Hindawi
Security and Communication Networks
Volume 2022, Article ID 3693748, 17 pages
https://doi.org/10.1155/2022/3693748

*Research Article*

# A Certificateless-Based Authentication and Key Agreement Scheme for IIoT Cross-Domain

**Xiangyang Wang**,[1] **Chunxiang Gu**,[1,2] **Fushan Wei**,[1] **Siqi Lu**,[1] **and Zhaoxuan Li**[3]

[1]*Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, Henan 450001, China*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, Henan 450001, China*
[3]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*

Correspondence should be addressed to Chunxiang Gu; gcx5209@126.com

Received 7 March 2022; Revised 9 September 2022; Accepted 20 September 2022; Published 17 October 2022

Academic Editor: Zhen Wang



The Industrial Internet of Things (IIoT) improves productivity and intelligent manufacturing process through revolutionary technology. Due to the complexity of the manufacturing process, cross-domain access is inevitable. Recently, Meng et al. proposed a secure and efficient blockchain-assisted entity authentication mechanism BASA for IIoT cross-domain. In the BASA scheme, the authors utilized identity-based signature (IBS) to realize mutual authentication and the Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) exchange mechanism to negotiate the session key. Due to the inherent key escrow problem of identity-based cryptography (IBC), the key generation center (KGC) can obtain the session key negotiated between two entities distributed in different domains. When KGC is threatened, the security of the session key is worrying. Considering this security concern, based on the BASA scheme, in this article, we first show a secure and efficient certificateless public-key signature (CL-PKS) scheme with anonymity. Then, combined with the ECDHE key exchange mechanism, we give an efficient cross-domain authentication and key agreement scheme CL-BASA with the aid of consortium blockchain. After that, we make security verification by the formal analysis tool, Tamarin, which shows that our CL-BASA is secure. The evaluation demonstrates that our CL-BASA may have a slight disadvantage in storage overhead, but it has obvious advantages than competitor schemes in terms of communication overhead and computational overhead.

## 1. Introduction

IoT technology provides a platform to connect equipment and factory domains to promote the task of automated manufacturing, which greatly improves productivity and minimizes management costs. Due to the emergence of Industry 4.0 [1], Industrial Internet of Things (IIoT) is becoming an important enabling technology. However, considering the complexity of the current production process, it has become a trend that some related domains cooperate to complete the production of products.

To complete the production of products, devices distributed in different domains need to share and exchange data by an efficient communication mechanism. In general, two different domains may not trust each other. Therefore, it is not easy to establish a secure communication mechanism between entities distributed in different domains. In the existing cross-domain authentication schemes, most of them adopt traditional KPI technology and digital certificate to realize cross-domain authentication. In traditional public-key cryptography (PKC), the user's random public key is associated with the user's identity through a certificate. This inevitably leads to a large amount of storage and computational overhead for the certificate management. Due to the resource constraints of IoT devices, these cross-domain schemes are inappropriate under the IoT scenarios. To simplify the certificate management, Shamir [2] introduced the identity-based cryptosystem (IBC), in which some public

known information, such as e-mail address, is used as the user's public key. Therefore, the certificate of the random public key is no longer required.

For the tamper-proof and decentralized characteristics, blockchain is increasingly used to solve the problems of identity credibility and data credibility in the field of the IoT [3]. The consortium blockchain is called "shared authentication blockchain," and its essence is a distributed hosting accounting system. The system is controlled by multiple "authoritative" nodes designated by the organization. These nodes manage and send the whole system according to the consensus mechanism. Consortium blockchain can be used as a public and trusted platform for domain-specific information sharing to establish trust between different domains. At present, many scholars have studied the combination of blockchain and cross-domain communication, but these studies generally adopt the IBS scheme in the authentication stage. As we all know, IBC has obvious advantages in storage and computation overhead, but it has the problem of key escrow. Therefore, the malicious KGC can forge the signature of any signer. In the cross-domain scenario, the key escrow problem can directly lead KGC to calculate the session key negotiated between the parties. So, in the scenario with high security, IBC-based schemes are unreliable.

To solve the inherent key escrow problem of IBC, Al-Riyami and Paterson [4] proposed the certificateless public key cryptography (CL-PKC) in 2003, which not only eliminates the use of certificates in PKC but also solves the problem of key escrow in IBC. In the CL-PKC, the private key of the user is constructed by the secret value and the partial private key. The former is randomly chosen by the user, while the latter is generated by KGC. In the past few years, a large number of CL-PKC schemes have been proposed. In addition, in 2001, Brown et al. [5] proposed a modification scheme of the optimal mail certificate (OMC) scheme [6]; it was later called the Elliptic Curve Qu-Vanstone (ECQV) implicit certificate scheme [7]. In addition to the certificateless scheme, ECQV implicit certificate scheme can also avoid the problem of key escrow. ECQV implicit certificate schemes have found application in the IoT. For example, it becomes part of the encryption suite building block in the ZigBee smart energy standard [8]. In recent years, the applications of implicit certificate schemes based on ECQV in the scenario of IoT have also been widely studied. However, compared with the certificateless scheme, the implicit certificate scheme based on ECQV needs to store an implicit certificate, so the management and maintenance of the certificate will inevitably bring more overhead.

In this article, based on BASA [9] proposed by Meng et al., we adopt the certificateless signature scheme and ECDHE key exchange mechanism to design a cross-domain authentication and key agreement scheme CL-BASA with the aid of blockchain, which is suitable for IIoT cross-domain. In detail, we first design a secure and efficient certificateless short signature scheme CL-PKS with anonymity; then, during mutual authentication stage, consortium blockchain is used to provide the authenticity of anonymous identity and public key, while CL-PKS is exploited to ensure data integrality. Finally, during the key agreement stage, CL-PKS and ECDHE are combined to realize cross-domain key agreement between two entities distributed in different domains. Compared with the competitor schemes, our scheme may be slightly insufficient in storage overhead, but it is significantly better in terms of communication overhead and computational overhead.

Our contributions are as follows:

(i) We design a secure and efficient certificateless public-key signature scheme CL-PKS with anonymity. Based on the Elliptic Curve Computational Diffie-Hellman (ECCDH) problem, we prove that our signature scheme satisfies the existentially unforgeable one against adaptively chosen message attacks (EUF-CMA) in the random oracle model. See Appendix A for the detailed proof.

(ii) Based on the BASA, we use CL-PKS and ECDHE key exchange mechanism to design a cross-domain authentication scheme CL-BASA suitable for IIoT cross-domain. Compared with the BASA scheme, our CL-BASA scheme avoids the possible security threats caused by the key escrow problem and is superior to BASA in terms of communication overhead.

(iii) For the management of anonymous identity, we designed a simple identity management mechanism, which is designed based on the CL-PKS scheme with anonymity. Compared with the identity management mechanism in BASA, our mechanism reduces the overhead of identity management and the computation overhead of generating anonymous identity.

(iv) In order to illustrate the security of our CL-BASA, we make security verification by the formal analysis tool, Tamarin. The result of formal verification shows that our CL-BASA is secure. The evaluation demonstrates that our CL-BASA has more obvious advantages than competitor schemes in terms of communication overhead and computational overhead.

The rest of this article is organized as follows: Section 2 reviews the existing authentication mechanisms proposed for IoT. Section 3 introduces some basic knowledge. In Section 4, we show our certificateless public-key signature scheme CL-PKS and the security model. In Section 5, we introduce our authentication and key agreement mechanism CL-BASA for IIoT cross-domain. Section 6 is the security analysis. Section 7 is the security verification of our scheme. We give a performance evaluation in Section 8. Section 9 is the summary of our work.

## 2. Related Work

In this section, we introduce the authentication schemes that have been proposed for the IoT scenario and show them from three aspects, namely, certificate-based schemes, identity-based schemes, and certificateless-based schemes.

*2.1. Certificate-Based Schemes.* Due to the resource limitation of wireless sensor networks, security solutions based on the elliptic curve are often used in wireless sensor networks. Several implicit certificate schemes for wireless sensor networks were introduced [10, 11]. In addition, the applications in IoT scenario of ECQV implicit certificate schemes have been well studied. By combining Elliptic Curve Diffie-Hellman (ECDH) key exchange mechanisms and ECQV implicit certificate scheme, a key management protocol suitable for mobile and IIoT systems was proposed [12]. For robust key negotiation, lightweight node authentication, fast key reset, and effective replay attack protection, the ECQV implicit certificate scheme was used to make a lightweight security association suitable for IoT devices [13]. Considering the deficiency of the PAuthKey protocol, it does not clearly point out how to share the network key between IoT devices, and CA only authenticates each authorized device based on the network key rather than a single key. The authors propose a new ECQV implicit certificate-based protocol, which solves the problem of the PAuthKey protocol. The first nontrivial identity-based authentication (IBI) scheme [14] with implicit authentication is given by using the ECQV implicit certificate scheme. Compared with the traditional identity-based scheme, the method based on implicit certificates can resist key escrow. All the above research studies pay attention to authentication in the single-domain. Besides, there are also some research studies based on the ECQV implicit certificate scheme in IoT cross-domain scenarios. To maintain the reliable connection and accessibility of IoT devices distributed in different domains, the authors of [15] proposed an authentication mechanism based on the ECQV implicit certificate scheme. This two-stage authentication protocol allows sensor nodes and end-users to make mutual authentication and establish a secure connection. For realizing efficient and dynamic certificate distribution at a lower cost in the vehicle cloud network, the authors of [16] designed an effective certificate distribution mechanism based on the ECQV implicit certificate scheme in the vehicle cloud network. These schemes only make the research on cross-domain authentication in the distributed IoT architecture. Strictly speaking, the management domains in their study are not independent. Besides, they need to store implicit certificates and add additional storage overhead. The authors of [17] proposed a certificate-based scheme for cross-domain IoT, which is a lightweight authentication scheme based on consortium blockchain and designed a cryptocurrency-like digital token to build trust. This scheme has obvious advantages in computation and communication overhead, but it only realizes one-way authentication and provides additional overhead for certificate storage.

*2.2. Identity-Based Schemes.* Considering the advantages of identity-based cryptography (IBC), many identity-based authentication schemes have been proposed. An identity-based mutual authentication scheme for power line communication (PLC) was developed [18]. Similarly, the authors of [19] proposed an identity-based authentication scheme for cloud computing, which is considered more effective than the SSL authentication protocol. However, this scheme does not consider the mutual authentication of terminal devices. The authors of [9] proposed a blockchain-assisted entities authentication mechanism BASA for IIoT cross-domain. Specifically, the authors utilized a consortium blockchain to realize decentralization and build trust between different factory domains. Besides, identity-based signature (IBS) was used for identity authentication, and ECDHE was adopted to negotiate the session key. In their work, cross-domain identity authentication was completed under the cooperation of key generation center (KGC), authentication agent server (AAS), and blockchain agent server (BAS), which needs a lot of executions and interactions among the coordination elements and causes a large communication overhead. In the IIoT scenario, the authors of [20] proposed a cloud-based cross-domain data sharing platform based on multiple security gateways. These security gateways store information in a centralized cloud with the aid of blockchain. In this scheme, recent technologies such as blockchain, machine learning, ECDHE, IBS, and cloud computing are used to realize the trust of cross-domain communication between entities. But it is not secured for sharing data, and the authors only verify the user or organization to exchange the data. By combining the IoT platform and the Hyperledger Fabric framework, the authors of [21] proposed a framework sash for IoT data sharing. In this framework, the blockchain was used to store access control policies and make access control decisions. At the same time, identity-based encryption is used to provide encryption mandatory access control. The authors have not defined how to distribute the encryption key to decrypt the obtained data, and the use of the encryption key distribution agencies has brought the problems of centralized networks and reliance on authority. The authors of [22] built a secure, lightweight, and blockchain-based IoT cross-domain access control system by integrating licensed blockchain, attribute-based access control, and identity-based signature. The authors of [23] proposed a decentralized IoT authentication scheme $A^2$ chain, which is based on IBC. This scheme uses edge computing to disperse the processing of authentication requests and eliminate the burden of authentication and uses blockchain and sidechain technology to securely share authentication information. But this scheme only provides mutual authentication between the terminal device in domain $A$ and the edge authentication server in domain $B$ and not between the terminal device, and there is no session key negotiation. All abovementioned schemes utilize the blockchain technology to achieve decentralization, but the inherent key escrow problem of IBC can lead to KGC to decrypt any ciphertext of any user and forge a signature on any message. Therefore, IBC seems to be only applicable to small private networks with low-security requirements and inevitably faces some security challenges in cross-domain communication.

