# Cost-Effective and Anonymous Access Control for Wireless Body Area Networks

Fagen Li, *Member, IEEE*, Yanan Han, and Chunhua Jin

***Abstract*—Wireless body area networks (WBANs) are expected to play an important role in monitoring the health information and creating a smart, reliable, and ubiquitous healthcare system. Only authorized users can access the network since the collected data by the WBANs are used to diagnosed and treated. However, it is still a challenging task to design a cost-effective and secure access control scheme because of inherent characteristics of the WBANs, such as open medium channel, limited resources of sensor nodes, and the absence of fixed infrastructure. In this paper, we first propose a novel certificateless signcryption scheme, and then, design a cost-effective and anonymous access control scheme for the WBANs using the novel signcryption. The proposed access control scheme achieves anonymity, confidentiality, authentication, integrity, and nonrepudiation. Compared with existing three access control schemes for the WBANs, our proposed scheme has the least computational cost and total energy consumption for the controller.**

***Index Terms*—Access control, certificateless cryptography (CLC), security, signcryption, wireless body area networks (WBANs).**

## I. Introduction

WITH the rapid progress in medical sensors and wireless communication, wireless body area networks (WBANs) [1] has emerged as a new technology for creating a smart, reliable, and ubiquitous healthcare system. A typical WBAN consists of a number of implantable or wearable sensor nodes and a controller [2]. The sensor nodes are used to monitor a patient's vital signs (e.g., electrocardiogram, breathing rate, heart rate, and blood pressure) and environmental parameter (e.g., humidity, temperature, and light). There are two communication types for the WBANs. The first type is that the sensor nodes communicate with the controller. The second type is that the controller transmits the collected data to the healthcare staffs and network servers. In [3], the aforementioned two communication types are classified as intrabody communication and extrabody communication. The intrabody communication and extrabody communication are interconnected by the controller. The WBANs enhance the efficiency of healthcare since a patient does not need to visit the hospital frequently. This characteristic is especially favorable for countries that are lack of medical staffs and medical infrastructure. In addition, the clinical diagnosis and emergency medical response can be realized by using the WBANs. Therefore, the WBANs play an important role in creating a smart, reliable, and ubiquitous healthcare system. A good survey about the WBANs are given by Movassaghi *et al.* [4].

Since collected health data by the WBANs play a vital role in the medical diagnosis and treatment, only authorized users can access these data [5]. Without the access control, the health data may be abused, disseminated and even modified, which may violate the privacy of the patient and even result in a catastrophic consequence. Therefore, it is important to design an efficient access control scheme that can authorize, authenticate, and revoke a user to access the WBANs. However, it is still a challenging task to design a cost-effective and secure access control scheme because of inherent characteristics of the WBANs, such as open medium channel, limited resources of sensor nodes, and the absence of fixed infrastructure.

### A. Related Work

Since the health data stored in the WBANs play an important role in the medical diagnosis and treatment, we should solve the security issues in the WBANs before real development [5]. Recently, some secure schemes for the WBANs have been proposed from different sides. In 2013, Hu *et al.* [6] discussed how to provide a security mechanism for the extrabody communication. Their solution is to use the attribute-based encryption (ABE) [7]. However, the ABE may not be a good choice because the costly cryptographic operation is a heavy burden for resource-limited sensor nodes [5]. Lu *et al.* [8] gave a privacy preserving opportunistic framework for the WBANs. Zhao *et al.* [9] discussed the key management problem in the WBANs. In 2014, He *et al.* [10] discussed how to provide a security mechanism for the intrabody communication. They used the lightweight one-way hash chain to establish the key. Tan *et al.* [11] designed an identity-based encryption (IBE) scheme called IBE-Lite for the WBANs. He and Zeadally [12] gave an identity-based authentication protocol for the WBANs. As compared with the public key infrastructure (PKI) that uses a digital certificate to bind an identity and the corresponding public key, the identity-based cryptography (IBC) [13] does not need public key certificates. A user's public key is derived from its identity information, such as telephone numbers, email addresses, and IP addresses. The user's private key is generated by a trusted third party called private key generator (PKG). Authenticity of a public key is explicitly verified without a certificate. Therefore, the IBC eliminates the certificate management problem of the PKI, including distribution, storage, verification, and revocation. The lightweight IBC is very suitable for designing a

Manuscript received September 1, 2015; revised February 14, 2016; accepted April 20, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61073176, Grant 61272525, Grant 61302161, and Grant 61462048 and in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2013J069.

The authors are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: fagenli@uestc.edu.cn; 990756406@qq.com; 736201192@qq.com).

Digital Object Identifier 10.1109/JSYST.2016.2557850

security scheme for the WBANs.However, the lightweight IBC has a weakness called key escrow problem since the PKG knows all users' private keys. That is, the PKG can decrypt any ciphertext in an IBE scheme and forge a signature for any message in an identity-based signature scheme. Therefore, the IBC is only suitable for small networks, such as the WBANs, and is not suitable for large-scale networks, such as the Internet. That is, the IBC is a good choice for use in intrabody communication network and is not a good choice for use in extrabody communication network. In this paper, we investigate the scheme for the WBANs that controls the Internet users access to the WBANs, that is, the access control to the extrabody communication network. Therefore, total IBC cannot satisfy this goal. There are some good works conducted for access control in WBANs. In 2011, Cagalaban and Kim [14] proposed an access control scheme for the WBANs using identity-based signcryption (IBSC) [15] (hereafter, called CK). The novelty of the CK scheme [14] is the use of signcryption that can simultaneously authenticates the users and protects the query messages. Signcryption is a cryptographic technique that can obtain both the functions of public key encryption and digital signature in a logical single step, with a cost significantly lower than that required by the traditional encryption-then-signature or signature-then-encryption methods [16]. That is, a signcryption scheme can simultaneously achieve confidentiality, authentication, integrity, and nonrepudiation with a lower cost. However, there exists the key escrow weakness in [14] since it uses an IBC technique. In 2013, Hu *et al.* [17] developed a fuzzy attribute-based signcryption (FABSC) that can be used in data encryption, access control, and digital signature for the WBANs (hereafter, called HZLCL). The weakness of [17] is that costly cryptographic operation is needed in the FABSC. In 2014, Liu *et al.* [18] used certificateless signature to design the access control for the WBANs (hereafter, called LZCK). The contribution of LZCK has two points. The first point is to achieve the anonymity and the second point is to use certificateless cryptography (CLC) [19]. The CLC has neither public key certificates nor key escrow problem. The CLC still needs a trusted third party called key generating center (KGC) who is responsible for generating a partial private key using a master key and a user's identity. Then, the user generates some secret value and combines the secret value with the partial private key to obtain a full private key. Note that the KGC does not know the full private key since it does not know the secret value. In addition, He *et al.* [20] designed an anonymous authentication protocol for wireless medical sensor networks using smart card. A temporal credential-based mutual authentication and key agreement scheme using pseudoidentity is also proposed in [21]. Chaudhry *et al.* [22] discussed the authentication method for the telecare medicine information systems using biometrics. An improved two factor (password and smart card) authentication protocol for the telecare medical information systems was given in [23].

### B. Motivation and Contribution

The motivation of this paper is to find a new solution for the access control for the WBANs. This solution shall have neither key escrow problem nor public key certificates. Only authorized users must have access the WBANs and both the query messages and identities of the users must be protected. It is important to protect the query messages and identities of the users for preserving the privacy of the users [24]. Our solution is to use certificateless signcryption (CLSC). In this paper, we first propose a novel CLSC scheme, and then, design a cost-effective and anonymous access control scheme for the WBANs based on the CLSC scheme. Our proposed scheme achieves anonymity, confidentiality, authentication, integrity, and nonrepudiation. Compared with existing three access control schemes (CK [14], HZLCL [17], and LZCK [18]) for the WBANs, our proposed scheme has the least computational cost and total energy consumption for the controller.

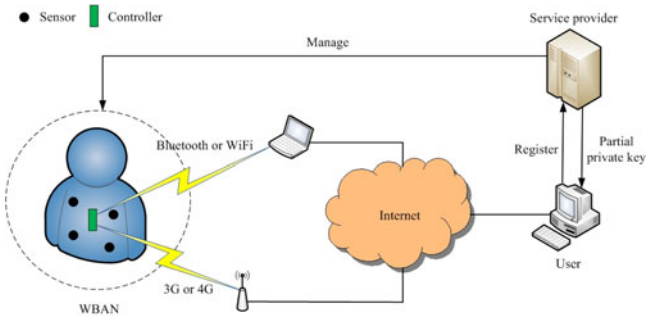


Fig. 1. Network model.

## II. Preliminaries

In this section, we give the network model, security requirements, and bilinear pairings.

### A. Network Model

Fig. 1 shows the overview of the network model that consists of three main entities, the WBAN of a patient, a user (e.g., a nurse, a doctor, a government agency, or an insurance company), and a service provider (SP) [2]. The WBAN is composed of some sensor nodes and a controller. The sensor nodes communicate with the controller and the controller communicates with not only the sensor nodes but also the Internet. The SP manages the WBAN that monitors the patient's vital signs and environmental parameter. The SP is responsible for the registration for the user and the WBAN and generates a partial private key for the user and the private keys for the WBAN (the registration can be online or offline). That is, the SP acts as the KGC in the CLC. We assume that the SP is trustworthy and curious. That is, we do not need to fully trust the SP since it only knows the partial private key of the user. This is an important advantage of CLC than IBC. If a user hopes to access the WBAN, it must be registered at the SP and obtain a partial private key. Then, the user sends a query message to the WBAN. The controller checks if the user is valid. If yes, the user is authorized to access the data of WBAN. Otherwise, the query request is rejected.

### B. Security Requirements

The communication between the user and the controller should satisfy five security properties, i.e., anonymity, confidentiality, authentication, integrity, and nonrepudiation [5]. Anonymity means that a third party cannot identify the identity of a user by a given ciphertext. Confidentiality is keeping query messages secret from the others except the user and the controller. Authentication is the assurance that only the authorized users can access the WBAN. Integrity is ensuring that the query messages from the user have not been modified by unauthorized entities. Nonrepudiation is preventing the denial of previous queries submitted by the user. That is, if the user has issued a query message to the WBAN, it cannot deny its action.

### C. Bilinear Pairings

Let $G_1$ and $G_2$ be two cyclic groups with same prime order $p$. $G_1$ is an additive group and $G_2$ is a multiplicative group. Let $P$ be a generator of $G_1$. A bilinear pairing is a map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ that satisfies the following properties [13].

1) *Bilinearity:* $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$, $a, b \in \mathbb{Z}_p^*$.
2) *Nondegeneracy:* There are $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$, where 1 is the identity element of $G_2$.
3) *Computability:* $\hat{e}(P, Q)$ can be efficiently computed for all $P$,$Q \in G_1$.

The modified Weil pairing and Tate pairing provide admissible maps of this kind. Please refer to [13] for details.

The security of our proposed scheme depends on the hardness of the following hard assumptions [25]–[27].

*Definition 1:* Given groups $G_1$ and $G_2$ of the same prime order $p$, a generator $P$ of $G_1$ and a bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$, the $q$-bilinear Diffie–Hellman inversion ($q$-BDHI) problem in $(G_1, G_2, \hat{e})$ is to compute $\hat{e}(P, P)^{1/\alpha}$ given $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$. Here, $\alpha \in \mathbb{Z}_p^*$.

