheme. That is, CA offloads the task of distributing pseudonyms to the ESs. In this way, the responsibility of generating pseudonyms is taken over by ESs. And partial information about pseudonyms is put on the blockchain for sharing, which allows CA to revoke the devices when malicious behavior is detected. This reduces the burden on the CA. Furthermore, not complete pseudonyms are uploaded to the blockchain. Hence, the number of transactions in the blockchain is reduced, and the efficiency of updating keys is increased. The major contributions of this article can be summarized as follows.

1) We design a mutual authentication scheme in IIoT using edge computing and blockchain, which can realize the authentication between SDs and different ESs to provide real-time services. At the same time, this protocol achieves some properties, such as anonymity and traceability.
2) To reduce the dependence on the CA, we design a pseudonym-based method. Specifically, we utilize the ESs to generate and distribute pseudonyms for the SDs without the involvement of the CA. At the same time, we reduce the storage overhead on the blockchain.
3) We conduct an informal security analysis and use an automatic cryptographic protocol verification tool called ProVerif to prove that our scheme is secure. And the performance analysis demonstrates that our scheme gets a tradeoff between security and efficiency.

The remainder of this article is structured as follows. Some related works are presented in Section II. Section III provides an introduction to relevant preliminary concepts. The system model, security goals and attack model are presented in Section IV. The details of our proposed scheme are described in Section V. And Section VI shows the security analysis. Section VII presents the performance evaluation. Finally, Section VIII summarizes this article.

II. Related Work

There are various solutions to achieve mutual authentication between a device and server, which can be broadly categorized into two types. The first type is conventional schemes that do not involve blockchain technology [16], [17], [18], [19], [20]. These schemes usually employ techniques, such as elliptic curve cryptography (ECC), edge computing, fog computing, and asymmetric encryption, to protect system security and privacy. The second type is blockchain-based schemes [21], [22], [23]. These schemes leverage the properties of blockchain technology, such as transparency and tamper-proofing, to facilitate key management and data sharing. Moreover, as blockchain is decentralized in nature, it can establish trust among different domains, enabling cross-domain authentication.

First, we discuss the conventional schemes. Rostampour et al. [24] presented a secure authentication scheme for IoT edge devices. In this scheme, authors only used ECC to implement the authentication process and they examined their proposed scheme's security via BAN logic and Scyther tool. However, this protocol cannot reveal the unique identity of an embedded device when malicious behavior is detected. Rangwani and Om [25] proposed an ECC-based secure user authentication scheme for cloud computing. This scheme addresses some problems, such as denial-of-service

and stolen smart card, which are found in [26]. However, it requires the cloud server to negotiate the symmetric key with the SD and user in advance, without considering if the key is updated. Alam and Kumar [27] presented an efficient authentication scheme for IoT devices, which is based on the elliptic curve discrete logarithm problem (ECDLP). The authors used hash function and XOR function to reduce computation cost. However, the authentication process of cloud servers and users requires the involvement of TA, which can be burdensome as the number of authentication requirements increases. With the increasing demand for low latency and QoS, the research focus has shifted from cloud computing to edge computing.

Amore et al. [28] presented a privacy-preserving authentication scheme for fog computing. However, the mutual authentication between a fog user and fog server relies on a list called SV, which is maintained by the registration authority (RA). It should be noted that the cost of communication involved in updating the list can be significantly high, and it cannot resist stolen verifier attacks. Later, a lightweight and privacy-preserving authentication scheme for edge computing was proposed in [29]. However, we found that RC's secret key can be calculated by the user and server during the registration phase. During the registration phase, the message $\{SID_u, d_u, r_u\}$ is sent to the user and the user can compute $d_{RC} = (SID_u - r_u)h_u^{-1}$, where h_u, d_{RC} denote the hash value and RC's private key, respectively. Jia et al. [30] proposed an efficient identity-based anonymous authentication scheme for edge computing, but the protocol does not achieve key management (i.e., key update and key revoke) and lacks conditional anonymity. Li et al. [31] proposed an anonymous authentication scheme for edge computing and the user anonymity is fully protected. However, registration center gets involved in the authentication process. Furthermore, the identity-based schemes suffer from the problem of leaking the privacy of devices, such as the real identity.

To provide conditional anonymity, group signature has been introduced in many schemes. Chaum and Heyst [32] invented the notion of group signature in 1991. It means that any message signed by a group member can be verified using the group public key, without exposing the real identity of the signer. Huang et al. [33] proposed an efficient certificateless group signature scheme for mobile edge computing. This scheme can achieve some features, such as anonymity, unforgeability, and traceability. However, group signature is known to have high computation and communication overhead. Additionally, it heavily relies on the group manager to achieve the key update and revocation.

Later, blockchain is introduced to share public information and achieve flexible key management. Wang et al. [6] proposed a blockchain-based anonymous authentication scheme for smart grid edge computing, where smart contract is utilized to record public keys to achieve flexible key management. However, the private keys of devices need to be updated by RA regularly. Shen et al. [34] proposed a cross-domain authentication method for IIoT based on blockchain. In this scheme, blockchain is used to establish trust among different domains, and the specific domain information is stored off-blockchain to eliminate the throughput bottleneck of blockchain. Lin et al. [35] presented a blockchain-based conditional privacy-preserving authentication scheme for vehicular networks. In the designed scheme, the authors employed a key derivation algorithm to get vehicles' private keys and the CA issues a certificate of corresponding public keys. Panda et al. [36] presented a blockchain-based authentication and key management protocol in distributed IoT using one-way hash chains. However, the protocol does not achieve key update and conditional anonymity. Zhang et al. [37] presented a blockchain-based reliable and privacy-preserving authentication protocol for fog-based IoT devices. In this protocol, the fog node (i.e., IoT device) sends multiple query requests to the full blockchain node which consists of some fake queries and one correct query. Thereby, the blockchain nodes are obfuscated from knowing which is the real request thus protecting the privacy of the user. Yang et al. [38] proposed a blockchain-based secure and lightweight authentication scheme for IoT. In this scheme, a modular square root algorithm is used to generate signatures effectively. Wang et al. [39] proposed a cross-domain authentication scheme for IoT that utilizes blockchain and dynamic accumulator. The proposed protocol transfers the authentication problem into a signature transitivity problem by utilizing an accumulator and a standard digital signature scheme. Wang et al. [12] proposed a blockchain-based authentication model of intelligent telehealth systems with multiserver edge computing architecture. The proposed scheme enables an authenticated server to assist a user in authenticating another server to reduce interactions.

In general, existing blockchain-based authentication schemes still suffer from issues, such as low efficiency in updating devices' keys and pseudonyms, as well as high-storage overhead on the blockchain. Thus, designing a blockchain-based cross-domain authentication for edge-computing-assisted IIoT remains challenging.

III. Preliminaries

In this section, we introduce some relevant background knowledge that will be used in our scheme, including computational hardness assumptions and blockchain. Table I provides an overview of all the notations used in this article.

A. Computational Hardness Assumption

The security of our scheme is based on the following computational hardness assumptions, i.e., elliptic curve discrete logarithm (ECDL) and elliptic curve Diffie–Hellman (ECDH).

Let $\mathbb{Z}_q^*$ be a finite field, which is determined by the prime number q. Let $\mathbb{G}$ be a cyclic additive group consisted of the points on the elliptic curve and the point at infinity. Let P be a generator of $\mathbb{G}$.

1) *ECDL:* For $a \in \mathbb{Z}_q^*$, given $P, aP \in \mathbb{G}$, it is hard to compute a.
2) *ECDH:* For $a, b \in \mathbb{Z}_q^*$, given $P, aP, bP \in \mathbb{G}$, it is hard to compute abP.

TABLE I
NOTATIONS

Notations	Descriptions
s	The secret key of RA
P_{pub}	The public key of RA
ES_i	The i-th Edge server
SD_j	The j-th Smart device
ID_i	The real identity of ES_i
ID_j	The real identity of SD_j
x_i,	The secret key of ES_i
PK_i	The public key of ES_i
ω	The long term private key of SD_j
Ω	The token of SD_j
$T_i(i = 1, ..., 4)$	Timestamp
TK	Symmetric key
$E_{TK}()$	Symmetric encryption with TK
$D_{TK}()$	Symmetric decryption with TK
$h_i(i = 1, ..., 6)$	Hash functions

B. Blockchain and Smart Contract

The blockchain (BC) is a decentralized database maintained by peer-to-peer nodes. It uses cryptographic primitives (e.g., signature algorithm and one-way hash function) and consensus mechanisms to ensure the integrity and consistency of data. The full node has the entire ledger and is able to verify whether the transaction is valid alone and upload data to the blockchain network. The consortium blockchain is a type of blockchain that consists of multiple parties that have reached an agreement. The members of consortium blockchain do not trust each other, but cooperate with each other to accomplish tasks under the consensus mechanism. Therefore, we can utilize consortium blockchain to assist the authentication in the cross-domain environment.

