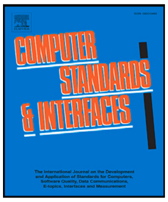
Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi



Securing Wireless Body Area Network with lightweight certificateless signcryption scheme using equality test[☆]

Zohaib Ali [a,e], Junaid Hassan [b], Muhammad Umar Aftab [a,*], Negalign Wake Hundera [b,c,d,**], Huiying Xu [c], Xinzhong Zhu [c,f,g,*]

[a] *Department of Computer Science, National University of Computer and Emerging Science, Islamabad, Chiniot-Faisalabad Campus, 35400, Pakistan*
[b] *School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China*
[c] *School of Computer Science and Technology, Zhejiang Normal University, Jinhua, 321004, China*
[d] *Zhejiang Institute of Optoelectronics, Jinhua, China*
[e] *Department of Computer Science, The University of Faisalabad, Faisalabad Campus, 38600, Pakistan*
[f] *AI Research Institute of Beijing Geekplus Technology Co., Ltd., Beijing, 100101, China*
[g] *Research Institute of Hangzhou Artificial Intelligence, Zhejiang Normal University, Hangzhou, 311231, China*

ARTICLE INFO

Keywords:
Internet of Things (IoT)
Cloud
Equality test
Wireless Body Area Network (WBAN)
Hyperelliptic curves (HEC)
Certificateless signcryption

ABSTRACT

The growth of Internet of Things (IoT) technologies, such as cloud computing, 5G communication, and wireless sensor networks, is driving a smarter and more connected future. Thousands of terabytes of data are uploaded to cloud servers each day for storage or computation. Due to data privacy, we cannot upload personal pictures, videos, locations, and medical records directly to the cloud because they will be at risk if compromised. Due to the untrusted nature of the cloud, data needs to be encrypted to ensure confidentiality before being outsourced to it. The data must first be decrypted before any operation can be performed, which can be resource-intensive and wasteful. Secure data transmission from sensors to an Internet host becomes a critical issue for the success of IoT. To address these issues, this paper introduces a lightweight certificateless signcryption scheme with an equality test (CLS-ET), which leverages the power of hyperelliptic curves. This scheme obtains the security goals of authentication, integrity, confidentiality, and non-repudiation in one logical step. Furthermore, this scheme enables us to verify whether two ciphertexts are encrypted with the same or different keys that contain the same information without decrypting them. Indistinguishability under adaptive chosen ciphertext attack (IND-CCA2), existential unforgeability under chosen message attack (EUF-CMA), and one-wayness under adaptive chosen ciphertext attack (OW-CCA2) level security have been achieved by the proposed scheme in the Random Oracle Model (ROM). Furthermore, we compared our proposed scheme with other existing state-of-the-art schemes. While maintaining security and functionality, our scheme reduces computation costs for encryption, decryption, and testing stages, thereby improving efficiency in resource-constrained IoT-enabled Wireless Body Area Networks.

1. Introduction

Cloud computing is gaining popularity due to recent technological advancements, such as the Internet of Things (IoT), 5G communication, and Wireless Sensor Networks (WSNs). WSNs, often integrated as part of IoT, consist of dedicated sensor nodes that monitor and record the data and transfer collected data to a central location. A Wireless Body Area Network (WBAN) is a specialized type of WSN designed for healthcare monitoring. It serves as a core component in numerous telehealth applications, including personalized healthcare and home-based mobile health services. WBANs can also utilize an equality test to evaluate a patient's health status [1]. An overview of a working WBAN with an equality test server is shown in Fig. 1. Patients are equipped with various sensors to collect real-time or continuous physiological health data, such as blood pressure, glucose levels, breathing rate,

[☆] This work was supported by the National Natural Science Foundation of China (62376252); Zhejiang Province Leading Geese Plan (2025C02025, 2025C01056); Zhejiang Province Province-Land Synergy Program (2025SDXT004-3).

[*] Corresponding authors.

[**] Corresponding author at: School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China.

E-mail addresses: umar.aftab@nu.edu.pk (M.U. Aftab), nigaccna21@zjnu.edu.cn (N.W. Hundera), zxz@zjnu.edu.cn (X. Zhu).

https://doi.org/10.1016/j.csi.2025.104070
Received 25 April 2025; Received in revised form 18 July 2025; Accepted 28 August 2025
Available online 17 September 2025

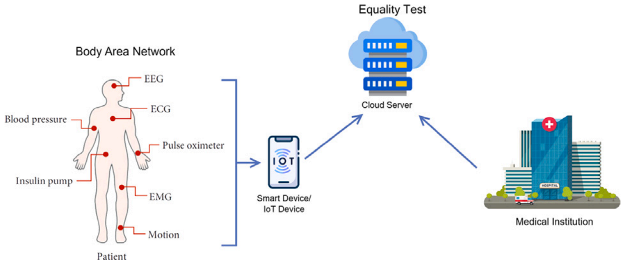


Fig. 1. Standard WBAN scenario.

electrocardiogram (ECG), and motion [2–4]. This data is then wirelessly transferred to an IoT device, where it is encrypted and sent to a cloud server for storage. The cloud server, which also receives encrypted data from medical institutions, performs an equality test to determine the patient's health status. If the test indicates equality, the patient's status is considered normal; if not, the patient's status is deemed abnormal. This process not only enhances the quality and efficiency of healthcare delivery but also reduces treatment costs.

Globally, cloud-based systems are widely used to manage and process vast amounts of data. In the e-healthcare sector, cloud computing has become the most common solution for managing and facilitating communication among IoT devices [5]. Cloud servers are utilized to store large volumes of data and perform computations on it. However, due to the untrusted nature of cloud environments, directly uploading personal data such as pictures, videos, location information, and medical records poses significant risks if the cloud server is compromised.

Data security and efficiency are significant challenges in resource-constrained IoT-enabled Wireless Body Area Networks (WBANs). The confidentiality of data can be achieved by applying encryption before outsourcing it to the cloud. However, there is a limitation in data recovery because of the "all-or-nothing" decryption characteristic [6]. Most IoT devices are battery-powered with limited storage and processing capabilities, which exacerbates these challenges. To improve this process, Boneh [7] introduced the notion of PKE-KS, which integrates keyword search with public key encryption and retrieves their information without decrypting the data. Using keyword search, the cloud can perform a test to check whether two ciphertexts carry the same information or not. However, this approach has a drawback as it does not work when two ciphertexts are encrypted with different public keys. Unfortunately, this scheme becomes unsuitable for cloud searching, due to the heterogeneous nature of IoT data. To address this issue, Yang [8] proposed the notion of public key encryption with an equality test (PKE-ET), which supports search operations among ciphertexts encrypted with both the same and distinct public keys. This method is more suitable for the heterogeneous nature of IoT data. Nevertheless, the scheme is built under the framework of a public key infrastructure (PKI), which requires digital certificates to verify the validity of public keys. To eliminate the need for certificates and enhance efficiency, Ma [9] introduced the first Identity-Based Encryption with Equality Test (IBEET). Building on this foundation, researchers have developed various schemes to address emerging security challenges. In response to threats posed by quantum computers, Z. Yang [10] proposed a lattice-based IBEET scheme to enhance cloud service security against quantum threats. Additionally, in response to the COVID-19 pandemic, Ramadan [11] introduced the WBAN-19 scheme for telemedicine systems, designed to secure telemedicine systems and reduce the widespread transmission of contagious diseases.

With the growing prevalence of IoT-enabled Wireless Sensor Networks (WSNs), the usage of cloud computing is also increasing. Due to the untrusted nature of the cloud, data must be encrypted before being outsourced. While Identity-Based Encryption (IBE) can eliminate the need for certificates, practical IBE schemes often rely on bilinear pairings as a mathematical tool [12,13]. However, bilinear operations are significantly more computationally expensive than point multiplication, which poses a challenge given the resource-constrained nature of IoT-enabled WSNs. Therefore, there is a pressing need to improve the efficiency of existing schemes in terms of computational cost and message overhead. In this paper, a Lightweight Certificateless Signcryption scheme with equality test for WBAN (CLS-ET) is proposed to address these challenges. Our scheme relies on hyperelliptic curve cryptography (HECC), which eliminates the need for pairing operations during the signcryption and unsigncryption stages. The experimental findings indicate that the bilinear pairing computation cost is much higher as compared to both the Rivest, Shamir, and Adleman (RSA) and elliptic curve cryptography (ECC) methods by 13.65 ms and 13.93 ms, respectively [14]. Furthermore, RSA's computation cost is higher than that of HECC by 14.42 ms [15]. HECC, with an 80-bit key size, provides a security level equivalent to a 1024-bit RSA key and a 160-bit ECC key, but with lower computational costs and communication overhead.

1.1. Related work

The growth of Internet of Things (IoT) technologies, including cloud computing and wireless sensor networks, is driving the creation of a smarter and more connected future. However, data privacy is a concern, and users often encrypt sensitive information before storing it in the cloud. Most IoT devices are resource-constrained, when it comes to memory, storage, energy, and processing power, and are mostly battery-powered. Various schemes have been put forward to guarantee the quick retrieval of encrypted data from the cloud.

In an effort to solve this problem, Boneh [7] proposed the first public key encryption scheme with the functionality of keyword search. Sadly, this technique becomes inconvenient for cloud searching, due to the diverse nature of IoT data. To solve this issue, Yang [8] proposed public key encryption with an equality test (PKE-ET) scheme. However, these schemes are constructed under a public key infrastructure (PKI), requiring digital certificates to confirm the validity of the public keys. The certificate management cost incurred by PKI is unfavorable for resource-constrained IoT-enabled WSNs with limited storage and computing capacity. This is because the demand for public key certificates (involving storage, distribution, and revocation) is high, and additional time is spent verifying a public key before it can be used.

Shamir [16] proposed an Identity-based encryption (IBE) scheme to eliminate certificate management. In IBE, each user uses their own identity (Name, Email, EMI number, etc.) as the public key. In order to ensure data confidentiality and achieve efficiency, Ma [9] presented the first identity-based encryption with the functionality to perform an equality test (IBEET) scheme. But this scheme faces user revocation and key escrow problems. Afterward, numerous studies regarding IBBEET have been published in the literature [17–19]. Most notably, Ramadan [20] proposed an ID-based encryption scheme, IBEET-RSA for Wireless Body Area Networks (WBANs). The scheme is built on RSA and has the security of OW-ID-CCA in the RO model. It presents a promising solution for ensuring medical data security and privacy in WBANs.