*2.3. Certificateless-Based Schemes.* To solve the inherent problem of key escrow in IBC, Al-Riyami and Paterson proposed the CL-PKC. The certificateless signature scheme

is one of the several feasible methods to provide data integrity and user identity security for the devices with limited resources. In the current research, the applications of the certificateless schemes in the IoT scenario are often combined with signcryption schemes. The authors of [24] proposed a short signcryption scheme S-ECSC based on the elliptic curve cryptosystem for the IoT scenarios, which provides confidentiality and unforgeability and ensures secure data storage and transmission. The wireless body area network (WBAN) is one of the emerging technologies of the IoT. The authors of [25] designed a lightweight and provable secure cross-domain access control scheme for WBAN. This scheme adopts a certificateless signcryption scheme on the application provider and an identity-based signcryption scheme on the WBAN. For the security and efficiency of data transmission, the authors of [26] constructed a secure and lightweight hybrid signcryption scheme for the IoT. This scheme effectively protects the secure communication between nodes with limited resources in the IoT. Unfortunately, all abovementioned research only provides a signcryption scheme that may be applicable to the IoT scenario but does not give a complete entity authentication scheme. Besides, there are still many research on the applications of certificateless schemes in the IoT scenario, which will not be repeated here. The signcryption scheme was first proposed by [27]. At the same time, the author pointed out that not all messages need to be encrypted and signed at the same time, that is, the encryption-only mode and the signature-only mode. We know that encryption is not indispensable in the key agreement process. Usually, the signature-only mode is needed to ensure the integrity of data. Therefore, we consider using the certificateless signature scheme in our scheme during the mutual authentication stage and key agreement stage.

## 3. Preliminaries

In this section, we introduce some basic knowledge. The list of the key notations used in this article is represented in Table 1.

Let $p$ be a prime number greater than 3; a nonsingular elliptic curve $E$ defined on the finite field $F_p$ is composed of points satisfying the equation $y^2 = x^3 + ax + b \bmod p$ and a point representing infinity denoted by $O$, where $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0 \bmod p$. Let $G$ be the set of all points on the elliptic curve, namely,

$$G = \left\{(x, y):\ y^2 = x^3 + ax + b \bmod p\right\} \cup O. \tag{1}$$

*3.1. Bilinear Pairing.* Let $G_1$ be an additive cyclic group generated by $P$, whose order is prime $q$, and $G_2$ is a multiplicative cyclic group with the same order. Bilinear pairing $e$: $G_1 \times G_1 \longrightarrow G_2$ satisfies the following properties:

(1) Bilinearity: for any $P, Q, R \in G_1$, we have $e(P + R, Q) = e(P, Q)e(R, Q)$ and $e(P, Q + R) = e(P, Q)e(P, R)$. Especially, given any $a, b \in Z_q^*$, we have

$$e(aP, bP) = e(P, P)^{ab} = e(P, abP) = e(abP, P). \tag{2}$$

(2) Nondegeneracy: there exists $P, Q \in G_1$, such that $e(P, Q) \neq 1$.

(3) Computability: given any $P, Q \in G_1$, there exists a polynomial-time algorithm to compute $e(P, Q) \in G_2$.

Table 1: Description of notations.

| Notation | Description |
|---|---|
| $G$ | A cyclic group of order $q$ |
| $P$ | The generator of group $G$ |
| $e$ | Bilinear map |
| $H(\cdot)$ | Hash function |
| Sign $(\cdot)$ | Signature function |
| param | The system public parameters |
| $Pk/Sk$ | The public/private key of entity or KGC |
| $U(V)$ | The entity of domain |
| ID | Unique identification |
| PID | The anonymous identity of entity |
| KGC | The key generation center of domain |
| BAS | The blockchain agent server of domain |

*3.2. Complexity Assumptions.* $G$ is an additive cyclic group generated by $P$, and its order is $q$. The difficult assumption of the ECCDH states that, for randomly chosen $a, b \in Z_q^*$, given $(P, aP, bP)$, it is not feasible to compute $abP$ for any polynomial-time algorithm with nonnegligible probability.

## 4. Certificateless Signature and Security Model

In this section, we briefly introduce the generic construction of the certificateless signature, security model, and our anonymous certificateless short signature scheme.

*4.1. Generic Construction of Certificateless Signature.* CL-PKS scheme involves three entities, namely, KGC, user/signer, and verifier. Generally, it consists of the following PPT algorithms: Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Secret-Key, Set-Public-Key, Sign, and Verify.

Setup: KGC takes the security parameters as the input and returns the master key $s$ and the system parameter param.

Partial-Private-Key-Extract: KGC takes the system parameters param, $s$, and user $ID$ as inputs. Then, KGC generates the partial private key $D_{\mathrm{ID}}$ and sends it to the user through a secure channel.

Set-Secret-Value: the user randomly chooses a secret value $r_{\mathrm{ID}}$.

Set-Secret-Key: the user takes the partial private key $D_{ID}$ and its secret value $r_{\mathrm{ID}}$ as inputs and then returns its complete private key $Sk_{\mathrm{ID}}$.

Set-Public- Key: the user takes param and its complete private key $Sk_{\mathrm{ID}}$ as inputs and returns its public key $Pk_{\mathrm{ID}}$.

Sign: the signer/user takes param, message $m$, and its complete private key $Sk_{\mathrm{ID}}$ as inputs. Then, it generates a signature $\sigma$ of the message $m$.

Verify: the verifier takes param, public key $Pk_{\mathrm{ID}}$, message $m$, user's $ID$, and signature $\sigma$ as inputs and returns *True* or *False* to indicate the validity of the signature.

*4.2. Security Model.* Al-Riyami and Paterson [4] proposed the concept of the certificateless public key signature (CL-PKS) and defined the first security model. In their model, the security evaluation of CL-PKS considers two types of adversaries, namely, type I adversary and type II adversary. Type I adversary is an external attacker who can replace any legitimate public key with the public key chosen by it. This adversary simulates the ability of the adversary to deceive the user to verify the signature with the public key chosen by it. Type I adversaries should not hold the partial private key of the user. While, the Type II adversary models a malicious KGC, which holds the secret master key. Thus, it means that it can obtain the partial private key of any user. However, the type II adversary cannot replace the public key of the target user. After that, some researchers studied the formal security model definition of the CL-PKS schemes. Especially, according to the attack ability of potential adversaries in the certificateless signature, Huang et al. [28] divided the adversaries into three new types adversaries for the first time, that is, normal adversary, strong adversary, and super adversary (in the order of attack power from low to high). By combining the type I adversary and type II adversary previously defined in the CL-PKS, the authors defined the security of the CL-PKS under different adversary scenarios. Here, we adopt the formal security model introduced by Huang et al. [28] and prove that our certificateless short signature scheme satisfies the existential unforgeability security for the super adversary under the random oracle model. Next, we show the detailed description of the security model. It is formalized through two games between the challengers and a super type I or type II adversary.

*4.2.1. Game I.* A challenger $\mathscr{B}_I$ runs the algorithm, CL.Setup, to generate the master key $s$ and the public parameters param. Then, it sends the param to the super type I adversary $\mathscr{A}_I$ and keeps master key $s$ secret. After that, $\mathscr{A}_I$ makes polynomial times queries for the following oracles to $\mathscr{B}_I$.

CreateUser: for some query on an identity ID, if this ID has already been created, $\mathscr{B}_I$ returns $(\mathrm{ID}, P_{\mathrm{ID}}, Pk_{\mathrm{ID}})$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ calls the algorithms Cl.Partial-Private-Key-Extract, CL.Set-User-Value, Set-User-Key, and Set-Public-Key, respectively, to obtain $(\mathrm{ID}, P_{\mathrm{ID}}, \mathrm{PID}_{\mathrm{ID}}, D_{\mathrm{ID}}, r_{\mathrm{ID}}, Pk_{\mathrm{ID}})$ and returns $(\mathrm{ID}, P_{\mathrm{ID}}, Pk_{\mathrm{ID}})$ to $\mathscr{A}_I$.

Partial-Private-Key-Extract: when receiving this query on $(\mathrm{ID}, R_{\mathrm{ID}})$, $\mathscr{B}_I$ returns the partial key $D_{\mathrm{ID}}$ to $\mathscr{A}_I$.

Public-Key-Replace: when receiving this query on $(\mathrm{ID}, r_{\mathrm{ID}}', Pk_{\mathrm{ID}}')$, $\mathscr{B}_I$ uses $Pk_{\mathrm{ID}}'$ to replace the original public key $Pk_{\mathrm{ID}}$.

Secret-Value-Extract: when receiving this query on ID, $\mathscr{B}_I$ returns $r_{\mathrm{ID}}$ to $\mathscr{A}_I$. If Public-Key-Replace query has been issued without providing a corresponding $r_{\mathrm{ID}}'$, we require that it is not permitted to query Secret-Value-Extract with ID.

Super-Sign: when receiving this query on $(\mathrm{ID}, \mathrm{PID}_{\mathrm{ID}}, m)$, $\mathscr{B}_I$ calls the algorithm, CL.Sign, and returns a signature $\sigma$ that satisfies CL.Verify $(\mathrm{param}, \mathrm{PID}_{\mathrm{ID}}, Pk_{\mathrm{ID}}, m, \sigma) = \mathrm{true}$ to $\mathscr{A}_I$. Note that $Pk_{\mathrm{ID}}$ is the latest public key.

After making all the queries, $\mathscr{A}_I$ submits a forged signature $\sigma^*$ on the message $m^*$ as the target identity $\mathrm{ID}^*$. $\mathscr{A}_I$ is said to succeed in Game I if the following conditions are satisfied:

(1) CL.Verify $(\mathrm{ID}^*, \mathrm{param}, \mathrm{PID}_{\mathrm{ID}^*}, Pk_{\mathrm{ID}^*}, m^*, \sigma^*) = \mathrm{true}$

(2) $\mathscr{A}_I$ did not make Super-Sign query with $(\mathrm{ID}^*, \mathrm{PID}_{\mathrm{ID}^*}, m^*)$

(3) $\mathscr{A}_I$ did not make Partial-Private-Key-Extract query with $(\mathrm{ID}^*, R_{\mathrm{ID}^*})$

*4.2.2. Game II.* This game is executed between a super type II adversary $\mathscr{A}_{II}$ and a challenger $\mathscr{B}_{II}$. This game proceeds as follows: a challenger $\mathscr{B}_{II}$ runs the algorithm CL.Setup to generate the master key $s$ and the public parameters param. Then, it sends param to the super type II adversary $\mathscr{A}_{II}$ and keeps master key $s$ secret. After that, $\mathscr{A}_{II}$ issues CreateUser, Partial-Private-Key-Extract, Public-Key-Replace, Secret-Value-Extract, and Super-Sign oracles as described in the Game I, and $\mathscr{B}_{II}$ makes the corresponding responses in the same way as $\mathscr{B}_I$ in the Game I.

After making all the queries, $\mathscr{A}_{II}$ submits a forged signature $\sigma^*$ on the message $m^*$ as the target identity $\mathrm{ID}^*$. $\mathscr{A}_{II}$ is said to succeed in Game II if the following conditions are satisfied:

(1) CL.Verify$(\mathrm{ID}^*, \mathrm{param}, \mathrm{PID}_{\mathrm{ID}^*}, Pk_{\mathrm{ID}^*}, m^*, \sigma^*) = \mathrm{true}$

(2) $\mathscr{A}_{II}$ did not make Super-Sign query with $(\mathrm{ID}^*, \mathrm{PID}_{\mathrm{ID}^*}, m^*)$

(3) $\mathscr{A}_{II}$ did not make Partial-Private-Key-Extract query with $(\mathrm{ID}^*, R_{\mathrm{ID}^*})$

(4) $\mathscr{A}_{II}$ did not make Public-Key-Replace query with $(\mathrm{ID}^*, r_{\mathrm{ID}^*}', Pk_{\mathrm{ID}^*}')$

*Definition 1.* A CL-PKS scheme is existentially unforgeable against adaptively chosen message attacks (EUF-CMA), if for any polynomial-time super adversary (super type I or super type II), the advantage of super adversary is negligible.

*4.3. Our Certificateless Signature Scheme.* In this section, we embed the anonymity into the generic certificateless signature scheme and give a certificateless short signature scheme with anonymity. Here is our certificateless scheme.

CL.Setup $(1^k)$:

(1) Select an elliptic curve $E$: $y^3 = x^2 + ax + b$, which is defined in a prime field $F_p$. Generate $(G_1, G_2, e)$, where $G_1$ is an additive group of order $q$, whose generator is $P$; $G_2$ is a multiplicative group of order $q$; bilinear pair $e$: $G_1 \times G_1 \longrightarrow G_2$.

(2) Randomly choose a number $s \in Z_q^*$ as the master key. Then, compute the master public key $P_{\text{pub}} = sP$.

(3) Select three collision resistance hash functions $H_1, H_2$: $\{0,1\}^* \longrightarrow Z_q^*$, $H_3$: $\{0,1\}^* \longrightarrow G_1$.

(4) Output the system parameter $\text{param} = \{G_1, G_2, e, q, P, P_{\text{pub}}, H_1, H_2, H_3\}$.