*Definition 2:* Given groups $G_1$ and $G_2$ of the same prime order $p$, a generator $P$ of $G_1$ and a bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$, the modified bilinear inverse Diffie–Hellman (mBIDH) problem in $(G_1, G_2, \hat{e})$ is to compute $\hat{e}(P, P)^{1/(\alpha+\gamma)}$ given $(P, \alpha P, \gamma)$. Here, $\alpha, \gamma \in \mathbb{Z}_p^*$.

*Definition 3:* Given a group $G_1$ of prime order $p$ and a generator $P$ of $G_1$, the $q$-strong Diffie–Hellman ($q$-SDH) problem in $G_1$ is to find a pair $(w, \frac{1}{\alpha+w} P) \in \mathbb{Z}_p^* \times G_1$ given $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$. Here, $\alpha \in \mathbb{Z}_p^*$.

*Definition 4:* Given a group $G_1$ of prime order $p$ and a generator $P$ of $G_1$, the modified inverse computational Diffie–Hellman (mICDH) problem in $G_1$ is to compute $(\alpha + \gamma)^{-1} P$ given $(P, \alpha P, \gamma)$. Here, $\alpha, \gamma \in \mathbb{Z}_p^*$.

## III. CLSC Scheme

In 2005, Barreto *et al.* [25] proposed an efficient IBSC scheme. However, this scheme cannot be used to design an access control scheme for the WBANs because it uses the IBC technique. In this section, we propose an efficient CLSC scheme based on [25]. The main notations of our proposed scheme are listed in Table I.

TABLE I
NOTATIONS

| Notations | Descriptions |
|---|---|
| $k$ | A security parameter |
| $G_1$ | A cyclic additive group |
| $G_2$ | A cyclic multiplicative group |
| $P$ | A generator of group $G_1$ |
| $\hat{e}$ | A bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ |
| $p$ | The order of group $G_1$ and $G_2$ |
| $n$ | The number of bits of a message to be sent |
| $s$ | A master secret key of the KGC |
| $P_{\text{pub}}$ | A master public key of the KGC |
| $g$ | A element in group $G_2$, where $g = \hat{e}(P, P)$ |
| $H_i()$ | A one way hash function ($i = 1, 2, 3, 4$) |
| $D_U$ | A partial private key of the user with identity $\text{ID}_U$ |
| $S_U$ | A full private key of the user with identity $\text{ID}_U$ |
| $\text{PK}_U$ | A public key of the user with identity $\text{ID}_U$ |
| $+$ | Addition operator |

### A. Efficient CLSC Scheme

Our proposed scheme consists of the following six algorithms. Here, we assume that the sender's identity is $\text{ID}_A$ and the receiver's identity is $\text{ID}_B$.

*1) Setup:* Given a security parameter $k$, the KGC chooses an additive group $G_1$ and a multiplicative $G_2$ of the same prime order $p$, a generator $P$ of $G_1$, a bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$, and four hash functions $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2: G_1 \rightarrow \mathbb{Z}_p^*$, $H_3: G_2 \rightarrow \{0,1\}^n$, and $H_4: \{0,1\}^n \times \{0,1\}^* \times G_1 \times G_2 \rightarrow \mathbb{Z}_p^*$. Here, $n$ is the number of bits of a message to be sent. The KGC randomly selects a master secret key $s \in \mathbb{Z}_p^*$ and computes the corresponding public key $P_{\text{pub}} = sP$. The KGC publishes the system parameters $\{G_1, G_2, p, \hat{e}, n, P, P_{\text{pub}}, g, H_1, H_2, H_3, H_4\}$ and keeps $s$ secret. Here, $g = \hat{e}(P, P)$.

*2) Extract-Partial-Private-Key:* A user submits its identity $\text{ID}_U$ to the KGC. The KGC computes the partial private key

$$D_U = \frac{1}{H_1(\text{ID}_U) + s} P$$

and sends $D_U$ to the user.

*3) Generate-User-Key:* A user with identity $\text{ID}_U$ randomly chooses $x_U \in \mathbb{Z}_p^*$ as the secret value and sets

$$\text{PK}_U = x_U(H_1(\text{ID}_U)P + P_{\text{pub}})$$

as the public key. The public key can be published without certification.

*4) Set-Private-Key:* Given a partial private key $D_U$, a secret value $x_U$, and a public key $\text{PK}_U$, the user with identity $\text{ID}_U$ sets a full private key

$$S_U = \frac{1}{x_U + H_2(\text{PK}_U)} D_U.$$

*5) Signcrypt:* Given a message $m \in \{0,1\}^n$, a sender's private key $S_A$, identity $\text{ID}_A$ and public key $\text{PK}_A$, and a receiver's identity $\text{ID}_B$ and public key $\text{PK}_B$, this algorithm works as follows:

1) choose $x$ from $\mathbb{Z}_p^*$ randomly;
2) compute $r = g^x$ and $c = m \oplus H_3(r)$;

3) compute $h = H_4(m, \mathrm{ID}_A, \mathrm{PK}_A, r)$;
4) compute $S = (x + h)S_A$;
5) compute $T = x(\mathrm{PK}_B + H_2(\mathrm{PK}_B)(H_1(\mathrm{ID}_B)P + P_{\mathrm{pub}}))$.

The ciphertext is $\sigma = (c, S, T)$.

*6) Unsigncrypt:* : Given a ciphertext $\sigma$, a sender's identity $\mathrm{ID}_A$ and public key $\mathrm{PK}_A$, and a receiver's private key $S_B$, this algorithm works as follows:

1) compute $r = \hat{e}(T, S_B)$;
2) recover $m = c \oplus H_3(r)$;
3) compute $h = H_4(m, \mathrm{ID}_A, \mathrm{PK}_A, r)$;
4) accept the message if and only if

$$r = \hat{e}(S, \mathrm{PK}_A + H_2(\mathrm{PK}_A)(H_1(\mathrm{ID}_A)P + P_{\mathrm{pub}}))g^{-h}$$

return error symbol $\perp$ otherwise.

### B. Security

Now, we prove that the proposed CLSC scheme satisfies confidentiality [indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2)] and unforgeability [existential unforgeability against adaptive chosen messages attacks (EUF-CMA)] [28]. For each notion, we should consider two types of adversaries, Type I and Type II [19], [28]. A Type I adversary models an attacker that is a common user and does not have the KGC's secret key. But it can adaptively replace users' public keys with (valid) public keys of its choice. A Type II adversary models an honest-but-curious KGC who knows the KGC's secret key. But it cannot replace users' public keys. We prove the proposed CLSC scheme according to the security model defined in [28].

*Theorem 1:* [Confidentiality] In the random oracle model, the proposed CLSC scheme satisfies the IND-CCA2 security under the $q$-BDHI and mBIDH assumptions.

*Proof:* This theorem follows from the following Lemmas 1 and 2. ■

*Lemma 1:* In the random oracle model, if there exists an adversary $\mathcal{A}_I$ that has a nonnegligible advantage $\epsilon$ against the Type I IND-CCA2 security of the proposed CLSC scheme when running in a time $t$ and carrying out $q_{\mathrm{ppk}}$ partial private key extraction queries, $q_{\mathrm{sk}}$ private key queries, $q_{\mathrm{pk}}$ public key queries, $q_{\mathrm{pkr}}$ public key replacement queries, $q_s$ signcryption queries, $q_u$ unsigncryption queries, and $q_{H_i}$ queries to oracles $H_i$ $(i = 1, 2, 3, 4)$, then we can construct an algorithm $\mathcal{C}$ that can solve the $q$-BDHI problem for $q = q_{H_1}$ with an advantage

$$\epsilon' \geq \frac{\epsilon}{q_{H_1}(q_{H_3} + 2q_{H_4})}\left(1 - \frac{q_s(q_s + q_{H_4})}{2^k}\right)\left(1 - \frac{q_u}{2^k}\right)$$

in a time $t' \leq t + O(q_s + q_u)t_p + O(q_{H_1}^2)t_m + O(q_u q_{H_4})t_e$, where $t_p$ is the cost for one pairing operation, $t_m$ is the cost for a scalar multiplication operation in $G_1$, and $t_e$ is the cost for an exponentiation operation in $G_2$.

*Proof:* In this proof, we show how $\mathcal{C}$ can use $\mathcal{A}_I$ as a subroutine to solve a random instance $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$ of the $q$-BDHI problem.

*Initial:* In a preparation phase, $\mathcal{C}$ chooses $\ell \in \{1, \ldots, q_{H_1}\}$, elements $e_\ell \in \mathbb{Z}_p^*$ and $w_1, \ldots, w_{\ell-1}, w_{\ell+1}, w_q \in \mathbb{Z}_p^*$ randomly. For $i = 1, \ldots, \ell - 1, \ell + 1, \ldots, q$, $\mathcal{C}$ sets $e_i = e_\ell - w_i$. Then, $\mathcal{C}$ uses its input to set a generator $Q \in G_1$ and an element $X = \alpha Q \in G_1$ such that it knows $q - 1$ pairs $(w_i, V_i = \frac{1}{\alpha + w_i}Q)$ for $i \in \{1, \ldots, q\}\backslash\{\ell\}$ as in [29]. To do so, $\mathcal{C}$ expands the polynomial

$$f(z) = \prod_{i=1, i\neq\ell}^{q}(z + w_i) = \sum_{j=0}^{q-1} c_j z^j.$$

A generator $Q$ and an element $X$ can be obtained as

$$Q = \sum_{j=0}^{q-1} c_j(\alpha^j P) = f(\alpha)P$$

and

$$X = \sum_{j=1}^{q} c_{j-1}(\alpha^j P) = \alpha f(\alpha)P = \alpha Q.$$

As in [29], the pairs $(w_i, V_i)$ for $i \in \{1, \ldots, q\}\backslash\{\ell\}$ can be gotten by expanding

$$f_i(z) = \frac{f(z)}{z + w_i} = \sum_{j=0}^{q-2} d_j z^j$$

and setting

$$V_i = \sum_{j=0}^{q-2} d_j(\alpha^j P) = f_i(\alpha)P = \frac{f(\alpha)}{\alpha + w_i}P = \frac{1}{\alpha + w_i}Q.$$

The public key of the KGC is set as $Q_{\mathrm{pub}} = -X - e_\ell Q = (-\alpha - e_\ell)Q$ and its corresponding secret key is implicitly set to $s = -\alpha - e_\ell \in \mathbb{Z}_p^*$. For all $i \in \{1, \ldots, q\}\backslash\{\ell\}$, we have $(e_i, -V_i) = (e_i, \frac{1}{e_i + s}Q)$. $\mathcal{C}$ gives $\mathcal{A}_I$ the system parameters with $Q, Q_{\mathrm{pub}} = (-\alpha - e_\ell)Q$ and $g = \hat{e}(Q, Q)$.