Key escrow and user revocations are the inherent problems with ID-based cryptography (IBC). In an effort to solve the user revocation problem, Sun [21] proposed a scheme that provides user revocation and consumes less bandwidth, storage, and other resources because both the ciphertext and key are short. The security of the scheme is accordant with the Chinese SM9 encryption standard and has the hardness assumption of the BDH (Bilinear Diffie–Hellman) problem. Subsequently, another scheme called RIBEET for wireless body area networks (WBANs) was proposed [22]. To solve the user key escrow problem, Elhabob [23] proposed CL-PKE-ET scheme for the Internet of Vehicles (IoV) environment. The scheme was based on the original CL-PKC scheme proposed by Al-Riyami [24]. The user's private key is split into two parts to resolve the key-escrow problem. Key generator centers (KGCs) create the first part, while users create the second part. The user can make the complete private key by combining them. The scheme has demonstrated IND-CCA and OW-CCA level security in the RO model. Elhabob [25] further proposed a pairing-free CL-PKE-ET protocol, which offers superior performance compared to its predecessor scheme. Additionally, Tian [26] proposed a lightweight certificateless encryption scheme with keyword search and equality test (CLAE-KS&ET), providing enhanced security against message recovery attacks for cloud environments while supporting secure ciphertext retrieval and comparison without decryption. However, most existing schemes are unsuitable for applications within IoT-enabled Wireless Sensor Networks (WSNs) due to their high computational costs, message overhead, and storage demands.

1.2. Contributions

1. We propose a novel Certificateless Signcryption scheme with Equality Test (CLS-ET) specifically designed for Wireless Body Area Networks (WBANs). In this scheme, the cloud server can use the ciphertext form of the patient's health and medical institution's data to perform an equality test and check whether the patient's status is normal or abnormal, indicating the need for medical attention.
2. Based on the syntax of [27], we propose a novel framework and concrete construction for the CLS-ET scheme specifically designed for WBANs. Our construction is optimized to meet the unique requirements of WBANs within IoT environments.
3. The proposed scheme employs Hyperelliptic Curve Cryptography (HECC) with an 80-bit key size, offering significant efficiency improvements over traditional elliptic curve cryptography (ECC) and bilinear pairing methods, which require larger key sizes 160-bits and 256-bits, respectively.
4. Our scheme effectively addresses the inherent key escrow issue associated with Identity-Based Encryption (IBE), enhancing the security and practicality of the proposed system.
5. We provide rigorous security proofs and analysis, demonstrating that our scheme achieves IND-CCA2, EUF-CMA, and OW-CCA2 levels of security within the Random Oracle Model (ROM).
6. Through extensive evaluation, we show that the proposed CLS-ET scheme outperforms existing state-of-the-art schemes in terms of computational efficiency and message overhead, while maintaining robust security. This makes our scheme particularly well-suited for resource-constrained IoT-enabled WSNs.

1.3. Paper organization

The rest of this paper is organized as follows. Preliminaries are given in Section 2. The framework and security model are presented in Section 3. The concrete construction of the scheme is detailed in Section 4. A security analysis is provided in Section 5. Details of the test environment and a comparative analysis of our proposed scheme against other existing state-of-the-art schemes are presented in Sections 6 and 7, respectively. Conclusions are drawn in Section 8.

2. Preliminaries

2.1. Hyperelliptic curve

The hyperelliptic curve (HEC) is a special class of algebraic curves introduced by Koblitz [28]. HEC can be considered a generalized or shorter key version of [29]. Unlike ECC, the points on HEC are not derived from a group [30]. In HEC, the additive Abelian group is computed from the divisor, which results in smaller parameters and key sizes compared to ECC. Despite these smaller parameters, HEC can perform all essential operations required in a public-key cryptosystem, including signature generation, encryption, decryption, and key exchange. Importantly, the hyperelliptic curve provides the same level of security as RSA, bilinear pairing, and elliptic curves, making it particularly well-suited for resource-constrained IoT environments [31].

A curve with a genus value of 1 is commonly referred to as an elliptic curve (EC). In contrast, hyperelliptic curves are defined over curves with a genus greater than 1 [32]. For instance, a curve with a genus of 1 over a finite field $\mathcal{F}_\mathfrak{q}$, the group order of the field $|\mathcal{F}_\mathfrak{q}|$ requires operands of length 160 bits. This requirement implies that $\mathfrak{g}. \log_2 q \approx 2^{160}$, where $\mathfrak{g}$ represents the genus of the curve within the finite field $\mathcal{F}_\mathfrak{q}$. Similarly, a curve with a genus of 2 is called a hyperelliptic curve (HEC) and requires 80-bit long operands within the field $\mathcal{F}_\mathfrak{q}$, where $\mathfrak{g} \cdot \log_2 q \approx 2^{80}$.

HEC is a special type of non-singular and projective curve. The hyperelliptic curve defined over the field $\mathcal{F}_\mathfrak{q}$ can be represented by points $(\mathfrak{w}, \mathfrak{v}) \in \mathcal{F}_\mathfrak{q}$, which satisfy the following equation:

$$HEC : \ \mathfrak{v}^2 + h(\mathfrak{w})\mathfrak{v} = f(\mathfrak{w}) \tag{1}$$

where f and h are both polynomials in the field $\mathcal{F}_\mathfrak{q}$ with $\deg(f) = 2\,g{+}1$ and $\deg(h) \leq g$. The curve also satisfies both Eq. (1) and the partial derivative equations $h'(\mathfrak{w}) = 0$ and $h'(\mathfrak{w})\mathfrak{v} + f'(\mathfrak{w}) = 0$.

2.2. Complexity assumptions

We have considered the following assumptions while conducting the analysis:

- $\mathcal{F}_\mathfrak{q}$ is a finite field with the order q, where $q \approx 2^{80}$.
- $\mathcal{D}$ is a divisor of the hyperelliptic curve (HEC) selected from the Jacobian group, which is the finite sum of points $\mathfrak{p}_i \in$ HEC as:

$$\mathcal{D} = \sum_{\mathfrak{p}_{i \in \mathrm{HEC}}} \mathfrak{m}_i \mathfrak{p}_i \tag{2}$$

where $\mathfrak{m}_i \in \mathcal{F}_\mathfrak{q}$.

Definition 1. Given $(\mathcal{D}, A = a \cdot \mathcal{D}, B = b \cdot \mathcal{D}) \in \mathcal{J}(\mathcal{C})$, compute $Z = ab\mathcal{D} \in \mathcal{G}$.

The HC-CDHP assumption holds if: No probabilistic polynomial-time (PPT) algorithm can solve the HC-CDHP by computing $ab\mathcal{D}$ from $(\mathcal{D}, a\mathcal{D}, b\mathcal{D})$ with non-negligible probability.

Definition 2. Suppose $\partial \in \{1, 2, 3, 4, 5, \dots, q-1\}$ is randomly picked. The value of Λ is calculated using Eq. (3).

$$\Lambda = \partial \cdot \mathcal{D} \tag{3}$$

The probability of finding the value of ∂ from Λ is negligible due to the Hyperelliptic Curve Discrete Logarithm Problem (HE-CDLP).

3. Framework and security model of CLS-ET

The syntax of our scheme, "A Lightweight Certificateless Signcryption scheme with Equality Test (CLS-ET) for WBANs", is based on the "Efficient CL-PKC-ET for IoV" scheme [27]. The scheme consists of eight algorithms: Setup, Private Number Generation, Partial Private Key Generation, Full Key Generation, Certificateless Signcryption, Trapdoor, Test, and Certificateless Unsigncrypt. The first three roles, Setup, Private Number Generation, and Partial Private Key Generation are handled by the Key Generation Center (KGC), which is responsible for performing these tasks. The second set of roles is assigned to the users, who can perform Full Key Generation, Signcryption, Unsigncryption, and Trapdoor generation. The final role is assigned to the Medical Record Management Server (MRMS), a cloud server responsible for storing and maintaining medical records from patients and medical institutions, as well as performing Equality Tests on the signcrypted data sent by patients and medical institutions. The functioning of the scheme is illustrated in Fig. 2, with the data flow described as follows: The KGC initializes the system parameters and distributes them to all entities. Each user sends their identity to the KGC (See step ①), after which the KGC generates and sends the partial private key to each user (See step ②). The user then generates a full private key from this partial key. Patients are equipped with various sensors that collect real-time medical data and transfer it to a smart device/IoT (See step ③). The collected data is then signcrypted and sent to the MRMS server (See step ④). The user generates a trapdoor using their private key and sends it to the MRMS along with the signcrypted data (See step ⑤). Similarly, medical institutions also send signcrypted data along with a trapdoor to the MRMS for the Equality Test. The MRMS has two primary tasks: record management and performing Equality Tests. It receives the signcrypted data and trapdoors from patients and medical institutions and checks if the equality holds (See step ⑥). It then sends the result, either “true” or “false”, to the Medical Institution (See step ⑦). If the result is “true”, the patient’s status is normal; if “false”, the patient’s status is abnormal and requires medical attention. In the case of a “false” result, the MRMS sends the patient’s signcrypted data to the Medical Institution for further analysis of the patient’s health (See step ⑧).

Table 1 lists notable notations used in this scheme, and the descriptions of each of these algorithms are provided below.

1. **Setup:** In this phase, the KGC randomly picks its master secret key α and publishes the system parameters params $= \{\text{HEC}, \mathcal{F}_q, \mathcal{G}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$.
2. **Private Number Generation:** In this phase, the KGC receives the users’ IDs, randomly selects β.
3. **Partial Private Key Generation:** This phase is performed by the KGC. It takes as input the user’s identity ID_u along with other parameters, and then returns and sends the partial private key ω to all users according to their IDs via a secure channel.
4. **Full Key Generation:** This algorithm is performed by the users. Each user receives their corresponding partial private key ω from the KGC, computes the full private and public keys, and then further checks the validity of the public and private keys.
5. **Certificateless Signcryption:** The sender executes this algorithm. It utilizes the sender’s private key Pr_{cls}, system parameters *params*, the recipient’s public encryption key EPb_{clus}, and the message M to produce the signcrypted output Ω. The algorithm integrates several security elements such as confidentiality, integrity, non-repudiation, and authentication into one logical process. The sender creates a ciphertext intended for the receiver.
6. **Trapdoor:** This algorithm is performed by the users. The algorithm takes input Pr_u and generates trapdoor T_u as output.
7. **Test:** The MRMS Server executes this algorithm by taking two pairs of ciphertext-trapdoor (CT_A, T_A) and (CT_B, T_B) from two users with ID_A and ID_B, respectively. If the equality holds, it returns an output of 1; otherwise, it returns 0.
8. **Certificateless Unsigncrypt:** This algorithm is performed by the receiver. It takes as input the signcrypted ciphertext Ω, public system parameters *params*, the receiver’s private key Pr_{clus} and returns the original message M. If Ω is not invalid, it will return the symbol $\perp$.

3.1. Security model

This section establishes the security models for the proposed CLS-ET scheme, specifically regarding the IND-CCA2, EUF-CMA, and OW-CCA2 security. Here, C refers to the challenger, while $\mathcal{A}_1$ and $\mathcal{A}_2$ represent Type-I and Type-II adversaries involved. Let us play some games between Challenger C and Adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ to prove the security of our proposed scheme.

