Journal of Machine Learning and Information Security
https://www.sciltp.com/journals/jmlis

Article

An Enhanced Security Data Transmission Scheme for Wireless Medical Sensor Network

Xi Chen [1,†], Chunqiang Hu [1,*,†], Yuwen Chen [2,†], Xiaofeng Xia [1,†], Bin Cai [1,†] and Jiguo Yu [3,†]

[1] School of Big Data & Software Engineering, Chongqing University, Chongqing 400030, China
[2] Institute of Biomedical and Health Sciences, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
[3] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
* Correspondence: chu@cqu.edu.cn
† These authors contributed equally to this work.

How To Cite: Chen, X.; Hu, C.; Chen, Y.; et al. An Enhanced Security Data Transmission Scheme for Wireless Medical Sensor Network. *Journal of Machine Learning and Information Security* **2026**, *2*(1), 3. https://doi.org/10.53941/jmlis.2026.100003

Received: 25 August 2025
Revised: 29 October 2025
Accepted: 16 December 2025
Published: 27 January 2026

Abstract: Wireless Medical Sensor Networks (WMSNs) facilitate the real-time collection and transmission of patients' physiological data, enabling remote healthcare and intelligent medical services. However, the inherent openness of wireless communication renders sensitive data vulnerable to security threats such as interception, tampering, and replay attacks. To mitigate these issues, this paper introduces a novel certificateless aggregate signcryption scheme based on elliptic curve cryptography. The proposed scheme eliminates the complexities associated with certificate management, ensures data confidentiality and unforgeability, and incorporates an anonymity mechanism to safeguard client identities. Moreover, an efficient invalid signature detection algorithm is introduced, which utilizes an authentication key to swiftly identify malicious nodes in the event of aggregate verification failure, thereby minimizing redundant computations and improving system robustness. Under the random oracle model, formal security proofs demonstrate the scheme's resilience against adaptive chosen-ciphertext and forgery attacks. Experimental results indicate that the proposed scheme not only achieves lower communication overhead but also maintains competitive computational efficiency compared to existing schemes, all while delivering stronger security assurances.

Keywords: wireless medical sensor networks; certificateless signcryption scheme; invalid signature node

1. Introduction

Wireless Medical Sensor Networks (WMSNs) are emerging healthcare systems based on Internet of Things (IoT) technology, typically composed of miniature, resource-constrained sensor devices deployed on or inside the patient's body to collect physiological data, such as heart rate, blood pressure, electrocardiograms, and blood glucose levels [1, 2]. These data are wirelessly transmitted to healthcare professionals or centralized systems (e.g., application provider centers) for real-time monitoring, diagnosis, and treatment [3,4]. Doctors can provide remote healthcare services based on this data, while patients can receive personalized health management plans. WMSNs not only enhance the accessibility and efficiency of healthcare services but also play an important role in emergency care and chronic disease management [5].

Although WMSNs provide users with convenient and efficient healthcare services, they inevitably introduce potential security and privacy risks while enhancing the quality of medical care. Since client data are typically transmitted through open and not fully trusted communication channels, adversaries may exploit various attack vectors, such as data interception, replay attacks, malicious injection, and tampering, to compromise the confidentiality, integrity, and availability of sensitive information [6–9]. Consequently, ensuring security and privacy during data transmission without imposing excessive overhead on the system has become a critical challenge that WMSNs must urgently address.

Among various security mechanisms for securing data transmission, identity authentication is widely recognized as a crucial technology for ensuring trusted communication [10–19]. It provides undeniable and unforgeable guarantees during data transmission, thereby preventing unauthorized entities from impersonating legitimate users and ensuring the security and trustworthiness of data sources [20]. Nevertheless, when applied to WMSNs, existing authentication schemes still suffer from the following limitations.

(1) *High computational and communication overheads:* Traditional identity authentication schemes typically depend on complex cryptographic algorithms and certificate management, which impose significant computational demands. Moreover, the authentication process often involves multiple rounds of communication, increasing bandwidth consumption and latency, particularly in scenarios involving large-scale data transmission. Such overhead is unsuitable for medical sensor devices, which are generally resource-constrained in terms of computational capacity and energy.
(2) *Insufficient privacy protection:* Many current schemes do not sufficiently preserve client identity privacy. In wireless environments, transmitted data that lacks encryption or anonymization are susceptible to eavesdropping or tracking, potentially exposing sensitive user identity information. This issue is especially critical in healthcare applications, where the leakage of patient information may lead to severe consequences.