CL.Set-User-Value $(\text{ID}_U, P_{\text{pub}})$:

(1) Randomly choose $r_U \in Z_q^*$ and compute $R_U = r_U P$

(2) Output $(R_U, r_U)$

CL.Extract-Partial-Key$(\text{ID}_U, P_{\text{pub}}, R_U, s)$:

(1) Randomly select $k \in Z_q^*$ and compute $P_U = kP$, $\text{PID}_U = H_1(\text{ID}_U, P_U, R_U)P$, $D_U = s\text{PID}_U$.

(2) Let $Pk_U = R_U$ and output $(\text{PID}_U, D_U)$.

CL.Set-User-Key $(\text{ID}_U, P_{\text{pub}}, \text{PID}_U, D_U, r_U)$:

(1) Save $\text{PID}_U$ as anonymous identity, and let $Sk_U = \langle D_U, r_U \rangle$

CL.Set-Public-Key $(\text{ID}_U, P_{\text{pub}}, \text{PID}_U, R_U)$:

(1) Let $Pk_U = R_U$

CL.Sign $(\text{param}, \text{PID}_U, Sk_U, m)$:

(1) Compute $h_2 = H_2(\text{PID}_U, P_{\text{pub}}, Pk_U, m)$, $h_3 = H_3(\text{PID}_U, P_{\text{pub}}, Pk_U, m)$

(2) Generate signature $\sigma = (h_2 D_U + r_U h_3)$

CL.Verify $(\text{param}, \text{PID}_U, Pk_U, m)$:

(1) Compute $h_2 = H_2(\text{PID}_U, P_{\text{pub}}, Pk_U, m)$, $h_3 = H_3(\text{PID}_U, P_{\text{pub}}, Pk_U, m)$

(2) Verify $e(\sigma, P) = e(h_2 \text{PID}_U, P_{\text{pub}}) e(h_3, Pk_U)$, if it holds, the signature $\sigma$ is accepted

Correctness: if the anonymous identity is $\text{PID}_U$, public key $Pk_U$ and signature $\sigma$ are generated according to the above signature scheme. The following equation ensures the correctness of the signature scheme, that is,

$$\begin{aligned} e(\sigma, P) &= e(h_2 D_U + r_U h_3, P) \\ &= e(h_2 D_U, P) e(r_U h_3, P) \\ &= e(h_2 \text{PID}_U, P_{\text{pub}}) e(h_3, Pk_U). \end{aligned} \tag{3}$$

## 5. Our CL-BASA Scheme

In this section, we will introduce our authentication and key agreement scheme CL-BASA for IIot devices cross-domain communication and describe its main structure.

*5.1. The Layered Design of CL-BASA Scheme.* Here, we employ the layered architecture proposed in [9]. The difference is that considering that the servers in the agent layer are still the nodes in domain, we put the entity layer and the agent layer in [9] together as the entity layer. Our layered architecture is as follows:

As shown in Figure 1, in our CL-BASA scheme, IIoT devices, key generation center (KGC), and blockchain agent server (BAS) are all in the entity layer. The blockchain layer is used as a public security platform for sharing domain-specific information, which includes domain identifier, its binding value is composed of the uniform resource identifier (URI), and the hash value is calculated on the actual domain-specific data. The URI points to the actual files stored in the storage layer. The following is a brief introduction to the functions of each layer:

The entity layer includes IIoT devices, KGC, and BAS. IIoT devices are manufacturing facilities that are responsible for specific tasks. KGC generates the anonymous identity and the partial private key for each device in its domain and cooperates with BAS to update the information of the device (including anonymous identity and public key) to the consortium blockchain; $\text{BAS}^A$ and $\text{BAS}^B$ represent blockchain agent servers (such as edge servers) in domain A and domain B, respectively. Their functions include the following: updating the devices information on the consortium blockchain in cooperation with KGC and responding to the requests for domain-specific information of other domains.

The functions of the blockchain in our scheme are the same with [9], so we make a brief introduction. For a detailed introduction, please refer to [9]. The blockchain layer represents the consortium blockchain used underneath. It is a globally distributed ledger, which is composed of blocks encapsulating transactions. These blocks carry domain-specific information related to different management domains. This information is shared between each domain and used for cross-domain authentication. The global ledger is maintained by a preselected set of nodes, and each node represents the BAS of a management domain. The format of domain-specific information written into the ledger is shown in Figure 2.

$\text{ID}_{\text{domain}}$ represents the unique identifier of a domain, which is used to distinguish other domains. The URI here points to a file hosted on the cloud service, which stores the details of domain-specific information. The hash value is calculated according to the actual file of specific-domain information. It is used to verify the integrity of the actual specific-domain information file, for preventing it from being modified by malicious adversaries or cloud service providers.

What the storage layer stores are the files that the URI in the blockchain layer points to and is hosted in the cloud server outside the blockchain. The file contains the following information: domain name, domain master public key, domain system parameters, anonymous identity, and public key list of the domain entities.

*5.2. Our CL-BASA Scheme.* In this section, based on our anonymous certificateless short signature designed in Section 4 and ECDHE key exchange mechanism, we show a

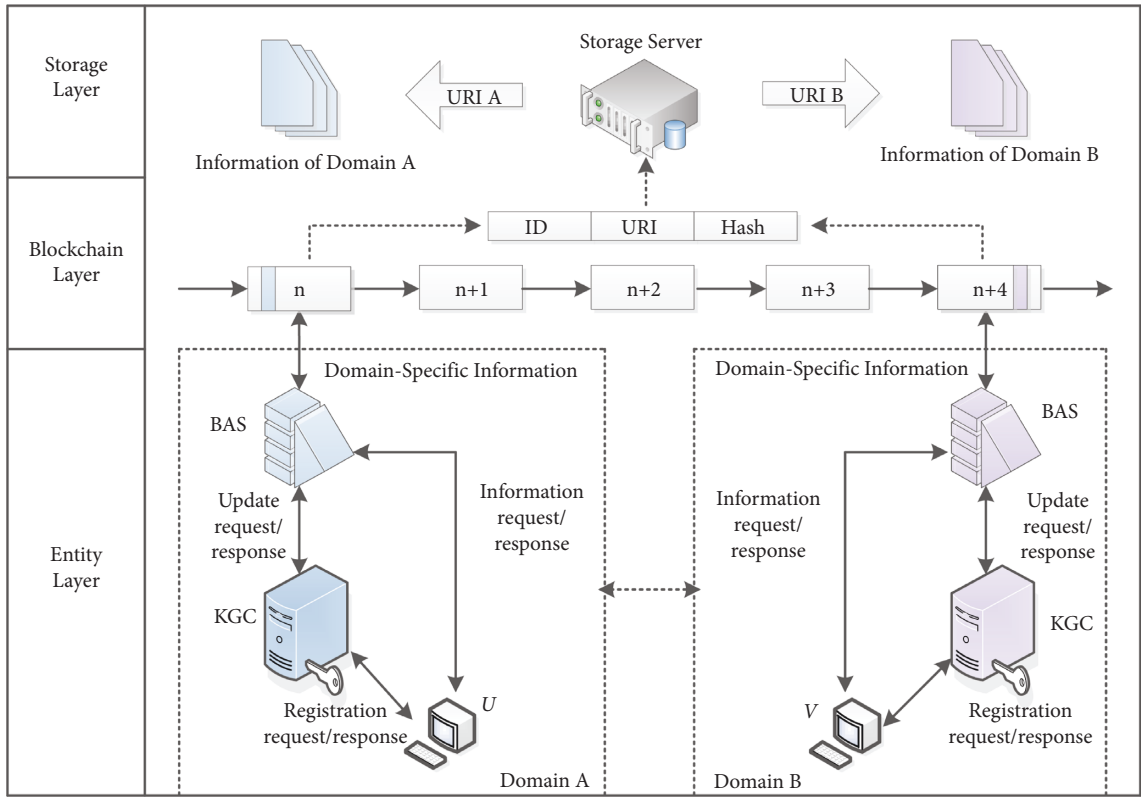


Figure 1: Layered architecture of our CL-BASA scheme.

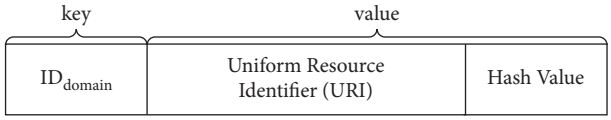


Figure 2: Data fields indicating domain-specific information encapsulated into transaction [9].

two-stage cross-domain authentication and key agreement scheme CL-BASA with the aid of blockchain. The proposed scheme is shown in Figure 3. $U$ and $V$ represent two cross-domain IIoT entities; $KGC^A$ represents the key generation centers of domain $A$ ($KGC^B$ in domain $B$, respectively); $BAS^A$ represents blockchain agent servers (such as edge servers) in domain $A$ ($BAS^B$ in domain $B$, respectively).

Our CL-BASA scheme consists of two stages, namely, registration stage and authentication and key agreement stage. These two stages are described in detail as follows.

*5.2.1. Registration Stage.* Devices in each domain perform this phase via a secure channel when they are deployed. Then, the user and KGC perform the key generation process of our short signature scheme (see Figure 4) and generate the anonymous identity and public/private key pair for each user. During this process, with the cooperation of KGC, BAS updates the user's anonymous identity and public key on the consortium blockchain.

Taking user $U$ as an example, first, $U$ randomly generates the public key $R_U$. Then, $U$ sends $(U, R_U)$ to $KGC^A$ through a secure channel to request registration. $KGC^A$ generates $P_U$ randomly after receiving $(U, R_U)$, generates $U's$ anonymous identity $PID_U$ and partial private key $D_U$, and saves $R_U$ as the public key $Pk_U$ of $U$. $KGC^A$ sends an update request to $BAS^A$ and receives an update response. After $BAS^A$ updates the information of user $U$ in the blockchain, $KGC^A$ sends $(PID_U, D_U)$ to user $U$ through a secure channel. After receiving $(PID_U, D_U)$, $U$ saves $PID_U$ as its anonymous identity and generates the complete private key $Sk_U$.

Anonymous identity management mechanism: taking user $U$ as an example. In our proposed scheme, the anonymous identity of user $U$ is maintained and controlled by $KGC^A$. During user registration, $KGC^A$ sets a time window for each user's anonymous identity. The expire-time of the user's anonymous identity is the current time plus the corresponding time window. When the anonymous identity expires, $KGC^A$ sends a reregistration request to $U$ (step 0 in Figure 3). After receiving the reregistration request sent by $KGC^A$, $U$ executes the registration phase again (steps 1–4 in Figure 3). $KGC^A$ selects the random $P_U$ after receiving the registration request of $U$ and calculates the new anonymous identity and some partial private key for user $U$. Then, after cooperating with $BAS^A$ to update the information of $U$ in the blockchain, $KGC^A$ sends the anonymous identity and the partial private key of $U$ to user $U$, and $KGC^A$ sets an expiration time for the new anonymous identity of $U$. $P_U$ reserved here is to facilitate the maintenance and management of $U's$ anonymous identity by $KGC^A$, which can be regarded as the coding of the expire time of user $U's$ anonymous identity on the elliptic curve $E$.

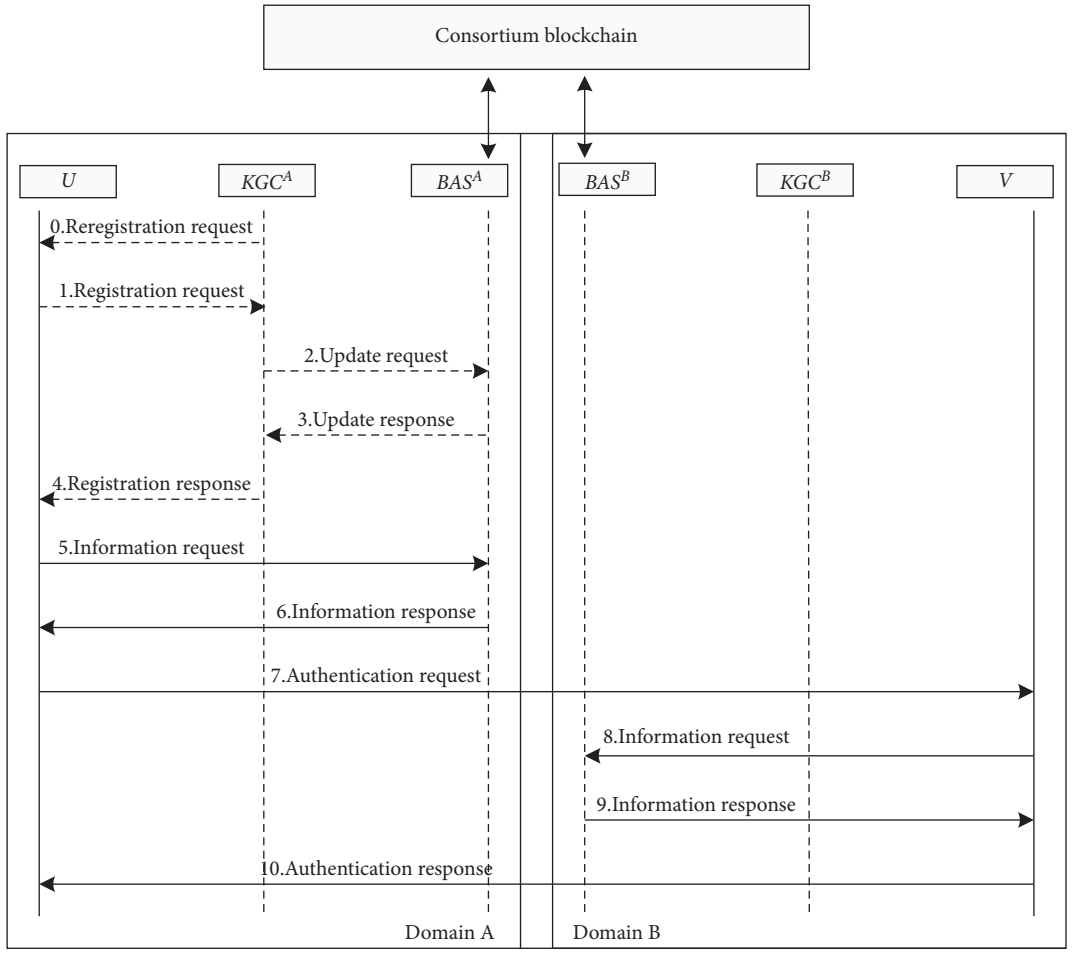


Figure 3: The authentication process of our CL-BASA scheme.