The smart contract is a self-executing computer program that automatically executes once conditions are met. Smart contracts are deployed into the blockchain network in advance and the content of the contract is open and transparent. Authorized nodes can trigger smart contracts to manage data by publishing transactions and query the data recorded on the blockchain by submitting the parameters. In our scheme, the blockchain is mainly used to establish trust among different servers for sharing cross-domain information. The data on the blockchain are immutable.

IV. Models and Security Goals

In this section, we first introduce the system model for the proposed scheme and then demonstrate the security goals and attack model.

A. System Model

As illustrated in Fig. 1, our system consists of RA, ES, SD, and blockchain (BC).

1) *RA:* The RA is a trust manager that generates system parameters and distributes key materials to the corresponding entities. Besides, it can utilize the blockchain to record the public keys and related parameters of the registered ESs and the identities of the revoked SDs.
2) *ES:* The ES is equipped with sufficient computation and storage resources, and is responsible for providing data analysis and services. Furthermore, it honestly follows the rules for generating pseudonyms for SDs. At the same time, each ES can query public keys through the blockchain and some high-reputation ESs, as blockchain nodes, can upload information about pseudonyms to the blockchain.
3) *SD:* The SD can be a terminal device (e.g., a drone) with limited computation and storage resources. It needs to connect with ESs to acquire specific services after the successful mutual authentication with ESs.
4) *BC:* The BC is a distributed ledger maintained by RA and ESs. It is responsible for recording the public keys and revoked devices. The information recorded on the BC is reliable.

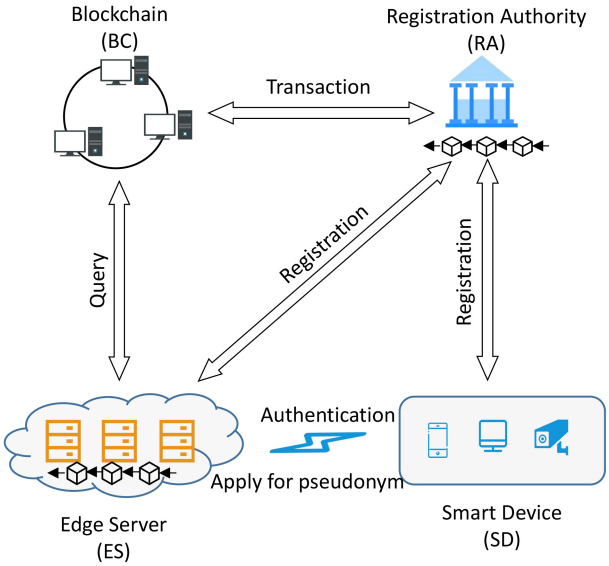


Fig. 1. System model.

B. Security Goals

The scheme is supposed to satisfy the following security goals.

1) *Mutual Authentication:* To protect the system security and avoid leaking the privacy of the devices, communicators need to validate each other's identity before sending sensitive message.
2) *Conditional Anonymity:* To protect the privacy of the devices, no other entity can know the real identity of the device except for the RA, which can reveal its identity.
3) *Session Key Agreement:* For further exchange of secret messages, a session key that can only be shared between the communicators should be produced during the mutual authentication process.
4) *Unlinkability:* No one except the RA can determine whether two different signatures are from the same device.
5) *No Online RA:* To minimize the communication overhead, it allows communicators to achieve mutual authentication and update the keys of devices without involving the RA.

6) *Forward Security:* To protect the previous communications from being leaked, it is necessary to ensure that even if an attacker obtains the secret key, they cannot obtain the session key used in previous communications.
7) *Resistance to Common Attacks:* To guarantee the security of the whole network, the scheme should withstand various typical attacks, including replay attacks, stolen verifier attacks, and impersonation attacks.

C. Attack Model

There are two types of attackers: 1) internal attackers and 2) external attackers. Internal attackers are those actively engaged in the scheme, such as ESs and SDs. External attackers are not involved in the scheme and do not have access to critical materials. Some typical attacks are as follows.

1) *Internal Attack:*
   a) *Semi-Trusted ESs:* In the proposed protocol, we suppose ESs as semi-trusted entities. They will execute the protocol honestly but will be curious about the identity of the requester and the information transmitted via the public channel. Furthermore, in cross-domain authentication, the ESs in different domains may not always trust each other. The absence of complete trust between interdomain ESs can lead to privacy leaks during cross-domain authentication processes for IIoT devices. The ES of the accessed domain might identify the true identity of the accessing device, capturing access activity records for thorough analysis and thereby compromising the device's privacy.
   b) *Compromised SDs:* SDs are resource-constrained and can be compromised. Once compromised, the SDs may leak confidential data or inject false data, disrupting subsequent data analysis. In cross-domain environments, a task may involve collaboration among multiple domains. If devices are compromised and subsequently move to different domains, they could transmit misleading information, potentially leading to erroneous data analysis and mission failure.
2) *External Attack:*
   a) *Replay Attack:* An attacker sends a message that the server has already accepted, aiming to deceive the server and successfully pass the authentication process. In cross-domain authentication, the attacker might intercept a message sent by a device in domain A and forward the message to domain B to achieve successful authentication.
   b) *Impersonation Attack:* By impersonating a legitimate entity (i.e., SD), an attacker can interact with the server and successfully pass the authentication process. Similarly, in cross-domain environments, an attacker might impersonate ESs across multiple domains, interacting with devices to pass the authentication and obtain relevant information about tasks.
   c) *Eavesdroping Attack:* An attacker can eavesdrop on the information exchanged between the device and the server through the public channel, thereby gaining access to private information. Additionally, when a task involves multiple domains, an attacker in a cross-domain environment can infer sensitive details (e.g., production quantity) by eavesdropping on communications across these domains.

V. Proposed Scheme

In this section, we describe the specifics of the proposed authentication protocol for SDs and ESs.

A. Overview

Our protocol consists of five phases: 1) setup; 2) registration; 3) pseudonym application; 4) authentication; and 5) revocation. We provide a high-level explanation of the blockchain-based protocol.

The RA sets up the system and generates the private key and public key. Then it publishes the system parameters. The new edge sever and device need to register with RA before interacting with each other. The edge sever can obtain the pseudonym materials during the registration process and then SD can apply for new keys and pseudonyms from ES. After the mutual authentication with ES, the device can acquire the service from the server. The RA can revoke a device and record it on the blockchain.

B. System Setup

The RA performs the system setup phase at the start of the system deployment, as explained in the following.

1) RA selects a cyclic additive group $\mathbb{G}$ with a generator P and a prime order q on an elliptic curve $E(\mathbb{F}_q)$ over the finite field $\mathbb{F}_q$.
2) RA randomly selects a number $s \in \mathbb{Z}_q^*$ and set it as its private key. Then RA computes the corresponding public key $P_{\text{pub}} = sP$.
3) *RA Selects Hash Functions:* $h_1 : \mathbb{G} \to \mathbb{Z}_q^*, h_2 : \mathbb{Z}_q^* \to \mathbb{Z}_q^*, h_3 : \{0,1\}^* \to \mathbb{Z}_q^*, h_4 : \{0,1\}^* \times \mathbb{G} \to \mathbb{Z}_q^*, h_5 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*, h_6 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_q^*$.
4) RA keeps s secretly and publishes the public parameters $\{\mathbb{G}, q, P, P_{\text{pub}}, h_i (1 \le i \le 6)\}$.

Then, RA initializes the blockchain according to the configuration file, and the successfully registered ESs can join the blockchain network to facilitate collaboration through the consensus mechanism. Smart contracts are deployed in the blockchain to manage cross-domain information. The related smart contract algorithms are shown in Algorithms 1–3. Algorithm 1 updates the information of ESs, adds newly registered servers to the smart contract, and marks ESs as invalid if they exhibit malicious behavior. Algorithm 2 records information for SDs applying for pseudonyms and, if a device displays malicious behavior, it is

Algorithm 1 Update ESMT

Input: parameters{*ID*, *PK*, *R*, *status*}
Output: bool

```
struct ESM {
    byte32 ID
    uint256[4] PK
    uint256[4] R
    string status }
struct ESM[] ESMT
if msg.sender ≠ RA then
    return false
else
    i = 1
    for ; i ≤ ESMT.size(); i++ do
        if ESMT[i].ID == ID then
            function Update_ESM(ID, PK, R, status)
            return true
        end if
    end for
    if i > ESMT.size()  then
        function Add_New_ESM(ID, PK, R, status)
        return true
    end if
end if
```

Algorithm 2 Update SDMT

Input: parameters{Ω, *t*, *z*, *status*}
Output: bool

```
struct SDM {
    byte32 Ω
    DateTime t
    uint256[2] z
    string status}
struct SDM[] SDMT
if status == invaild && msg.sender ≠ RA  then
    return false;
else if msg.sender ≠ server then
    return false;
else
    i = 1
    for ;i ≤ SDMT.size();i++ do
        if SDMT[i].Ω == Ω  then
            function Update_SDM(Ω, t, z, status)
            return true
        end if
    end for
    if i > SDMT.size() then
        function Add_New_SDM(Ω, t, z, status)
        return true
    end if
end if
```

marked as invalid to prevent further pseudonym applications. Algorithm 3 is utilized to query the public key of registered ESs.