To overcome these security and privacy challenges, we propose an innovative certificateless aggregate signcryption scheme. The proposed design not only enhances data security but also substantially reduces both computational and communication costs. By removing the need for complex certificate management, it lowers system overhead and simplifies implementation, thereby improving both deployability and operational efficiency. In addition, the scheme incorporates an anonymity mechanism that conceals clients' real identities, strengthening privacy protection without compromising authentication reliability. In practice, attackers may inject invalid signatures to disrupt aggregate verification, leading to unnecessary resource expenditure. To address this, we further design an efficient invalid signature detection algorithm capable of rapidly identifying and localizing malicious nodes upon verification failure, thus maintaining verification efficiency and system robustness. Overall, the proposed scheme not only resists common security attacks but also achieves an effective balance between security and efficiency, making it well-suited for practical WMSNs environments. The main contributions of this work are summarized as follows.

(1) *A novel certificateless aggregate signcryption scheme:* This paper proposes a novel certificateless aggregate signcryption scheme to ensuresdata confidentiality and unforgeability while incorporating an anonymity mechanism to safeguard clients' real identities from disclosure. It achieves low computational cost and communication overhead, making it suitable for resource-constrained environments.
(2) *Invalid signature detection algorithm:* This paper proposes an invalid signature node detection algorithm. By introducing an authentication key into the signcryption process, the algorithm enables rapid and accurate localization of invalid signature nodes in the event of aggregate verification failure. This design effectively avoids the additional overhead caused by repeated verification and significantly enhances system availability in the presence of malicious nodes.

The remaining portions of this paper are organized as follows. In Section 2, we describe the related work of this paper. Section 3 introduces the preliminaries of cryptography, framework, and security model. In Section 4, we introduce the proposed signcryption scheme. In Section 5, we describe the security analysis of the proposed signcryption scheme. Section 6 describes the performance analysis. Conclusions can be found in Section 7.

2. Related Work

Certificateless aggregate signcryption (CLASC) schemes have gained broad recognition for their ability to simplify certificate management and mitigate key escrow risks, thereby ensuring confidentiality and unforgeability in data transmission. Additionally, CLASC improves verification efficiency, rendering it a promising solution for secure communication in resource-constrained environments [21]. In recent years, numerous CLASC schemes have been developed to enhance data transmission security across various network architectures, including wireless sensor networks, vehicular networks, and IoT.

Yang et al. [22] presented a CLASC scheme under the random oracle model to strengthen security in vehicular networks. Basudan et al. [23] introduced a fog computing–based CLASC scheme for vehicular data. However, both approaches rely on bilinear pairings, which incur considerable communication overhead and may impair scalability and real-time performance. In contrast, Yu et al. [24] proposed an ECC-based CLASC scheme tailored to tackle certificate management and key escrow issues in 5G-IoT multi-device authentication. By leveraging elliptic curve cryptography (ECC), their scheme reduces computational costs without compromising security. Similarly, Dai et al. [25] devised a pairing-free ECC-based CLASC scheme that guarantees confidentiality, authenticity,

non-repudiation, and privacy in vehicular networks, offering a more efficient alternative for such demanding settings. Ren et al. [26] combined ECC with neural networks to create a low-complexity scheme for protecting patient data in WMSNs, incorporating Levenshtein entropy coding to improve security while minimizing overhead. Wang et al. [27] integrated ECC with blockchain to introduce a pairing-free scheme that counters key escrow, public key replacement attacks, and high overhead in vehicular networks. Chen et al. [28] further developed a pairing-free ECC scheme named SPF-CLASC, which ensures identity anonymity, confidentiality, integrity, and resistance to full key exchange attacks, all while reducing overhead for resource-limited IoT nodes. Zhang et al. [29] proposed an ECC-based certificateless aggregate signature scheme with strengthened security against Type I and Type II attacks, addressing vulnerabilities that jeopardize data privacy and integrity in constrained networks. Collectively, these schemes have advanced secure data transmission in multiple application domains. Nevertheless, certain limitations persist—for instance, the schemes by Yu et al. [24] and Ren et al. [26] are vulnerable to replay attacks. Moreover, none of the aforementioned schemes adequately addresses the scenario where aggregate verification fails.