*Phase 1*: $\mathcal{C}$ simulates $\mathcal{A}_I$'s challenger in the Type I IND-CCA2 game. $\mathcal{C}$ keeps four lists $L_1$, $L_2$, $L_3$, and $L_4$ to simulate oracles $H_1$, $H_2$, $H_3$, and $H_4$, respectively. $\mathcal{C}$ should maintain the consistency and avoid collision for these answers. In addition, $\mathcal{C}$ maintains a list $L_k$ that is initially empty to keep the public key information. We assume that $H_1$ queries are different, that $\mathcal{A}_I$ will first ask $H_1(\mathrm{ID})$ before ID is used in the other queries and that the target identity $\mathrm{ID}_B$ is submitted to $H_1$ at some point. In addition, we think the sender's identity is different to the receiver's identity by irreflexivity assumption [30].

1) *$H_1$ queries:* These queries are indexed by a counter $\nu$ that is initially set to 1. For a $H_1(\mathrm{ID}_\nu)$ query, $\mathcal{C}$ returns $e_\nu$ as the answer, inserts $(\mathrm{ID}_\nu, e_\nu)$ into the list $L_1$ and increments $\nu$.
2) *$H_2$ queries:* When $\mathcal{A}_I$ asks a $H_2(\mathrm{PK}_i)$ query, $\mathcal{C}$ checks if the value of $H_2$ has been defined for the input $\mathrm{PK}_i$. If it has been defined, $\mathcal{C}$ returns previously defined value. Otherwise, $\mathcal{C}$ chooses $h_{2,i}$ from $\mathbb{Z}_p^*$ randomly, returns $h_{2,i}$ as an answer and inserts $(\mathrm{PK}_i, h_{2,i})$ into the list $L_2$.
3) *$H_3$ queries:* When $\mathcal{A}_I$ asks a $H_3(r_i)$ query, $\mathcal{C}$ checks if the value of $H_3$ has been defined for the input $r_i$. If it has been defined, $\mathcal{C}$ returns previously defined value. Otherwise, $\mathcal{C}$ chooses $h_{3,i}$ from $\{0, 1\}^n$ randomly, returns $h_{3,i}$ as an answer and inserts $(r_i, h_{3,i})$ into the list $L_3$.
4) *$H_4$ queries:* For a $H_4(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i)$ query, $\mathcal{C}$ checks if the value of $H_4$ has been defined for the input $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i)$. If it has been defined, $\mathcal{C}$ returns the

previously defined value. Otherwise, $\mathcal{C}$ returns a random $h_{4,i} \in \mathbb{Z}_p^*$ as the answer. In addition, to answer the following queries, $\mathcal{C}$ simulates $H_3$ oracle to obtain $h_{3,i} = H_3(r_i) \in \{0,1\}^n$ and sets $c_i = m_i \oplus h_{3,i}$ and $\xi_i = r_i \cdot \hat{e}(Q,Q)^{h_{4,i}}$. Finally, $\mathcal{C}$ inserts the tuple $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i, h_{4,i}, c_i, \xi_i)$ into the list $L_4$.

5) *Partial private key extraction queries:* $\mathcal{A}_I$ chooses an identity $\mathrm{ID}_i$ and asks its partial private key. If $i = \ell$, then $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ knows that $H_1(\mathrm{ID}_i) = e_i$ and returns the partial private key $-V_i = \frac{1}{e_i+s}Q$.
6) *Private key queries:* $\mathcal{A}_I$ can ask a private key query on an identity $\mathrm{ID}_i$. If $i = \ell$, then $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ knows the partial private key $-V_i = \frac{1}{e_i+s}Q$. Then, $\mathcal{C}$ searches $L_k$ for the entry $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$ ($\mathcal{C}$ generates a new key pair if this entry does not exist) and returns $S_i = -\frac{1}{x_i+h_{2,i}}V_i$.
7) *Public key queries:* $\mathcal{A}_I$ selects an identity $\mathrm{ID}_i$ and sends it to $\mathcal{C}$. If the list $L_k$ contains a tuple $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$, then $\mathcal{C}$ returns $\mathrm{PK}_i$ to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ chooses a random $x_i \in \mathbb{Z}_p^*$, sets $\mathrm{PK}_i = x_i(e_iQ + Q_{\mathrm{pub}})$, inserts $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$ into the list $L_k$ and returns $\mathrm{PK}_i$ to $\mathcal{A}_I$.
8) *Public key replacement queries:* $\mathcal{A}_I$ may replace a public key $\mathrm{PK}_i$ with a value chosen by it. For a public key replacement query for $(\mathrm{ID}_i, \mathrm{PK}_i)$, $\mathcal{C}$ updates the list $L_k$ with tuple $(\mathrm{ID}_i, \mathrm{PK}_i, \perp)$. Here, $\perp$ denotes an unknown value.
9) *Signcryption queries:* $\mathcal{A}_I$ can make a signcryption query by submitting a message $m$, a sender's identity $\mathrm{ID}_i$ and a receiver's identity $\mathrm{ID}_j$. If $i \neq \ell$, $\mathcal{C}$ knows the sender's private key $S_i$ and can answer this query according to the steps of signcryption algorithm. If $i = \ell$, $\mathcal{C}$ knows the receiver's private key $S_j$ since $j \neq \ell$ by the irreflexivity assumption [30]. In this case, $\mathcal{C}$ first randomly chooses $\theta, h \in \mathbb{Z}_p^*$, compute $S = \theta S_j, T = \theta(\mathrm{PK}_i + h_{2,i}(e_iQ + Q_{\mathrm{pub}})) - h(\mathrm{PK}_j + h_{2,j}(e_jQ + Q_{\mathrm{pub}}))$ and $r = \hat{e}(T, S_j)$. Then, $\mathcal{C}$ defines the hash value $H_4(m, \mathrm{ID}_i, \mathrm{PK}_i, r)$ to $h$. Finally, $\mathcal{C}$ computes $c = m \oplus H_3(r)$ and returns $\sigma = (c, S, T)$ to the adversary. $\mathcal{C}$ fails if $H_4$ is already defined but this only happens with probability $(q_s + q_{H_4})/2^k$.
10) *Unsigncryption queries:* $\mathcal{A}_I$ can make an unsigncryption query by submitting a ciphertext $\sigma = (c, S, T)$, a sender's identity $\mathrm{ID}_i$ and a receiver's identity $\mathrm{ID}_j$. If $j \neq \ell$, then $\mathcal{C}$ knows the receiver's private key $S_j$ and can answer this query according to the steps of unsigncryption algorithm. If $j = \ell$, $\mathcal{C}$ knows the sender's private key $S_i$ since $i \neq \ell$ by the irreflexivity assumption [30]. For all valid ciphertexts, we have $\log_{S_i}(S - hS_i) = \log_{\mathrm{PK}_j + h_{2,j}(e_jQ + Q_{\mathrm{pub}})} T$, where $h = H_4(m, \mathrm{ID}_i, \mathrm{PK}_i, r)$. Therefore, we have

$$\hat{e}(T, S_i) = \hat{e}(\mathrm{PK}_j + h_{2,j}(e_jQ + Q_{\mathrm{pub}}), S - hS_i).$$

$\mathcal{C}$ first sets $\xi = \hat{e}(S, \mathrm{PK}_i + h_{2,i}(e_iQ + Q_{\mathrm{pub}}))$, and then, searches the list $L_4$ for the entries of the form $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i, h_{4,i}, c, \xi)$ indexed by $i \in \{1, \ldots, q_{H_4}\}$. If there does not exist such an entry, $\sigma$ is rejected. Otherwise, $\mathcal{C}$ further checks whether the following equation holds for the corresponding indexes:

$$\frac{\hat{e}(T, S_i)}{\hat{e}(\mathrm{PK}_j + h_{2,j}(e_jQ + Q_{\mathrm{pub}}), S)} = \hat{e}(\mathrm{PK}_j + h_{2,j}(e_jQ + Q_{\mathrm{pub}}), S_i)^{-h_{4,i}}.$$

If the unique $i \in \{1, \ldots, q_{H_4}\}$ that satisfies this aforementioned equation is found, $\mathcal{C}$ returns the matching plaintext $m_i$. Otherwise, $\sigma$ is rejected. For all unsigncryption queries, the probability to reject a valid ciphertext is less than or equal to $\frac{q_u}{2^k}$.

*Challenge:* $\mathcal{A}_I$ generates two equal length messages $(m_0, m_1)$, a sender's identity $\mathrm{ID}_A$ and a receiver's identity $\mathrm{ID}_B$ on which it hopes to be challenged. If $\mathrm{ID}_B \neq \mathrm{ID}_\ell$, $\mathcal{C}$ fails. Otherwise, $\mathcal{C}$ chooses $c^* \in \{0,1\}^n$, $\lambda \in \mathbb{Z}_p^*$, $S^* \in G_1$ randomly and sets $T^* = -\lambda x_B Q - \lambda h_{2,B} Q$. $\mathcal{C}$ returns the ciphertext $\sigma^* = (c^*, S^*, T^*)$ to $\mathcal{A}_I$. If we define $\rho = \lambda/\alpha$ and since $s = -\alpha - e_\ell$, we have

$$\begin{aligned}
T^* &= -\lambda x_B Q - \lambda h_{2,B} Q \\
&= -\rho\alpha x_B Q - \rho\alpha h_{2,B} Q \\
&= (e_\ell + s)\rho x_B Q + (e_\ell + s)\rho h_{2,B} Q \\
&= \rho e_\ell x_B Q + \rho x_B Q_{\mathrm{pub}} + \rho e_\ell h_{2,B} Q + \rho h_{2,B} Q_{\mathrm{pub}} \\
&= \rho x_B(e_\ell Q + Q_{\mathrm{pub}}) + \rho h_{2,B}(e_\ell Q + Q_{\mathrm{pub}}) \\
&= \rho \mathrm{PK}_B + \rho h_{2,B}(e_\ell Q + Q_{\mathrm{pub}}) \\
&= \rho(\mathrm{PK}_B + h_{2,B}(e_\ell Q + Q_{\mathrm{pub}})).
\end{aligned}$$

$\mathcal{A}_I$ cannot identify that $\sigma^*$ is not a valid ciphertext unless it makes a $H_3$ or $H_4$ query on $\hat{e}(Q,Q)^\rho$.

*Phase 2:* $\mathcal{A}_I$ is allowed to make a polynomially bounded number of queries adaptively again as in the phase 1 with the limitation that: 1) it cannot make a private key query on $\mathrm{ID}_B$; 2) it cannot make a partial private key extraction query on $\mathrm{ID}_B$ if the public key of $\mathrm{ID}_B$ has been replaced before the challenge phase; and 3) it cannot make an unsigncryption query on $(\sigma^*, \mathrm{ID}_A, \mathrm{ID}_B)$ to obtain the corresponding message unless the public key $\mathrm{PK}_B$ has been replaced after the challenge phase. $\mathcal{C}$ answer $\mathcal{A}_I$'s queries according to the same method as in the phase 1.

*Guess:* $\mathcal{A}_I$ outputs a guess bit $\beta'$, which is ignored by $\mathcal{C}$.