Algorithm 3 Query ESMT

Input: parameters{*ID*}
Output: parameters{*ID*, *PK*, *R*, *status*} or false

```
i = 1
for ;i ≤ SDMT.size();i++ do
    if ESMT[i].ID == ID then
        return{ID, PK, R, status}
    end if
end for
return false
```

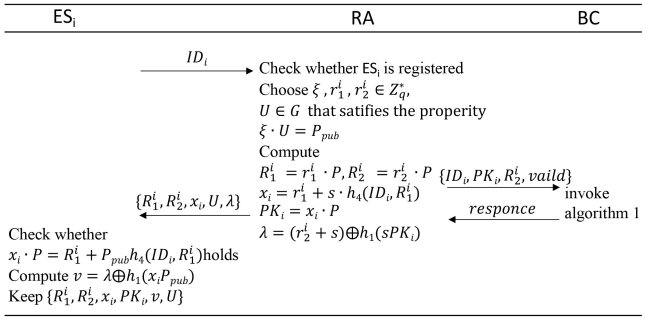


Fig. 2. ES registration phase.

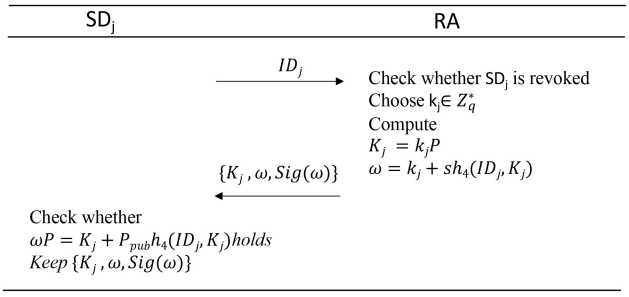


Fig. 3. SD registration phase.

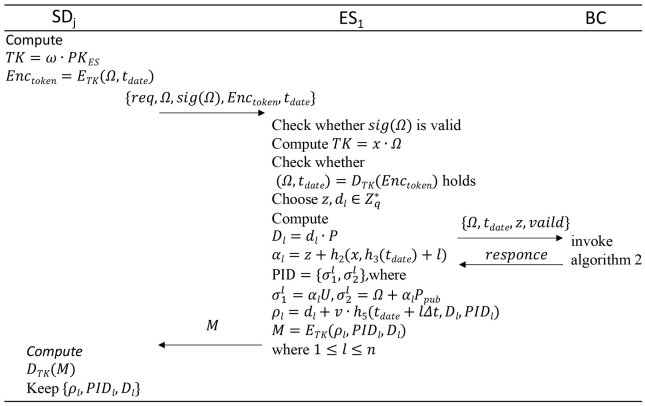


Fig. 4. Pseudonym application phase.

C. Registration

In this phase, each ES and SD is required to register with RA to get its secret key and corresponding public key, respectively, where the registration process should be performed in a secure and private channel. Figs. 2 and 3 show the main steps of the ES registration phase and SD registration phase, respectively.

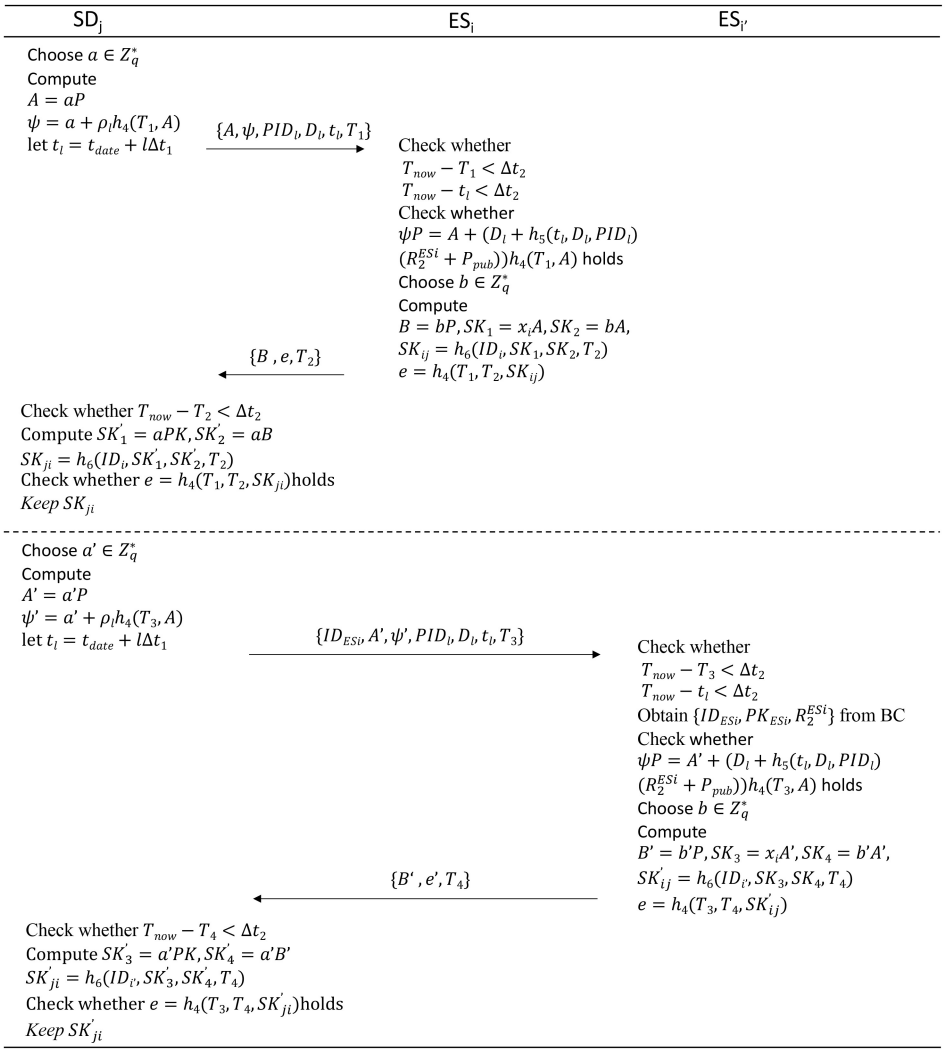


Fig. 5. Authentication phase.

1) Edge Server Registration:

1) Upon receiving the real identity ID_i from ES_i, RA first checks whether ID_i has been previously registered. If it has, RA will refuse this request. If not, RA will execute the following steps.
2) RA selects several random numbers $\xi_i, r_1^i, r_2^i \in \mathbb{Z}_q^*$. In addition, there is an element $U \in \mathbb{G}$ that satisfies the following property: $\xi_i U = P_{\text{pub}}$. Then RA calculates $R_1^i = r_1^i P, R_2^i = r_2^i P, x_i = r_1^i + sh_4(ID_i, R_1^i), PK_i = x_i P$ and $\lambda = (r_2^i + s) \oplus h_1(sPK_i)$, where λ is the parameter for ES_i to generate pseudonyms for SDs, x_i is the private key belonging to ES_i and PK_i is the corresponding public key. RA sends $m_1 = \{R_1^i, R_2^i, x_i, U, \lambda\}$ to ES_i in a secure channel. RA stores $\{ID_i, \xi_i\}$ in its local database, and uploads $\{ID_i, PK_i, R_2^i, valid\}$ to the blockchain by invoking the smart contract 1, i.e., Update ES materials table (ESMT).
3) Upon receiving the message m_1, ES_i checks whether $x_i P = R_1^i + P_{\text{pub}} h_4(ID_i, R_1^i)$ is satisfied, if so, ES_i calculates $PK_i = x_i P, v = \lambda \oplus h_1(x_i P_{\text{pub}})$, and stores $\{R_1^i, R_2^i, x_i, PK_i, v, U\}$ in its local database.

2) Smart Device Registration:

1) SD SD_j sends its identity ID_j to RA. First, RA checks whether the SD is revoked, if not, executes the following steps.
2) RA chooses a random number $k_j \in \mathbb{Z}_q^*$, and calculates $K_j = k_j P, \omega = k_j + sh_4(ID_j, K_j)$ and $\Omega = \omega P$, where Ω is deemed as a token belonging to SD_j. Finally, RA sends $m_2 = \{K_j, \omega, sig(\Omega)\}$ to SD_j in a secure channel, where $sig(\Omega)$ is a signature signed on Ω with the private key s of RA.
3) Upon receiving the message m_2, SD_j computes $\Omega = \omega P$ and checks whether the condition $\Omega = K_j + P_{\text{pub}} h_4(ID_j, K_j)$ is satisfied. If so, SD_j stores $\{K_j, \omega, \Omega, sig(\Omega)\}$ in its local database.