The handling of aggregate verification failures remains a common challenge in CLASC systems, particularly when invalid signatures are present within a batch. Conventional approaches often discard the entire batch, resulting in inefficient processing of otherwise valid signatures. To tackle this issue, Huang et al. [30] and Xiong et al. [31] proposed binary search–based and elementary symmetric polynomial–based detection methods, respectively, to identify invalid signature nodes. While promising, these techniques involve substantial computational costs during detection, which may limit their scalability in larger networks. Haturn et al. [32] and Wang et al. [33] subsequently introduced fault-tolerant aggregate signature schemes and their improved variants to locate invalid nodes, though these also entail high computational overhead. Li et al. [34] designed a lightweight ECC-based two-dimensional matrix scheme for rapid detection of illegal signcryption in power systems. However, this method requires constructing a matrix of size $a \times b = n$, which becomes inefficient when n is a prime number, thereby constraining its detection performance.

A comprehensive review of the literature highlights two major shortcomings in current CLASC schemes: (i) inadequate protection against advanced attacks; and (ii) the absence of efficient invalid signature detection and deduplication mechanisms, which impede the timely removal of malicious nodes and degrade system performance. To address these issues, this paper proposes an ECC-based CLASC scheme that enhances the security of data transmission while balancing computational and communication efficiency, thereby ensuring the protection of sensitive information in WMSNs. Furthermore, an efficient detection algorithm is designed to quickly locate invalid nodes during aggregate verification, thus sustaining system robustness. By effectively balancing security and efficiency, the proposed scheme offers a more practical solution for secure communication in WMSNs.

3. Preliminaries

This section outlines the underlying cryptographic primitives and formal models for our proposal. We begin with ECC and the requisite hardness assumptions, followed by the system framework, security model, and defined security goals.

3.1. Cryptography

The basic equations of elliptic curves are usually expressed as:

$$E(\mathbb{F}_q) : y^2 \equiv x^3 + ax + b \pmod{q} \tag{1}$$

where $E(\mathbb{F}_q)$ represents an elliptic curve defined over a finite field $\mathbb{F}_q$, y and x are coordinate points on the curve, $a, b \in F_q$, and $4a^3 + 27b^2 \neq 0 \pmod{q}$, and q is prime number. The (x, y) and the infinity point O form a group G.

Elliptic Curve Discrete Logarithm Assumption (ECDL): Given a known tuple $(P, Q = aP), a \in \mathbb{Z}_q^*, P, Q \in G$, where $\mathbb{Z}_q^* = \{0, 1, ..., q-1\}$ and a is not known. a cannot be computed in polynomial time.

Computation Diffie-Hellman Assumption (CDH): Given a known tuple $(P, aP, bP), a, b \in \mathbb{Z}_q^*$, where a and b are not known. And abP cannot be computed in polynomial time.

3.2. Framework

There are 4 participants consists of this system: (1) Key Generator Center (KGC); (2) Application Provider Center (APC); (3) Area Server (AS); (4) Client (C), as illustrated in the Figure 1.

(1) KGC: KGC serves as a trusted network management center. Its primary function is to register the identity of participants and distribute partial private keys. Prior to joining the system, clients and APC are required to register with KGC. When the system detects the malicious client, the KGC will trace the true identity of the malicious client.

(2) APC: APC provides corresponding healthcare services to clients. APC has a server with powerful storage and computing capabilities for storing data transmitted by the client. When the client interacts with APC, APC provides healthcare services based on the stored data and current client requirements.

(3) AS: The primary role of the AS is to receive revocation notifications, update the public and private keys of this region, remove the associated ID, and broadcast this information. The AS aggregates the signatures and data uploaded by multiple client users within the region and transmits them to the APC for storage.

(4) C: The client is an intelligent mobile device (edge device) equipped with sensors, typically carried by patients, to collect their physiological information. It then transmits the sensitive information to the APC to assist remote doctors in monitoring the patients' physical conditions.