$\mathcal{C}$ fetches a random entry $(r_i, h_{3,i})$ or $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i, h_{4,i}, c_i, \xi_i)$ from $L_3$ or $L_4$. Since the list $L_3$ includes no more than $q_{H_3} + q_{H_4}$ records, the selected entry will contain the correct element $r_i = \hat{e}(Q,Q)^\rho = \hat{e}(P,P)^{f(\alpha)^2\lambda/\alpha}$ with probability $1/(q_{H_3} + 2q_{H_4})$. As in [25], the $q$-BDHI problem can be solved by noting that, if $\xi^* = \hat{e}(P,P)^{1/\alpha}$, then

$$\hat{e}(Q,Q)^{1/\alpha} = \xi^{*^{(c_0^2)}} \hat{e}\left(\sum_{j=0}^{q-2} c_{j+1}(\alpha^j P), c_0 P\right) \hat{e}\left(Q, \sum_{j=0}^{q-2} c_{j+1}(\alpha^j) P\right).$$

This finishes the description of the whole simulation. Now, we analyze $\mathcal{C}$'s advantage. Define the events $E_1$, $E_2$, $E_3$, $E_4$, and $E_5$ as follows.

$E_1$: $\mathcal{A}_I$ does not select $\mathrm{ID}_\ell$ as the receiver's identity in the challenge phase.

$E_2$: $\mathcal{A}_I$ has asked a private key query on the identity $\mathrm{ID}_\ell$.

$E_3$: $\mathcal{A}_I$ has asked a partial private key extraction query on the identity $\mathrm{ID}_\ell$ and the public key of $\mathrm{ID}_\ell$ has been replaced before the challenge phase.

$E_4$: $\mathcal{C}$ aborts in a signcryption query because of a collision on $H_4$.

$E_5$: $\mathcal{C}$ aborts in an unsigncryption query because of rejecting a valid ciphertext.

According to aforementioned analysis, we know that the probability of $\mathcal{C}$ not aborting is

$$\Pr[\neg\mathrm{abort}] = \Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3 \wedge \neg E_4 \wedge \neg E_5].$$

We know that $\Pr[\neg E_1] = 1/q_{H_1}$, $\Pr[E_4] \leq q_s(q_s + q_{H_4})/2^k$, and $\Pr[E_5] \leq q_u/2^k$. In addition, we know that $\neg E_1$ implies $\neg E_2$ and $\neg E_3$. So, we have

$$\Pr[\neg\mathrm{abort}] \geq \frac{1}{q_{H_1}}\left(1 - \frac{q_s(q_s + q_{H_4})}{2^k}\right)\left(1 - \frac{q_u}{2^k}\right).$$

In addition, $\mathcal{C}$ chooses the correct element from the list $L_3$ or $L_4$ with probability $1/(q_{H_3} + 2q_{H_4})$. Therefore, we have

$$\epsilon' \geq \frac{\epsilon}{q_{H_1}(q_{H_3} + 2q_{H_4})}\left(1 - \frac{q_s(q_s + q_{H_4})}{2^k}\right)\left(1 - \frac{q_u}{2^k}\right).$$

The bound on $\mathcal{C}$'s computation time can be obtained from the fact that $\mathcal{C}$ needs $O(q_{H_1}^2)$ point multiplication operations in $G_1$ in the preparation phase, $O(q_s + q_u)$ pairing operations and $O(q_u q_{H_4})$ exponentiation operations in $G_2$ in the unsigncryption queries.

*Lemma 2:* In the random oracle model, if there exists an adversary $\mathcal{A}_{II}$ that has a nonnegligible advantage $\epsilon$ against the Type II IND-CCA2 security of the proposed CLSC scheme when running in a time $t$ and carrying out $q_{sk}$ private key queries, $q_{\mathrm{pk}}$ public key queries, $q_s$ signcryption queries, $q_u$ unsigncryption queries, and $q_{H_i}$ queries to oracles $H_i$ $(i = 1, 2, 3, 4)$, then we can construct an algorithm $\mathcal{C}$ that can solve the mBIDH problem with an advantage

$$\epsilon' \geq \frac{\epsilon}{q_{H_1}(q_{H_3} + 2q_{H_4})}\left(1 - \frac{q_s(q_s + q_{H_4})}{2^k}\right)\left(1 - \frac{q_u}{2^k}\right)$$

in a time $t' \leq t + O(q_s + q_u)t_p + O(q_u q_{H_4})t_e$, where $t_p$ is the cost for one pairing operation and $t_e$ is the cost for an exponentiation operation in $G_2$.

*Proof:* In this proof, we show how $\mathcal{C}$ can use $\mathcal{A}_{II}$ as a subroutine to solve a random instance $(P, \alpha P, \gamma)$ of the mBIDH problem.

*Initial:* $\mathcal{C}$ gives $\mathcal{A}_{II}$ the system parameters *params* with $P_{\mathrm{pub}} = sP$ and secret keys $s$. Here, $s$ is randomly selected by $\mathcal{C}$.

*Phase 1*: $\mathcal{C}$ simulates $\mathcal{A}_{II}$'s challenger in the Type II IND-CCA2 game. $\mathcal{C}$ keeps four lists $L_1$, $L_2$, $L_3$, and $L_4$ to simulate oracles $H_1$, $H_2$, $H_3$, and $H_4$, respectively. $\mathcal{C}$ should maintain the consistency and avoid collision for these answers. In addition, $\mathcal{C}$ maintains a list $L_k$ that is initially empty to keep the public key information. We assume that $H_1$ queries are different and that $\mathcal{A}_{II}$ will first ask $H_1(\mathrm{ID})$ before ID is used in the other queries. In addition, we think the sender's identity is different to the receiver's identity by irreflexivity assumption [30]. $\mathcal{C}$ chooses a random number $\ell \in \{1, 2, \ldots, q_{H_1}\}$ and answers $\mathcal{A}_{II}$'s queries as follows.

1) $H_1$ *queries:* $\mathcal{A}_{II}$ chooses an identity $\mathrm{ID}_i$ and submits it to $\mathcal{C}$, $\mathcal{C}$ randomly chooses $e_i \in \mathbb{Z}_p^*$, inserts $(\mathrm{ID}_i, e_i)$ into the list $L_1$ and answers $H_1(\mathrm{ID}_i) = e_i$.
2) $H_2$ *queries:* When $\mathcal{A}_{II}$ asks a $H_2(\mathrm{PK}_i)$ query, $\mathcal{C}$ checks if the value of $H_2$ has been defined for the input $\mathrm{PK}_i$. If it has been defined, $\mathcal{C}$ returns previously defined value. Otherwise, $\mathcal{C}$ checks if $\mathrm{PK}_i = e_i\alpha P + s\alpha P$ (i.e., $i = \ell$). If yes, $\mathcal{C}$ returns $h_{2,\ell} = \gamma$ and inserts $(\mathrm{PK}_\ell, \gamma)$ into the list $L_2$. If no, a random $h_{2,i} \in \mathbb{Z}_p^*$ is returned and $(\mathrm{PK}_i, h_{2,i})$ is inserted into the list $L_2$.
3) $H_3$ *queries:* When $\mathcal{A}_{II}$ asks a $H_3(r_i)$ query, $\mathcal{C}$ checks if the value of $H_3$ has been defined for the input $r_i$. If it has been defined, $\mathcal{C}$ returns previously defined value. Otherwise, $\mathcal{C}$ chooses $h_{3,i}$ from $\{0,1\}^n$ randomly, returns $h_{3,i}$ as an answer and inserts $(r_i, h_{3,i})$ into the list $L_3$.
4) $H_4$ *queries:* For a $H_4(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i)$ query, $\mathcal{C}$ checks if the value of $H_4$ has been defined for the input $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i)$. If it has been defined, $\mathcal{C}$ returns the previously defined value. Otherwise, $\mathcal{C}$ returns a random $h_{4,i} \in \mathbb{Z}_p^*$ as the answer. In addition, to answer the following queries, $\mathcal{C}$ simulates $H_3$ oracle on its own to obtain $h_{3,i} = H_3(r_i) \in \{0,1\}^n$ and sets $c_i = m_i \oplus h_{3,i}$ and $\xi_i = r_i \cdot \hat{e}(P, P)^{h_{4,i}}$. Finally, $\mathcal{C}$ inserts the tuple $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i, h_{4,i}, c_i, \xi_i)$ into the list $L_4$.
5) *Private key queries:* $\mathcal{A}_{II}$ can ask a private key query on an identity $\mathrm{ID}_i$. If $i = \ell$, then $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ runs $H_1$ oracle to obtain $(\mathrm{ID}_i, e_i)$. Then, $\mathcal{C}$ searches $L_k$ for the entry $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$ ($\mathcal{C}$ generates a new key pair if this entry does not exist) and returns

$$S_i = \frac{1}{x_i + h_{2,i}}\frac{1}{e_i + s}P.$$

6) *Public key queries:* $\mathcal{A}_{II}$ chooses an identity $\mathrm{ID}_i$ and sends it to $\mathcal{C}$. If $i \neq \ell$, $\mathcal{C}$ chooses a random $x_i \in \mathbb{Z}_p^*$, sets public key $\mathrm{PK}_i = x_i(e_i P + P_{\mathrm{pub}})$, inserts $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$ into the list $L_k$ and returns $\mathrm{PK}_i$ to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ returns $\mathrm{PK}_\ell = e_\ell\alpha P + s\alpha P$ and inserts $(\mathrm{ID}_\ell, \mathrm{PK}_\ell, \perp)$ into the list $L_k$.
7) *Signcryption queries:* $\mathcal{A}_{II}$ can make a signcryption query by submitting a message $m$, a sender's identity $\mathrm{ID}_i$, and a receiver's identity $\mathrm{ID}_j$. If $i \neq \ell$, $\mathcal{C}$ knows the sender's private key $S_i$ and can answer this query according to the steps of signcryption algorithm. If $i = \ell$, $\mathcal{C}$ knows the receiver's private key $S_j$ since $j \neq \ell$ by the irreflexivity assumption [30]. In this case, $\mathcal{C}$ first randomly chooses $\theta, h \in \mathbb{Z}_p^*$, compute $S = \theta S_j$, $T = \theta(\mathrm{PK}_i + h_{2,i}(e_i P + P_{\mathrm{pub}})) - h(\mathrm{PK}_j + h_{2,j}(e_j P + P_{\mathrm{pub}}))$ and $r = \hat{e}(T, S_j)$. Then, $\mathcal{C}$ defines the hash value $H_4(m, \mathrm{ID}_i, \mathrm{PK}_i, r)$ to $h$. Finally, $\mathcal{C}$ computes $c = m \oplus H_3(r)$ and returns $\sigma =$

$(c, S, T)$ to the adversary. $\mathcal{C}$ fails if $H_4$ is already defined but this only happens with probability $(q_s + q_{H_4})/2^k$.