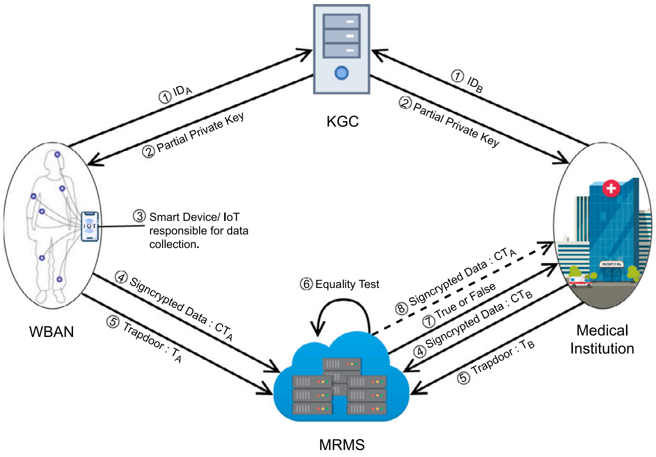


Fig. 2. Architecture of a lightweight certificateless signcryption scheme with equality test for the WBANs.

Table 1
Notation used in this scheme.

Notation	Meaning
params	System parameters
KGC	Key Generation Center
$\mathcal{H}_i$	ith one-way hash function, where $i = 1, 2, 3$
$\mathcal{F}_q$	Finite field $\mathcal{F}_q$ of order q
q	Large prime number
α	The secret key of KGC
D	HEC’s Divisor
$\mathcal{G}$	The cyclic group of prime order q
$\mathcal{N}_{\text{non}}$	Fresh nonce value
ID_{cls}, ID_{clus}	Identity of the Sender (signcrypter) and receiver (unsigncrypter)
Pb_{cls}, Pb_{clus}	Public key of the Sender (signcrypter) and receiver (unsigncrypter)
Pr_{cls}, Pr_{clus}	Private key of the Sender (signcrypter) and receiver (unsigncrypter)
CT_u, T_u	Pairs of ciphertext-trapdoor from users
M	Message
Ω	Ciphertext (signcrypted message)
$\perp$	Decryption failure

Definition 3. It is possible for a signcryption scheme to achieve IND-CCA2, if there exists $\mathcal{A}_1$ adversary, who can query $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, $\mathcal{H}_4$, setup $\mathcal{Q}_{\text{setup}}$, private number generation $\mathcal{Q}_{\text{PNG}}$, partial private key generation $\mathcal{Q}_{\text{PPKG}}$, full key generation $\mathcal{Q}_{\text{FKG}}$, certificateless-signcryption $\mathcal{Q}_{\text{cls}}$, and certificateless-unsigncryption $\mathcal{Q}_{\text{clus}}$ oracles for $\mathcal{Q}_{h1}$, $\mathcal{Q}_{h2}$, $\mathcal{Q}_{h3}$, $\mathcal{Q}_{h4}$, $\mathcal{Q}_{setup}$, $\mathcal{Q}_{PNG}$, $\mathcal{Q}_{PPKG}$, $\mathcal{Q}_{FKG}$, $\mathcal{Q}_{cls}$, and $\mathcal{Q}_{clus}$, respectively, who is capable of winning the IND-CCA2 game in time ϵ with a success probability τ in a probabilistic polynomial time.

- **IND-CCA2 Game:**
  Setup: The setup algorithm is run by the challenger C and takes the security parameter ψ as input. C picks a random number as a secret key $a = \alpha$. Then pick four hash functions $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, and $\mathcal{H}_4$. Finally, C send some public parameters, such as $\psi \in \{\text{HEC}, \mathcal{F}_q, G, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$ to $\mathcal{A}_1$.
  Phase 1: $\mathcal{A}_1$ issues $\mathcal{H}_i$ hash queries as $(i = 1, 2, 3, 4)$, setup $\mathcal{Q}_{\text{setup}}$, private number generation $\mathcal{Q}_{\text{PNG}}$, partial private key generation $\mathcal{Q}_{\text{PPKG}}$, full key generation $\mathcal{Q}_{\text{FKG}}$, certificateless-signcryption $\mathcal{Q}_{\text{cls}}$, and certificateless unsigncryption $\mathcal{Q}_{\text{clus}}$ queries for sender identity ID_s and randomly chosen message M. In response to these queries, C generates a private number, partial private key, and full

key for sender identity ID_s and also answers the certificateless-signcryption and certificateless unsigncryption queries and sends the results to $\mathcal{A}_1$.
Challenge: $\mathcal{A}_1$ chooses two equal lengths but dissimilar types of messages M_1 and M_2 and the sender's identity ID'_s and sends it to C. C runs private number generation, partial private key generation, and full key generation algorithms. Then randomly selects a bit $f \in \{0,1\}$ to produce certificateless-signcryption ciphertext Ω' and sends it to $\mathcal{A}_1$. Note that M_1, M_2, and ID'_s should be fresh and not from the pair (M_1, ID'_s) or (M_2, ID'_s).
Phase 2: In this phase, $\mathcal{A}_1$ makes the same queries as aforementioned in phase 1, except the certificateless unsigncryption query $\mathcal{Q}_{\mathrm{clus}}$ for the targeted ciphertext Ω'.
Guess: $\mathcal{A}_1$ outputs a bit $f' \in \{0,1\}$, and if $f' = f$, then $\mathcal{A}_1$ has succeeded. If $f' \neq f$, then the algorithm terminates without any output. The advantage of $\mathcal{A}_1$ winning the game is negligible.

Definition 4. It is possible for a signcryption scheme to achieve EUF-CMA, if there exists $\mathcal{A}_1$ adversary, who can query $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, $\mathcal{H}_4$, setup $\mathcal{Q}_{\mathrm{setup}}$, private number generation $\mathcal{Q}_{\mathrm{PNG}}$, partial private key generation $\mathcal{Q}_{\mathrm{PPKG}}$, full key generation $\mathcal{Q}_{\mathrm{FKG}}$, and certificateless-unsigncryption $\mathcal{Q}_{\mathrm{clus}}$ oracles for $\mathcal{Q}_{h1}$, $\mathcal{Q}_{h2}$, $\mathcal{Q}_{h3}$, $\mathcal{Q}_{h4}$, $\mathcal{Q}_{setup}$, $\mathcal{Q}_{PNG}$, $\mathcal{Q}_{PPKG}$, $\mathcal{Q}_{FKG}$, and $\mathcal{Q}_{clus}$, respectively, who is capable of winning the EUF-CMA game in time ϵ with a success probability τ in a probabilistic polynomial time.

- **EUF-CMA Game:**
Setup: In this phase, C executes similar tasks as performed in the Game IND-CCA2 setup phase.
Attack: $\mathcal{A}_1$ issues $\mathcal{Q}_{\mathrm{setup}}$, $\mathcal{Q}_{\mathrm{PNG}}$, $\mathcal{Q}_{\mathrm{PPKG}}$, $\mathcal{Q}_{\mathrm{FKG}}$, and $\mathcal{Q}_{\mathrm{cls}}$ queries for sender identity ID_s and randomly chosen message M. In response to these queries, C generates a private number, partial private key, and full key for sender identity ID_s and also runs the certificateless-signcryption algorithm to generate ciphertext Ω, and sends it to $\mathcal{A}_1$.
Forgery: In response to the message M', $\mathcal{A}_1$ outputs a certificateless-signcrypted ciphertext and message pair (Ω', M'). For a sender with identity ID'_s and message M', $\mathcal{A}_1$ can win the game if Ω' is a valid certificateless-signcrypted ciphertext, provided that the sender's private key and a tuple (M', ID'_s) have not been accessed before through any query.

Definition 5. In the proposed WBAN-19 scheme, the plaintext remains secure even when the adversary possesses the trapdoor and the corresponding ciphertext. Thus, the scheme is considered one-wayness under adaptive chosen ciphertext attack (OW-CCA2) in the random oracle model, provided the advantage of adversary $\mathcal{A}_2$ in distinguishing between messages is negligible

- **OW-CCA2 Game:**
Setup: The setup algorithm is run by the challenger C and takes the security parameter ψ as input. C picks a random number as a secret key $a = \alpha$. Then pick four hash functions $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, and $\mathcal{H}_4$. Finally, C send some public parameters, such as $\psi \in \{\mathrm{HEC}, \mathcal{F}_q, G, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$ to $\mathcal{A}_2$.
Phase 1: $\mathcal{A}_2$ issues $\mathcal{H}_i$ hash queries as $(i = 1,2,3,4)$, setup $\mathcal{Q}_{\mathrm{setup}}$, private number generation $\mathcal{Q}_{\mathrm{PNG}}$, partial private key generation $\mathcal{Q}_{\mathrm{PPKG}}$, full key generation $\mathcal{Q}_{\mathrm{FKG}}$, Trapdoor-Queries $\mathcal{Q}_{\mathrm{trapdoor}}$, and certificateless unsigncryption $\mathcal{Q}_{\mathrm{clus}}$ queries for sender identity ID_s and randomly chosen message M. In response to these queries, C generates a private number, partial private key, and full key for sender identity ID_s and also answers the trapdoor and certificateless unsigncryption queries and sends the results to $\mathcal{A}_2$.
Challenge: The challenger, C, randomly selects a plaintext message $M' \in M$ and computes the corresponding ciphertext tuple Ω using the signcrypt algorithm. The ciphertext tuple Ω is then sent to the adversary $\mathcal{A}_2$.
Phase 2: The challenger answers similarly to Phase 1; however, during this phase, $\mathcal{A}_2$ is restricted from making queries related to the secret key and the plaintext message.
Guess: The adversary $\mathcal{A}_2$ outputs a guess M' for the original plaintext message M.

4. Proposed scheme

In this section, an efficient Certificateless Signcryption scheme with the functionality of an equality test for WBAN is introduced. The deployment of our scheme is shown in Fig. 3. A detailed mathematical explanation of each algorithm employed in the scheme is provided below.