D. Pseudonym Application

To avoid the single point of failure and relieve the burden of RA, we offload the task of generating pseudonyms to ES. Fig. 4 shows the main steps of the pseudonym application phase.

1) First, SD_j sends the request and its token to ES_i to validate the authenticity. The details are as following.

a) SD_j obtains ID_{ESi} and the corresponding public key PK_{ESi} of the ES (denoted by ES_i) which is close to SD_j by querying the blockchain.
b) Then SD_j calculates $TK = \omega PK_{ESi}, CT_{\text{token}} = E_{TK}(\Omega, t_{\text{date}})$, where E_{TK} is symmetric encryption with key TK and t_{date} is the time when SD_j applies for pseudonyms.
c) SD_j sends $m_3 = \{\Omega, \text{sig}(\Omega), CT_{\text{token}}, t_{\text{date}}\}$ to ES_i.

we need to note that the signature $\text{sig}(\Omega)$ is to prevent malicious users from sending useless messages to occupy the resources of the servers and the CT_{token} is to ensure that only the user who owns the token can apply for pseudonyms to avoid man-in-the-middle attacks.

2) After successfully validating the authenticity of the token, ES_i calculates the pseudonyms and uploads related information to the blockchain. The details are as following.
   a) Checks whether the signature is valid. Aborts if the check fails.
   b) ES_i calculates $TK = x_i\Omega$ and verifies if $(\Omega, t_{\text{date}}) = D_{TK}(CT_{\text{token}})$ holds, where D_{TK} is symmetric decryption with key TK. Aborts if the check fails.
   c) ES_i chooses several random numbers $z, d_l \in \mathbb{Z}_q^*$, and calculates $D_l = d_lP, \alpha_l = z + h_2(x_i, h_3(t_{\text{date}}) + l), \sigma_1^l = \alpha_l U, \sigma_2^l = \Omega + \alpha_l P_{\text{pub}}, PID_l = \{\sigma_1^l, \sigma_2^l\}, \rho_l = d_l + vh_5(t_{\text{date}} + l\Delta t_1, D_l, PID_l)$, where $1 \le l \le n$, n represents the number of pseudonyms and Δt_1 represents the period when the pseudonym is valid. Then ES_i will upload $\{\Omega, t_{\text{date}}, z, valid\}$ to the blockchain by invoking the smart contract 2, i.e., Update SD materials table (SDMT).
3) Finally, ES_i sends the ciphertext of the pseudonyms to SD_j. SD_j gets the pseudonyms by decrypting the ciphertext. The details are as following.
   a) ES_i computes $M = E_{TK}(\{\rho_l, PID_l, D_l\})$ $(1 \le l \le n)$, and sends M to SD_j. ES_i will delete all the related information about pseudonyms in its local database when it finishes the service. There are two reasons for that: a) we can protect the pseudonyms from being leaked when the server is compromised and b) we can save storage because the number of devices is large.
   b) Upon receiving the ciphertext M, SD_j obtains the materials $\{\rho_l, PID_l, D_l\}(1 \le l \le n)$ by computing $D_{TK}(M)$.

E. Authentication

A SD needs to authenticate with ESs in different domains as it may move to different edge networks to perform tasks. Therefore, there are two scenarios to consider. The first case is that the SD authenticates with the ES which is responsible for generating pseudonyms for it. The other one is that the SD authenticates with ESs where the device is located after it has moved. Fig. 5 shows the main steps of the authentication phase.

1) When the SD SD_j is not moving or the server where the device is located has generated pseudonyms for it, SD_j needs to carry out this phase to authenticate the server ES_i and to negotiate a session key for further communication. The details of this phase are as follows.
   a) SD_j selects a random number $a \in \mathbb{Z}_q^*$ and computes $A = aP, \psi = a + \rho_l h_4(T_1, A)$, where T_1 is the current timestamp.
   b) Let $t_l = t_{\text{date}} + l\Delta t_1$, SD_j sends the message $m_4 = \{A, \psi, PID_l, D_l, t_l, T_1\}$ to ES_i.
   c) Upon receiving the message m_4, ES_i first verifies if $T_{\text{now}} - T_1 < \Delta t_2$ and $T_{\text{now}} - t_l < \Delta t_2$ holds, where T_{now} is the current timestamp and Δt_2 represents the maximum tolerable interval. Then ES_i checks whether PID_l is revoked and checks whether $\psi P = A + (D_l + h_5(t_l, D_l, PID_l)(R_2^{ESi} + P_{\text{pub}}))h_4(T_1, A)$ is satisfied. Rejects if the check fails.
   d) ES_i selects a random number $b \in \mathbb{Z}_q^*$ and calculates $B = bP, SK_1 = x_iA, SK_2 = bA, SK_{ij} = h_6(ID_i, SK_1, SK_2, T_2), e = h_4(T_1, T_2, SK_{ij})$, where T_2 is the current timestamp. The value SK_{ij} is a session key and e is responsible for the authenticity of the session key. ES_i sends the message $m_5 = \{B, T_2, e\}$ to SD_j
   e) Upon receiving the message m_5, SD_j first checks whether $T_{\text{now}} - T_2 \le \Delta t_2$ is satisfied. Rejects if the check fails. Then SD_j calculates $SK_1' = aPK_i, SK_2' = aB, SK_{ji} = h_6(ID_i, SK_1', SK_2', T_2)$ and sets SK_{ji} as the session key if $e = h_4(T_1, T_2, SK_{ji})$ holds.
2) When the SD SD_j has moved, it needs to carry out this phase to authenticate ES $ES_{i'}$ and establish a session key for further communication. The details are as follows.
   a) SD_j chooses a random number $a' \in \mathbb{Z}_q^*$ and calculates $A' = a'P, \psi' = a' + \rho_l h_4(T_3, A')$, where T_3 is the current timestamp.
   b) Let $t_l = t_{\text{date}} + l\Delta t_1$, SD_j sends the message $m_6 = \{ID_{ESi}, A', \psi', PID_l, D_l, t_l, T_3\}$ to $ES_{i'}$.
   c) Upon receiving m_4, $ES_{i'}$ first verifies if $T_{\text{now}} - T_3 < \Delta t, T_{\text{now}} - t_l < \Delta t_2$ holds, where T_{now} is the current timestamp. Then $ES_{i'}$ checks whether PID_l is revoked and retrieves the corresponding tuple $(ID_{ESi}, PK_{ESi}, R_2^{ESi})$ from the blockchain according to ID_{ESi}.
   d) Checks whether $\psi'P' = A' + (D_l + h_5(t_l, D_l, PID_l)(R_2^{ESi} + P_{\text{pub}}))h_4(T_3, A')$ is satisfied. Rejects if the check fails.
   e) Performs the same process from step 4 to step 5 in the above phase.

F. Revocation

ES sends the corresponding pseudonym $\{ID_{ESi}, PID_l\}$ to RA when there is a malicious device called SD_j. RA retrieves the tuple $\{ID_{ESi}, \xi_{ESi}\}$ from its local database according ID_{ESi} and calculates $\Omega = \sigma_2^l - \xi_{ESi}\sigma_1^l$. And RA will record $\{\Omega, ID_{SD_j}\}$ on the blockchain to prevent SD_j from reregistering and applying for pseudonyms again. In addition, RA retrieves the tuple $\{\Omega, t_{\text{date}}, z\}$ and calculates the successive pseudonyms PID_l which are still valid if not revoked. To save storage

TABLE II
COMPARISON OF SECURITY PROPERTIES

Security Properties	[6]	[12]	[24]	Ours
Mutual Authentication	✓	✓	✓	✓
Conditional Anonymity	✓	✓	×	✓
Session Key Agreement	✓	✓	✓	✓
Unlinkability	×	×	×	✓
No Online RA	✓	✓	✓	✓
Efficient Update	×	×	×	✓
Stolen Verifier Attack	✓	✓	×	✓

✓: satisfying the security property
×: not satisfying the security property

overhead, we use cuckoo filter to record the revoked devices instead of a revocation list. If there is a data structure which has better performance in storing and querying, it can be replaced. Specifically, RA prepares a cuckoo filter whose size is dependent on the max number of revoked devices. Note that the data recorded on the filter is the fingerprint of σ_1^l. Because σ_1^l and PID_l correspond to each other and the space can be saved. When these pseudonyms expire, RA executes the delete operation to remove the corresponding fingerprints from the filter. Note that the filter is in the blockchain.

VI. SECURITY ANALYSIS

In this section, we first analyze the security of our proposed scheme by using the random oracle model. Then, we discuss how the proposed protocol meets the requirements presented in Section III, and the comparison with other relevant schemes [6], [12], and [24] are listed in Table II. Finally, we utilize ProVerif to prove the security of the scheme.