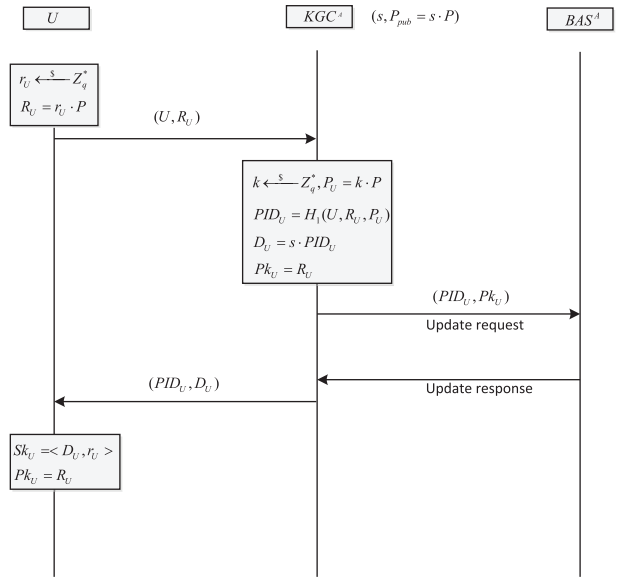


Figure 4: The registration process of our CL-BASA.

*5.2.2. Authentication and Key Agreement Stage.* In this section, we introduce the authentication and key agreement process of our scheme in detail. We assume that the user's anonymous identity is valid. Otherwise, the user re-executes the registration phase. When user $U$ in domain A needs to communicate with user $V$ in domain B, user $U$ first interacts

with $BAS^A$ to obtain the domain-specific information of domain B and the public information of user $V$, that is, steps 5 and 6 in Figure 3.

In our authentication mechanism, the certificateless signature of Section 4 is used. When requesting authentication, an entity needs to provide the following information: the information of the signature private key corresponding to the claimed public key; the authenticity of identity and public key. The former can be reflected by using the claimed public key to verify the validity of the signature, while the latter is realized through the domain-specific information of each domain shared in the consortium blockchain.

Assuming the initiator of authentication is the user $U$, the mutual authentication of $U$ and $V$ needs to be completed before cross-domain communication. The mutual authentication process is shown in Figure 5. The format of Token is as follows:

$$\begin{aligned} \text{Token}_{UV} &= N_U \| \text{ID}_{\text{domain}A} \| \text{PID}_U \| \text{Text}_U \| \text{Sign}_{Sk_U}\left(N_U \| \text{ID}_{\text{domain}A} \| \text{PID}_U \| \text{Text}_U\right), \\ \text{Token}_{VU} &= N_V \| \text{ID}_{\text{domain}B} \| \text{PID}_V \| \text{Text}_V \| \text{Sign}_{Sk_V}\left(N_V \| \text{ID}_{\text{domain}B} \| \text{PID}_V \| \text{Text}_V\right), \end{aligned} \tag{4}$$

where $N_U$ is the nonrepeated random number randomly selected by $U$; $\text{Sign}_{Sk_U}(X)$ is the signature of message $X$ by $U$ through the signature private key; $\text{Text}_U$ is an extended field.

Next, we show the authentication and key agreement process. During the authentication process, we utilize the Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) key exchange mechanism to negotiate the session key, and the ECDHE key exchange mechanism is shown in Figure 6. See Figure 7 for the detailed authentication and key agreement process.

We assume that the authentication process is initiated by $U$ in domain A. If the anonymous identity of $U$ expires, $U$ will re-execute the registration phase. Otherwise, $U$ will send a request $(\text{PID}_U, \text{reqinf}_A)$ to $BAS^A$ through a secure channel to obtain the system parameters of domain $B$ and the anonymous identity and public key information of $V$. Here, $\text{ID}_{\text{domain}B}$ represents the unique identifier of domain $B$, $\text{descrip}_V$ indicates the function of the device $V$ in domain $B$, and $TS_A$ represents the timestamp of domain $A$. After receiving the information request, $BAS^A$ will verify the validation of $TS_A$ and $\text{PID}_U$. If they are valid, $BAS^A$ will search the public information of domain $B$ in the consortium chain and send $(\text{ID}_{\text{BAS}^A}, \text{respinf}_A)$ to $U$ through a secure channel. $\text{respinf}_A$ contains the system parameters of domain $B$, the anonymous identity and public key information of $V$. Then, after receiving the information response, $U$ randomly selects the nonrepeated random number $N_U$ and $r_U'$ and then calculates $Pk_U' = r_U'P$ as the ephemeral public key. $U$ generates the message $m_U = N_U \| \text{ID}_{\text{domain}A} \| \text{PID}_U \| Pk_U' \|$ and uses the private key $Sk_U$ to generate the signature $\sigma_U$ of message $m_U$ by the signature algorithm. $U$ sends $(m_U, \sigma_U)$ to $V$ in domain $B$.

After receiving the message $(m_U, \sigma_U)$, according to the plaintext information in $m_U$, $V$ will send $m_U$ to $BAS^B$ through a secure channel to request the system parameters of domain A and the public key information of $U$ and make a request for verifying the anonymous identity of $U$. If $\text{PID}_U$ is valid, $BAS^B$ returns the verification result, the system parameters of the domain A and the public key of $U$ to $V$ through a secure channel. This process is similar to the communication between $U$ and $BAS^A$. After receiving the response from $BAS^B$, $V$ uses $U's$ public key to verify the signature $\sigma_U$ according to the signature verification algorithm. If the signature $\sigma_U$ is valid, $V$ accepts the message $(m_U, \sigma_U)$. Then, $V$ randomly selects the nonrepeated random number $N_V$ and $r_V'$ and calculates the ephemeral public key $Pk_V' = r_V'P$. $V$ generates the message $m_V = N_V \| \text{ID}_{\text{domain}}B \| \text{PID}_V \| Pk_V'$ and uses the private key $Sk_V$ to generate the signature $\sigma_V$ of message $m_V$ by the signature algorithm. Then, $V$ sends $(m_V, \sigma_V)$ to $U$ and calculates the session key $Sk_{UV} = r_V'Pk_U' = r_V'r_U'P$.

After receiving the message $(m_V, \sigma_V)$, $U$ verifies the authenticity of $V's$ anonymous identity. If the verification passes, $U$ uses $V's$ public key to verify the signature $\sigma_V$ according to the signature verification algorithm. If the signature $\sigma_V$ is valid, $U$ accepts the message $(m_V, \sigma_V)$. Then, $U$ calculates the session key $Sk_{UV} = r_U'Pk_V' = r_U'r_V'P$.

## 6. Security Analysis

We use certificateless signature to realize the mutual authentication of entities. We prove that our certificateless public-key signature scheme satisfies EUF-CMA. Appendix A can be seen for the detailed security proof. In addition, the tamper-proof feature of blockchain provides an effective way to ensure the authenticity of anonymous identities and public keys. As for the establishment of the session key, we use the ECDHE key agreement mechanism, which satisfies PFS. The combination of the certificateless signature and ECDHE can resist man-in-the-middle attacks. The privacy security of the entity identity depends on the hash function, which generates the anonymous identity.

## 7. Security Verification

In order to further illustrate the security of our CL-BASA, we make security verification by the formal analysis tool, Tamarin [29]. Tamarin is a powerful tool for symbolic modeling and analysis of security protocols. It supports a variety of algebraic properties, including encryption and decryption operations, operation combination laws, Diffie-Hellman power operations, and bilinear operations, and has

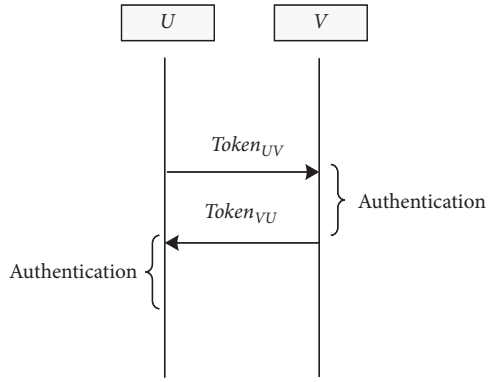


Figure 5: Mutual authentication model.

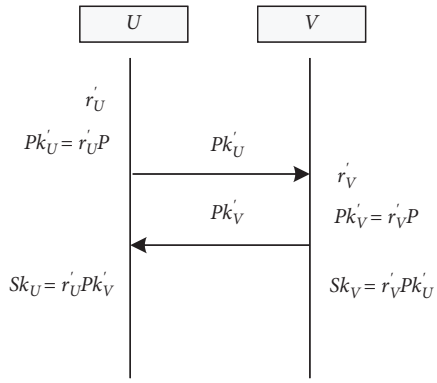


Figure 6: ECDHE key exchange mechanism.

built-in support for equational theories such as the one modeling Diffie-Hellman Key exchanges. For more information, please refer to references [29, 30]. The formal analysis results of our scheme can be seen in Figure 8.

In the authentication and key agreement phase, we take confidentiality, weak agreement, perfect forward security, noninjective agreement, and injective agreement as the security goals for the formal verification. These security attributes have been verified. However, it should also be noted that we simplify our certificateless signature $\sigma = h_2 D_U + r_U h_3$ to $\sigma = r_U \cdot h_3$ in our formal verification. The reason why we make this modification is that Tamarin tool cannot effectively handle the addition operation, which is also clearly pointed out in [31]. In addition, it depends on Tamarin's improvement for more accurate modeling.

## 8. Performance Evaluation

We theoretically compare our CL-BASA scheme with BASA [9], $A^2$Chain [22], $ES^3A$ [32], CPAL [33], and LCCH [34]in terms of communication overhead, computational overhead, and storage overhead. For comparison purposes, we use the same cryptographic parameter settings, namely, the prime256v1 elliptic curve.

*8.1. Communication Overhead.* We evaluate the communication overhead of these schemes by accumulating all key payloads during the authentication process between the IoT device and the server. The size of entity ID, signature, timestamp, random number, and cryptographic key is denoted as $S_{\mathrm{ID}}$, $S_{\mathrm{Sig}}$, $S_{TS}$, $S_N$, and $S_{\mathrm{Key}}$. The values of $S_{\mathrm{ID}}$, $S_{TS}$, and $S_N$ are 32, 4, and 4 bytes, respectively. While, the values of $S_{\mathrm{Sig}}$ and $S_{\mathrm{Key}}$ should be determined according to the specific scheme.

In our evaluation, since the same elliptic curve is considered, the domain public parameter param is not considered in the message payload. In addition, we assume that the field descrip occupies 8 bytes. In our scheme, $U$ sends a 76-byte information request to $\mathrm{BAS}^A$ to obtain the anonymous identity and public key of $V$. $\mathrm{BAS}^A$ returns a 140-byte information response. $U$ sends a 160-byte authentication request to $V$. After receiving this request, $V$ sends a 140-byte information request to $\mathrm{BAS}^B$ for verifying the anonymous identity of $U$ and obtaining the public key of $U$. $\mathrm{BAS}^B$ returns a 140 bytes information response. Then, after verifying the signature of $U$, $V$ sends a 160 bytes authentication response to $U$. The communication bandwidth cost for authentication and key agreement achieves 816 bytes. Therefore, compared with BASA [9], $A^2$chain [22], $ES^3A$ [32], CPAL [33], and LCCH [34] whose communication cost is 1344 bytes, 1232 bytes, 1336 bytes, 1232 bytes, and 2016 bytes, respectively, the communication overhead of our scheme is much smaller. Here, we need to point out that the 1232 bytes of $A^2$chain [22] is the communication overhead to realize the mutual authentication.