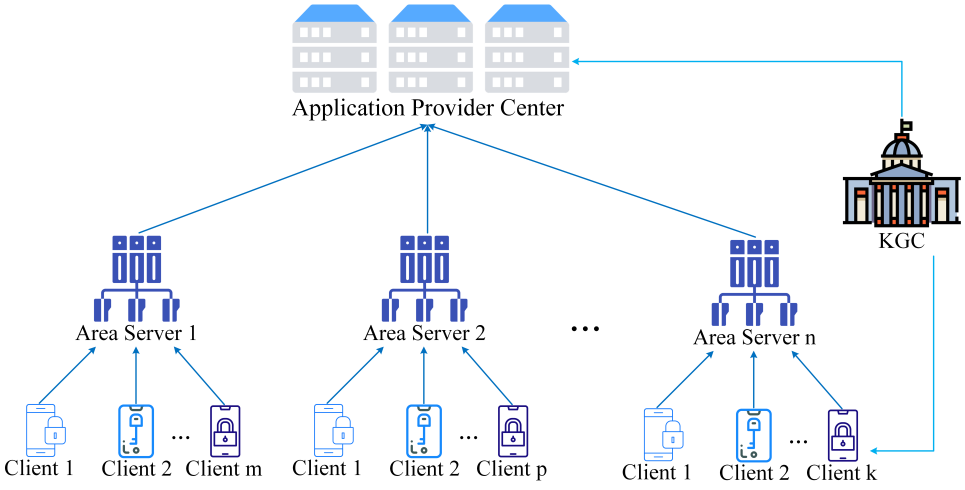

Figure 1. The framework of the transmission system.

Meanwhile, relevant assumptions are provided as follows:

(1) Trusted participants: The system assumes that the KGC, APC, and AS are trusted and operate without malicious intent. These entities play crucial roles in key distribution, data aggregation, and service provision within the network.

(2) Malicious clients: Clients, while trusted at the time of registration, may turn malicious during operation. A compromised or malicious client can inject invalid signatures or attempt to breach privacy. The system includes mechanisms for invalid signature detection to identify and isolate such clients efficiently.

(3) Communication channel: We assume that communication between the clients, AS, and APC occurs over potentially insecure, public wireless channels. This exposes the system to various attacks, such as eavesdropping and replay attacks, which the proposed signcryption scheme mitigates through strong encryption and integrity mechanisms.

3.3. Security Model

The proposed signcryption scheme is susceptible to attacks from two types of adversaries.

(1) Type I adversary: A Type I adversary $\mathcal{A}_I$ is capable of performing public key replacement attacks, where the adversary can substitute the public key of a legitimate participant with a malicious one. However, this adversary does not have access to the system's master private key, and hence cannot directly decrypt data or generate valid signatures for arbitrary messages. This limitation ensures that, although public key replacement attacks may disrupt the integrity of communications, they cannot fully compromise the confidentiality or unforgeability of the system's signcryption scheme.

(2) A Type II adversary $\mathcal{A}_{II}$, in contrast, has access to the system's master private key. This adversary can decrypt messages, potentially compromising the confidentiality of the transmitted data. However, this adversary is unable to perform public key replacement attacks, meaning that it cannot impersonate other users or interfere with the authenticity of the public keys used in the system.

A novel certificateless signcryption scheme has proposed in this paper can achieve confidentiality against adaptive chosen-ciphertext attacks from adversaries $\mathcal{A}_I$ and $\mathcal{A}_{II}$, as well as unforgeability against adaptive chosen message attacks, in the random oracle model (ROM). Table 1 summarizes the security game between Type I and Type II adversaries under ROM.

Table 1. Security game for the proposed scheme.

Security	**Confidentiality/Unforgeability**	
Game	Security game for Type I	Security game for Type II
	Initialization: $\mathcal{C}$ generates system master private key and system parameters,$\mathcal{C}$ broadcasts system parameters.	Initialization: $\mathcal{C}$ generates system master private key and system parameters, $\mathcal{C}$ broadcasts system parameters and sends master private key to $\mathcal{A}$.
Description	Query phase: $\mathcal{A}$ launches some queries, $\mathcal{C}$ answers these queries. 1. $\mathcal{A}$ launches queries on $\tilde{H}_1$, $H_1 \sim H_3$, $\mathcal{C}$ answers the query result. 2. $\mathcal{A}$ launches query on public key, $\mathcal{C}$ answers the query result. 3. $\mathcal{A}$ launches queries on partital private key, private key, public key replacement, and signcryption, $\mathcal{C}$ answers the query result.	
	Challenge phase: $\mathcal{A}$ selects two plaintexts and two challenge idenities, then $\mathcal{C}$ generates challenge information. Guess phase: $\mathcal{A}$ outputs the guessing result, if the result is correct, the ECDH problem is solved. Forge phase: $\mathcal{A}$ outputs the target signature, if $\mathcal{A}$ wins the game, $\mathcal{A}$ can solve the ECDL problem.	
Notes	Type II cannot launch public key replacement query. Confidentiality: Query phase, Challenge phase, and Guess phase. Unforgeability: Query phase and Forge phase. $\mathcal{A}$ and $\mathcal{C}$ represent adversary and challenger, respectively.	