- **Setup:** In this phase, the Key Generation Center (KGC) picks a random number as a secret key $\alpha \in \{1,2,3,\dots,q-1\}$. Then, KGC freely creates a set of public parameters, such as params = $\{\mathrm{HEC}, \mathcal{F}_q, \mathcal{G}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$ and $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4$ are random one-way hash functions defined as: $\mathcal{H}_1 : \{0,1\}' \to \mathcal{F}_q$, $\mathcal{H}_2 : \mathcal{G} \to \{0,1\}^k$, $\mathcal{H}_3 : \mathcal{G} \to \{0,1\}^n$, and $\mathcal{H}_4 : \{0,1\}' \to \mathcal{F}_q$, where $k = \lceil \log_2 q \rceil$. Let $\Phi : \mathcal{F}_q \to \{0,1\}^k$ be the function that maps an integer to its binary representation padded to length k, and $\Phi^{-1} : \{0,1\}^k \to \mathcal{F}_q$ as its inverse.
- **Private Number Generation:** In this phase, KGC receives the User's IDs and randomly selects $\beta \in \{1,2,3,\dots,q-1\}$ as a private number.
- **Partial Private Key Generation:** In this phase, KGC calculates $\omega = \alpha \cdot \beta \pmod q$ as a partial private key, and sends it to all the users according to their IDs through a secure channel.
- **Full Key Generation:** Users receive their corresponding partial private key ω, and then it picks two random number $x, y \in \{1,2,3,\dots,q-1\}$ as its private key and calculates $\Delta = y \cdot \omega \pmod q$ after that it calculate their public encryption key $\mathrm{EPb}_u = y \cdot \mathcal{D}$ and their full public and private keys as follows:

$$\mathrm{Pb}_u = \mathcal{H}_1(\omega \| \mathrm{ID}_u) \cdot \mathcal{D} + x \cdot \mathcal{D}$$

and

$$\mathrm{Pr}_u = x \cdot \mathrm{Pb}_u.$$

The private key pair = (y, Pr_u), and the public key pair = $(\mathrm{EPb}_u, \mathrm{Pb}_u)$ of the user.
- **Certificateless Signcryption:** In this phase, the certificateless signcrypter takes as input its own private key Pr_A, the receiver's encryption public key EPb_B, as well as a plaintext message M. It outputs a signcrypted tuple $\Omega = \{U, \Gamma, Y, \mu, V\}$ by following these steps.
First, this algorithm randomly picks two numbers $s_n, \eta_n \in \{1,2,3,\dots,q-1\}$ and computes:
Compute $m_n = \mathcal{H}_4(M_n)$

$$R_n = (\eta_n \cdot \Delta) \cdot \mathrm{Pb}_n$$

$$Z_n = \eta_n \cdot \mathrm{EPb}_B$$

$$Y_n = \mathcal{H}_4(M_n) \cdot \Delta \cdot \mathrm{Pr}_A$$

$$\mu_n = \eta_n \cdot \omega \pmod q$$

$$U_n = \eta_n \cdot \mathcal{D}$$

$$\Gamma_n = \Phi(\eta \cdot m_n) \oplus \mathcal{H}_2(s_n \cdot R_n)$$

$$V_n = (M_n \| \mathcal{N}_{\mathrm{nonce}}) \oplus \mathcal{H}_3(Y_n) \oplus \mathcal{H}_3(Z_n)$$

After making all the calculations, the certificateless signcrypter sends the tuple $\Omega_n = \{U_n, \Gamma_n, Y_n, \mu_n, V_n\}$ through a secure channel to the certificateless unsigncrypter.

- **Certificateless Unsigncryption:** In this phase, the certificateless unsigncrypter receives the tuple $\Omega_n = \{U_n, \Gamma_n, Y_n, \mu_n, V_n\}$ and takes its own private key y_B, and calculates:

  $Z'_n = y_B \cdot U_n$

  If $Z'_n = Z_n$, then calculate:

  $M' = V \oplus \mathcal{H}_3(Y) \oplus \mathcal{H}_3(Z')$

  Otherwise, abort this algorithm.
- **Trapdoor:** For the given (CT_A, ID_A) and (CT_B, ID_B). The trapdoors for users A and B are calculated by this algorithm, as follows:

  $T_A = s_A \cdot \mathrm{Pr}_A \quad \text{and} \quad T_B = s_B \cdot \mathrm{Pr}_B,$

  where s_A and s_B are the random numbers chosen during signcryption by A and B, respectively.
- **Test:** For the given (CT_A, T_A) and (CT_B, T_B). The entity ET runs this algorithm as follows:

  1. Compute: $\Theta_A = \mathcal{H}_2(\mu_A \cdot T_A)$ and $\Theta_B = \mathcal{H}_2(\mu_B \cdot T_B)$.
  2. Compute $\chi'_A = \Gamma_A \oplus \Theta_A$ and $\chi'_B = \Gamma_B \oplus \Theta_B$.
  3. Compute $\chi_A = \Phi^{-1}(\chi'_A)$ and $\chi_B = \Phi^{-1}(\chi'_B)$.
  4. ET checks if $\chi_B \cdot U_A = \chi_A \cdot U_B$ holds. If the equivalence holds, then the server will return 1. If not, the server will return 0.

- **Correctness:** The proposed scheme demonstrates consistency through the following proof:
  (1) Signcryption:

  $Z'_n = y_B \cdot U_n$

  $= y_B \cdot (\eta_n \cdot \mathcal{D}) = (\eta_n \cdot y_B) \cdot \mathcal{D} = Z_n$

  and finally, calculates M'_n:

  $M'_n = V_n \oplus \mathcal{H}_3(Y_n) \oplus \mathcal{H}_3(Z'_n)$

  $= (M_n \| \mathcal{N}_{\text{nonce}}) \oplus \mathcal{H}_3(Y_n) \oplus \mathcal{H}_3(Z_n) \oplus \mathcal{H}_3(Y_n) \oplus \mathcal{H}_3(Z'_n)$

  $= M_n \| \mathcal{N}_{\text{nonce}}$

  since $Z'_n = Z_n$, so $\mathcal{H}_3(Z'_n) = \mathcal{H}_3(Z_n)$.
  (2) Equality Test:

  $\chi'_n = \Gamma_n \oplus \mathcal{H}_2(\mu_n \cdot T_n)$

  $= \Phi(\eta \cdot m) \oplus \mathcal{H}_2(R_n \cdot s_n) \oplus \mathcal{H}_2(\mu_n \cdot T_n)$

  $= \Phi(\eta \cdot m) \oplus \mathcal{H}_2(Pb_n \cdot \eta_n \cdot \Delta \cdot s_n) \oplus \mathcal{H}_2(\eta_n \cdot \omega \cdot s_n Pr_n)$

  $= \Phi(\eta \cdot m) \oplus \mathcal{H}_2(Pb_n \cdot \eta_n \cdot y \cdot \omega \cdot s_n) \oplus \mathcal{H}_2(\eta_n \cdot \omega \cdot s_n \cdot y \cdot Pb_n)$

  $= \Phi(\eta \cdot m) \oplus \mathcal{H}_2(Pb_n \cdot \eta_n \cdot y \cdot \omega \cdot s_n) \oplus \mathcal{H}_2(Pb_n \cdot \eta_n \cdot y \cdot \omega \cdot s_n)$

  $= \Phi(\eta \cdot m)$

  where $m_n = \mathcal{H}_4(M_n)$, and since $s_n \cdot R_n = \mu_n \cdot T_n$ (as shown in analysis, due to commutative scalars), $\mathcal{H}_2(s_n \cdot R_n) = \mathcal{H}_2(\mu_n \cdot T_n)$, so:

  $\chi'_n = \Phi(\eta_n \cdot m_n) \oplus 0 = \Phi(\eta_n \cdot m_n)$

  Then,

  $\chi_n = \Phi^{-1}(\chi'_n) = \eta_n \cdot m_n$

  Now, check:

  $\chi_B \cdot U_A = (\eta_B \cdot m_B) \cdot (\eta_A \cdot \mathcal{D}) = (\eta_A \eta_B m_B) \cdot \mathcal{D}$

  $\chi_A \cdot U_B = (\eta_A \cdot m_A) \cdot (\eta_B \cdot \mathcal{D}) = (\eta_A \eta_B m_A) \cdot \mathcal{D}$

  Equality holds if $\eta_A \eta_B m_B = \eta_A \eta_B m_A$, so $m_B = m_A$, i.e., $\mathcal{H}_4(M_B) = \mathcal{H}_4(M_A)$, implying $M_A = M_B$ with high probability if $\mathcal{H}_4$ is collision-resistant.

5. Security analysis

In the random oracle model, we can achieve IND-CCA2, EUF-CMA, and OW-CCA2 using a cyclic group $\mathcal{G}$ of prime order q and Divisor $\mathcal{D}$, as shown in the following two theorems [33,34]. In this section, we discuss the formal security analyses of our proposed scheme based on the Hyperelliptic Curve Computational Diffie–Hellman Problem (HC-CDHP) assumptions.

Theorem 1. *If a probabilistic polynomial-time (PPT) adversary $\mathcal{A}_1$ can break the IND-CCA2 security of the proposed CLS-ET scheme with a non-negligible advantage ϵ, then a challenger $\mathcal{C}$ can be constructed to solve the Hyperelliptic Curve Computational Diffie–Hellman Problem (HC-CDHP) with an advantage.*

$$\epsilon' \geq \frac{(2\epsilon - \mathcal{Q}_{clun}/q^2)}{(\mathcal{Q}_{h2} + \mathcal{Q}_{h3})}$$

$$\tau' \approx \tau + \tau_\lambda(\mathcal{Q}_{cls} + \mathcal{Q}_{clus} + \mathcal{Q}_{h2} + \mathcal{Q}_{h3})$$

where τ_λ is the average oracle query running time.

Proof. We will show that if such an adversary $\mathcal{A}_1$ exists, a challenger $\mathcal{C}$ can use $\mathcal{A}_1$ as a subroutine to solve an instance of the HC-CDH problem. The challenger $\mathcal{C}$ receives an instance of the HC-CDHP: a tuple $(\mathcal{D}, A = a \cdot \mathcal{D}, B = b \cdot \mathcal{D}) \in \mathcal{J}(C)$, where $\mathcal{D}$ is a base divisor and a, b are unknown scalars. $\mathcal{C}$'s goal is to compute $Z = ab\mathcal{D}$. The challenger $\mathcal{C}$ is interacting with $\mathcal{A}_1$ as.

Setup: This algorithm runs by the challenger $\mathcal{C}$ and takes the security parameter params as input and does the following steps to generate some public parameters. It defines the Hyper-elliptic curve HEC$/\mathcal{F}_q$ over prime finite field $\mathcal{F}_q$. Let $\mathcal{G}$ be a cyclic group over $\mathcal{F}_q$ where $\mathcal{D}$ is the Divisor of $\mathcal{G}$. Challenger picks a random number as a secret key $a = \alpha$. It also chooses four cryptographic hash functions, denoted as $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, and $\mathcal{H}_4$ modeled as a random oracle model (ROM). Finally, $\mathcal{C}$ sends some public parameters for encryption and decryption, such as params $\in \{\text{HEC}, \mathcal{F}_q, \mathcal{G}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$, to $\mathcal{A}_1$.

Phase 1: In this phase, the adversary $\mathcal{A}_1$ issues some queries to challenger $\mathcal{C}$, and $\mathcal{C}$ maintains four hash lists $list_1$, $list_2$, $list_3$, $list_4$ and answers their queries as follows.