8) *Unsigncryption queries:* $\mathcal{A}_{II}$ can make an unsigncryption query by submitting a ciphertext $\sigma = (c, S, T)$, a sender's identity $\mathrm{ID}_i$, and a receiver's identity $\mathrm{ID}_j$. If $j \neq \ell$, then $\mathcal{C}$ knows the receiver's private key $S_j$ and can answer this query according to the steps of unsigncryption algorithm. If $j = \ell, \mathcal{C}$ knows the sender's private key $S_i$ since $i \neq \ell$ by the irreflexivity assumption. For all valid ciphertexts, we have $\log_{S_i}(S - hS_i) = \log_{\mathrm{PK}_j + h_{2,j}(e_j P + P_{\mathrm{pub}})} T$, where $h = H_4(m, \mathrm{ID}_i, \mathrm{PK}_i, r)$. Therefore, we have

$$\hat{e}(T, S_i) = \hat{e}(\mathrm{PK}_j + h_{2,j}(e_j P + P_{\mathrm{pub}}), S - hS_i).$$

$\mathcal{C}$ first sets $\xi = \hat{e}(S, \mathrm{PK}_i + h_{2,i}(e_i P + P_{\mathrm{pub}}))$, and then, searches the list $L_4$ for the entries of the form $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i, h_{4,i}, c, \xi)$ indexed by $i \in \{1, \ldots, q_{H_4}\}$. If there does not exist such an entry, $\sigma$ is rejected. Otherwise, $\mathcal{C}$ further checks whether the following equation holds for the corresponding indexes

$$\frac{\hat{e}(T, S_i)}{\hat{e}(\mathrm{PK}_j + h_{2,j}(e_j P + P_{\mathrm{pub}}), S)} = \hat{e}(\mathrm{PK}_j + h_{2,j}(e_j P + P_{\mathrm{pub}}), S_i)^{-h_{4,i}}.$$

If the unique $i \in \{1, \ldots, q_{H_4}\}$ that satisfies this aforementioned equation is found, then $\mathcal{C}$ returns the matching message $m_i$. Otherwise, $\sigma$ is rejected. For all unsigncryption queries, the probability to reject a valid ciphertext is less than or equal to $\frac{q_u}{2^k}$.

*Challenge:* $\mathcal{A}_{II}$ generates two equal length messages $(m_0, m_1)$, a sender's identity $\mathrm{ID}_A$ and a receiver's identity $\mathrm{ID}_B$ on which it hopes to be challenged. If $\mathrm{ID}_B \neq \mathrm{ID}_\ell$, $\mathcal{C}$ fails. Otherwise $\mathcal{C}$ randomly chooses $c^* \in \{0,1\}^n, \lambda \in \mathbb{Z}_p^*, S^* \in G_1$, and sets $T^* = \lambda P$. $\mathcal{C}$ returns the ciphertext $\sigma^* = (c^*, S^*, T^*)$ to $\mathcal{A}_{II}$. $\mathcal{A}_{II}$ cannot identify that $\sigma^*$ is not a valid ciphertext unless it makes a $H_3$ or $H_4$ query on $\hat{e}(T^*, S_B)$.

*Phase 2:* $\mathcal{A}_{II}$ is allowed to make a polynomially bounded number of queries adaptively again as in the phase 1 with the limitation that: 1) it cannot ask a private key query on $\mathrm{ID}_B$; (2) it cannot ask an unsigncryption query on $(\sigma^*, \mathrm{ID}_A, \mathrm{ID}_B)$ to obtain the corresponding message. $\mathcal{C}$ answer $\mathcal{A}_{II}$'s queries according to the same method as in the phase 1.

*Guess:* $\mathcal{A}_{II}$ produces a bit $\beta'$, which is ignored by $\mathcal{C}$.

$\mathcal{C}$ fetches a random entry $(r_i, h_{3,i})$ or $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i, h_{4,i}, c_i, \xi_i)$ from $L_3$ or $L_4$. Since the list $L_3$ includes no more than $q_{H_3} + q_{H_4}$ records, the chosen entry will contain the right element $r_i = \hat{e}(T^*, S_B)$ with probability $1/(q_{H_3} + 2q_{H_4})$. The mBIDH problem can be solved by noting that, if

$$\hat{e}(T^*, S_B) = \hat{e}\left(\lambda P, \frac{1}{\alpha + \gamma}\frac{1}{e_i + s} P\right)$$

we have

$$\hat{e}(P, P)^{\frac{1}{\alpha+\gamma}} = r_i^{\frac{e_i + s}{\lambda}}.$$

This finishes the description of the whole simulation. Now, we analyze $\mathcal{C}$'s advantage. Define the events $E_1$, $E_2$, $E_3$, and $E_4$ as follows.

$E_1$: $\mathcal{A}_{II}$ does not select $\mathrm{ID}_\ell$ as the receiver's identity in the challenge phase.

$E_2$: $\mathcal{A}_{II}$ has asked a private key query on the identity $\mathrm{ID}_\ell$.

$E_3$: $\mathcal{C}$ aborts in a signcryption query because of a collision on $H_4$.

$E_4$: $\mathcal{C}$ aborts in an unsigncryption query because of rejecting a valid ciphertext.

According to aforementioned analysis, we know that the probability of $\mathcal{C}$ not aborting is

$$\Pr[\neg\mathrm{abort}] = \Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3 \wedge \neg E_4].$$

From the aforementioned analysis, we know that $\Pr[\neg E_1] = 1/q_{H_1}$, $\Pr[E_3] \leq q_s(q_s + q_{H_4})/2^k$ and $\Pr[E_4] \leq q_u/2^k$. In addition, we know that $\neg E_1$ implies $\neg E_2$. So, we have

$$\Pr[\neg\mathrm{abort}] \geq \frac{1}{q_{H_1}}\left(1 - \frac{q_s(q_s + q_{H_4})}{2^k}\right)\left(1 - \frac{q_u}{2^k}\right).$$

In addition, $\mathcal{C}$ chooses the correct element from the list $L_3$ or $L_4$ with probability $1/(q_{H_3} + 2q_{H_4})$. Therefore, we have

$$\epsilon' \geq \frac{\epsilon}{q_{H_1}(q_{H_3} + 2q_{H_4})}\left(1 - \frac{q_s(q_s + q_{H_4})}{2^k}\right)\left(1 - \frac{q_u}{2^k}\right).$$

The bound on $\mathcal{C}$'s computation time can be obtained from the fact that $\mathcal{C}$ needs $O(q_s + q_u)$ pairing operations and $O(q_u q_{H_4})$ exponentiation operations in $G_2$ in the unsigncryption queries.□

*Theorem 2:* [Unforgeability] In the random oracle model, the proposed CLSC scheme satisfies the EUF-CMA security under the $q$-SDH and mICDH assumptions.

*Proof:* This theorem follows from the following Lemmas 3 and 4. □

*Lemma 3:* In the random oracle model, if there exists an adversary $\mathcal{F}_I$ that has an advantage $\epsilon \geq 10(q_s + 1)(q_s + q_{H_4})/2^k$ against the Type I EUF-CMA security of the proposed CLSC scheme when running in a time $t$ and carrying out $q_{\mathrm{ppk}}$ partial private key extraction queries, $q_{sk}$ private key queries, $q_{\mathrm{pk}}$ public key queries, $q_{\mathrm{pkr}}$ public key replacement queries, $q_s$ signcryption queries, $q_u$ unsigncryption queries, and $q_{H_i}$ queries to oracles $H_i$ $(i = 1, 2, 3, 4)$, then we can construct an algorithm $\mathcal{C}$ that can solve the $q$-SDH problem for $q = q_{H_1}$ in a time

$$t' \leq 120686 q_{H_1} q_{H_4} \frac{t + O((q_s + q_u)t_p + q_u q_{H_4} t_e)}{\epsilon(1 - 1/2^k)(1 - q/2^k)} + O(q^2 t_m)$$

where $t_p$ is the cost for one pairing operation, $t_e$ is the cost for an exponentiation operation in $G_2$, and $t_m$ is the cost for a scalar multiplication operation in $G_1$.

*Proof:* In this proof, we show how $\mathcal{C}$ can use $\mathcal{F}_I$ as a subroutine to solve a random instance $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$ of the $q$-SDH problem.

*Initial:* $\mathcal{C}$ randomly chooses $w_1, w_2, \ldots, w_{q-1} \in \mathbb{Z}_p^*$ and uses $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$ to set a generator $Q \in G_1$ and $Q_{\mathrm{pub}} = \alpha Q \in G_1$ such that it knows $q - 1$ pairs $(w_i, V_i = \frac{1}{\alpha + w_i} Q)$ for $i \in \{1, \ldots, q-1\}$ as in Lemma 1. $\mathcal{C}$ also randomly chooses a challenge identity $\mathrm{ID}_\ell$ and gives $\mathcal{F}_I$ the $\mathrm{ID}_\ell$ and system parameters with $Q$, $Q_{\mathrm{pub}} = \alpha Q$, and $g = \hat{e}(Q, Q)$.

*Attack*: $\mathcal{C}$ simulates $\mathcal{F}_I$'s challenger in the Type I EUF-CMA game. Similar to Lemma 1, $\mathcal{C}$ keeps four lists $L_1$, $L_2$, $L_3$, and $L_4$ to simulate oracles $H_1$, $H_2$, $H_3$, and $H_4$, respectively. $\mathcal{C}$

also maintains a list $L_k$ to keep the public key information. We also assume that $H_1$ queries are different and that $\mathcal{F}_I$ will first ask $H_1(\mathrm{ID})$ before ID is used in the other queries. $H_2$ queries, $H_3$ queries, $H_4$ queries, and public key replacement queries are treated according to the same method in Lemma 1. The other queries are explained as follows.

1) $H_1$ *queries:* These queries are indexed by a counter $\nu$ that is initially set to 1. If $\mathrm{ID} = \mathrm{ID}_\ell$, $\mathcal{C}$ answers $\mathcal{F}_I$ a random value $w_\ell \in \mathbb{Z}_p^*$. Otherwise, $\mathcal{C}$ answers $\mathcal{F}_I$ the $w_\nu$ and increases $\nu$. In both case, $\mathcal{C}$ inserts $(\mathrm{ID}, w)$ (here, $w = w_\ell$ or $w_\nu$) into the list $L_1$.
2) *Partial private key extraction queries:* $\mathcal{F}_I$ chooses an identity $\mathrm{ID}_i$ and asks its partial private key. If $i = \ell$, then $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ knows that $H_1(\mathrm{ID}_i) = w_i$ and returns the partial private key $V_i = \frac{1}{\alpha + w_i} Q$.
3) *Private key queries:* $\mathcal{F}_I$ can ask a private key query on an identity $\mathrm{ID}_i$. If $i = \ell$, then $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ knows the partial private key $V_i = \frac{1}{\alpha + w_i} Q$. Then, $\mathcal{C}$ searches $L_k$ for the entry $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$ ($\mathcal{C}$ generates a new key pair if this entry does not exist) and returns $S_i = \frac{1}{x_i + h_{2,i}} V_i$.
4) *Public key queries:* $\mathcal{F}_I$ selects an identity $\mathrm{ID}_i$ and sends it to $\mathcal{C}$. If $L_k$ contains a tuple $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$, then $\mathcal{C}$ returns $\mathrm{PK}_i$ to $\mathcal{F}_I$. Otherwise, $\mathcal{C}$ chooses a random $x_i \in \mathbb{Z}_p^*$, sets $\mathrm{PK}_i = x_i(w_i Q + Q_{\mathrm{pub}})$, inserts $(\mathrm{ID}_i, \mathrm{PK}_i, x_i)$ into the list $L_k$ and returns $\mathrm{PK}_i$ to $\mathcal{F}_I$.
5) *Signcryption queries:* $\mathcal{F}_I$ can make a signcryption query by submitting a message $m$, a sender's identity $\mathrm{ID}_i$ and a receiver's identity $\mathrm{ID}_j$. If $i \neq \ell$, $\mathcal{C}$ knows the sender's private key $S_i$ and can answer this query according to the steps of signcryption algorithm. If $i = \ell$, $\mathcal{C}$ knows the receiver's private key $S_j$ since $j \neq \ell$ by the irreflexivity assumption [30]. In this case, $\mathcal{C}$ first randomly chooses $\theta, h \in \mathbb{Z}_p^*$, compute $S = \theta S_j$, $T = \theta(\mathrm{PK}_i + h_{2,i}(w_i Q + Q_{\mathrm{pub}})) - h(\mathrm{PK}_j + h_{2,j}(w_j Q + Q_{\mathrm{pub}}))$ and $r = \hat{e}(T, S_j)$. Then, $\mathcal{C}$ defines the hash value $H_4(m, \mathrm{ID}_i, \mathrm{PK}_i, r)$ to $h$. Finally, $\mathcal{C}$ computes $c = m \oplus H_3(r)$ and returns $\sigma = (c, S, T)$ to the adversary. $\mathcal{C}$ fails if $H_4$ is already defined but this only happens with probability $(q_s + q_{H_4})/2^k$.
6) *Unsigncryption queries:* $\mathcal{F}_I$ can make an unsigncryption query by submitting a ciphertext $\sigma = (c, S, T)$, a sender's identity $\mathrm{ID}_i$ and a receiver's identity $\mathrm{ID}_j$. If $j \neq \ell$, then $\mathcal{C}$ knows the receiver's private key $S_j$ and can answer this query according to the steps of the unsigncryption algorithm. If $j = \ell$, $\mathcal{C}$ knows the sender's private key $S_i$ since $i \neq \ell$ by the irreflexivity assumption [30]. For all valid ciphertexts, we have $\log_{S_i}(S - hS_i) = \log_{\mathrm{PK}_j + h_{2,j}(w_j Q + Q_{\mathrm{pub}})} T$, where $h = H_4(m, \mathrm{ID}_i, \mathrm{PK}_i, r)$. Therefore, we have