A. Formal Security Proof Using Random Oracle Model

We propose a security model for our proposed scheme. The model is defined by a game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the adversary can make the following queries.

1) *Setup:* The challenger $\mathcal{C}$ first generates system parameters and private key, and then sends public parameters to the adversary $\mathcal{A}$.
2) *Hash Oracle:* When the adversary $\mathcal{A}$ makes a query with information m, the challenger $\mathcal{C}$ selects a random number x and stores $\{m, x\}$ in the corresponding list(L_{h4}orL_{h5}). Then $\mathcal{C}$ sends x to $\mathcal{A}$.
3) *Secret Key Oracle:* When the adversary $\mathcal{A}$ invokes the query with PID, the challenger $\mathcal{C}$ can generate the secret key. Then, $\mathcal{C}$ sends the secret key to $\mathcal{A}$. Let Q_s denote all the secret key queries that $\mathcal{A}$. has made.
4) *Output:* The adversary $\mathcal{A}$ returns a signature $\{A, \psi, PID, D, t, T\}$. $\mathcal{A}$ wins and outputs 1 if verif$(A, \psi, PID, D, t, T) = 1$ and the secret key is not included in Q_s.

Theorem 1: If the adversary $\mathcal{A}$ can break the proposed scheme Γ, the challenger $\mathcal{C}$ can solve the ECDL problem.

Proof: If the adversary $\mathcal{A}$ can successfully forge a signature $\{A, \psi, PID, D, t, T\}$, then the challenger $\mathcal{C}$ is able to break the ECDL problem in polynomial time by utilizing $\mathcal{A}$ as a subroutine. We let $\{P, PK = skP | P \in \mathbb{G}, sk \in Z_q^*\}$ be an instance of ECDL problem, the purpose of $\mathcal{C}$ is to compute the value sk. $\mathcal{A}$ can make following queries supported by $\mathcal{C}$.

1) *Setup:* The challenger $\mathcal{C}$ first generates system parameters $\{\mathbb{G}, q, P, P_{\text{pub}}, h_4, h_5\}$, and then sends public parameters to the adversary $\mathcal{A}$.
2) *h_4 Oracle:* The challenger $\mathcal{C}$ maintains a list L_{h4} which is initialized to be empty. When the adversary $\mathcal{A}$ makes a query with information $<T, A>$, the challenger $\mathcal{C}$ first checks whether $<T, A, x_{h4}>$ exists in the list, returns x_{h4} if it exists. If not, $\mathcal{C}$ selects a random number x_{h4} and inserts $<T, A, x_{h4}>$ into L_{h4}. Then, $\mathcal{C}$ returns x_{h4} to $\mathcal{A}$.
3) *h_5 Oracle:* The challenger $\mathcal{C}$ maintains a list L_{h5} which is initialized to be empty. When the adversary $\mathcal{A}$ makes a query with information $<t, D, PID>$, the challenger $\mathcal{C}$ first checks whether $<t, D, PID, x_{h5}>$ exists in the list, returns x_{h5} if it exists. If not, $\mathcal{C}$ selects a random number x_{h5} and inserts $<t, D, PID, x_{h5}>$ into L_{h5}. Then, $\mathcal{C}$ returns x_{h5} to $\mathcal{A}$.
4) *Secret Key Oracle:* When the adversary $\mathcal{A}$ makes a query with information PID, the challenger $\mathcal{C}$ chooses random numbers ρ and d, where ρ is as the secret key corresponding to PID. Then, $\mathcal{C}$ computes $D = dP$ and queries $x_{h5} = h_5(t, D, PID)$ through the list L_{h5}. Finally, $\mathcal{C}$ sends $<PID, D, \rho>$ to $\mathcal{A}$.
5) *Output:* The adversary $\mathcal{A}$ chooses a random number a and computes $A = aP$. Then $\mathcal{A}$ make a h_4 query with message A and computes $\psi = a + \rho h_4(T, A)$. Finally, $\mathcal{A}$ generates a signature $\{A, \psi, PID, D, t, T\}$.

According to forking lemma [40], $\mathcal{A}$ is able to obtain another signature $\{A, \psi', PID, D, t, T\}$ by executing the above operations with a different result of x_{h5} but the same inputs $\{t, D, PID\}$. Therefore, it is easy to obtain the following equation:

$$\psi' = a + \left(d + v x'_{h5}\right) x_{h4}. \tag{1}$$

Then, the challenger $\mathcal{C}$ can get v by computing

$$\begin{aligned} &\frac{(\psi - \psi')}{(x_{h5} - x'_{h5}) x_{h4}} (\text{mod}\, q) \\ &= \frac{a + (d + v x_{h5}) x_{h4} - a - \left(d + v x'_{h5}\right) x_{h4}}{(x_{h5} - x'_{h5}) x_{h4}} \\ &= v. \end{aligned} \tag{2}$$

Therefore, the challenger $\mathcal{C}$ outputs the $(\psi - \psi')(x_{h5} - x'_{h5})^{-1} x_{h4}^{-1}$ as the solution of the ECDL problem. We let q_{h4}, q_{h5} and q_s denote the number of h_4 queries, h_5 queries and secret key queries, respectively. Let E_1 denote the event that there is no conflict between secret key query and hash query, E_2 denote the event that $\mathcal{A}$ can output a valid forgery. Therefore, $\mathcal{A}$ can return a valid forgery with the probability

$$\text{acc} = \Pr[E_1 E_2] = \Pr[E_1] \Pr[E_2 | E_1]. \tag{3}$$

Since $\mathcal{A}$ makes $q_{h4} + q_{h5} + q_s$ queries, the probability of the event that bad $\leftarrow$ true is $([q_{h4} + q_{h5} + q_s]/q)$ for one secret key query and the probability for q_s queries is $([q_s(q_{h4} + q_{h5} + q_s)]/q)$. Therefore, $\Pr[E_1] = 1 - \Pr[\text{bad} = \text{true}] = 1 - ([q_s(q_{h4} + q_{h5} + q_s)]/q)$.

$\Pr[E_2|E_1]$ represents the probability that $\mathcal{A}$ returns a valid forgery when $\mathcal{C}$ does not terminate due to the $\mathcal{A}$'s queries. Thus, by assumption $\mathcal{A}$ is able to return a valid forgery with probability at least ϵ.

Therefore, $\mathcal{A}$ can return a valid signature with the probability $\text{acc} = \epsilon - ([q_s(q_{h4} + q_{h5} + q_s)]/q)$, which can be derived from the following relations:

$$\Pr[E_1|E_2] = \frac{\Pr[E_1E_2]}{P[E_1]} \geq \epsilon \tag{4}$$

$$\begin{aligned}\Pr[E_1E_2] &\geq \epsilon Pr[E_1] \\ &= \epsilon\left(1 - \frac{q_s(q_{h4} + q_{h5} + q_s)}{q}\right) \\ &= \epsilon - \epsilon\frac{q_s(q_{h4} + q_{h5} + q_s)}{q} \\ &\geq \epsilon - \frac{q_s(q_{h4} + q_{h5} + q_s)}{q}.\end{aligned} \tag{5}$$

According to the General Forking Lemma [41], $\mathcal{C}$ can output two valid signatures by utilizing $\mathcal{A}$ as a subroutine with the probability $frk \geq \text{acc}([\text{a}\,cc/\hat{q}] - [1/\hat{h}])$, where $\hat{q}$ denotes the number of hash queries, and $\hat{h}$ denotes the number of replies to queries to random oracles. Therefore, $\mathcal{C}$ is able to break the ECDL problem with the probability

$$\left(\epsilon - \frac{q_s(q_{h4} + q_{h5} + q_s)}{q}\right)\left(\frac{\epsilon - \frac{q_s(q_{h4}+q_{h5}+q_s)}{q}}{q_{h4} + q_{h5}} - \frac{1}{q}\right) \tag{6}$$

where ϵ is nonnegligible. However, this is in contradiction to the ECDL problem. Therefore, the scheme is secure under the random oracle model.