- $\mathcal{H}_1$ *Queries*: $\mathcal{C}$ preserves the $list_1$ of tuple $(\omega, \text{ID}_u) \in list_1$, and upon $\mathcal{H}_1$ Query, it checks (ω, ID_u) exists in $list_1$ or not, if exists, then the value of $\mathcal{H}_1$ from the $list_1$ is returned otherwise, a random $\mathcal{H}_1 \in \{1, 2, 3, \dots, q-1\}$ is returned and added $(\omega, \text{ID}_u, \mathcal{H}_1)$ to the $list_1$.
- $\mathcal{H}_2$ *Queries*: $\mathcal{C}$ preserves the $list_2$ of tuple $(R_i) \in list_2$, and upon $\mathcal{H}_2$ Query, if R_i exists in the list, returns the stored $\mu_i = H_2(R_i)$ to A. Otherwise randomly pick $\mu_i \in \{0, 1\}'$ and returns $\mu_i = H_2(R_i)$ and added $(R_i, \mu_i, \mathcal{H}_2)$ to the $list_2$.
- $\mathcal{H}_3$ *Queries*: $\mathcal{C}$ preserves the $list_3$ of tuple $(Y_i) \in list_3$, and upon $\mathcal{H}_3$ Query, if Y_i exists in the list, returns the stored $\lambda_i = H_3(Y_i)$ to A. Otherwise randomly pick $\lambda_i \in \{0, 1\}'$ and returns $\lambda_i = H_3(Y_i)$ and added $(Y_i, \lambda_i, \mathcal{H}_3)$ to the $list_3$.
- $\mathcal{H}_4$ Queries: $\mathcal{C}$ preserves the list_4 of tuples $\mathcal{H}_4(M) \in \text{list}_4$, and upon a $\mathcal{H}_4$ query, it checks whether $\mathcal{H}_4(M)$ exists in list_1. If it exists, the value of $\mathcal{H}_4$ from list_4 is returned; otherwise, a random $\mathcal{H}_4 \in \{1, 2, 3, \dots, q-1\}$ is returned and $\mathcal{H}_4(M)$ is added to list_4.
- $\mathcal{PNG}$ *Queries*: When $\mathcal{A}_1$ queries for a private number for an identity ID_u: $\mathcal{C}$ generates a random integer $\beta_u \in \{1, 2, \dots, q-1\}$. It stores the pair (ID_u, β_u) in the list L_{PNG} and returns β_u to $\mathcal{A}_1$.
- $\mathcal{PPKG}$ *Queries*: When $\mathcal{A}_1$ queries for the partial private key for an identity ID_u:

  - If $ID_u = ID'_s$, $\mathcal{C}$ must abort the simulation. The EUF-CMA security model forbids the adversary from requesting the private key (or its components) of the identity it intends to attack.

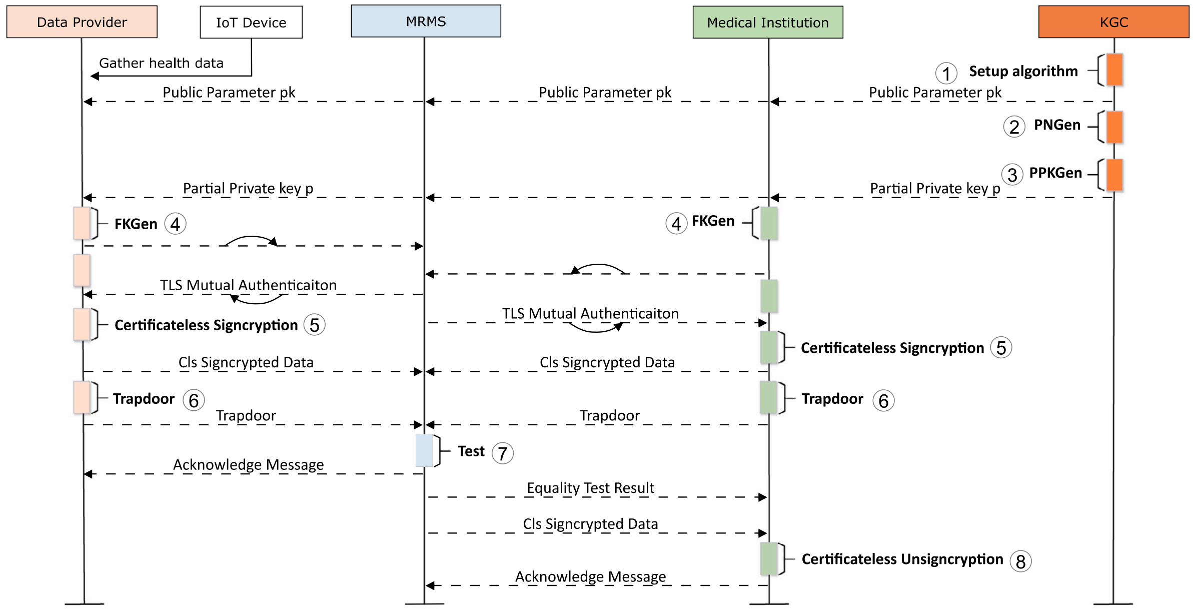


Fig. 3. Deployment of the proposed CLS-ET scheme.

  - If $ID_u \neq ID'_s$, $\mathcal{C}$ generates a random integer $\omega'_u \in \{1, 2, \dots, q-1\}$, stores the tuple (ID_u, ω'_u) in L_{Key}, and returns ω'_u to $\mathcal{A}_1$.

- $\mathcal{FKG}$ *Queries*: When $\mathcal{A}_1$ requests the public key for an identity ID_u:
  - If $ID_u = ID'_s$, the challenger embeds the second part of the HC-CDHP instance. It sets the public key $Pb'_s = B = b\mathcal{D}$. It returns Pb'_s to $\mathcal{A}_1$. $\mathcal{C}$ does not know the corresponding private key, which is expected.
  - If $ID_u \neq ID'_s$, $\mathcal{C}$ generates a complete, valid key pair. It chooses random secrets $x_u, y_u \in \{1, 2, \dots, q-1\}$. It uses the simulated partial private key ω'_u from the $\mathcal{Q}_{\text{PPKG}}$ simulation to compute the public key $Pb_u = \mathcal{H}_1(\omega'_u \| ID_u)\mathcal{D} + x_u\mathcal{D}$. The full private key is $Pr_u = x_u \cdot Pb_u$. $\mathcal{C}$ stores all these components in L_{Key} and returns the public parts to $\mathcal{A}_1$.

- Certificateless Signcryption Queries: When $\mathcal{A}_1$ issues a signcryption query,
  - $\mathcal{C}$ checks if $\text{ID}_u = \text{ID}'_u$, then $\mathcal{C}$ embed the HC-CDH problem part in Pb_u as $\text{Pb}_u = \mathcal{H}_1(\omega \| \text{ID}_u) \cdot \mathcal{D} + B$ and compute the remaining ciphertext part as calculated in the real signcryption algorithm and return the certificateless-signcryption tuple $\Omega' = \{U', \Gamma', Y', \mu', V'\}$ to $\mathcal{A}_1$.
  - $\mathcal{C}$ checks if $\text{ID}_u \neq \text{ID}'_u$, then it picks two numbers $s, \eta \in \{1, 2, 3, \dots, q-1\}$ and computes: $m = \mathcal{H}_4(M)$, $R = (\eta \cdot \Delta) \cdot \text{Pb}_n$, $Z = \eta \cdot \text{EPb}_B$, $Y = \mathcal{H}_1(M) \cdot \Delta \cdot \text{Pr}_A$, $\mu = \eta \cdot \omega \pmod{q}$, $U = \eta \cdot \mathcal{D}$, $\Gamma = \text{bin}(\eta \cdot m), \oplus \mathcal{H}_2(s \cdot R)$ where $m_n = \mathcal{H}_4(M_n)$, $V = (M \| \mathcal{N}_{\text{nonce}}) \oplus \mathcal{H}_3(Y) \oplus \mathcal{H}_3(Z)$, and sends the certificateless-signcryption tuple $\Omega = \{U, \Gamma, Y, \mu, V\}$ to $\mathcal{A}_1$.

- Certificateless Unsigncryption Queries: When $\mathcal{A}_1$ issues a unsigncryption query, $\mathcal{C}$ checks if $\text{ID}_u \neq \text{ID}'_u$, then M is returned. Otherwise, the following steps are performed:
  1. Calculates $Z' = y_B \cdot U$.
  2. Finally, computes the message $M' = V \oplus \mathcal{H}_3(Y) \oplus \mathcal{H}_3(Z')$ and return it to $\mathcal{A}_1$.

Challenge: M_1 and M_2 are two equal-length but dissimilar messages chosen by $\mathcal{A}_1$. $\mathcal{A}_1$ also chooses the sender's identity ID'_s and sends it to $\mathcal{C}$. Upon receiving the messages M_1 and M_2 and identity ID'_s, $\mathcal{C}$ randomly selects a bit $f \in \{0, 1\}$ and produces a certificateless-signcryption tuple $\Omega' = \{U', \Gamma', Y', \mu', V'\}$ for the message M_f using the following process: First, $\mathcal{C}$ embed the HC-CDH problem part in Pb_u as $\text{Pb}_u = \mathcal{H}_1(\omega \| \text{ID}_u) \cdot \mathcal{D} + B$, then it randomly picks two numbers $s, \eta \in \{1, 2, 3, \dots, q-1\}$ and computes: Compute $m' = \mathcal{H}_4(M)$, $R' = (\eta \cdot \Delta) \cdot \text{Pb}_n$, $Z' = \eta \cdot \text{EPb}_B$, $Y' = \mathcal{H}_1(M) \cdot \Delta \cdot \text{Pr}_A$, $\mu' = \eta \cdot \omega \pmod{q}$, $U' = \eta \cdot \mathcal{D}$, $\Gamma' = \Phi(\eta \cdot m) \oplus \mathcal{H}_2(s \cdot R)$, $V' = (M \| \mathcal{N}_{\text{nonce}}) \oplus \mathcal{H}_3(Y) \oplus \mathcal{H}_3(Z)$, after making all the calculations, the certificateless-signcrypter sends the tuple $\Omega' = \{U', \Gamma', Y', \mu', V'\}$ to $\mathcal{A}_1$.

Phase 2: In this phase, $\mathcal{A}_1$ made the identical queries as aforementioned in Phase 1, except the certificateless unsigncryption query $\mathcal{Q}_{\text{clus}}$ for the targeted ciphertext $\Omega' = \{U', \Gamma', Y', \mu', V'\}$. $\mathcal{C}$ answers all the queries upon receiving them from $\mathcal{A}_1$ except $\mathcal{Q}_{\text{clus}}$ with ID'_{cls}.

Guess: $\mathcal{A}_1$ outputs a bit $f' \in \{0, 1\}$, and if $f' = f$, then it is clear that $\mathcal{A}_1$ has succeeded and can calculates $Z' = y_B \cdot U_n$, $Y_n = \mathcal{H}_4(M_n) \cdot \Delta \cdot Pr_A$, and $V_n = (M_n, \mathcal{N}_{\text{nonce}}) \oplus \mathcal{H}_3(Y_n) + \mathcal{H}_3(Z_n)$, and Finally, can computes the message $M'_n = V_n \oplus \mathcal{H}_3(Y_n) \oplus \mathcal{H}_3(Z'_n)$. The following equation can achieve the HC-CDHP solution: $T = (\eta y \beta)^{-1}(R - \eta \Delta \mathcal{H}_1(\omega \| ID) \cdot \mathcal{D})$, it is easy to deduce that $T = ab\mathcal{D}$ if $R = (\eta \cdot \Delta) \cdot \text{Pb}$ Therefore, the CLS-ET scheme is secure against IND-CCA2.

Theorem 2. *If the HC-CDHP assumption holds in the Jacobian group of the hyperelliptic curve, then the proposed CLS-ET scheme is secure against EUF-CMA in the Random Oracle Model (ROM).*

Proof. Let $\mathcal{A}_1$ be a probabilistic polynomial-time (PPT) adversary that can break the EUF-CMA security of the scheme with a non-negligible advantage ϵ. We will construct a challenger $\mathcal{C}$ that can use $\mathcal{A}_1$ to solve an instance of the HC-CDHP. The challenger $\mathcal{C}$ is given an HC-CDHP instance, which is a tuple $(\mathcal{D}, A = a\mathcal{D}, B = b\mathcal{D})$, and its goal is to compute $Z = ab\mathcal{D}$.