$$\hat{e}(T, S_i) = \hat{e}(\mathrm{PK}_j + h_{2,j}(w_j Q + Q_{\mathrm{pub}}), S - hS_i).$$

$\mathcal{C}$ first sets $\xi = \hat{e}(S, \mathrm{PK}_i + h_{2,i}(w_i Q + Q_{\mathrm{pub}}))$, and then, searches the list $L_4$ for the entries of the form $(m_i, \mathrm{ID}_i, \mathrm{PK}_i, r_i, h_{4,i}, c, \xi)$ indexed by $i \in \{1, \ldots, q_{H_4}\}$. If there does not exist such an entry, $\sigma$ is rejected. Otherwise, $\mathcal{C}$ further checks whether the following equation holds for the corresponding indexes

$$\frac{\hat{e}(T, S_i)}{\hat{e}(\mathrm{PK}_j + h_{2,j}(w_j Q + Q_{\mathrm{pub}}), S)} = \hat{e}(\mathrm{PK}_j + h_{2,j}(w_j Q + Q_{\mathrm{pub}}), S_i)^{-h_{4,i}}.$$

If the unique $i \in \{1, \ldots, q_{H_4}\}$ that satisfies this afoementioned equation is found, $\mathcal{C}$ returns the matching plaintext $m_i$. Otherwise, $\sigma$ is rejected. For all unsigncryption queries, the probability to reject a valid ciphertext is less than or equal to $\frac{q_u}{2^k}$.

Now, we begin to use the forking lemma [31]. However, since this lemma is only suitable for identity-less chosen message attack, we need coalesce the identity $\mathrm{ID}_\ell$ and a message $m$ and form a "generalized" forged message $(\mathrm{ID}_\ell, m)$ to hide the identity aspect. If $\mathcal{F}_I$ wins the Type I EUF-CMA game, we can construct a Las Vegas machine $\mathcal{F}_I'$ that outputs two signed messages $((\mathrm{ID}_\ell, m), h, S)$ and $((\mathrm{ID}_\ell, m), h^*, S^*)$ with the same commitment and $h \neq h^*$. To get the result of $q$-SDH problem using $\mathcal{F}_I'$, we constructs an algorithm $\mathcal{C}$ as follows.

1) $\mathcal{C}$ obtains two different signatures $((\mathrm{ID}_\ell, m), h, S)$ and $((\mathrm{ID}_\ell, m), h^*, S^*)$ by executing $\mathcal{F}_I'$.
2) $\mathcal{C}$ computes $V_\ell = (x_\ell + H_2(\mathrm{PK}_\ell))(h - h^*)^{-1}(S - S^*) = \frac{1}{\alpha + w_\ell} Q = \frac{f(\alpha)}{\alpha + w_\ell} P$.
3) $\mathcal{C}$ can use long division method and express the polynomial $f$ as $f(z) = \psi(z)(z + w_\ell) + \psi_{-1}$ for some $\psi(z) = \sum_{i=0}^{q-2} \psi_i z^i$ and some $\psi_{-1} \in \mathbb{Z}_p^*$. Then, $\frac{f(z)}{z + w_\ell}$ can be expressed as

$$\frac{f(z)}{z + w_\ell} = \psi(z) + \frac{\psi_{-1}}{z + w_\ell} = \sum_{i=0}^{q-2} \psi_i z^i + \frac{\psi_{-1}}{z + w_\ell}.$$

So, $\mathcal{C}$ obtains

$$\frac{1}{\alpha + w_\ell} P = \frac{1}{\psi_{-1}} \left( V_\ell - \sum_{i=0}^{q-2} \psi_i (\alpha^i P) \right).$$

4) $\mathcal{C}$ outputs $(w_\ell, \frac{1}{\alpha + w_\ell} P)$ as the result of $q$-SDH problem.

According to the forking lemma in [31] and the relationship between the chosen identity attack and the given identity attack [32], if $\mathcal{F}_I$ succeeds in a time $t$ with probability $\epsilon \geq 10(q_s + 1)(q_s + q_{H_4})/2^k$, then $\mathcal{C}$ can obtain the result of $q$-SDH problem in

$$t' \leq 120686 q_{H_1} q_{H_4} \frac{t + O((q_s + q_u) t_p + q_u q_{H_4} t_e)}{\epsilon(1 - 1/2^k)(1 - q/2^k)} + O(q^2 t_m).$$

□

*Lemma 4:* In the random oracle model, if there exists an adversary $\mathcal{F}_{II}$ that has a nonnegligible advantage $\epsilon \geq 10(q_s + 1)(q_s + q_{H_4})/2^k$ against the Type II EUF-CMA security of the proposed CLSC scheme when running in a time $t$ and carrying out $q_{sk}$ private key queries, $q_{\mathrm{pk}}$ public key queries, $q_s$ signcryption queries, $q_u$ unsigncryption queries, and $q_{H_i}$ queries to oracles $H_i$ $(i = 1, 2, 3, 4)$, then we can construct an algorithm $\mathcal{C}$ that can solve the mICDH problem in a time

$$t' \leq 120686 q_{H_1} q_{H_4} \frac{t + O((q_s + q_u) t_p + q_u q_{H_4} t_e)}{\epsilon(1 - 1/2^k)}$$

where $t_p$ is the cost for one pairing operation and $t_e$ is the cost for an exponentiation operation in $G_2$.

*Proof:* We also uses forking lemma [31] to show how $\mathcal{C}$ can use $\mathcal{F}_{II}$ as a subroutine to solve a random instance $(P, \alpha P, \gamma)$ of the mICDH problem.

*Initial:* $\mathcal{C}$ randomly chooses a challenge identity $\mathrm{ID}_\ell$ and gives $\mathcal{F}_{II}$ the ID $_\ell$ and system parameters with $P_{\mathrm{pub}} = sP$ and secret keys $s$. Here, $s$ is randomly selected by $\mathcal{C}$.

*Attack*: $\mathcal{C}$ simulates $\mathcal{F}_{II}$'s challenger in the Type II EUF-CMA game. $\mathcal{C}$ keeps four lists $L_1$, $L_2$, $L_3$, and $L_4$ to simulate oracles $H_1$, $H_2$, $H_3$, and $H_4$, respectively. $\mathcal{C}$ also maintains a list $L_k$ to keep the public key information. We assume that $H_1$ queries are different and that $\mathcal{F}_{II}$ will first ask $H_1(\mathrm{ID})$ before ID is used in the other queries. $H_1$ queries, $H_2$ queries, $H_3$ queries, $H_4$ queries, private key queries, public key queries, signcryption queries, and unsigncryption queries are treated according to the same method in Lemma 2.

Similar to Lemma 3, if $\mathcal{F}_{II}$ wins the Type II EUF-CMA game, we can construct a Las Vegas machine $\mathcal{F}'_{II}$ that outputs two signed messages $((\mathrm{ID}_\ell, m), h, S)$ and $((\mathrm{ID}_\ell, m), h^*, S^*)$ with the same commitment and $h \neq h^*$. To get the result of mICDH problem using $\mathcal{F}'_{II}$, we constructs an algorithm $\mathcal{C}$ as follows.

1) $\mathcal{C}$ obtains two different signatures $((\mathrm{ID}_\ell, m), h, S)$ and $((\mathrm{ID}_\ell, m), h^*, S^*)$ by executing $\mathcal{F}'_{II}$.
2) $\mathcal{C}$ computes $V_\ell = (h - h^*)^{-1}(S - S^*) = \frac{1}{\alpha+\gamma}\frac{1}{e_\ell+s}P$.
3) $\mathcal{C}$ outputs $(e_\ell + s)V_\ell$ as the result of mICDH problem.

According to the forking lemma in [31] and the relationship between the chosen identity attack and the given identity attack [32], if $\mathcal{F}_{II}$ succeeds in a time $t$ with probability $\epsilon \geq 10(q_s+1)(q_s+q_{H_4})/2^k$, then $\mathcal{C}$ can obtain the result of mICDH problem in

$$t' \leq 120686 q_{H_1} q_{H_4} \frac{t + O((q_s+q_u)t_p + q_u q_{H_4} t_e)}{\epsilon(1-1/2^k)}.$$

□

## IV. Certificateless Access Control Scheme

In this section, we design an efficient certificateless access control scheme for the WBANs using the proposed CLSC scheme. Our proposed scheme uses identity-based access control model [33] that is a simple and a practical access control model that associates access privilege with specific users. The access control scheme is composed of four phases: the initialization phase, the registration phase, the authentication and authorization phase, and the revocation phase. We summarize the proposed access control scheme in Fig. 2.

### A. Initialization Phase

In this phase, the SP runs Setup algorithm and manages a WBAN. The controller is assigned an identity $\mathrm{ID}_B$, a public key $\mathrm{PK}_B$ and a private key $S_B$ (the SP may run Extract-Partial-Private-Key, Generate-User-Key, and Set-Private-Key algorithms). The delivery of private key can be online or offline. If the online method is adopted, we can use a secure socket layer to insure the confidentiality of the private key.

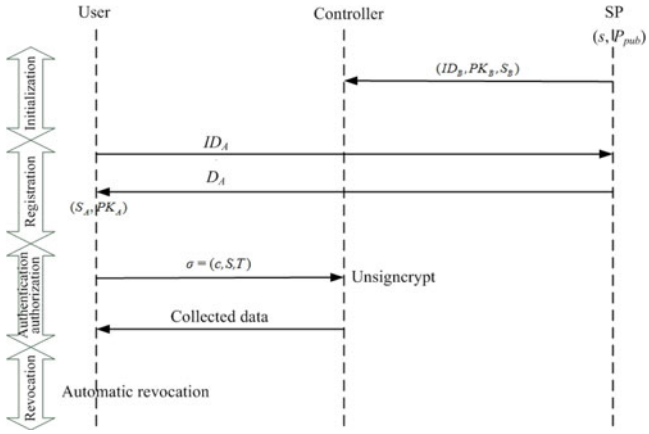


Fig. 2. Certificateless access control scheme.