B. Informal Security Analysis

1) *Mutual Authentication:* If an attacker intends to successfully pass the authentication with *ES*, the attacker must present a signature $\psi = a + \rho_l h_4(T, A)$, where $\rho_l = (d_l + vh_5(t_l, D_l, PID_l))$. According to the forking lemma, the attack can also provide another valid signature $\psi^{'} = a + (d_l + vh_5^{'})h_4$. Therefore, there will be a solution $(\psi - \psi^{'})(h_5 - h_5^{'})^{-1}h_4^{-1}$ of the ECDL problem. However, this is in contradiction to the ECDL problem.
   If an attacker, pretending to be a valid ES, aims to successfully pass the verification with *SD*, it needs to compute the hash value $h_4(T_1, T_2, SK_{ij})$, where $SK_{ij} = h_6(ID_i, T_2, SK_1, SK_2), SK_1 = x_iA, SK_2 = bA$. Thus, the attacker can compute $SK_1' = SK_1 = x_iA$.Then it can obtain a solution (x_ia) $P = SK_1'$ to the instance (x_iP, aP, x_iaP) which contradicts to the ECDH problem. Therefore, the mutual authentication is successfully achieved.
2) *Conditional Anonymity:* During the authentication with ES, SD dose not send its real identity to the verifier but its pseudonym. Therefore, the communicators and attackers cannot obtain the information about SD's identity. However, the RA can know who the SD is by computing $\Omega = \sigma_2^l - \xi_{ES}\sigma_1^l$ using the secret value ξ_{ES}, where Ω associates with the identity.
3) *Session Key Agreement:* ES_i and SD_j computes $SK_{ij} = h_6(ID_i, SK_1, SK_2, T_2)$ and $SK_{ji} = h_6(ID_i, SK_1', SK_2', T_2)$ independently. Afterward SD_j confirms the validity of the session key SK_{ji} by verifying the equation $e = h_4(T_1, T_2, SK_{ji})$.
4) *Unlinkability:* SD owns a number of pseudonyms and each pseudonym contains random number and timestamp. Therefore, attackers cannot distinguish whether two messages belong to the same device.
5) *No Online RA:* It is obvious that the mutual authentication between devices and servers dose not rely on the RA. What is more, there is no need for the RA to update the pseudonyms and temporary keys for devices.
6) *Forward Security:* Due to the session key SK_{ij} and SK_{ji} is computed by ES_i and SD_j independently, only the attacker who can break the ECDH assumption can get the random numbers a and b to generate the right session key.
7) *Resistance to Common Attacks:* The proposed scheme can resist the following common attacks.
   a) *Reply Attack:* Given the randomness and timestamp, it is easy to find whether a replay has occurred by checking the freshness of the timestamp.
   b) *Stolen Verifier Attack:* Since the information about ESs recorded on the BC and there is not a list recording the devices' identities, the proposed scheme is secure against the stolen verifier attack.
   c) *Impersonation Attack:* If an attacker intends to impersonate a legitimate entity (e.g., *SD*), it must break the mutual authentication security, which is proven in Sections VI-A and VI-B. Therefore, the proposed scheme is secure against the impersonation attack.

C. Formal Security Verification Using ProVerif

In this section, we present the results of formal security verification using ProVerif. The ProVerif is an automatic cryptographic protocol verification tool that can prove the security of various schemes. It can achieve some security properties, such as reachability, correspondence assertions, and observational equivalences. Since all SDs and ESs are equal, the simulation result cannot be affected by the number of devices and servers. Therefore, we assume that there are only one ES and one device. Similarly, we assume that devices own one key and one pseudonym instead of a batch. The simulation code is described in [42]. Fig. 6 shows the simulation results.

In Fig. 6, (1), (2), and (3) are the results of reachability query that show the attacker cannot obtain the private key of RA and the session key. The result (4) shows the strong anonymity of M which denotes the pseudonyms belonging to a device. In the simulation code, four events have been identified denoted as event authES(SK), event authSD(SK), event regisES and event regisSD. In Fig. 6, (5), (6), and (7) show the results of correspondence assertions that indicate the mutual authentication between ES and SD is successfully achieved.

```
--------------------------------------------------------------------------------
Verification summary:
(1)Query not attacker(s[]) is true.
(2)Query not attacker(SK_ij[]) is true.
(3)Query not attacker(SK_ji[]) is true.
(4)Non-interference M is true.
(5)Query inj-event(authSD(SK)) ==> inj-event(authES(SK)) is true.
(6)Query inj-event(authES(SK)) ==> inj-event(regisSD) is true.
(7)Query inj-event(authSD(SK)) ==> inj-event(regisES) is true.
--------------------------------------------------------------------------------
```

Fig. 6. Simulation results from the ProVerif tool.

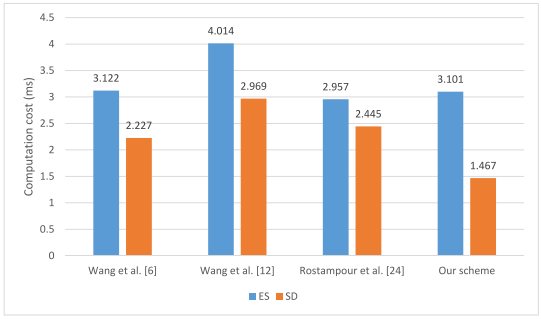


Fig. 7. Comparison of computation cost.

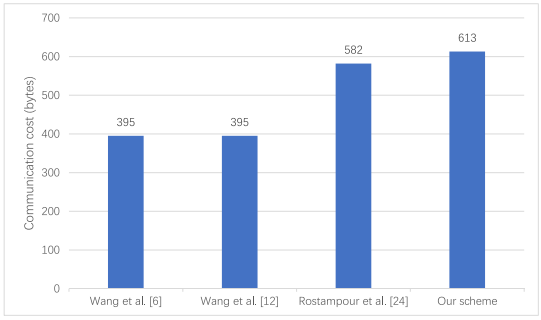


Fig. 8. Comparison of communication cost.

TABLE III
COMPARISON OF COMPUTATIONAL COST (MS)

Schemes	ES	SD
Ref. [6]	$6T_{Gm} + 2T_{Ga} + 5T_h$	$4T_{Gm} + T_{Ga} + 4T_h$
Ref. [12]	$8T_{Gm} + 3T_{Ga} + 5T_h$	$6T_{Gm} + 2T_{Ga} + 5T_h$
Ref. [24]	$6T_{Gm} + T_{Ga}$	$7T_{Gm} + T_{Ga}$
Ours	$5T_{Gm} + 3T_{Ga} + 4T_h$	$3T_{Gm} + 3T_h$

TABLE IV
COMPARISON OF COMMUNICATION COST (BYTE)

Schemes	Communication cost	Length(byte)
Ref. [6]	$3\|G\| + 2\|Z_q\| + 2\|T\|$	395
Ref. [12]	$3\|G\| + 2\|Z_q\| + 2\|T\|$	395
Ref. [24]	$6\|G\|$	582
Ours	$5\|G\| + 2\|Z_q\| + \|ID\| + 3\|T\|$	613

VII. PERFORMANCE ANALYSIS

In this section, we analyze the performance of our proposed scheme from three aspects which are computation cost, communication cost and on-chain cost. And we compare it with three related schemes [6], [12], and [24]. In Table III, we demonstrate the comparison of computation overhead to analyze the cost of these schemes on both SD and server. Additionally, the comparison of communication overhead is shown in Table IV. They are the theoretical analysis of the proposed scheme and the other three schemes, and they can point out the reasons for the differences among these schemes.

Wang et al. [6] built a consortium blockchain using Hyperledger Composer version V0.20.7, running on an x86_64 GNU/Linux system with 1 core and 2 GB RAM for executing smart contract operations. And we refer their experimental results, and the results are illustrated in Table V. We use the cryptographic library called MIRACL Core to test the execution time of cryptography operations under the Ubuntu 18.04 with Intel Skylake CPU @ 2.2 GHz and 2 GB bytes memory provided by a cloud platform. The experimental results are depicted in Figs. 7 and 8.

A. Computation Cost

We use T_{Gm}, T_{Ga} and T_h to represent the execution time of scalar multiplication operation in $\mathbb{G}$, point addition operation in $\mathbb{G}$, and one-way hash function operation. We omit some overhead of light operations, such as XOR operation, addition operation, and multiplication operation in $\mathbb{Z}_q^*$.

We mainly focus on the authentication phase because the registration is executed for only once and the corresponding cost has little influence on the whole system. First, we consider the computation cost on server side. In [6], the computation cost on server side requires six scalar multiplication operations, two point addition operations and five hash function operations. Therefore, the computation cost is about $6T_{Gm} + 2T_{Ga} + 5T_h \approx 3.122$ ms. In [12], the server requires eight scalar multiplication operations, three point addition operations and five hash function operations. The computation cost is about $8T_{Gm} + 3T_{Ga} + 5T_h \approx 4.014$ ms. In [24], the server needs to perform six scalar multiplication operations and one point addition operation. Thus, the computation cost is about $6T_{Gm} + T_{Ga} \approx 2.957$ ms. Since the computation overhead is similar in both cases of the authentication phase in our scheme, we consider the second one. And the computation cost on server side in our scheme needs five scalar multiplication operations, three point addition operations and four hash function operations. The corresponding cost is about $5T_{Gm} + 3T_{Ga} + 4T_h \approx 3.101$ ms. Then, we consider computation time cost on SD side. In [6], the computation cost on device side requires four scalar multiplication operations, one point addition operation and four hash function operations. Therefore, the computation cost is about $4T_{Gm} + T_{Ga} + 4T_h \approx 2.227$ ms. In [12], the device requires six scalar multiplication operations, two point addition operations and five hash function operations. The computation cost is about $6T_{Gm} + 2T_{Ga} + 5T_h \approx 2.969$ ms. In [24], the device needs to perform seven scalar multiplication operations and one point addition operation. Thus, the computation cost is about $7T_{Gm} + T_{Ga} \approx 2.445$ ms. And the computation cost on device side in our scheme needs three scalar multiplication operations

TABLE V
TIME COST (IN SECONDS) OF THE SMART CONTRACT IN [6]

Operations	Update	Delete	Query
Max Time	2.681	2.590	0.312
Min Time	1.989	2.086	0.138
Average Time	2.335	2.338	0.225

and three hash function operations. The corresponding cost is about $3T_{Gm} + 3T_h \approx 1.467$ ms. The computation overhead comparison of our scheme with [6], [12], and [24] is shown in Fig. 7. And Table III shows the specific data.