Setup: $\mathcal{C}$ takes the HC-CDHP instance $(\mathcal{D}, A, B)$. It sets the KGC's master public key $P_{\text{pub}} = A = a\mathcal{D}$. This implicitly sets the KGC's master secret key α to the unknown value a. $\mathcal{C}$ randomly selects a target identity ID'_s from the space of possible identities. This is the identity for which $\mathcal{A}_1$ will attempt to create a forgery. $\mathcal{C}$ initializes empty lists to simulate the random oracles: L_1, L_2, L_3, L_4 for the hash functions $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4$, respectively. It also initializes a list L_{PNG} to track private numbers and L_{Key} to track generated keys. $\mathcal{C}$ sends the system parameters `params` (including P_{pub}) to the adversary $\mathcal{A}_1$.

Queries (Attack Phase) $\mathcal{A}_1$ can issue a polynomial number of queries, which $\mathcal{C}$ answers as follows.

- *Hash Queries*: $\mathcal{C}$ respond to all the $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4$, hash queries as in Theorem 1.
- $\mathcal{PNG}$ *Queries*: When $\mathcal{A}_1$ queries for a private number for an identity ID_u: $\mathcal{C}$ generates a random integer $\beta_u \in \{1, 2, \dots, q-1\}$. It stores the pair (ID_u, β_u) in the list L_{PNG} and returns β_u to $\mathcal{A}_1$.
- $\mathcal{PPKG}$ *Queries*: When $\mathcal{A}_1$ queries for the partial private key for an identity ID_u:
  - If $ID_u = ID'_s$, $\mathcal{C}$ must abort the simulation. The EUF-CMA security model forbids the adversary from requesting the private key (or its components) of the identity it intends to attack.
  - If $ID_u \neq ID'_s$, $\mathcal{C}$ generates a random integer $\omega'_u \in \{1, 2, \dots, q-1\}$, stores the tuple (ID_u, ω'_u) in L_{Key}, and returns ω'_u to $\mathcal{A}_1$.
- $\mathcal{FKG}$ *Queries*: When $\mathcal{A}_1$ requests the public key for an identity ID_u:
  - If $ID_u = ID'_s$, the challenger embeds the second part of the HC-CDHP instance. It sets the public key $Pb'_s = B = b\mathcal{D}$. It returns Pb'_s to $\mathcal{A}_1$. $\mathcal{C}$ does not know the corresponding private key, which is expected.
  - If $ID_u \neq ID'_s$, $\mathcal{C}$ generates a complete, valid key pair. It chooses random secrets $x_u, y_u \in \{1, 2, \dots, q-1\}$. It uses the simulated partial private key ω'_u from the Q_{PPKG} simulation to compute the public key $Pb_u = \mathcal{H}_1(\omega'_u \| ID_u)\mathcal{D} + x_u\mathcal{D}$. The full private key is $Pr_u = x_u \cdot Pb_u$. $\mathcal{C}$ stores all these components in L_{Key} and returns the public parts to $\mathcal{A}_1$.
- Certificateless Signcryption Queries: When $\mathcal{A}_1$ issues a signcryption query,
  - $\mathcal{C}$ checks if $\text{ID}_u = \text{ID}'_u$, then $\mathcal{C}$ embeds the second part of the HC-CDHP instance. It sets the public key $Pb'_s = B = b\mathcal{D}$, and compute the remaining ciphertext part as calculated in the real signcryption algorithm and return the certificateless-signcryption tuple $\Omega = \{U, \Gamma, Y, \mu, V\}$ to $\mathcal{A}_1$.
  - If $ID_s \neq ID'_s$, $\mathcal{C}$ has all the necessary key components (stored in L_{Key}) for the sender ID_s. It follows the certificateless signcryption algorithm in the paper to generate a valid tuple Ω and returns it to $\mathcal{A}_1$.

Forgery: After making its queries, the adversary $\mathcal{A}_1$ outputs a forged signcryption tuple $\Omega' = (U', \Gamma', Y', \mu', V')$ for a new message M' under the target sender identity ID'_s. For the forgery to be valid: The signcryption tuple Ω' must be verifiable as correct. $\mathcal{A}_1$ must not have queried the partial private key for ID'_s. $\mathcal{A}_1$ must not have requested a signcryption for the pair (M', ID'_s) from the Q_{cls} oracle.

Analysis: Now, the challenger $\mathcal{C}$ uses the forged tuple Ω' to compute $ab\mathcal{D}$. From the forgery, $\mathcal{C}$ parses the component Y'. According to the signcryption algorithm, this component is calculated as: $Y' = \mathcal{H}_1(M') \cdot \Delta' \cdot Pr'_s$. Once $\mathcal{C}$ determines the combined term $K = (\mathcal{H}_1(M') \cdot y'_s \cdot x'_s \cdot \beta'_s)$, it can compute the solution to the HC-CDHP instance by calculating: $ab\mathcal{D} = K^{-1} \cdot Y'$. Since the adversary $\mathcal{A}_1$ can produce a valid forgery with non-negligible probability ϵ, the challenger $\mathcal{C}$ can successfully solve the HC-CDHP instance with a related non-negligible probability. This contradicts the assumption that the HC-CDHP is hard. Therefore, no such adversary $\mathcal{A}_1$ can exist, and the CLS-ET scheme is secure against EUF-CMA.

Theorem 3. *If the HC-CDHP assumption holds in the Jacobian group of the hyperelliptic curve, then the proposed CLS-ET scheme is secure against OW-CCA2 in the Random Oracle Model (ROM).*

Proof. Assume that $\mathcal{A}_2$ is a PPT adversary capable of breaking the OW-CCA2 security of our scheme with non-negligible advantage ϵ. Suppose there exists a challenger $\mathcal{C}$ who claims to solve HC-CDHP in polynomial time using $\mathcal{A}_2$ as a subroutine. Then, $\mathcal{A}_2$ and $\mathcal{C}$ engage in the following OW-CCA2 security game as.

Setup: The challenger $\mathcal{C}$ receives an instance of the HC-CDHP: a tuple $(\mathcal{D}, A = a \cdot \mathcal{D}, B = b \cdot \mathcal{D}) \in \mathcal{J}(C)$, where $\mathcal{D}$ is a base divisor and a, b are unknown scalars. $\mathcal{C}$'s goal is to compute $Z = ab\mathcal{D}$. This algorithm runs by the challenger $\mathcal{C}$ and picks a random number as a secret key $a = \alpha$. It also chooses four cryptographic hash functions, denoted as $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, and $\mathcal{H}_4$ modeled as a random oracle model (ROM). Finally, $\mathcal{C}$ sends some public parameters for encryption and decryption, such as params $\in \{\text{HEC}, \mathcal{F}_q, \mathcal{G}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$, to $\mathcal{A}_2$.

Query Phase: $\mathcal{A}_2$ adaptively queries the following oracles:

- $\mathcal{H}_1$ *Query*: If $(\omega, ID_u, h_1) \in L_{\mathcal{H}_1}$, return h_1. Else, pick $h_1 \in \mathbb{F}_q$, store (ω, ID_u, h_1), and return h_1.
- $\mathcal{H}_2$ *Query*: If $(R_i, \mu_i, h_2) \in L_{\mathcal{H}_2}$, return h_2. Else, pick $h_2 \in \mathbb{F}_q$, store (R_i, μ_i, h_2), and return h_2.
- $\mathcal{H}_3$ *Query*: If $(Y_i, \lambda_i, h_3) \in L_{\mathcal{H}_3}$, return h_3. Else, pick $h_3 \in \mathbb{F}_q$, store (Y_i, λ_i, h_3), and return h_3.
- $\mathcal{H}_4$ *Query*: If $(M, h_4) \in L_{\mathcal{H}_4}$, return h_4. Else, pick $h_4 \in \mathbb{F}_q$, store (M, h_4), and return h_4.

- $\mathcal{PNG}$ *Queries*: When $\mathcal{A}_2$ queries for a private number for an identity ID_u: $\mathcal{C}$ generates a random integer $\beta_u \in \{1, 2, \dots, q-1\}$. It stores the pair (ID_u, β_u) in the list L_{PNG} and returns β_u to $\mathcal{A}_2$.
- $\mathcal{PPKG}$ *Queries*: When $\mathcal{A}_2$ queries for the partial private key for an identity ID_u:
  - If $ID_u = ID'_s$, $\mathcal{C}$ must abort the simulation. The EUF-CMA security model forbids the adversary from requesting the private key (or its components) of the identity it intends to attack.
  - If $ID_u \neq ID'_s$, $\mathcal{C}$ generates a random integer $\omega'_u \in \{1, 2, \dots, q-1\}$, stores the tuple (ID_u, ω'_u) in L_{Key}, and returns ω'_u to $\mathcal{A}_2$.
- $\mathcal{FKG}$ *Queries*: When $\mathcal{A}_2$ requests the public key for an identity ID_u:
  - If $ID_u = ID'_s$, the challenger embeds the second part of the HC-CDHP instance. It sets the public key $Pb'_s = B = b\mathcal{D}$. It returns Pb'_s to $\mathcal{A}_2$. $\mathcal{C}$ does not know the corresponding private key, which is expected.
  - If $ID_u \neq ID'_s$, $\mathcal{C}$ generates a complete, valid key pair. It chooses random secrets $x_u, y_u \in \{1, 2, \dots, q-1\}$. It uses the simulated partial private key ω'_u from the Q_{PPKG} simulation to compute the public key $Pb_u = \mathcal{H}_1(\omega'_u \| ID_u)\mathcal{D} + x_u\mathcal{D}$. The full private key is $Pr_u = x_u \cdot Pb_u$. $\mathcal{C}$ stores all these components in L_{Key} and returns the public parts to $\mathcal{A}_2$.

- Trapdoor Queries: When $\mathcal{A}_2$ submits a ciphertext tuple $\Omega \neq \Omega'$, the challenger $\mathcal{C}$ returns the trapdoor $T_n = s \cdot \Pr$ to $\mathcal{A}_2$.
- Certificateless Unsigncryption Queries: When $\mathcal{A}_2$ issues a unsigncryption query, $\mathcal{C}$ checks if $\text{ID}_u \neq \text{ID}'_u$, then M is returned. Otherwise, the following steps are performed:
  1. Calculates $Z' = y_B \cdot U$.
  2. Finally, computes the message $M' = V \oplus \mathcal{H}_3(Y) \oplus \mathcal{H}_3(Z')$ and return it to $\mathcal{A}_2$.