### B. Registration Phase

A user needs to register with the SP to obtain the access privilege of the WBAN. The user first sends its identity $\mathrm{ID}_A$ to the SP, and then, the SP checks if the identity is valid. If this identity is not valid, the SP rejects this registration request. Otherwise, the SP sets an expiration date $ED$ and runs Extract-Partial-Private-Key algorithm to generate a partial private key

$$D_A = \frac{1}{H_1(\mathrm{ID}_A||ED) + s}P.$$

Here, $||$ is a concatenation symbol. The user can check if

$$\hat{e}(D_A, H_1(\mathrm{ID}_A||ED)P + P_{\mathrm{pub}}) = g$$

holds. If the equation holds, the $D_A$ is valid. Otherwise, the $D_A$ is not valid. After receiving $D_A$, the user runs Generate-User-Key and Set-Private-Key to obtain a full private key $S_A$ and a public key $\mathrm{PK}_A$. The user publishes the public key without certification.

### C. Authentication and Authorization Phase

When the user with identity $\mathrm{ID}_A$ hopes to access the monitoring data of the WBAN, it first generates a query message $m$. To resist the replay attack and to achieve the anonymity, the user concatenates the query message, a timestamp TS and the user's identity and public key to form a new message $m' = m||\mathrm{TS}||\mathrm{ID}_A||\mathrm{PK}_A$. Then, the user runs Signcrypt algorithm that takes as input the new message $m'$, the user's private key $S_A$, identity $\mathrm{ID}_A$, and public key $\mathrm{PK}_A$ and the controller's identity $\mathrm{ID}_B$ and public key $\mathrm{PK}_B$, and output a ciphertext $\sigma = (c, S, T)$. Finally, the user sends the ciphertext $\sigma$ to the controller. When receiving the query request from the user, the controller first computes $r = \hat{e}(T, S_B)$ and recovers $m||TS||\mathrm{ID}_A||\mathrm{PK}_A = c \oplus H_3(r)$. Then, the controller computes $h = H_4(m, \mathrm{ID}_A, \mathrm{PK}_A, r)$ and checks if

$$r = \hat{e}(S, \mathrm{PK}_A + H_2(\mathrm{PK}_A)(H_1(\mathrm{ID}_A||ED)P + P_{\mathrm{pub}}))g^{-h}$$

holds. If the aforementioned equation does not hold, it rejects the query request. Otherwise, the user is authorized to access data of the WBAN. In this case, the controller encrypts the collected

TABLE II
COMPARISON OF PERFORMANCE

| Schemes | Computational Cost | | Controller Communication Cost* | |
|---|---|---|---|---|
| | User | Controller | Receive | Transmit |
| CK [14] | 1*P*+3*M* | 3*P*+*M* | $2\|G_1\| + \|\text{ID}\| + \|m\|$ | — |
| HZLCL [17] | 5*P*+*M* | $(2\|m\|+5)M$ | — | $4\|G_1\| + \|G_2\| + \|\text{ID}\|$ |
| LZCK [18] | 1*E*+4*M* | 3*P*+2*E*+1*M* | $4\|G_1\| + 2\|\mathbb{Z}_p^*\| + \|\text{ID}\| + \|m\|$ | — |
| Ours | 1*E*+4*M* | 2*P*+1*E*+2*M* | $3\|G_1\| + \|\text{ID}\| + \|m\|$ | — |

*The timestamp is omitted in the communication cost for all schemes.

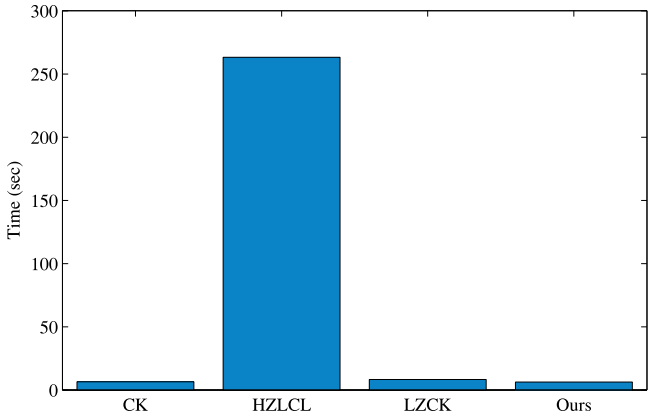


Fig. 3. Computational time of the controller.

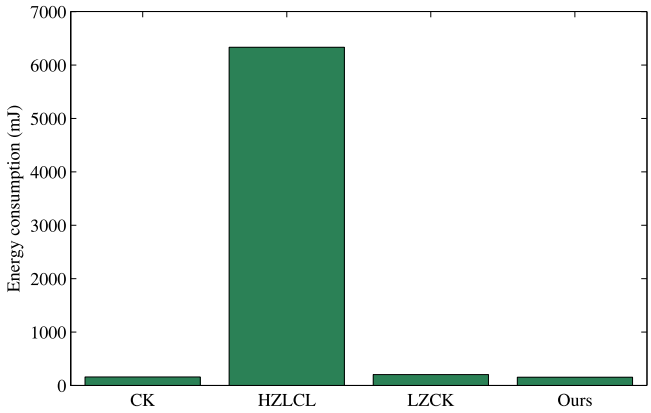


Fig. 4. Energy consumption of the controller.

data using a symmetric cipher (such as AES [34]) with the session key $H_3(r)$. This session key has been established between the user and the controller because of the CLSC scheme. This session key is only known by the user and the controller, which assures the confidentiality for future communication between them.

Because this access control scheme uses the proposed signcryption that is proved to have unforgeability in Theorem 2, an adversary cannot forge a ciphertext $\sigma$ that the controller accepts. So, our access control scheme achieves authentication, integrity and nonrepudiation. In addition, similar to [15], our access control scheme signcrypts both identity $\text{ID}_A$ and public key $\text{PK}_A$, an adversary cannot learn the identity information of the user. Therefore, our access control scheme achieves the anonymity.

### D. Revocation

The registration is revoked automatically by the expiration date ED. For example, if the expiration date ED is "2015-12-31," the user only can access the WBAN before December 31, 2015. That is, the partial private key and full private key of the user automatically become invalid after December 31, 2015. If we must revoke the user's access privilege before the expiration date due to some reasons, the SP can send a revoked identity message to the controller. The controller should keep a list of revoked identities to identify the validity of users. Of course, we also can use Tsai and Tseng's method [35] to revoke the users.

## V. ANALYSIS OF THE ACCESS CONTROL SCHEME

In this section, we evaluate the performance and security of the proposed access control scheme. First, we compare the main computational cost and communication cost of our proposed scheme with those of CK [14], HZLCL [17], and LZCK [18] that are shown in Table II. We denote by *P* the pairing operation, *M* the point multiplication operation in $G_1$, and *E* the exponentiation operation in $G_2$. The other operations are ignored in Table II since the three operations consume the most running time of the whole scheme. Let $|x|$ be the number of bits of $x$. Table II shows that our proposed scheme has less computational cost than the other three schemes for the controller. For the user part, our proposed scheme has less computational cost than CK and HZLCL and has the same computational cost as LZCK. For the communication cost of the controller, our proposed scheme is lower than HZLCL and LZCK and is higher than CK.

We give a quantitative analysis for CK [14], HZLCL [17], LZCK [18], and our proposed scheme. Here, we only consider the controller part since its resource is limited. We adopt the experimental results in [36] on MICA2 that is equipped with an ATmega128 8-bit processor clocked at 7.3728 MHz, 4-KB RAM, and 128-KB ROM. From [36], we know that a pairing operation takes 1.9 s and a exponentiation operation in $G_2$ takes 0.9 s using the supersingular curve $y^2 + y = x^3 + x$ with an embedding degree 4 and implementing $\eta_T$ pairing: $E(\mathbb{F}_{2^{271}}) \times E(\mathbb{F}_{2^{271}}) \rightarrow \mathbb{F}_{2^{4\cdot 271}}$, which is equivalent to the 80-bit security level. In addition, from [37], we know that a point multiplication operation takes 0.81 s. According to the results in [36] and [37], the computational time on the controller of CK, HZLCL, LZCK, and our proposed scheme are $3 * 1.9 + 1 * 0.81 = 6.51$ s, $(2|m| + 5) * 0.81$ s, $3 * 1.9 + 2 * 0.9 + 1 * 0.81 = 8.31$ s, and $2 * 1.9 + 1 * 0.9 + 2 * 0.81 = 6.32$ s, respectively. As in [36] and [38], we assume that the power level of MICA2 is 3.0 V, the current drawn in active mode is 8.0 mA, the current drawn in receiving mode is 10 mA, the current drawn in transmitting mode is 27 mA, and the data rate is 12.4 kbps. For energy consumption, according to the method in [24] and [39], a pairing operation consumes $3.0 * 8.0 * 1.9 = 45.6$ mJ, an exponentiation operation in $G_2$ consumes $3.0 * 8.0 * 0.9 = 21.6$

TABLE III
COMPARISON OF SECURITY

| Schemes | Security | | | | | | | Environment |
|---|---|---|---|---|---|---|---|---|
| | Anonymity | Confidentiality | Authentication | Integrity | Nonrepudiation | No Certificate | No Key Escrow | |
| CK [14] | × | √ | √ | √ | √ | √ | × | IBC |
| HZLCL [17] | × | √ | √ | √ | √ | √ | × | IBC |
| LZCK [18] | √ | × | √ | √ | √ | √ | √ | CLC |
| Ours | √ | √ | √ | √ | √ | √ | √ | CLC |

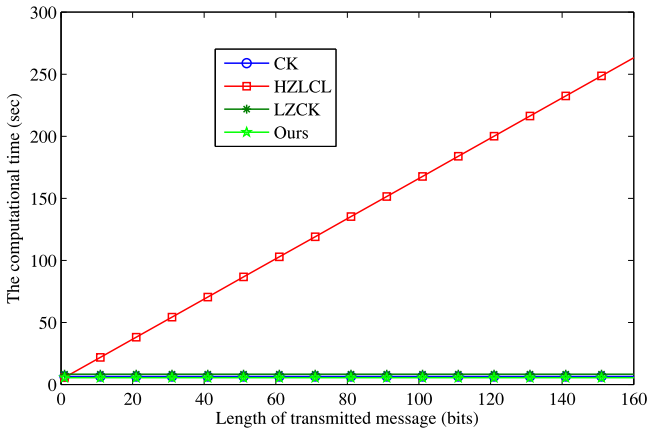


Fig. 5. Computational time versus length of transmitted message.