From Table III, we can see that our scheme requires several multiplication operations in $\mathbb{G}$. That is, due to the need to develop a method that protects the real identity of SD from being exposed to the ES responsible for generating pseudonyms. By doing so, the real identity of SD can only be revealed by the RA that owns the secret value ξ.

B. Communication Cost

As the BLS12381 curve is used in our scheme, the size of the element in $\mathbb{Z}_q^*$ and $\mathbb{G}$ is 48 bytes and 97 bytes, respectively. The size of timestamp and identity are 4 bytes and 20 bytes, respectively. Table IV demonstrates the communication costs of these schemes. $|G|$ and $|Z_q|$ denote the size of the element in $\mathbb{G}$ and $\mathbb{Z}_q^*$. $|T|$ and $|ID|$ denote the size of the timestamp and the identity, respectively.

In [6], it needs to transmit $3|G| + 2|Z_q| + 2|T|$. The communication cost is $3 \times 97 + 2 \times 48 + 2 \times 4 = 395$ bytes. In [12], it needs to transmit $3|G| + 2|Z_q| + 2|T|$. The communication cost is $3 \times 97 + 2 \times 48 + 2 \times 4 = 395$ bytes. In [24], it needs to transmit $6|G|$. The communication cost is $6 \times 97 = 582$ bytes. Since the communication overhead is similar in both cases of the authentication phase in our scheme, we consider the second one. Our scheme needs to transmit $5|G| + 2|Z_q| + |ID| + 3|T|$. The communication cost is $5 \times 97 + 2 \times 48 + 20 + 3 \times 4 = 613$ bytes.

From Fig. 8, we can see that the communication overhead of our scheme is a little higher than the other schemes. The communication overhead in our scheme is primarily due to the pseudonym $\{PID, D\}$ generated by ES. In real-world scenarios, temporary keys and pseudonyms with expiration time are regularly distributed to protect the privacy of devices. Thus, the frequency of updating keys will be high due to the large number of devices. In contrast to other schemes where the task of updating keys is usually undertaken by RA, our scheme utilizes ESs to generate pseudonyms and update keys for devices. The overhead incurred is affordable and does not pose a significant challenge.

C. On-Chain Operation Time and Storage Cost

As Rostampour et al. [24] did not utilize blockchain, the comparisons relate only to our proposed protocol [6], [12]. In our scheme, the on-chain operations include update operation, delete operation and query operation. From Table V, we can see that the time taken for update and delete operation is approximately 2.335 s and 2.338 s, respectively. However, the cost of query operation is significantly lower than that of update and delete operation, at around 0.225 s. In the first case of the authentication process, ES performs one read operation to check whether the device is revoked. In the second case of the authentication process, ES performs two read operations in our scheme. One read operation is to determine whether the device is revoked, and the other one is to look up the public key of ES when the device moves to another edge network. In [6] and [12], the server needs to perform one read operation. The comparison result can be seen in Fig. 9. In real-world scenarios, ES has much more computation resources than the simulation platform and the cost of on-chain operation will be less.

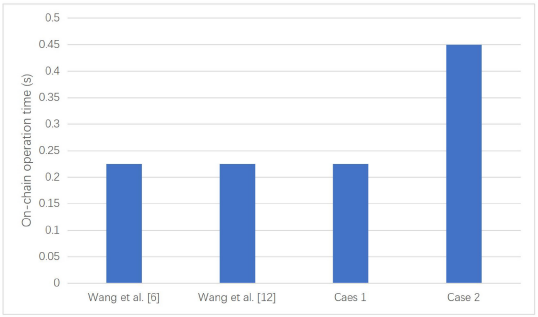


Fig. 9. Comparison of on-chain operation time cost.

Additionally, since the content on the blockchain cannot be deleted, the ledger will grow indefinitely. Therefore, it is vital to evaluate the storage overhead on the blockchain. In [6], there is an entry existing on the blockchain for every registered device. The entry contains $\{PID_i, C_i, R_i, ET_i\}$. Thus, the storage overhead on the blockchain for one device is calculated as $48+32+97+4 = 181$ bytes. In [12], the entry on the blockchain is consisted of $\{PID, X, R, T\}$ for each device. The corresponding storage overhead is $32+97+97+4 = 230$ bytes. In our scheme, the content recorded on the blockchain is $\{\Omega, t_{\text{date}}, z\}$. And the corresponding storage overhead is $97+4+4 = 105$ bytes. There is a difference between our scheme and the other two schemes that we only need to store one entry for multiple pseudonym services, while they need to store one entry for each service. In other words, when we apply for ten pseudonyms, we only store one entry on the blockchain, while they need to store ten entries. Suppose that a device requests five pseudonyms and corresponding keys at once in our scheme. The comparison of storage cost on the blockchain is illustrated in Fig. 10.

VIII. CONCLUSION

To protect the security and privacy of devices in an edge-computing environment, in this study, we propose a blockchain-based mutual authentication and session key agreement for cross-domain IIoT. Specifically, we utilize blockchain to build trust among ESs to share cross-domain information. Furthermore, a pseudonym-based privacy-preserving method is designed. ESs can generate pseudonyms for devices but do not know the device's real identity. In addition, the device and the server can authenticate each other without revealing the device's real identity. However, a RA can easily revoke

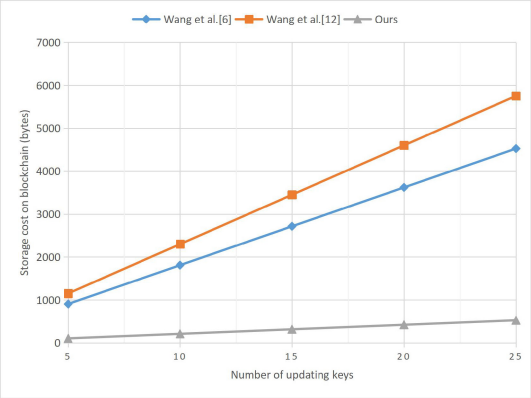


Fig. 10. Comparison of storage cost on blockchain.

the device. The security analysis and experimental results demonstrate that the proposed scheme is efficient and practical. In the future, we will investigate a cross-domain authentication scheme between IIoT devices and ESs completely free from trusted third parties.

ACKNOWLEDGMENT

The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this article.

REFERENCES

[1] L. Babun, K. Denney, Z. B. Celik, P. McDaniel, and A. S. Uluagac, "A survey on IoT platforms: Communication, security, and privacy perspectives," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108040.

[2] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, "A review and state of art of Internet of Things (IoT)," *Arch. Comput. Methods Eng.*, vol. 29, no. 3, pp. 1395–1413, 2021.

[3] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things architecture, possible applications and key challenges," in *Proc. 10th Int. Conf. Front. Inf. Technol.*, 2012, pp. 257–260.

[4] W. Z. Khan, M. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial Internet of Things: Recent advances, enabling technologies and open challenges," *Comput. Electr. Eng.*, vol. 81, Jan. 2020, Art. no. 106522.

[5] I. Mohiuddin and A. Almogren, "Security challenges and strategies for the IoT in cloud computing," in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, 2020, pp. 367–372.

[6] J. Wang, L. Wu, K.-K. R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1984–1992, Mar. 2020.

[7] A. Shahidinejad, M. Ghobaei-Arani, A. Souri, M. Shojafar, and S. Kumari, "Light-edge: A lightweight authentication protocol for IoT devices in an edge-cloud environment," *IEEE Consum. Electron. Mag.*, vol. 11, no. 2, pp. 57–63, Mar. 2022.

[8] X. Yang et al., "Secure and lightweight authentication for mobile-edge computing-enabled WBANs," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12563–12572, Jul. 2022.

[9] T.-Y. Wu, Q. Meng, L. Yang, X. Guo, and S. Kumari, "A provably secure lightweight authentication protocol in mobile edge computing environments," *J. Supercomput.*, vol. 78, no. 12, pp. 13893–13914, 2022.