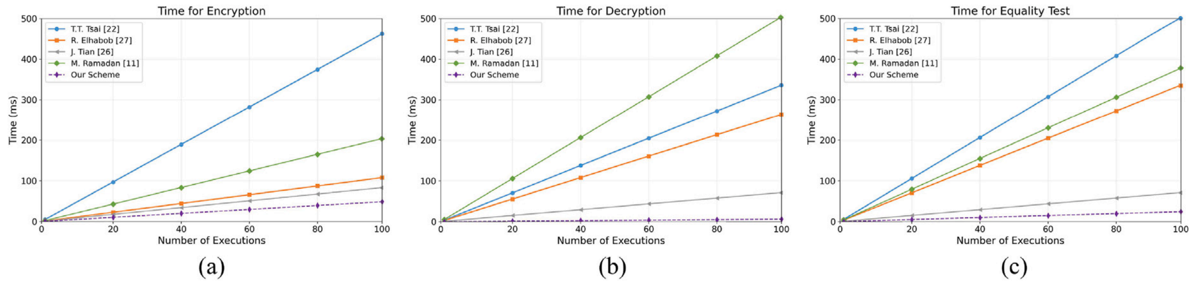


Fig. 4. Comparison of computational cost: (a) Time for encryption, (b) Time for decryption, and (c) Time for test.

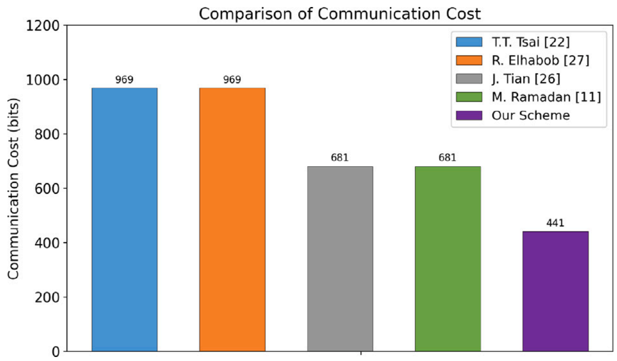


Fig. 5. Comparison of communication cost.

Challenge: M_1 and M_2 are two equal-length but dissimilar messages chosen by $\mathcal{A}_2$. $\mathcal{A}_2$ also chooses the sender's identity ID'_s and sends it to C. Upon receiving the messages M_1 and M_2 and identity ID'_s, C randomly selects a bit $f \in \{0,1\}$ and produces a certificateless-signcryption tuple $\Omega' = \{U', \Gamma', Y', \mu', V'\}$ for the message M_f using the following process: First, C embed the HC-CDH problem part in Pb_u as $\mathrm{Pb}_u = \mathcal{H}_1(\omega \| \mathrm{ID}_u) \cdot \mathcal{D} + B$, then it randomly picks two numbers $s, \eta \in \{1, 2, 3, \ldots, q-1\}$ and computes: Compute $m' = \mathcal{H}_4(M)$, $R' = (\eta \cdot \Delta) \cdot \mathrm{Pb}_n$, $Z' = \eta \cdot \mathrm{EPb}_B$, $Y' = \mathcal{H}_1(M) \cdot \Delta \cdot \mathrm{Pr}_A$, $\mu' = \eta \cdot \omega \pmod{q}$, $U' = \eta \cdot \mathcal{D}$, $\Gamma' = \Phi(\eta \cdot m) \oplus \mathcal{H}_2(s \cdot R)$, $V' = (M \| \mathcal{N}_{\text{nonce}}) \oplus \mathcal{H}_3(Y) \oplus \mathcal{H}_3(Z)$, after making all the calculations, the certificateless-signcrypter sends the tuple $\Omega' = \{U', \Gamma', Y', \mu', V'\}$ to $\mathcal{A}_2$.

Phase 2: In this phase, $\mathcal{A}_2$ made the identical queries as aforementioned in Phase 1, except the certificateless unsigncryption query $\mathcal{Q}_{\text{clus}}$ for the targeted ciphertext $\Omega' = \{U', \Gamma', Y', \mu', V'\}$. C answers all the queries upon receiving them from $\mathcal{A}_2$ except $\mathcal{Q}_{\text{clus}}$ with $\mathrm{ID}'_{\text{cls}}$.

Guess: $\mathcal{A}_2$ outputs a bit $f' \in \{0,1\}$, and if $f' = f$, then it is clear that $\mathcal{A}_2$ has succeeded and can calculates $Z' = y_B \cdot U_n$, $Y_n = \mathcal{H}_4(M_n) \cdot \Delta \cdot Pr_A$, and $V_n = (M_n, \mathcal{N}_{\text{nonce}}) \oplus \mathcal{H}_3(Y_n) + \mathcal{H}_3(Z_n)$, and Finally, can computes the message $M'_n = V_n \oplus \mathcal{H}_3(Y_n) \oplus \mathcal{H}_3(Z'_n)$. The following equation can achieve the HC-CDHP solution: $T = (\eta y \beta)^{-1}(R - \eta \Delta \mathcal{H}_1(\omega \| ID) \cdot \mathcal{D})$, it is easy to deduce that $T = abD$ if $R = (\eta \cdot \Delta) \cdot \mathrm{Pb}$ Therefore, the CLS-ET scheme is secure against OW-CCA2.

6. Test environment

6.1. Measurement tools

We used the charm-crypto library [35], the PBC library [36], and the G2HEC library [37] for benchmarking purposes. The code is mostly written in Python and C coding languages. The charm-crypto library is a Python-based library that internally utilizes other libraries such as the PBC library [36], the GMP library [38], and the OpenSSL library [39] to provide secure arithmetic operations and cryptographic parameters necessary for cryptographic schemes. We chose to use the charm-crypto library because it contains a vast collection of cryptographic primitives and functions, greatly simplifying the implementation process. For the calculation of hyperelliptic curve divisor multiplication operation, the libg2hec library [37] version 1.0.1 was used, which internally utilizes the NTL library [40]. Python was chosen as the programming language because it is fully compatible with the charm-crypto library and offers ease of use and simplicity. To ensure maximum compatibility, we are using version 0.50 of the charm-crypto library and Python version 3.7, as this is the latest version of Python that is compatible with the charm-crypto library.

Table 2
Average running times of various operations.

Symbols	Operations	Laptop 4200U @ 2.6 GHz
Mul	Point multiplication	0.11912 ms
Exp	Exponential operation	0.42102 ms
Pairing	Pairing operation	1.26010 ms
Dmul	Divisor multiplication	0.06069 ms

6.2. Measurement environment

A laptop PC with an Intel Core i5-4200U Processor running @ 2.6 GHz with 8 GB DDR3L RAM was configured as a test environment. The system was installed with the Linux-based Ubuntu 19.10 64-bit operating system, which comes with pre-installed Python 3.7. Ubuntu OS was chosen because it is a free and open-source operating system known for its user-friendly interface and extensive community support.

6.3. Measurement technique

To compute the computational cost of different cryptographic operations such as Pairing, Point Multiplication (Mul), Exponentiation (Exp), and HEC Divisor Multiplication (Dmul), we run the benchmark test. After getting the costs of these cryptographic operations, as provided in Table 2, the number of different types of operations each scheme uses is counted. Then, to compute the computational cost of each scheme, we multiplied the computed value of each cryptographic operation by the number of different operations each scheme uses to get the overall cost of each scheme.

7. Performance analysis

In this section, a comparison is made between the proposed CLS-ET scheme and other alternative schemes proposed by T.T. Tsai [22], R. Elhabob [27], J. Tian [26], and M. Ramadan [11] in terms of computation cost, communication cost, and functionality.

7.1. Computational cost

The Table 3 compares different schemes based on their computational costs for Encryption/Signcryption, Decryption/Unsigncryption, and Equality test phases. The detailed features of each scheme are presented in Table 5. When compared with the schemes of T.T. Tsai [22],

Table 3
Detailed comparison of computational cost in milliseconds.

Schemes	Encryption	Decryption	Test
T.T. Tsai [22]	2 ·Pairing + 5 ·Exp (4.625 ms)	2 ·Pairing + 2 ·Exp (3.362 ms)	4 ·Pairing (5.040 ms)
R. Elhabob [27]	2 ·Exp + 2 ·Mul (1.080 ms)	2 ·Pairing + 1 ·Mul (2.639 ms)	2 ·Pairing + 2 ·Exp (3.362 ms)
J. Tian [26]	7 ·Mul (0.834 ms)	6 ·Mul (0.715 ms)	6 ·Mul (0.715 ms)
M. Ramadan [11]	3 ·Mul + 1 ·Exp + 1 ·Pairing (2.039 ms)	4 ·Pairing (5.040 ms)	3 ·Pairing (3.780 ms)
Our scheme	8 ·Dmul (0.486 ms)	1 ·Dmul (0.061 ms)	4 ·Dmul (0.243 ms)

Table 4
Communication cost comparison in bits.

Schemes	Communication cost	Communication cost in bits
T.T. Tsai [22]	1(ID) + 1(K) + 2(M) + 1(TD) + 1(ET)	1(256) + 1(256) + 2(100) + 1(256) + 1(1) (969 bits)
R. Elhabob [27]	1(ID) + 1(K) + 2(M) + 1(TD) + 1(ET)	1(256) + 1(256) + 2(100) + 1(256) + 1(1) (969 bits)
J. Tian [26]	1(ID) + 1(K) + 2(M) + 1(TD) + 1(ET)	1(160) + 1(160) + 2(100) + 1(160) + 1(1) (681 bits)
M. Ramadan [11]	1(ID) + 1(K) + 2(M) + 1(TD) + 1(ET)	1(160) + 1(160) + 2(100) + 1(160) + 1(1) (681 bits)
Our scheme	1(ID) + 1(K) + 2(M) + 1(TD) + 1(ET)	1(80) + 1(80) + 2(100) + 1(80) + 1(1) (441 bits)

Table 5
Feature comparison of different proposed schemes.

Schemes	Cryptosystem	KS	ET	Fix KEP	Fix CMP
T.T. Tsai [22]	IBC-based	✓	✓	×	✓
R. Elhabob [27]	CLC-based	✓	✓	✓	✓
J. Tian [26]	CLC-based	✓	✓	✓	✓
M. Ramadan [11]	IBC-based	✓	✓	×	✓
Our scheme	CLC-based	✓	✓	✓	✓

R. Elhabob [27], J. Tian [26], and M. Ramadan [11], our scheme reduces the computation cost in the Encryption/Signcryption phase by 89.49%, 55.00%, 41.72%, and 76.16% respectively. Similarly, during the Decryption/Unsigncryption phase, our scheme achieves a reduction in computation cost by 98.19%, 97.69%, 91.47%, and 98.79%, respectively. Moreover, our scheme reduces the computation cost in the Equality Test phase, by 95.18%, 92.77%, 66.01% and 93.57%, respectively. Our proposed scheme employs Hyperelliptic Curve Cryptography (HECC) with an 80-bit key size and does not require any Pairing operations. By analyzing Table 3 and the accompanying Fig. 4, it becomes evident that Our scheme stands with significantly reduced computational requirements in comparison to the other schemes. It achieves Encryption/Signcryption with 8 · Dmul (0.486 ms), Decryption/Unsigncryption with 1 · Dmul (0.061 ms), and Equality Testing with 4 · Dmul (0.243 ms), making it more efficient for resource-constrained IoT-enabled WSNs.