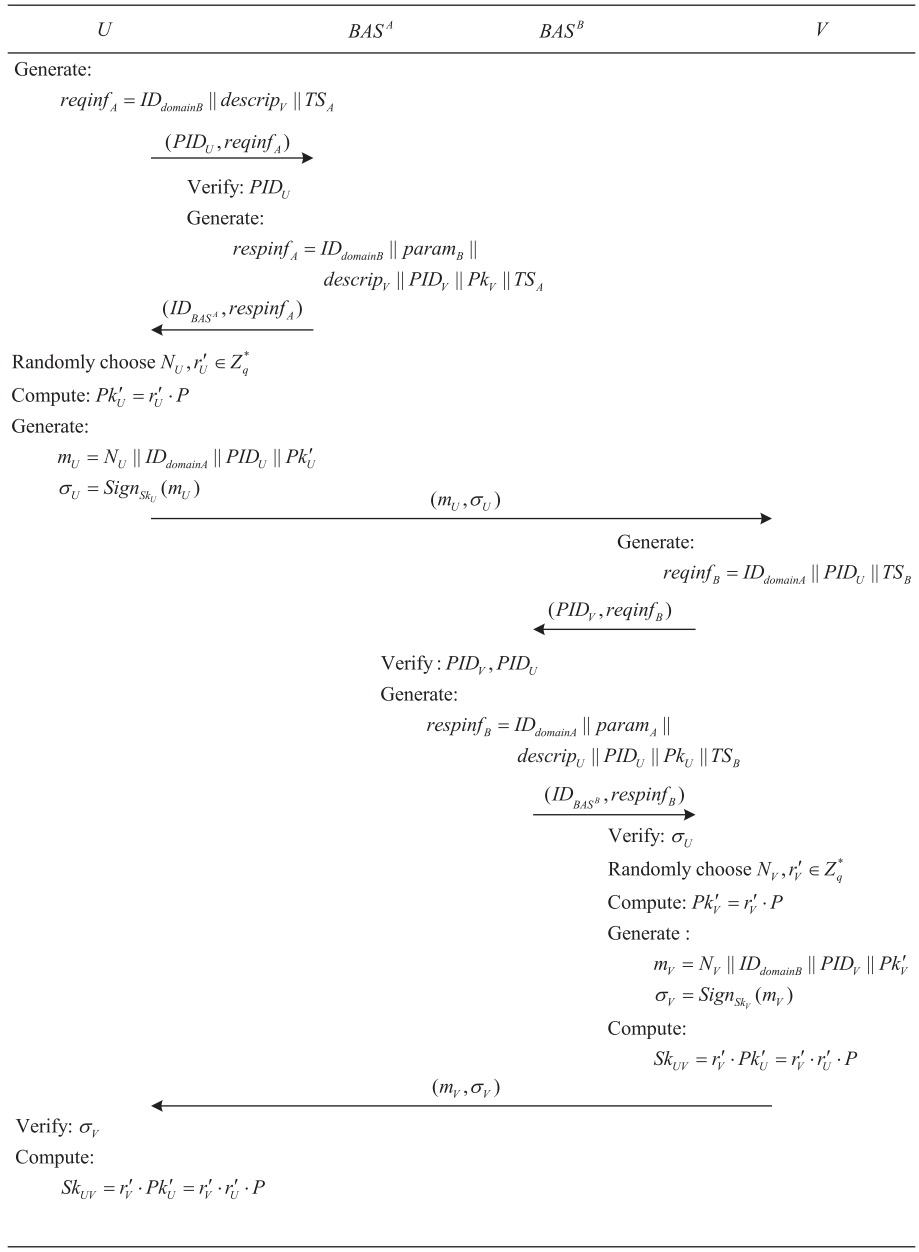


Figure 7: The authentication and key agreement process of our CL-BASA scheme.

```
==========================================================================
summary of summaries:

analyzed: certificateless_cross_domain.spthy

  secrecy (all-traces): verified (4 steps)
  weak_agreement (all-traces): verified (4 steps)
  Secret_FPS (all-traces): verified (4 steps)
  noninjective_agreement (all-traces): verified (4 steps)
  injective_agreement (all-traces): verified (4 steps)

==========================================================================
```

Figure 8: The formal analysis results of our CL-BASA.

*8.2. Computational Overhead.* In order to evaluate the computational overhead, we count cryptographic operations, including point multiplication in a group, point addition in the point group of the elliptic curve, scalar multiplication, exponential operation, bilinear pairing operation, and AES encryption/decryption operations, which are denoted by PM, PA, SM, EXP, BP, and AES, respectively. In our comparison, we no longer distinguish in detail the groups of those operations but make a simple comparison from the perspective of the number of corresponding operations. As for other operations, such as hash operation, integer addition, and multiplication, they cost little time, so they are not considered here. The result is shown in Table 2.

When the elliptic curve parameters are determined, in [35], the authors show the cost of the basic operations in relation to that of elliptic curve scalar multiplication. If *SM* is used as the benchmark, the cost of *BP* is about 150 times that of *SM* and the cost of EXP is about 36 times that of *SM*. In addition, we assume that the cost of *PA* is less than that of *SM*, and the cost of *PM* is also less than that of *SM*.

As can be seen from Table 2, since our CL-BASA scheme requires slightly more bilinear pairing operations than BASA [9] and $A^2$Chain [22], it may be inferior to the BASA [9] in computational overhead. However, compared with $ES^3A$ [32], CPAL [33], and LCCH [34], it still has great advantages. Compared with $A^2$chain [22], it is worth noting that the cryptographic operations in Table 2 are that the cost of one-way authentication. If $A^2$chain [22] realizes the mutual authentication of the IoT devices in the way, our scheme has obvious advantages in terms of the computational overhead. In addition, the reason why BASA [9] has lower computational overhead is that the bilinear pairing $e(P_1, P_{\text{pub}-s})$ (in their Algorithm 1 and Algorithm 2) is precomputed when the system is initialized and transmitted to IIoT devices, namely, the bilinear pairing $e(P_1, P_{\text{pub}-s})$ is only computed and stored as a constant value in IIoT devices. $A^2$chain [22] has a similar situation.

*8.3. Storage Overhead.* In terms of storage overhead, we only count in the most necessary data that should be stored in local device for authentication, such as identity, anonymous identity, public key, private key, and other information that needs to be stored in addition to system public parameters. In our scheme, we have the size of identity $S_{\text{ID}} = 32$ bytes, anonymous identity $S_{\text{PID}} = 32$ bytes, public key $S_{\text{CLPk}} = 32$ bytes, and private key $S_{\text{CLSk}} = 64$ bytes. The total storage overhead of our scheme, BASA [9], $A^2$chain [22], $ES^3A$ [32], CPAL [33], and LCCH [34] are shown in Table 3.

As shown in Table 3, compared with CPAL [33] and LCCH [34], our CL-BASA has slightly higher storage overhead; our scheme has the same storage overhead as BASA [9] and has an obvious advantage over $A^2$chain [22] and $ES^3A$ [32] in terms of storage overhead. $S_c$ represents the storage overhead of the bilinear pairing $e(P_1, P_{\text{pub}-s})$ stored in IIoT devices, and it occupies 32 bytes. $S_{\text{Sig}}$ is the storage overhead of the signature stored in IoT devices, which occupies 96 bytes. $S_{\text{Cert}}$ is the cost of storing certificate, which occupies 557 bytes [17].

Table 2: Comparison of the computational overhead.

| | AU + KN |
|---|---|
| BASA [9] | 18SM + 4PA + 8Exp + 4BP |
| $A^2$chain [22] | 8SM + 2PA + 4EXP + 4BP |
| $ES^3A$ [32] | 36SM + 4AES + 5Exp + 15BP |
| CPAL [33] | 27PM + 46EXP + 14BP |
| LCCH [34] | 41PM + 71EXP + 22BP |
| Our CL-BASA | 14SM + 2PA + 6BP |

Table 3: Comparison of the storage overhead.

| Scheme | Total cost | Total rough cost (bytes) |
|---|---|---|
| BASA [9] | $S_{\text{ID}} + S_{\text{PID}} + 2S_{Sk} + S_c$ | 160 |
| $A^2$chain [22] | $S_{\text{ID}} + S_{Sk} + S_c + S_{\text{Sig}}$ | 192 |
| $ES^3A$ [32] | $S_{\text{ID}} + S_{Pk} + S_{Sk} + S_{\text{Cert}}$ | 653 |
| CPAL [33] | $S_{\text{ID}} + S_{Pk} + S_{Sk}$ | 96 |
| LCCH [34] | $S_{\text{ID}} + S_{\text{PID}} + S_{Pk} + S_{Sk}$ | 128 |
| Our CL-BASA | $S_{\text{ID}} + S_{\text{PID}} + S_{Pk} + S_{\text{CLSk}}$ | 160 |

## 9. Conclusion

Considering the resource constraints of IoT devices, the research of cross-domain authentication and key agreement scheme in the IoT scenario has always been a subject of great concern. Based on BASA [9], in this article, we first design a secure and efficient certificateless short signature scheme with anonymity. Then, we propose a new secure authentication and key agreement mechanism CL-BASA for IIoT cross-domain, by combining our certificateless short signature and the ECDHE key exchange mechanism. Specifically, we avoid the inherent problem of key escrow in IBS by utilizing certificateless short signatures. In addition, with the aid of blockchain, we use the consortium jointly maintained by BASA in each domain to ensure the authenticity of entity anonymous identity and public key. Finally, we combine our certificateless short signature and the ECDHE key exchange mechanism to negotiate the session key. The evaluation demonstrates that our CL-BASA may have a slight disadvantage in storage overhead, but it has obvious advantages than competitor schemes in terms of communication overhead and computational overhead.

## Appendix

## A Security Proof of Our CL-PKS Scheme

Next, we prove that our certificateless signature (CL-PKS) scheme proposed in Section 4 satisfies the existential unforgeability security according to the security model proposed in Section 4.2. Specifically, we show that our scheme can resist super type I adversary and super type II adversary through Lemma a1 and Lemma a2. Theorem a1 shows that our CL-PKS scheme satisfies the existential unforgeability security.

**Lemma A1.** *In the random oracle model, if there is a super type I adversary $\mathcal{A}_I$, which successfully breaks our CL-PKS scheme in polynomial-time with negligible probability $\varepsilon_I$.*

*Then, there is a polynomial-time challenger $\mathscr{B}_I$ successfully solving CDH problem, and the advantage of $\mathscr{B}_I$ successfully solving CDH problem is as follows:*

$$\mathrm{Adv}^{\mathrm{CDH}}_{\mathscr{B}_I} \geq \left(1-\frac{q_{H_1}}{q}\right)^{q_{cu}}\left(1-\frac{1}{q_{cu}}\right)^{q_{cp}}\left(1-\frac{1-\xi}{q_{cu}}\right)\left(1-\frac{q_{H_2}}{q}\right)\left(1-\frac{q_{H_3}}{q}\right)\frac{1-\xi}{q_{cu}}\varepsilon_I, \tag{A.1}$$

*where $q_{H_1}$, $q_{H_2}$, $q_{H_3}$, $q_{cu}$, and $q_{cp}$ denote the number of adversary $\mathscr{A}_I$ querying to oracles $H_1$, $H_2$, $H_3$, Create-User, and Partial-Private-Key-Extract, respectively.*

Proof of Lemma A1. *Let $\mathscr{A}_I$ be a super type I adversary who can break our CL-PKS scheme, and its probability of success is $\varepsilon_I$. $\mathscr{B}_I$ acts as the simulator (challenger); we construct by the forgery of $\mathscr{A}_I$ to solve the CDH problem. Note $(X = aP, Y = bP) \in G_1 \times G_1$ is a random example of CDH problem as the input of $\mathscr{B}_I$. Next, we show how $\mathscr{A}_I$ can be used by $\mathscr{B}_I$ to solve the CDH problem. $\mathscr{B}_I$ maintains a list $L_{ID} = \{(ID, P_{ID}, PID_{ID}, D_{ID}, r_{ID}, Pk_{ID})\}$ for user ID. The anticollision hash functions are simulated as the random oracles $H_1$, $H_2$, and $H_3$. At the same time, $\mathscr{B}_I$ maintains the lists $L_1$, $L_2$, and $L_3$ to record the adversary $\mathscr{A}_I'$s queries to oracles $H_1$, $H_2$, and $H_3$. All lists are initialized to be empty. The game between $\mathscr{A}_I$ and $\mathscr{B}_I$ is as follows:*

Setup: $\mathscr{B}_I$ randomly chooses a target identity $\mathrm{ID}_t$, where $t \in \{1, 2, \ldots, q_{cu}\}$. Then, $\mathscr{B}_I$ sends system parameter $\mathrm{param} := \{G_1, G_2, e, q, P, P_{\mathrm{pub}} = X, H_1, H_2, H_3\}$ to $\mathscr{A}_I$.