3.4. Security Goals

The proposed CLASC scheme needs to meet the following security goals:

(1) *Confidentiality:* It is essential to ensure that the data remains unreadable to attackers during transmission and that no sensitive information is disclosed.

(2) *Unforgeability:* An attacker, without authorization (i.e., without knowledge of the full key), cannot successfully forge a new legitimate signature.

(3) *Anonymity:* The scheme ensures that client identities are kept anonymous during data transmission, protecting against identity leakage, even in the presence of a malicious entity observing the communication.

(4) *Integrity and Non-repudiation:* The use of cryptographic signing mechanisms ensures that once a message is signed, its authenticity can be verified by the receiving entities, preventing repudiation of the transmission by the client.

4. Proposed Scheme

In this section, we will introduce the proposed signcryption scheme.

4.1. Certificateless Aggregate Signcryption Scheme

This subsection presents our designed certificateless aggregate signcryption scheme (CLASS), which serves as the cornerstone of secure data transmission. The CLASS includes 7 algorithms: System Setup, Secret Key Generation, Partial Private Key Generation, Full Key Generation, Signcryption, Aggregation, and Verify.

(1) *System Setup* (1^κ): Initialize the system, generate system private key, system public key, and publish system parameters.

- (a) According to the security parameter κ, KGC selects a group G with q and P, which represents a large prime order and generator of G, respectively.
- (b) KGC randomly selects $s_C \in \mathbb{Z}_q^*, s_A \in \mathbb{Z}_q^*$ as the system private key, and calculates $P_{C-pub} = s_C P, P_{A-pub} = s_A P$ as the Client's system public key and APC's system public key, respectively. The purpose of selecting different system private keys is to prevent one party from using its parameters to impersonate the identity of another party.
- (c) KGC selects several hash functions: $H, \tilde{H}_1, H_1 \sim H_7$, where $H :\mapsto \{0,1\}^l, \tilde{H}_1, H_1 \sim H_7 :\mapsto \mathbb{Z}_q^*$.
- (d) KGC publishes the system parameters $params = \{G, q, P, P_{C-pub}, P_{A-pub}, H, \tilde{H}_1, H_1 \sim H_7\}$, and stores (s_C, s_A) securely.

(2) *Secret Key Generation*:

- (a) C_i randomly selects $x_{C_i} \in \mathbb{Z}_q^*$ as the secret key, and calculates $X_{C_i} = x_{C_i} P$, then sends (ID_{C_i}, X_{C_i}) to KGC.
- (b) APC randomly selects $x_A, x_{mac} \in \mathbb{Z}_q^*$ as the secret key, and calculates $X_A = x_A P, X_{mac} = x_{mac} P$, then sends (ID_A, X_A) to KGC.

(3) *Partial Private Key Generation*:

- (a) Upon receiving (ID_{C_i}, X_{C_i}) from C, KGC randomly selects $r_{C_i} \in \mathbb{Z}_q^*$, calculates $R_{C_i} = r_{C_i} P$, generates anonymity $PID_{C_i} = ID_{C_i} \oplus H(r_{C_i} P_{C-pub}, T_C)$, where T_C denotes the expiration time of PID_{C_i}. KGC calculates $h_{C_i} = H_1(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub})$, $ppk_{C_i} = r_{C_i} + s_C h_{C_i} \mod q$.

KGC sends $(PID_{C_i}, ppk_{C_i}, R_{C_i}, h_{C_i})$ to C_i. C_i calculates $h_{C_i} = H_1(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub})$, if $ppk_{C_i}P = R_{C_i} + h_{C_i}P_{C-pub}$ holds, C_i accepts ppk_{C_i}.

(b) Upon receiving (ID_A, X_A), KGC randomly selects $r_A \in \mathbb{Z}_q^*$, calculates $R_A = r_A P, h_A = \tilde{H}_1(ID_A, R_A, X_A, P_{A-pub})$, $ppk_A = r_A + s_A h_A \mod q$. KGC sends (ppk_A, R_A, h_A) to APC. APC calculates $h_A = H_1(ID_A, R_A, X_A, P_{A-pub})$, if $ppk_A P = R_A + h_A P_{A-pub}$ holds, APC accepts ppk_A.