7.2. Communication cost

The communication cost of various schemes is compared in Table 4. In all schemes, the message length was consistently set at 100 bits. Upon analyzing Table 4 and the accompanying Fig. 5, it becomes evident that our scheme significantly outperforms others in terms of communication cost reduction, and when compared with M. Ramadan [11] and J. Tian [26], our scheme boasts an impressive 35.24% reduction. Moreover, in comparison with T.T. Tsai [22] and R. Elhabob [27], our scheme demonstrates a remarkable 54.49% reduction in communication cost. These results highlight the significant efficiency of our scheme in terms of communication costs.

7.3. Property comparison

The Table 5 shows the comparison of different proposed schemes for addressing issues related to the use of cloud computing in IoT-enabled WSNs. The schemes are evaluated based on the type of cryptosystem used, as well as various features such as keyword search (KS) and equality testing (ET). The Table 5 also indicates whether each scheme addresses key escrow (KEP) and certificate management problems (CMP) and whether it supports equality testing. Our proposed scheme encompasses all major features and, being based on Certificateless Cryptography (CLC), effectively addresses key escrow and certificate management problems.

8. Conclusion

In this work, we proposed a lightweight certificateless signcryption scheme with equality test (CLS-ET) for WBANs. Our scheme incorporates the notions of certificateless Signcryption with the Equality Test, enabling the test between two ciphertexts encrypted under the same or different public keys. Our scheme is constructed under the certificateless cryptosystem (CLC), thereby addressing the Certificate management problem. Moreover, our proposed scheme fixes the inherent key escrow problem of ID-based encryption (IBE). We performed a security analysis on our proposed scheme and achieved IND-CCA2, EUF-CMA, and OW-CCA2 levels of security in the Random Oracle Model (ROM). Furthermore, we compared our proposed scheme with other existing state-of-the-art schemes. By minimizing computational costs and communication costs while maintaining security and functionality, our scheme exhibits significantly lower computational costs for encryption, decryption, and testing stages, thus enhancing efficiency in resource-constrained IoT-enabled WSNs.

CRediT authorship contribution statement

Zohaib Ali: Writing – review & editing, Writing – original draft, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Junaid Hassan:** Writing – review & editing, Methodology, Funding acquisition, Data curation. **Muhammad Umar Aftab:** Visualization, Validation, Supervision, Funding acquisition. **Negalign Wake Hundera:** Visualization, Validation, Supervision, Project administration. **Huiying Xu:** Visualization, Validation, Supervision, Funding acquisition. **Xinzhong Zhu:** Visualization, Validation, Supervision, Funding acquisition.

Ethical compliance

This research does not involve human participants, animals, or sensitive personal data. Hence, ethical approval was not required. The work complies with the ethical standards of research and publishing.

Funding information

This work was supported by the National Natural Science Foundation of China (62376252); Zhejiang Province Leading Geese Plan (2025C02025, 2025C01056); Zhejiang Province Province-Land Synergy Program (2025SDXT004-3).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] M. Luo, Y. Pei, M. Qiu, Cross domain heterogeneous signcryption scheme with equality test for WBAN, Wirel. Pers. Commun. 130 (2) (2023) 1107–1122.

[2] H. Chen, J. Wang, X. Dong, C. Zhao, Security design of ECG telemonitoring systems, in: 2020 International Conference on Computer Engineering and Application, ICCEA, IEEE, 2020, pp. 707–711.

[3] T.V.N. Rao, L. Mothukuri, S. Bhavana, IoT networks for real-time healthcare monitoring systems, in: Analyzing Current Digital Healthcare Trends using Social Networks, IGI Global, 2024, pp. 143–158.

[4] J. Hassan, D. Shehzad, I. Ullah, F. Algarni, M.U. Aftab, M. Asghar Khan, M.I. Uddin, A lightweight proxy Re-encryption approach with certificate-based and incremental cryptography for fog-enabled E-healthcare, Secur. Commun. Netw. 2021 (2021) 1–17.

[5] A. Souri, Y. Zhao, M. Gao, A. Mohammadian, J. Shen, E. Al-Masri, A trust-aware and authentication-based collaborative method for resource management of cloud-edge computing in social internet of things, IEEE Trans. Comput. Soc. Syst. (2023).

[6] H.A. Al Hamid, S.M.M. Rahman, M.S. Hossain, A. Almogren, A. Alamri, A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography, IEEE Access 5 (2017) 22313–22328.

[7] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, in: C. Cachin, J.L. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2004, pp. 506–522.

[8] G. Yang, C.H. Tan, Q. Huang, D.S. Wong, Probabilistic public key encryption with equality test, in: J. Pieprzyk (Ed.), Topics in Cryptology - CT-RSA 2010, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2010, pp. 119–131.

[9] S. Ma, Identity-based encryption with outsourced equality test in cloud computing, Inform. Sci. 328 (2016) 389–402.

[10] Z. Yang, D. He, L. Qu, Q. Ye, An efficient identity-based encryption with equality test in cloud computing, IEEE Trans. Cloud Comput. (2023).

[11] M. Ramadan, S. Raza, Secure equality test technique using identity based signcryption for telemedicine systems, IEEE Internet Things J. (2023).

[12] S. Ma, Z. Ye, Q. Huang, C. Jiang, Controllable forward secure identity-based encryption with equality test in privacy-preserving text similarity analysis, Inform. Sci. 660 (2024) 120099.

[13] H. Okano, K. Emura, T. Ishibashi, T. Ohigashi, T. Suzuki, Implementation of a strongly robust identity-based encryption scheme over type-3 pairings, Int. J. Netw. Comput. 10 (2) (2020) 174–188.

[14] C. Zhou, Z. Zhao, W. Zhou, Y. Mei, et al., Certificateless key-insulated generalized signcryption scheme without bilinear pairings, Secur. Commun. Netw. 2017 (2017).

[15] A. ur Rahman, I. Ullah, M. Naeem, R. Anwar, H. Khattak, S. Ullah, et al., A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve, Int. J. Adv. Comput. Sci. Appl. 9 (5) (2018).

[16] A. Shamir, Identity-based cryptosystems and signature schemes, in: G.R. Blakley, D. Chaum (Eds.), Advances in Cryptology, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 1985, pp. 47–53.

[17] S. Dong, Z. Zhao, B. Wang, W. Gao, S. Zhang, SM9 identity-based encryption with designated-position fuzzy equality test, Electronics 13 (7) (2024) 1256.

[18] X.-J. Lin, Q. Wang, L. Sun, H. Qu, Identity-based encryption with equality test and datestamp-based authorization mechanism, Theoret. Comput. Sci. 861 (2021) 117–132.

[19] J. Lu, H. Li, J. Huang, S. Ma, M.H.A. Au, Q. Huang, An Identity-Based Encryption with Equality Test scheme for healthcare social apps, Comput. Stand. Interfaces 87 (2024) 103759.

[20] M. Ramadan, Y. Liao, F. Li, S. Zhou, H. Abdalla, IBEET-RSA: Identity-based encryption with equality test over RSA for wireless body area networks, Mob. Netw. Appl. 25 (2020) 223–233.

[21] Y. Sun, P. Chatterjee, Y. Chen, Y. Zhang, Efficient identity-based encryption with revocation for data privacy in internet of things, IEEE Internet Things J. 9 (4) (2022) 2734–2743, Conference Name: IEEE Internet of Things Journal.

[22] T.-T. Tsai, H.-Y. Lin, H.-C. Chang, An efficient revocable identity-based encryption with equality test scheme for the wireless body area network, J. Sens. 2022 (2022) e1628344, Publisher: Hindawi.

[23] R. Elhabob, Y. Zhao, I. Sella, H. Xiong, An efficient certificateless public key cryptography with authorized equality test in IIoT, J. Ambient. Intell. Humaniz. Comput. 11 (3) (2020) 1065–1083.

[24] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, in: C.-S. Laih (Ed.), Advances in Cryptology - ASIACRYPT 2003, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2003, pp. 452–473.

[25] R. Elhabob, M. Taha, H. Xiong, M.K. Khan, S. Kumari, P. Chaudhary, Pairing-free certificateless public key encryption with equality test for Internet of Vehicles, Comput. Electr. Eng. 116 (2024) 109140.

[26] J. Tian, Y. Lu, J. Li, Lightweight searchable and equality-testable certificateless authenticated encryption for encrypted cloud data, IEEE Trans. Mob. Comput. 23 (8) (2024) 8431–8446.

[27] R. Elhabob, Y. Zhao, I. Sella, H. Xiong, Efficient certificateless public key cryptography with equality test for internet of vehicles, IEEE Access 7 (2019) 68957–68969.

[28] T. Wollinger, J. Pelzl, C. Paar, Cantor versus Harley: optimization and analysis of explicit formulae for hyperelliptic curve cryptosystems, IEEE Trans. Comput. 54 (7) (2005) 861–872.

[29] I. Ullah, M.A. Khan, M.H. Alsharif, R. Nordin, An anonymous certificateless signcryption scheme for secure and efficient deployment of Internet of vehicles, Sustainability 13 (19) (2021) 10891.

[30] I. Ullah, N. Ul Amin, M. Zareei, A. Zeb, H. Khattak, A. Khan, S. Goudarzi, A lightweight and provable secured certificateless signcryption approach for crowdsourced IIoT applications, Symmetry 11 (11) (2019) 1386.

[31] X. Fan, T. Wollinger, G. Gong, Efficient explicit formulae for genus 3 hyperelliptic curve cryptosystems over binary fields, IET Inf. Secur. 1 (2) (2007) 65–81.

[32] M.A. Khan, I.M. Qureshi, I. Ullah, S. Khan, F. Khanzada, F. Noor, An efficient and provably secure certificateless blind signature scheme for flying ad-hoc network based on multi-access edge computing, Electronics 9 (1) (2019) 30.

[33] H.-Y. Lin, RPCAE: a novel revocable proxy convertible authenticated encryption scheme, Int. J. Inf. Secur. 14 (2015) 431–441.

[34] F. Li, P. Xiong, Practical secure communication for integrating wireless sensor networks into the internet of things, IEEE Sens. J. 13 (10) (2013) 3677–3684.

[35] GitHub - JHUISI/charm: charm: A framework for rapidly prototyping cryptosystems — github.com, 2025, https://github.com/JHUISI/charm. (Accessed 5 January 2025).

[36] PBC library - pairing-based cryptography - about — crypto.stanford.edu, 2025, https://crypto.stanford.edu/pbc/. (Accessed 5 January 2025).

[37] GitHub - syncom/libg2hec: A genus 2 crypto C++ library — github.com, 2025, https://github.com/syncom/libg2hec. (Accessed 5 January 2025).

[38] The GNU MP bignum library — gmplib.org, 2025, https://gmplib.org/. (Accessed 5 January 2025).

[39] GitHub - openssl/openssl: TLS/SSL and crypto library — github.com, 2025, https://github.com/openssl/openssl. (Accessed 5 January 2025).

[40] NTL: A library for doing number theory — libntl.org, 2025, https://libntl.org/. (Accessed 5 January 2025).