Oracles: $\mathscr{A}_I$ queries the following oracles with polynomial time and receives the corresponding response from $\mathscr{B}_I$.

Create-User: when $\mathscr{A}_I$ makes a query to this oracle on ID, $\mathscr{B}_I$ first checks list $L_{\mathrm{ID}} = \{(\mathrm{ID}, P_{\mathrm{ID}}, \mathrm{PID}_{\mathrm{ID}}, D_{\mathrm{ID}}, r_{\mathrm{ID}}, Pk_{\mathrm{ID}})\}$. If there is an entry for this query, $\mathscr{B}_I$ returns $(\mathrm{ID}, P_{\mathrm{ID}}, Pk_{\mathrm{ID}})$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ creates a new entry and adds it into the list $L_{\mathrm{ID}}$ as the following. If $\mathrm{ID} \neq \mathrm{ID}_t$, $\mathscr{B}_I$ randomly chooses $k$, $r_{\mathrm{ID}}$, $t_1 \in Z_q^*$, and computes $P_{\mathrm{ID}} = kP$, $R_{\mathrm{ID}} = r_{\mathrm{ID}}P$, $\mathrm{PID}_{\mathrm{ID}} = t_1P$, $D_{\mathrm{ID}} = s(t_1P) = t_1P_{\mathrm{pub}}$, and $Pk_{\mathrm{ID}} = R_{\mathrm{ID}}$. Then, $\mathscr{B}_I$ adds the entry $(\mathrm{ID}, P_{\mathrm{ID}}, \mathrm{PID}_{\mathrm{ID}}, D_{\mathrm{ID}}, r_{\mathrm{ID}}, Pk_{\mathrm{ID}})$ into the list $L_{\mathrm{ID}}$ and returns $(\mathrm{ID}, P_{\mathrm{ID}}, Pk_{\mathrm{ID}})$. If $\mathrm{ID} = \mathrm{ID}_t$, $\mathscr{B}_I$ randomly chooses $k$, $r_{\mathrm{ID}_t} \in Z_q^*$ and computes $P_{\mathrm{ID}_t} = kP$, $R_{\mathrm{ID}_t} = r_{\mathrm{ID}_t}P$, $\mathrm{PID}_{\mathrm{ID}_t} = \perp$, $D_{\mathrm{ID}_t} = \perp$, and $Pk_{\mathrm{ID}_t} = R_{\mathrm{ID}_t}$. Then, $\mathscr{B}_I$ adds the entry $(\mathrm{ID}, P_{\mathrm{ID}_t}, \mathrm{PID}_{\mathrm{ID}_t}, D_{\mathrm{ID}_t}, r_{\mathrm{ID}_t}, Pk_{\mathrm{ID}_t})$ into the list $L_{\mathrm{ID}_t}$ and returns $(\mathrm{ID}, P_{\mathrm{ID}_t}, Pk_{\mathrm{ID}_t})$.

Oracle $H_1$: when $\mathscr{A}_I$ makes a query to this oracle on $(\mathrm{ID}, P_{\mathrm{ID}}, Pk_{\mathrm{ID}})$, we use the proof technique of Coron [28] to answer this query, namely, if the ID is not created, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ flips a coin $T \in \{0, 1\}$, $P(T = 0) = \xi$, $P(T = 1) = 1 - \xi$. If $\mathrm{ID} \neq \mathrm{ID}_t$, $\mathscr{B}_I$ checks $L_1$. If there is an entry corresponding to $(\mathrm{ID}, P_{\mathrm{ID}}, Pk_{\mathrm{ID}})$, $\mathscr{B}_I$ returns the corresponding value to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ looks up $L_{\mathrm{ID}}$, adds $(\mathrm{ID}, Pk_{\mathrm{ID}}, \mathrm{PID}_{\mathrm{ID}}, t_1, T)$ to $L_1$ and returns $\mathrm{PID}_{\mathrm{ID}}$ to $\mathscr{A}_I$. If $\mathrm{ID} = \mathrm{ID}_t$ and $T = 0$, $\mathscr{B}_I$ executes the same operations as the case of $\mathrm{ID} \neq \mathrm{ID}_t$. If $\mathrm{ID} = \mathrm{ID}_t$ and $T = 1$, let $\mathrm{PID}_{\mathrm{ID}_t} = t_1Y$, where $t_1 = H_1(\mathrm{ID}_t, P_{\mathrm{ID}_t}, Pk_{\mathrm{ID}_t})$, $\mathscr{B}_I$ adds $(\mathrm{ID}_t, P_{\mathrm{ID}_t}, Pk_{\mathrm{ID}_t}, \mathrm{PID}_{\mathrm{ID}_t}, t_1, T)$ into $L_1$ and returns to $\mathrm{PID}_{\mathrm{ID}_t}$ to $\mathscr{A}_I$. If there is an entry $(\mathrm{ID}_t, P_{\mathrm{ID}_t}), Pk_{\mathrm{ID}_t}, H_1(ID_t, P_{\mathrm{ID}_t}, Pk_{\mathrm{ID}_t}))$ in the $L_1$ and $t_1 \neq H_1(\mathrm{ID}_t, P_{\mathrm{ID}_t}, Pk_{\mathrm{ID}_t})$, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$ and aborts the game.

Partial-Private-Key-Extract: when $\mathscr{A}_I$ queries this oracle $(\mathrm{ID}, R_{\mathrm{ID}})$, if the ID is not created, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$. Otherwise, if $\mathrm{ID} \neq \mathrm{ID}_t$, $\mathscr{B}_I$ queries $L_{\mathrm{ID}}$ and returns $D_{\mathrm{ID}}$ to $\mathscr{A}_I$; if $\mathrm{ID} = \mathrm{ID}_t$, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$ and aborts the game.

Public-Key-Replace: when $\mathscr{A}_I$ queries this oracle $(\mathrm{ID}, r_{\mathrm{ID}}', Pk_{\mathrm{ID}}')$, if the ID is not created, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ uses $(\mathrm{ID}, r_{\mathrm{ID}}', Pk_{\mathrm{ID}}')$ to replace the original public key $(\mathrm{ID}, r_{\mathrm{ID}}, Pk_{\mathrm{ID}})$ and updates the entry $(\mathrm{ID}, P_{\mathrm{ID}}, \mathrm{PID}_{\mathrm{ID}}, D_{\mathrm{ID}}, r_{\mathrm{ID}}', Pk_{\mathrm{ID}}')$, where $r_{\mathrm{ID}}'$ may be $\perp$.

Secret-Value-Extract: when $\mathscr{A}_I$ queries this oracle on ID, if the ID is not created, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ queries $L_{\mathrm{ID}}$ and returns $r_{\mathrm{ID}}$ to $\mathscr{A}_I$. If $\mathscr{A}_I$ has issued *Public-Key-Replace* query without providing a corresponding $r_{\mathrm{ID}}'$, we require that $\mathscr{A}_I$ is not permitted to query *Secret-Value-Extract* with ID.

Oracle $H_2$: when $\mathscr{A}_I$ queries this oracle $(\mathrm{PID}_{\mathrm{ID}}, P_{\mathrm{pub}}, Pk_{\mathrm{ID}}, m)$, if the ID is not created, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ looks up list $L_2$. If there is a corresponding entry, $\mathscr{B}_I$ returns the previously defined value to $\mathscr{A}_I$. If there is no corresponding entry, $\mathscr{B}_I$ randomly selects $t_2 \in Z_q^*$ and then returns $t_2$ to $\mathscr{A}_I$ and adds $(\mathrm{PID}_{\mathrm{ID}}, P_{\mathrm{pub}}, Pk_{\mathrm{ID}}, m, H_2, t_2)$ into the list $L_2$.

Oracle $H_3$: when $\mathscr{A}_I$ queries this oracle $(\mathrm{PID}_{\mathrm{ID}}, P_{\mathrm{pub}}, Pk_{\mathrm{ID}}, m)$, if the ID is not created, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ looks up list $L_3$. If there is a corresponding entry, $\mathscr{B}_I$ returns the previously defined value to $\mathscr{A}_I$. If there is no corresponding entry, $\mathscr{B}_I$ randomly selects $t_3 \in Z_q^*$ and then returns $t_3P$ to $\mathscr{A}_I$ and adds $(\mathrm{PID}_{\mathrm{ID}}, P_{\mathrm{pub}}, Pk_{\mathrm{ID}}, m, H_3, t_3P)$ into the list $L_3$.

Super-Sign: when $\mathscr{A}_I$ queries this oracle $(\mathrm{ID}, \mathrm{PID}_{\mathrm{ID}}, m)$, if the ID is not created, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$. Otherwise, $\mathscr{B}_I$ looks up list $L_{\mathrm{ID}}$, $L_1$, $L_2$, and $L_3$ and performs the following operations:

(1) If $\mathrm{ID} \neq \mathrm{ID}_t$, $\mathscr{B}_I$ returns the signature $\sigma = t_2D_{\mathrm{ID}} + t_3Pk_{\mathrm{ID}}$ to $\mathscr{A}_I$

(2) If $\mathrm{ID} = \mathrm{ID}_t$ and $T = 0$, $\mathscr{B}_I$ returns the signature $\sigma = t_2(t_1P_{\mathrm{pub}}) + t_3Pk_{\mathrm{ID}_t}$ to $\mathscr{A}_I$

(3) If $\mathrm{ID} = \mathrm{ID}_t$ and $T = 1$, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$ and aborts the game

Forgery: through the above queries, $\mathscr{A}_I$ outputs a signature forgery $(\mathrm{ID}^*, \mathrm{PID}_{\mathrm{ID}^*}, m^*, \sigma^*)$. Assuming $\sigma^*$ is a valid signature, then $\mathscr{A}_I$ wins the game. If $\mathrm{ID} \neq \mathrm{ID}_t$, $\mathscr{B}_I$ returns $\perp$ to $\mathscr{A}_I$ and aborts the game. Otherwise, $\mathscr{B}_I$ performs the following:

(1) $\mathscr{B}_I$ checks list $L_1$; if $T = 0$, $\mathscr{B}_I$ returns $\perp$ and aborts the game.

(2) If $T = 1$, $\mathscr{B}_I$ solves the CDH problem by relying on the forgery $\sigma^*$ of $\mathscr{A}_I$. Due to $\sigma^*$ is a valid signature, the following formula holds:

$$\begin{aligned} & e(\sigma^*, P) \\ &= e(t_2^* t_1 Y, X) e(h_3^*, Pk_{\mathrm{ID}^*}) \\ &= e(t_2^* t_1 (ab)P, P) e(t_3^* Pk_{\mathrm{ID}^*}, P). \end{aligned} \tag{A.2}$$

So, we can get $\sigma^* = t_3^* Pk_{\mathrm{ID}^*} + t_2^* t_1 (ab)P$. Finally, $\mathscr{B}_I$ solves the CDH problem by relying on $\mathscr{A}_I'$s forger $\sigma^*$, that is, $(ab)P = (t_2^* t_1)^{-1} (\sigma^* - t_3^* Pk_{\mathrm{ID}^*})$.

Next, for a random example $(X = aP, Y = bP) \in G_1 \times G_1$. We analyze the probability of the challenger $\mathscr{B}_I$ outputting the correct solution of CDH problem. The conditions for challenger $\mathscr{B}_I$ success are as follows:

(1) $E_1$: $\mathscr{B}_I$ does not abort the game

(2) $E_2$: $\sigma^*$ is a valid signature on $(\mathrm{ID}^*, \mathrm{PID}_{\mathrm{ID}^*}, m^*, \sigma^*)$

The advantage of $\mathscr{B}_I$ is

$$\mathrm{Adv}_{\mathscr{B}_I}^{\mathrm{CDH}} = \Pr[E_1 \wedge E_2] = \Pr[E_1] \Pr[E_2 | E_1]. \tag{A.3}$$

Assuming that the super type I adversary $\mathscr{A}_I$ successfully breaks our CL-PKS scheme with probability $\varepsilon_I$, we can get $\Pr[E_2|E_1] = \varepsilon_I$. Next, we analyze the probability of event $E_1$ occurring. When the following events occur at the same time, challenger $\mathscr{B}_I$ will not abort the game.

(1) In the Create-User query, there is no collision during the query of $H_1$. The probability of this event occurrence is at least $(1 - q_{H_1}/q)^{q_{cu}}$.

(2) The adversary $\mathscr{A}_I$ does not query to Partial-Private-Key-Extract with $(\mathrm{ID}_t, R_{\mathrm{ID}_t})$. The probability of this event occurrence is $(1 - 1/q_{cu})^{q_{ep}}$.