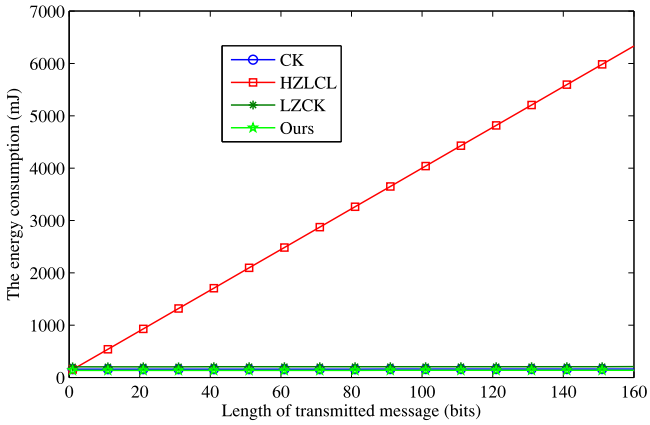


Fig. 6. Energy consumption versus length of transmitted message.

mJ, and a point multiplication operation consumes $3.0 * 8.0 * 0.81 = 19.44$ mJ. Therefore, the computational energy cost on the controller of CK, HZLCL, LZCK, and our proposed scheme are $3 * 45.6 + 1 * 19.44 = 156.24$ mJ, $(2|m| + 5) * 19.44 = 38.88|m| + 97.2$ mJ, $3 * 45.6 + 2 * 21.6 + 1 * 19.44 = 199.44$ mJ, and $2 * 45.6 + 1 * 21.6 + 2 * 19.44 = 151.68$ mJ, respectively.

For the communication cost, we assume that $|m| = 160$ bits and $|\mathrm{ID}| = 80$ bits. Since we use a curve over the binary field $\mathbb{F}_{2^{271}}$ with the $G_1$ of the 252 bits prime order. The size of an element in group $G_1$ is 542 bits and can be reduced to 34 bytes by standard compression technique [36]. The size of an element in group $G_2$ is 1084 bits. So, in CK, the controller needs to receive

$$2|G_1| + |\mathrm{ID}| + |m|\mathrm{bits} = 2 * 34 + 10 + 20\,\mathrm{bytes} = 98\,\mathrm{bytes}$$

messages. In HZLCL, the controller needs to transmit

$$4|G_1| + |G_2| + |\mathrm{ID}|\mathrm{bits} = 4 * 34 + 136 + 10\,\mathrm{bytes} = 282\,\mathrm{bytes}$$

messages. In LZCK, the controller needs to receive

$$\begin{aligned} & 4|G_1| + 2|\mathbb{Z}_p^*| + |\mathrm{ID}| + |m|\mathrm{bits} \\ = \; & 4 * 34 + 2 * 32 + 10 + 20\,\mathrm{bytes} = 230\,\mathrm{bytes} \end{aligned}$$

messages. In our proposed scheme, the controller needs to receive

$$3|G_1| + |\mathrm{ID}| + |m|\mathrm{bits} = 3 * 34 + 10 + 20\,\mathrm{bytes} = 132\,\mathrm{bytes}$$

messages. From [36], we know the controller consumes $3 * 27 * 8/12400 = 0.052$ mJ and $3 * 10 * 8/12400 = 0.019$ mJ to transmit and receive one byte messages, respectively. Therefore, in CK, HZLCL, LZCK, and our proposed scheme, communication energy consumption of the controller are $0.019 * 98 = 1.86$ mJ, $0.052 * 282 = 14.66$ mJ, $0.019 * 230 = 4.37$ mJ, and $0.019 * 132 = 2.51$ mJ. The total energy consumption of CK, HZLCL, LZCK, and our proposed schemes are $156.24 + 1.86 = 158.1$ mJ, $38.88 * 160 + 97.2 + 14.66 = 6,332.66$ mJ, $199.44 + 4.37 = 203.81$ mJ, and $151.68 + 2.51 = 154.19$ mJ, respectively.

The computational time and total energy consumption on the controller are summarized in Figs. 3 and 4, respectively (here, we assume $|m| = 160$ bits). From Figs. 3 and 4, we find that our proposed scheme has the least computational time and total energy consumption among the four schemes.

We give the relationship between the computational time and the length of transmitted message in Fig. 5 and the relationship between the total energy consumption and the length of transmitted message in Fig. 6. We find that the computational time and energy consumption increase quickly when the length of the transmitted message become bigger in HZLCL. However, the other three schemes have a little effect on the length of the transmitted message.

We compare the security properties of the four schemes in Table III. A symbol √ means that the scheme satisfies the security property and a symbol × means that the scheme does not satisfy the security property. Both CK and HZLCL cannot achieve the anonymity and have the key escrow problem. LZCK cannot achieve the confidentiality. Our proposed scheme satisfies all security properties. In addition, our proposed scheme has

neither key escrow problem nor public key certificates since it is based on the CLC environment.

## VI. Conclusion

In this paper, we first proposed a new CLSC scheme and proved its security in the random oracle model. Then, we gave a certificateless access control scheme for the WBANs using the proposed signcryption. Compared with existing three access control schemes for the WBANs, our proposed scheme has the least computational time and total energy consumption on the controller. In addition, our proposed scheme is based on the CLC environment that has neither key escrow problem nor public key certificates.

## References


[1] O. Salem, Y. Liu, A. Mehaoua, and R. Boutaba, "Online anomaly detection in wireless body area networks for reliable healthcare monitoring," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 5, pp. 1541–1551, Sep. 2014.

[2] D. He, S. Chan, and S. Tang, "A novel and lightweight system to secure wireless medical sensor networks," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 1, pp. 316–326, Jan. 2014.

[3] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless Netw.*, vol. 17, no. 1, pp. 1–18, Jan. 2011.

[4] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: A survey," *IEEE Commun. Surveys Tutorials*, vol. 16, no. 3, pp. 1658–1686, Aug. 2014.

[5] M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 51–58, Feb. 2010.

[6] C. Hu, F. Zhang, X. Cheng, X. Liao, and D. Chen, "Securing communications between external users and wireless body area networks," in *Proc. 2nd ACM Workshop Hot Topics Wireless Netw. Security Privacy*, Budapest, Hungary, 2013, pp. 31–35.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, Berkeley, CA, USA, pp. 321–334, 2007.

[8] R. Lu, X. Lin, and X. Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 614–624, Mar. 2013.

[9] H. Zhao, J. Qin, and J. Hu, "An energy efficient key management scheme for body sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2202–2210, Nov. 2013.

[10] D. He, S. Chan, Y. Zhang, and H. Yang, "Lightweight and confidential data discovery and dissemination for wireless body area networks," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 2, pp. 440–448, Mar. 2014.

[11] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "IBE-Lite: A lightweight identity-based cryptography for body sensor networks," *IEEE Trans. Inform. Technol. Biomed.*, vol. 13, no. 6, pp. 926–932, Nov. 2009.

[12] D. He and S. Zeadally, "Authentication protocol for an ambient assisted living system," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 71–77, Jan. 2015.

[13] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[14] G. Cagalaban and S. Kim, "Towards a secure patient information access control in ubiquitous healthcare systems using identity-based signcryption," in *Proc. 13th Int. Conf. Adv. Commun. Technol.*, Seoul, Korea, pp. 863–867, 2011.

[15] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Public Key Cryptography-PKC 2005* (Lecture Notes in Computer Science), vol. 3386. Belin, Germany: Springer, 2005, pp. 362–379.

[16] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost(encryption)," in *Advances in Cryptology-CRYPTO'97*, (Lecture Notes in Computer Science), vol. 1294. Berlin, Germany: Springer, 1997, pp. 165–179.

[17] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: A fuzzy attribute-based signcryption scheme," *IEEE J. Select. Areas Commun.*, vol. 31, no. 9, pp. 37–46, Sep. 2013.

[18] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 332–342, Feb. 2014.

[19] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT 2003* (Lecture Notes in Computer Science), vol. 2894. Berlin, Germany: Springer-Verlag, 2003, pp. 452–474.

[20] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Syst.*, vol. 21, no. 1, pp. 49–60, Feb. 2015.

[21] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inform. Sci.*, vol. 311, pp. 263–277, Nov. 2015.

[22] S. A. Chaudhry, K. Mahmood, H. Naqvi, and M. K. Khan, "An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography," *J. Med. Syst.*, vol. 39, no. 11, pp. 1–12, Nov. 2015.

[23] S. A. Chaudhry, H. Naqvi, T. Shon, M. Sher, and M. S. Farash, "Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems," *J. Med. Syst.*, vol. 39, no. 6, pp. 1–11, Jun. 2015, doi:10.1007/s10916-015-0244-0.

[24] C. Ma, K. Xue, and P. Hong, "Distributed access control with adaptive privacy preserving property for wireless sensor networks," *Security Commun. Netw.*, vol. 7, no. 4, pp. 759–773, Apr. 2014.

[25] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J. J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Advances in Cryptology-ASIACRYPT 2005* (Lecture Notes in Computer Science), vol. 3788. Berlin, Germany: Springer, 2005, pp. 515–532.

[26] K. Y. Choi, J. H. Park, J. Y. Hwang, and D. H. Lee, "Efficient certificateless signature schemes," in *Applied Cryptography and Network Security-ACNS 2007* (Lecture Notes in Computer Science), vol. 4521. Berlin, Germany: Springer, 2007, pp. 443–458.

[27] Y. H. Kim, H. Lee, J. H. Park, L. T. Yang, and D. H. Lee, "Key establishment scheme for sensor networks with low communication cost," in *Autonomic and Trusted Computing-ATC 2007* (Lecture Notes in Computer Science), vol. 4610. Berlin, Germany: Springer, 2007, pp. 441–448.

[28] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proc. ACM Symp. Inform., Comput. Commun. Security-ASIACCS*, Tokyo, Japan, 2008, pp. 369–372.

[29] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Advances in Cryptology-EUROCRYPT 2004*, (Lecture Notes in Computer Science), vol. 3027, Springer-Verlag, 2004, pp. 56–73.

[30] X. Boyen, "Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography," in *Advances in Cryptology-CRYPTO 2003*, (Lecture Notes in Computer Science), vol. 2729. Berlin, Germany: Springer, 2003, pp. 383–399.

[31] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.

[32] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Public Key Cryptography-PKC 2003*, (Lecture Notes in Computer Science), vol. 2567. Berlin, Germany: Springer, 2003, pp. 18–30.

[33] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure, or: Assigning roles to strangers," in *Proc. IEEE Symp. Security Privacy*, Berkeley, CA, USA, 2000, pp. 2–14.

[34] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the Advanced Encryption Standard.* Berlin, Germany: Springer, 2002.

[35] T. T. Tsai and Y. M. Tseng, "Revocable certificateless public key encryption," *IEEE Syst. J.*, vol. 9, no. 3, pp. 824–833, Sep. 2015.

[36] K. A. Shim, Y. R. Lee, and C. M. Park, "EIBAS: An efficient identity-based broadcast authentication scheme in wireless sensor networks," *Ad Hoc Netw.*, vol. 11, no. 1, pp. 182–189, Jan. 2013.

[37] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems-CHES 2004*, (Lecture Notes in Computer Science), vol. 3156. Berlin, Germany: Springer-Verlag, 2004, pp. 119–132.

[38] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks," *Comput. Commun.*, vol. 31, no. 4, pp. 659–667, Mar. 2008.

[39] K. A. Shim, "$S^2$DRP: Secure implementations of distributed reprogramming protocol for wireless sensor networks," *Ad Hoc Netw.*, vol. 19, pp. 1–8, Aug. 2014.