(4) *Full Key Generation*:

(a) C_i obtains the full private key $sk_{C_i} = (x_{C_i}, ppk_{C_i})$ and full public key $pk_{C_i} = (X_{C_i}, R_{C_i})$. C_i publishes pk_{C_i}.

(b) APC obtains the full private key $sk_A = (x_A, ppk_A)$ and full public key $pk_A = (X_A, R_A)$. APC publishes pk_A and X_{mac}, where X_{mac} represents the authentication public key.

(5) *Signcryption*: Let M_i is the physiological data collected by C_i, t_i is the current timestamp. C_i randomly selects $f_{i1}, f_{i2} \in \mathbb{Z}_q^*$, calculates $f_i = f_{i1}x_{C_i} + f_{i2}ppk_{C_i}$, $F_i = f_i P$, $V_i = f_i(X_A + R_A + h_A P_{A-pub})$, $C_i = H_2(ID_A, V_i, F_i, t_i) \oplus M_i, h_{3i} = H_3(PID_{C_i}, M_i, F_i, X_{C_i}, R_{C_i}, t_i), u_i = f_i + h_{3i}(x_{C_i} + x_{s_i}) + ppk_{C_i}$, where x_{s_i} represents the private key of the AS_i. C_i calculates $MAC_i = H_2(ID_A, f_i X_{mac}, F_i, t_i) \oplus H_7(u_i, PID_{C_i}, M_i)$. C_i outputs the signcryption $\sigma_i = (F_i, u_i, C_i, MAC_i, t_i)$.

(6) *Aggregate*: AS_i aggregates the received information $\sigma_i, PID_{C_i}, i = 1, 2, ..., n$. AS_i sets $T = (T_i)_{i=1}^n, \eta = \sum_{i=1}^n u_i$, and outputs aggregate information $A_{s_i} = \{F_1, ..., F_n, C_1, ..., C_n, MAC_i, ..., MAC_n, \eta, T\}$.

(7) *Verify*: Upon receiving A_{si}, APC first checks the validity of T, if T is not vaild, terminate. Otherwise, APC calculates $h_{C_i} = H_1(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, RT_{C_i}), V_i = (x_A + ppk_A)F_i, M_i = H_2(ID_A, V_i, F_i, t_i) \oplus C_i, h_{3i} = H_3(PID_{C_i}, M_i, F_i, X_{C_i}, R_{C_i}, t_i), i = 1, 2, ..., n$. If $\eta P = \sum_{i=1}^n F_i + \sum_{i=1}^n h_{3i}(X_{C_i} + Q_{s_i}) + \sum_{i=1}^n (R_{C_i} + h_{C_i}P_{C-pub})$ holds, APC accepts the AS_i and stores the data. Otherwise, APC starts the invalid signature detection algorithm and searches for invalid signature nodes.

4.1.1. Correctness

$$\begin{aligned} V_i &= f_i(X_A + R_A + h_A P_{A-pub}) \\ &= f_i(x_A + r_A + s_A h_A)P \\ &= (x_A + r_A + s_A h_A)F_i \end{aligned} \tag{2}$$

$$\begin{aligned} \eta P &= \sum_{i=1}^n (f_i + h_{3i}(x_{C_i} + x_{s_i}) + ppk_{C_i})P \\ &= \sum_{i=1}^n (f_i P) + \sum_{i=1}^n h_{3i}(x_{C_i} + x_{s_i})P + \sum_{i=1}^n ppk_{C_i}P \\ &= \sum_{i=1}^n F_i + \sum_{i=1}^n h_{3i}(X_{C_i} + Q_{s_i}) + \\ &\sum_{i=1}^n (R_{C_i} + h_{C_i}P_{C-pub}) \end{aligned} \tag{3}$$

4.1.2. Invalid Signature Detection

Aggregate verification can save a significant amount of computational time and enhance verification efficiency. However, aggregate verification faces a challenge: when one or more invalid signatures are present in a batch of signatures, it leads to the failure of aggregate verification. Invalid signatures can stem from various reasons, such as packet loss, wireless channel interference, or the involvement of malicious attackers [30]. In most aggregate signcryption schemes, the common approach is to reject the entire batch of data upon aggregation failure. Nevertheless, physiological data is highly valuable, and it is undesirable to reject an entire batch of data due to a single invalid signature. Instead, we should exclude the invalid signatures and accept the data containing other valid signatures.