(3) In the Super-Sign query, the event $\mathrm{ID} = \mathrm{ID}_t \wedge T = 1$ does not happen. The probability is $(1 - (1 - \xi)/q_{cu})$.

(4) In the Forgery query, $\mathrm{ID}^* \neq \mathrm{ID}_t \vee T = 0$ does not happen. The probability is $(1 - \xi)/q_{cu}$.

(5) Both $t_2^*$ and $t_3^*$ are found in $L_2$ and $L_3$. Due to the randomness of the random oracle, if $(\mathrm{ID}^*, \mathrm{PID}_{\mathrm{ID}^*}, m^*, \sigma^*)$ is a valid forgery, then the probability of existing $t_2^*$ and $t_3^*$ is at least $(1 - q_{H_2}/q)(1 - q_{H_3}/q)$.

So, we can get the following:

$$Pr[E_1] \geq \left(1 - \frac{q_{H_1}}{q}\right)^{q_{cu}} \left(1 - \frac{1}{q_{cu}}\right)^{q_{ep}} \left(1 - \frac{(1-\xi)}{q_{cu}}\right) \left(1 - \frac{q_{H_2}}{q}\right) \left(1 - \frac{q_{H_3}}{q}\right) \frac{(1-\xi)}{q_{cu}}. \tag{A.4}$$

Through the above analysis, we have the following:

$$\mathrm{Adv}_{\mathscr{B}_I}^{\mathrm{CDH}} \geq \left(1 - \frac{q_{H_1}}{q}\right)^{q_{cu}} \left(1 - \frac{1}{q_{cu}}\right)^{q_{ep}} \left(1 - \frac{(1-\xi)}{q_{cu}}\right) \left(1 - \frac{q_{H_2}}{q}\right) \left(1 - \frac{q_{H_3}}{q}\right) \frac{(1-\xi)}{q_{cu}} \varepsilon_I. \tag{A.5}$$

Therefore, if there is a polynomial-time super Type I adversary that breaks our CL-PKS scheme, we can find an attacker successfully solving the CDH problem. □

**Lemma A2.** *In the random oracle model, if there is a super type II adversary $\mathscr{A}_{II}$, which successfully breaks our CL-PKS scheme in polynomial-time with negligible probability $\varepsilon_{II}$. Then, there is a polynomial-time challenger $\mathscr{B}_{II}$ successfully solving the CDH problem, and the advantage of $\mathscr{B}_{II}$ successfully solving CDH problem is as follows:*

$$\mathrm{Adv}_{\mathscr{B}_{II}}^{\mathrm{CDH}} \geq \left(1 - \frac{q_{H_1}}{q}\right)^{q_{cu}} \left(1 - \frac{1}{q_{cu}}\right)^{q_{rp}} \left(1 - \frac{1}{q_{cu}}\right)^{q_{es}} \left(1 - \frac{(1-\xi)}{q_{cu}}\right) \left(1 - \frac{q_{H_2}}{q}\right) \left(1 - \frac{q_{H_3}}{q}\right) \frac{(1-\xi)}{q_{cu}} \varepsilon_{II}, \tag{A.6}$$

*where $q_{H_1}$, $q_{H_2}$, $q_{H_3}$, $q_{cu}$, and $q_{rp}$ denote the numbers of adversary $\mathscr{A}_{II}$ querying to oracle $H_1$, $H_2$, $H_3$, Create-User, Public-Key-Replace, and Secret-Value-Extract, respectively.*

Proof of Lemma A2. *Let $\mathscr{A}_{II}$ be a super type II adversary breaking our CL-PKS scheme, and its probability of success is $\varepsilon_{II}$. $\mathscr{B}_{II}$ acts as the simulator (challenger) we construct by the forgery of $\mathscr{A}_{II}$ to solve the CDH problem. Note $(X = aP, Y = bP) \in G_1 \times G_1$ is a random example of CDH problem as the input of $\mathscr{B}_{II}$. Next, we show how $\mathscr{A}_{II}$ can be used by $\mathscr{B}_{II}$ to solve the CDH problem. $\mathscr{B}_{II}$ maintains a list $L_{ID} = \{(ID, P_{ID}, PID_{ID}, D_{ID}, r_{ID}, Pk_{ID}, T)\}$ for user ID. The anticollision hash functions are simulated as the random oracles $H_1$, $H_2$, and $H_3$. At the same time, $\mathscr{B}_{II}$ maintains the*

*lists $L_1$, $L_2$, and $L_3$ to record the adversary $\mathscr{A}_{II}$'s queries to oracles $H_1$, $H_2$, and $H_3$. All lists are initialized to be empty. The game between $\mathscr{A}_{II}$ and $\mathscr{B}_{II}$ is as follows:*

Setup: $\mathscr{B}_{II}$ randomly chooses randomly chooses $s \in Z_q^*$ as the master secret key and calculates the master public key $P_{\text{pub}} = sP$. Then, $\mathscr{B}_{II}$ randomly chooses a target identity $\text{ID}_t$, where $t \in \{1, 2, \ldots, q_{cu}\}$. Then, $\mathscr{B}_{II}$ sends system parameter $\text{param} := \{G_1, G_2, e, q, P, P_{\text{pub}} = X, H_1, H_2, H_3\}$ to $\mathscr{A}_{II}$.

Oracles: $\mathscr{A}_{II}$ queries the following oracles with polynomial time and receives the corresponding response from $\mathscr{B}_{II}$.

Create-User: when $\mathscr{A}_{II}$ makes a query to this oracle with ID, we use the proof technique of Coron [28] to answer this query. $\mathscr{B}_{II}$ flips a coin $T \in \{0, 1\}$, $P(T = 0) = \xi$, $P(T = 1) = 1 - \xi$. $\mathscr{B}_{II}$ first checks list $L_{\text{ID}} = \{(\text{ID}, P_{\text{ID}}, \text{PID}_{\text{ID}}, D_{\text{ID}}, r_{\text{ID}}, Pk_{\text{ID}}, T)\}$. If there is an entry corresponding to this query, $\mathscr{B}_{II}$ returns $(\text{ID}, P_{\text{ID}}, Pk_{\text{ID}})$ to $\mathscr{A}_I I$. Otherwise, $\mathscr{B}_{II}$ creates a new entry and adds it into the list $L_{\text{ID}}$ as the following: if $\text{ID} \neq \text{ID}_t$, $\mathscr{B}_{II}$ randomly chooses $k$, $r_{\text{ID}}$, $t_1 \in Z_q^*$ and computes $P_{\text{ID}} = kP$, $R_{\text{ID}} = r_{\text{ID}}P$, $\text{PID}_{\text{ID}} = t_1 P$, $D_{\text{ID}} = s(t_1 P) = t_1 P_{\text{pub}}$, where $t_1 = H_1(\text{ID}, R_{\text{ID}}, P_{\text{ID}})$. Let $Pk_{\text{ID}} = R_{\text{ID}}$. Then, $\mathscr{B}_{II}$ adds the entry $(\text{ID}, P_{\text{ID}}, \text{PID}_{\text{ID}}, D_{\text{ID}}, r_{\text{ID}}, Pk_{\text{ID}}, T)$ into the list $L_{\text{ID}}$ and returns $(\text{ID}, P_{\text{ID}}, Pk_{\text{ID}})$ to $\mathscr{A}_{II}$. If $\text{ID} = \text{ID}_t$ and $T = 0$, $\mathscr{B}_{II}$ performs the same operations just as the case of $\text{ID} \neq \text{ID}_t$ and returns $(\text{ID}_t, P_{\text{ID}_t}, Pk_{\text{ID}_t})$ to $\mathscr{A}_{II}$. If $\text{ID} = \text{ID}_t$ and $T = 1$, $\mathscr{B}_{II}$ randomly chooses $k$, $r_{ID_t} \in Z_q^*$, and computes $(P_{\text{ID}_t}) = k_t P$, $R_{\text{ID}_t} = r_{\text{ID}_t} X$, $\text{PID}_{\text{ID}_t} = t_1 P$, $D_{\text{ID}_t} = s(t_1 P) = t_1 P_{\text{pub}}$, where $t_1 = H_1(\text{ID}_t, R_{\text{ID}_t}, P_{\text{ID}_t})$. Let $Pk_{\text{ID}_t} = R_{\text{ID}_t}$. $\mathscr{B}_{II}$ adds the entry $(ID, P_{ID_t}, PID_{ID_t}, D_{ID_t}, r_{ID_t}, Pk_{ID_t}, T)$ into the list $L_{ID_t}$ and returns $(ID, P_{ID_t}, Pk_{ID_t})$ to $\mathscr{A}_{II}$.

Oracle $H_1$: when $\mathscr{A}_{II}$ makes a query to this oracle with $(ID, P_{ID}, Pk_{ID})$, if the $ID$ is not created, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$. Otherwise, $\mathscr{B}_{II}$ checks $L_1$. If there is an entry corresponding to $(ID, P_{ID}, Pk_{ID})$, $\mathscr{B}_{II}$ returns the corresponding value to $\mathscr{A}_{II}$. Otherwise, $\mathscr{B}_{II}$ looks up $L_{ID}$ and adds $(ID, P_{ID}, Pk_{ID}, H_1, \text{PID}_{\text{ID}})$ to $L_1$ and returns $\text{PID}_{\text{ID}}$ to $\mathscr{A}_{II}$.

Partial-Private-Key-Extract: when $\mathscr{A}_{II}$ queries this oracle $(\text{ID}, R_{\text{ID}})$, if the $ID$ is not created, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$. Otherwise, $\mathscr{B}_{II}$ looks up $L_{ID}$ and returns $D_{ID}$ to $\mathscr{A}_{II}$.

Public-Key-Replace: when $\mathscr{A}_{II}$ queries this oracle $(ID, r_{ID}', Pk_{ID}')$, if the $ID$ is not created, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$. Otherwise, if $ID = ID_t$, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$ and aborts the game. If $ID \neq ID_t$, $\mathscr{B}_{II}$ uses $(ID, r_{ID}', Pk_{ID}')$ to replace the original $(ID, r_{ID}, Pk_{ID})$ and updates the entry $(ID, P_{ID}, \text{PID}_{\text{ID}}, D_{ID}, r_{ID}', Pk_{ID}', T)$, where the $r_{ID}'$ may be $\perp$.

Secret-Value-Extract: when $\mathscr{A}_{II}$ queries the oracle $ID$, if the $ID$ is not created, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$. Otherwise, if $ID \neq ID_t$, $\mathscr{B}_{II}$ queries $L_{ID}$ and returns $r_{ID}$ to $\mathscr{A}_{II}$. If $ID = ID_t$, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_I I$ and aborts the game.

Oracle $H_2$: when $\mathscr{A}_{II}$ queries this oracle $(PID_{ID}, P_{pub}, Pk_{ID}, m)$, if the $ID$ is not created, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$. Otherwise, $\mathscr{B}_{II}$ looks up list $L_2$. If there is a corresponding entry, $\mathscr{B}_{II}$ returns the previously defined value to $\mathscr{A}_{II}$. If there is no corresponding entry, $\mathscr{B}_{II}$ randomly selects $t_2 \in Z_q^*$ and then returns $t_2$ to $\mathscr{A}_{II}$ and adds $(\text{PID}_{ID}, P_{\text{pub}}, Pk_{ID}, m, H_2, t_2)$ into the list $L_2$.

Oracle $H_3$: when $\mathscr{A}_{II}$ queries this oracle $(\text{PID}_{\text{ID}}, P_{\text{pub}}, Pk_{ID}, m)$, if the $ID$ is not created, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$. Otherwise, $\mathscr{B}_{II}$ looks up list $L_3$. If there is a corresponding entry, $\mathscr{B}_{II}$ returns the previously defined value to $\mathscr{A}_{II}$. If there is no corresponding entry, $\mathscr{B}_{II}$ randomly selects $t_3 \in Z_q^*$ and then returns $t_3 Y$ to $\mathscr{A}_{II}$ and adds $(\text{PID}_{\text{ID}}, P_{\text{pub}}, Pk_{ID}, m, H_3, t_3 Y)$ into the list $L_3$.

Super-Sign: when $\mathscr{A}_{II}$ queries this oracle $(ID, \text{PID}_{\text{ID}}, m)$, if the $ID$ is not created, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$. Otherwise, $\mathscr{B}_{II}$ looks up list $L_{ID}$, $L_1$, $L_2$, and $L_3$ and performs the following operations:

(1) If $ID \neq ID_t$, $\mathscr{B}_{II}$ returns the signature $\sigma = t_2 D_{ID} + t_3 Pk_{ID}$ to $\mathscr{A}_{II}$

(2) If $ID = ID_t$ and $T = 0$, $\mathscr{B}_{II}$ returns the signature $\sigma = t_2(t_1 P_{\text{pub}}) + t_3 Pk_{ID_t}$ to $\mathscr{A}_{II}$

(3) If $ID = ID_t$ and $T = 1$, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$ and aborts the game

Forgery: through the above queries, $\mathscr{A}_{II}$ outputs a signature forgery $(ID^*, \text{PID}_{ID^*}, m^*, \sigma^*)$. Assuming $\sigma^*$ is a valid signature, $\mathscr{A}_{II}$ wins the game. If $ID \neq ID_t$, $\mathscr{B}_{II}$ returns $\perp$ to $\mathscr{A}_{II}$ and aborts the game. Otherwise, $\mathscr{B}_{II}$ performs the following:

(1) $\mathscr{B}_{II}$ checks list $L_{ID}$, if $T = 0$, $\mathscr{B}_{II}$ returns $\perp$ and aborts the game.