[10] A. Irshad, M. Sher, H. F. Ahmad, B. A. Alzahrani, S. A. Chaudhry, and R. Kumar, "An improved multi-server authentication scheme for distributed mobile cloud computing services," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 12, pp. 5529–5552, 2016.

[11] G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin, "A blockchain-based mutual authentication scheme for collaborative edge computing," *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 1, pp. 146–158, Feb. 2022.

[12] W. Wang, H. Huang, L. Xue, Q. Li, R. Malekian, and Y. Zhang, "Blockchain-assisted handover authentication for intelligent telehealth in multi-server edge computing environment," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 102024.

[13] Q. Fan, J. Chen, L. J. Deborah, and M. Luo, "A secure and efficient authentication and data sharing scheme for Internet of Things based on blockchain," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102112.

[14] D. R.-J. G.-J. Rydning, J. Reinsel, and J. Gantz, *The Digitization of the World From Edge to Core*, vol. 16, Int. Data Corp., Framingham, MA, USA, 2018.

[15] Y. Yang, L. Wei, J. Wu, C. Long, and B. Li, "A blockchain-based multidomain authentication scheme for conditional privacy preserving in vehicular Ad-Hoc network," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8078–8090, Jun. 2021.

[16] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor authenticated key agreement scheme for Industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3801–3811, Mar. 2021.

[17] Y. Zhang, D. He, P. Vijayakumar, M. Luo, and X. Huang, "SAPFS: An efficient symmetric-key authentication key agreement scheme with perfect forward secrecy for Industrial Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 11, pp. 9716–9726, Jun. 2023.

[18] J. Li et al., "EPA-CPPA: An efficient, provably-secure and anonymous conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *Veh. Commun.*, vol. 13, pp. 104–113, Jul. 2018.

[19] J. Li, W. Zhang, V. Dabra, K.-K. R. Choo, S. Kumari, and D. Hogrefe, "AEP-PPA: An anonymous, efficient and provably-secure privacy-preserving authentication protocol for mobile services in smart cities," *J. Netw. Comput. Appl.*, vol. 134, pp. 52–61, May 2019.

[20] Y. Chang, J. Li, N. Lu, W. Shi, Z. Su, and W. Meng, "Practical privacy-preserving scheme with fault tolerance for smart grids," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 1990–2005, Jan. 2024.

[21] J. Liu, X. Li, Q. Jiang, M. S. Obaidat, and P. Vijayakumar, "BUA: A blockchain-based unlinkable authentication in VANETs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[22] A. Maria, V. Pandi, J. D. Lazarus, M. Karuppiah, and M. S. Christo, "BBAAS: Blockchain-based anonymous authentication scheme for providing secure communication in VANETs," *Secur. Commun. Netw.*, vol. 2021, Feb. 2021, Art. no. 6679882.

[23] X. Li, Y. Wang, P. Vijayakumar, D. He, N. Kumar, and J. Ma, "Blockchain-based mutual-healing group key distribution scheme in unmanned aerial vehicles Ad-Hoc network," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11309–11322, Nov. 2019.

[24] S. Rostampour, M. Safkhani, Y. Bendavid, and N. Bagheri, "ECCbAP: A secure ECC-based authentication protocol for IoT edge devices," *Pervasive Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101194.

[25] D. Rangwani and H. Om, "A secure user authentication protocol based on ECC for cloud computing environment," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3865–3888, Jan. 2021.

[26] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Gener. Comput. Syst.*, vol. 91, pp. 475–492, Feb. 2019.

[27] I. Alam and M. Kumar, "A novel protocol for efficient authentication in cloud-based IoT devices," *Multimedia Tools Appl.*, vol. 81, no. 10, pp. 13823–13843, 2022.

[28] A. B. Amor, M. Abid, and A. Meddeb, "A privacy-preserving authentication scheme in an edge-fog environment," in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA)*, 2017, pp. 1225–1231.

[29] K. Kaur, S. Garg, G. Kaddoum, M. Guizani, and D. N. K. Jayakody, "A lightweight and privacy-preserving authentication protocol for mobile edge computing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

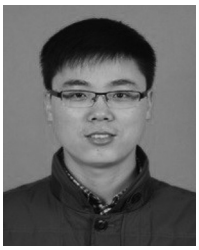
[30] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Syst. J.*, vol. 14, no. 1, pp. 560–571, Mar. 2020.

[31] Y. Li, Q. Cheng, X. Liu, and X. Li, "A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing," *IEEE Syst. J.*, vol. 15, no. 1, pp. 935–946, Mar. 2021.

[32] D. Chaum and E. V. Heyst, "Group signatures," in *Proc. Workshop Theory Appl. Cryptogr. Techn.*, 1991, pp. 257–265.

[33] H. Huang, Y. Wu, F. Xiao, and R. Malekian, "An efficient signature scheme based on mobile edge computing in the NDN-IoT environment," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 5, pp. 1108–1120, Oct. 2021.

[34] M. Shen et al., "Blockchain-assisted secure device authentication for cross-domain Industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, May 2020.
[35] C. Lin, D. He, X. Huang, N. Kumar, and K.-K. R. Choo, "BCPPA: A blockchain-based conditional privacy-preserving authentication protocol for vehicular Ad Hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7408–7420, Dec. 2021.
[36] S. S. Panda, D. Jena, B. K. Mohanta, S. Ramasubbareddy, M. Daneshmand, and A. H. Gandomi, "Authentication and key management in distributed IoT using blockchain technology," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12947–12954, Aug. 2021.
[37] C. Zhang, L. Zhu, and C. Xu, "BPAF: Blockchain-enabled reliable and privacy-preserving authentication for fog-based IoT devices," *IEEE Consum. Electron. Mag.*, vol. 11, no. 2, pp. 88–96, Mar. 2021.
[38] X. Yang et al., "Blockchain-based secure and lightweight authentication for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3321–3332, Mar. 2021.
[39] L. Wang, Y. Tian, and D. Zhang, "Toward cross-domain dynamic accumulator authentication based on blockchain in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2858–2867, Apr. 2022.
[40] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, pp. 361–396, Mar. 2000.
[41] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 390–399.
[42] Y. Zhu. "Open-source code of the implementation of the BMASKA using the proverif tool." github. 2023. [Online]. Available: https://github.com/yhzhu11/proverif/blob/main/BMASKA.pv



Jie Cui (Senior Member, IEEE) was born in Henan, China, in 1980. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2012.

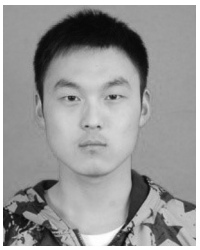
He is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University, Hefei. He has over 150 scientific publications in reputable journals, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON MULTIMEDIA, academic books, and international conferences. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking.

Yihu Zhu received the B.Eng. degree from Anhui Polytechnic University, Wuhu, China, in 2020.

He is currently a research student with the School of Computer Science and Technology, Anhui University, Hefei, China. His research focuses on the security of Industrial Internet of Things.

Hong Zhong (Member, IEEE) was born in Anhui, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2005.

She is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University, Hefei. She has over 200 scientific publications in reputable journals, such as IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS AND IEEE TRANSACTIONS ON BIG DATA, academic books, and international conferences. Her research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking.



Qingyang Zhang (Member, IEEE) was born in Anhui, China, in 1992. He received the B.Eng. degree and Ph.D. degree in computer science from Anhui University, Hefei, China, in 2014 and 2021,respectively.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. He has over 30 scientific publications in reputable journals (e.g., PROCEEDINGS OF THE IEEE, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and IEEE TRANSACTIONS ON COMPUTERS) and international conferences. His research interest includes edge computing, computer systems, and security.

Chengjie Gu received the Ph.D. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2012.

From 2012 to 2017, he was an Innovation Team Leader with the 38th Research Institute, China Electronics Technology Group Corporation, Beijing, China, and conducted research and development in the communication and networking sector. He is currently the Dean of the School of Public Security and Emergency, Anhui University of Science and Technology, Huainan, China, and the President of the Security Research Institute, New H3C Group. He has completed postdoctoral research with the University of Science and Technology of China, Hefei, China. He is a High-Level Innovation Leader of Anhui Province and a Cybersecurity Expert of Zhejiang, China. His research interest includes network security and trusted network architecture.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009.

He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University, and Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai, China. He has published over 100 research papers in refereed international journals and conferences, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION SECURITY AND FORENSIC, and Usenix Security Symposium. His work has been cited more than 10 000 times at Google Scholar. His main research interests include cryptography and information security, in particular, cryptographic protocols.

Prof. He is the recipient of the 2018 IEEE Sysems Journal Best Paper Award and the 2019 IET Information Security Best Paper Award. He is in the editorial board of several international journals, such as *Journal of Information Security and Applications*, *Frontiers of Computer Science*, and *Human-Centric Computing and Information Sciences*.