(2) If $T = 1$, $\mathscr{B}_{II}$ solves the CDH problem by relying on the forgery $\sigma^*$ of $\mathscr{A}_{II}$. Due to $\sigma^*$ is a valid signature, the following formula holds:

$$\begin{aligned} & e(\sigma^*, P) \\ &= e\left(t_2^* \text{PID}_{\text{ID}^*}, P_{\text{pub}}\right) e\left(t_3^* Y, r_{ID^*} X\right) \\ &= e\left(t_2^* D_{ID^*}, P\right) e\left(r_{ID^*} t_3^* (ab)P, P\right). \end{aligned} \tag{A.7}$$

So, we can get $\sigma^* = t_2^* D_{ID^*} + r_{ID^*} t_3^* (ab)P$. Finally, $\mathscr{B}_{II}$ solves the CDH problem by relying on $\mathscr{A}_{II}'s$ forger $\sigma^*$, that is, $(ab)P = (r_{ID^*} t_3^*)^{-1} (\sigma^* - t_2^* D_{ID^*})$.

Next, for a random example $(X = aP, Y = bP) \in G_1 \times G_1$. We analyze the probability of the challenger $\mathscr{B}_{II}$ outputting the correct solution of CDH problem. The conditions for challenger $\mathscr{B}_{II}$ success are as follows:

(1) $E_1$: $\mathscr{B}_{II}$ does not abort the game

(2) $E_2$: $\sigma^*$ is a valid signature on $(ID^*, \text{PID}_{\text{ID}^*}, m^*, \sigma^*)$

The advantage of $\mathscr{B}_{II}$ is as follows:

$$\text{Adv}_{\mathscr{B}_{II}}^{C\,DH} = \Pr[E_1 \wedge E_2] = \Pr[E_1] \Pr[E_2 | E_1]. \tag{A.8}$$

Assuming that the super type II adversary $\mathscr{A}_{II}$ successfully breaks our CL-PKS scheme with probability $\varepsilon_{II}$, we can get $\Pr[E_2|E_1] = \varepsilon_{II}$. Next, we analyze the probability of event $E_1$ occurring. When the following events occur at the same time, challenger $\mathscr{B}_{II}$ will not abort the game.

(1) In the Create-User query, there is no collision during the query of $H_1$. The probability of this event occurrence is at least $(1 - q_{H_1}/q)^{q_{cu}}$.

(2) The adversary $\mathscr{A}_{II}$ does not query to *Public-Key-Replace* with $(ID_t, Pk_{ID_t}')$. The probability of this event occurrence is $(1 - 1/q_{cu})^{q_{rp}}$.

(3) The adversary $\mathcal{A}_{II}$ does not query to Secret-Value-Extract with $ID_t$. The probability of this event occurrence is $(1-1/q_{cu})^{q_{es}}$.

(4) In the Super-Sign query, the event $ID = ID_t \wedge T = 1$ does not happen. The probability is $(1-(1-\xi)/q_{cu})$.

(5) In the Forgery query, $ID^* \neq ID_t \vee T = 0$ does not happen. The probability is $(1-\xi)/q_{cu}$.

(6) Both $t_2^*$ and $t_3^*$ are found in $L_2$, $L_3$. Due to the randomness of the random oracle, if $(ID^*, \mathrm{PID}_{ID^*}, m^*, \sigma^*)$ is a valid forgery, then the probability of existing $t_2^*$ and $t_3^*$ is at least $(1-q_{H_2}/q)(1-q_{H_3}/q)$.

So, we can get the following:

$$Pr[E_1] \geq \left(1-\frac{q_{H_1}}{q}\right)^{q_{cu}}\left(1-\frac{1}{q_{cu}}\right)^{q_{rp}}\left(1-\frac{1}{q_{cu}}\right)^{q_{es}}\left(1-\frac{1-\xi}{q_{cu}}\right)\left(1-\frac{q_{H_2}}{q}\right)\left(1-\frac{q_{H_3}}{q}\right)\frac{(1-\xi)}{q_{cu}}. \tag{A.9}$$

Through the above analysis, we have the following:

$$\mathrm{Adv}_{\mathcal{B}_{II}}^{\mathrm{CDH}} \geq \left(1-\frac{q_{H_1}}{q}\right)^{q_{cu}}\left(1-\frac{1}{q_{cu}}\right)^{q_{rp}}\left(1-\frac{1}{q_{cu}}\right)^{q_{es}}\left(1-\frac{(1-\xi)}{q_{cu}}\right)\left(1-\frac{q_{H_2}}{q}\right)\left(1-\frac{q_{H_3}}{q}\right)\frac{(1-\xi)}{q_{cu}}\varepsilon_{II}. \tag{A.10}$$

Therefore, if there is a polynomial-time super type II adversary that that breaks our CL-PKS scheme, we can find an attacker successfully solving the CDH problem.

From Definition 1, Lemma a1, and Lemma a2, we know that our certificateless short signature CL-PKS scheme satisfies the EUF-CMA secure, that is, the following Theorem A1 holds. □

**Theorem A1.** *Assuming that the CDH difficulty assumption is true, our proposed CL-PKS scheme satisfies EUF-CMA against any polynomial-time super adversary (type I or super type II) in the random oracle model.*

## Data Availability

There are no data supporting the results of this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors thank the students and teachers around them for their concerns. They provided a great help for the work of this article.

## References

[1] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng*, vol. 6, no. 4, pp. 239–242, 2014.

[2] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology-CRYPTO 1984. CRYPTO 1984. Lecture Notes in Computer Science*, G. R. Blakley and D. Chaum, Eds., pp. 47-53, Springer, Berlin, Germany, 1984.

[3] M. Samaniego, U. Jamsrandorj, and R. Deters, "Blockchain as a service for IoT," in Proceedings of the 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data, pp. 433-436, Chengdu, China, December 2016.

[4] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT 2003. ASIACRYPT 2003. Lecture Notes in Computer Science*, C. S. Laih, Ed., pp. 452-473, Springer, Berlin, Germany, 2003.

[5] D. Brown, R. Gallant, and S. Vanstone, "Provably secure implicit certificate schemes," in *Financial Cryptography-FC 2001. FC 2001*, P. Syverson, Ed., pp. 156-165, Springer, Berlin, Germany, 2002.

[6] L. A. Pintsov and S. A. Vanstone, "Postal revenue collection in the digital age," in *Financial Cryptography-FC 2000. FC 2000*, Y. Frankel, Ed., pp. 105–120, Springer, Berlin, Germany, 2001.

[7] Certicom Research, "Sec 4: elliptic curve qu-vanstone implicit certificate scheme (ECQV)," *Standard for Efficient Cryptography*, 2013.

[8] ZigBee Alliance, *ZigBee Smart Energy Standard. Revision 19*, San Ramon, CA, USA, 2014.

[9] S. Meng, H. Liu, L. Zhu et al., "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, 2020.

[10] K. Malasri and L. Wang, "Design and implementation of a secure wireless mote-based medical sensor network," *Sensors*, vol. 9, no. 8, pp. 6273–6297, 2009.

[11] R. Lu, X. Li, X. Liang, X. Shen, and X. Lin, "GRS: the green, reliability, and security of emerging machine to machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 28–35, 2011.

[12] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key management protocol with implicit certificates for IoT systems,"' Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems, IoT-ys@ MobiSys 2015, pp. 37-42, Florence, Italy, May 2015.

[13] C. S. Park, "A secure and efficient ECQV implicit certificate issuance protocol for the internet of things applications," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2215–2223, 2017.

[14] A. Braeken, J. J. Chin, and S. Y. Tan, "ECQV-IBI: identity-based identification with implicit certification," *Journal of*

*Information Security and Applications*, vol. 63, Article ID 103027, 2021.

[15] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications,"' Proceedings of the 2014 IEEE Wireless Communications and Networking Conference (WCNC), pp. 2728-2733, Istanbul, Turkey, April 2014.

[16] A. Teniou and B. A. Bensaber, "Efficient and dynamic elliptic curve qu-vanstone implicit certificates distribution scheme for vehicular cloud networks," *Security and Privacy*, vol. 1, pp. e11-1, 2018.

[17] Y. J. Zhang, Y. H. Luo, X. Chen et al., "A lightweight Authentication scheme based on consortium blockchain for cross-domain IoT," *Security and Communication Networks*, vol. 2022, Article ID 9686049, 15 pages, 2022.

[18] J. Heo, C. S. Hong, M. S. Choi, S. H. Ju, and Y. H. Lim, "Identity-based mutual device authentication schemes for PLC system,"' Proceedings of the 2008 IEEE International Symposium on Power Line Communications and Its Applications, pp. 47-51, Jeju, Korea, April 2008.

[19] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing,"' Proceedings of the 2009 IEEE International Conference on Cloud Computing, pp. 157-166, Berlin, Germany, December 2009.

[20] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Cross-domain secure data sharing using blockchain for industrial IoT," *Journal of Parallel and Distributed Computing*, vol. 156, pp. 176–184, 2021.

[21] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards secure and decentralized sharing of IoT data,"' Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), pp. 176-183, Atlanta, GA, USA, July 2019.

[22] X. D. Jia, N. Hu, S. Yin, Y. Zhao, C. Zhang, and X. D. Cheng, "A2 chain: a blockchain-based decentralized authentication scheme for 5G-enabled IoT," *Mobile Information Systems*, vol. 2020, Article ID 7602384, 2020.

[23] S. Sun, R. Du, S. Chen, and W. Li, "Blockchain-based IoT access control system: towards security, lightweight, and cross-domain," *IEEE Access*, vol. 9, pp. 36868–36878, 2021.

[24] X. Zhou, Z. Jin, Y. Fu, H. Zhou, and L. Qin, "Short signcryption scheme for the internet of things," *Informatica*, vol. 35, pp. 521–530, 2011.

[25] I. Ullah, S. Zeadally, N. U. Amin, M. Asghar Khan, and H. Khattak, "Lightweight and provable secure cross-domain access control scheme for internet of things (IoT) based wireless body area networks (WBAN)," *Microprocessors and Microsystems*, vol. 81, Article ID 103477, 2021.

[26] B. Gong, Y. Wu, Q. Wang, Y. H. Ren, and C. Guo, "A secure and lightweight certificateless hybrid signcryption scheme for Internet of Things," *Future Generation Computer Systems*, vol. 127, pp. 23–30, 2022.

[27] Y. Zheng, "Digital signcryption or how to achieve cost(-signature & encryption) ≪ cost(signature) + cost(encryption)," in *Advances in Cryptology-CRYPTO 1997. CRYPTO 1997. Lecture Notes in Computer Science*, B. S. Kaliski, Ed., pp. 165-179, Springer, Berlin, Germany, 1997.

[28] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signatures: new schemes and security models," *The Computer Journal*, vol. 55, no. 4, pp. 457–474, 2012.

[29] B. Schmidt, S. Meier, C. Cremers, and D. A. Basin, "Automated analysis of diffie-hellman protocols and advanced security properties,"' Proceedings of the 2012 IEEE 25th Computer Security Foundations Symposium, pp. 78-94, Cambridge, MA, USA, June 2012.

[30] The Tamarin Team, "Tamarin prover manual," 2022, https://tamarin-prover.github.io/manual/.

[31] D. Basin, L. Hirschi, and R. Sasse, "Symbolic analysis of identity-based protocols," in *Foundations of Security, Protocols, and Equational Reasoning. Lecture Notes in Computer Science*, J. Guttman, C. Landwehr, J. Meseguer, and D. Pavlovic, Eds., pp. 112-134, Springer, Berlin, Germany, 2019.

[32] J. Ni, X. Lin, and X. S. Shen, "Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 644–657, 2018.

[33] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: a conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet of Things Journal*, vol. 1, pp. 46–57, 2014.

[34] J. K. Liu, C.-K. Chu, S. S. M. Chow, X. Huang, M. H. Au, and J. Zhou, "Time-bound anonymous authentication for roaming networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 178–189, 2015.

[35] M.-H. Le and S. O. Hwang, "Certificate-based signcryption scheme without pairing: directly verifying signcrypted messages using a public key," *ETRI Journal*, vol. 38, pp. 724–734, 2016.