According to [30], let N_t indicate the number of clients managed by AS_i, N_b indicate the number of validation batches, and N_I indicate the number of invalid signatures. Let $P[k]$ indicate the probability that there are k invalid signatures in one aggregation validation, as follows.

$$P[k] = \frac{\binom{N_t - N_I}{Nb-k}\binom{N_I}{k}}{\binom{N_t}{N_b}}, \quad k = 0, 1, 2, ... \tag{4}$$

Let $\mathbb{E}$ indicate the event that re-aggregation verification is required to successfully verify all valid signatures. Therefore, the probability of $\mathbb{E}$ can be expressed as Equation (5),

$$\begin{aligned} P[\mathbb{E}] &= P[k=1] + p[k=2] + ... + p[k=n] \\ &= \frac{\sum_{k=1}^{n} \binom{N_t - N_I}{Nb-k}\binom{N_I}{k}}{\binom{N_t}{N_b}} \end{aligned} \tag{5}$$

From Equation (5), there is at least one invalid signature in an aggregate verification, causing aggregate verification to fail. Therefore, re-aggregate validation is required. Figure 2 intuitively how the relationship between the number of invalid signatures and that of requests in an aggregate. In Figure 2, assuming the number of invalid signatures is 0–120, the number that aggregate verification can verify simultaneously is 0–120. When $k = 1$, the probability of successful re-aggregation verification is about 0.33 at most. When there are two invalid signatures in the aggregate signature, the probability is about 0.22, and when the invalid signatures increases to 4, the probability drops to 0.04. This indicates that the greater the number of invalid signatures in an aggregation batch, the lower the probability of successful re-aggregation verification. To solve this problem, we design an invalid signature detection algorithm.

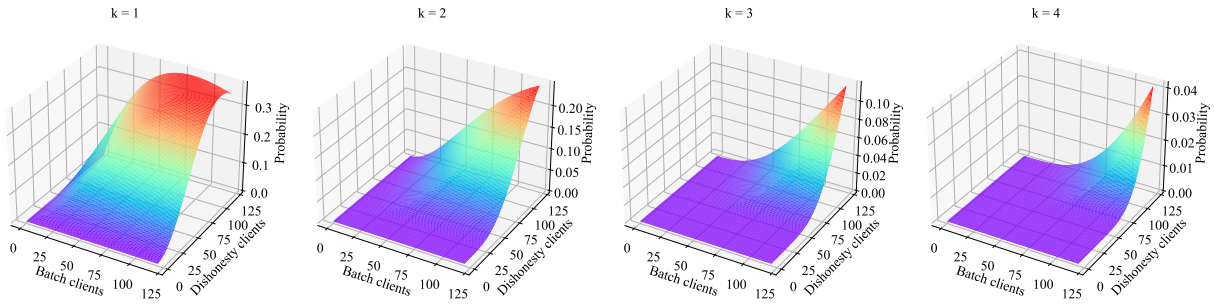

Figure 2. Re-batch verify probability.

When the aggregate verification fails, APC starts the invalid signature detection algorithm to check the client nodes that cause the aggregate signature to be invalid. And we assume that the aggregation information A_{si} sent by AS_i fails to pass the verification of APC. Firstly, APC extracts A_{si} and divides N_t signatures into m groups. Secondly, APC conducts re-aggregate verification on m groups and identifies the groups that did not pass the verification. Thirdly, APC detects group MAC values that fail verification and outputs invalid signature nodes. Finally, APC accepts valid signatures that have been verified.

As shown in Figure 3, we give a specific example to explain the invalid signature detection algorithm. We assume that $N_t = 2000$, $m = 10$, and the four invalid signature nodes are located at positions 1, 2, 1801, and 2000. APC divides 2000 signatures into 10 groups, then performs re-aggregation verification on each group and finds the group that failed the verification. After re-aggregate verification, After re-aggregation verification, APC determined that groups 2–9 were all valid signature groups, and performed MAC value detection on group 1 and group 10 to find invalid signature nodes. Finally, APC outputs invalid signature nodes and accepts valid signature nodes. For ease of reading, the invalid signature detection algorithm is summarized in Algorithm 1.

Algorithm 1 Invalid Signature Detection Algorithm.

INPUT
N_t signatures, m
OUTPUT
Invalid signature nodes, valid signature nodes
PROCEDURE
1: Create an invalid group Ig and a valid group Vg
2: Divide N_t signatures into m groups
3: Re-aggregate verification for each group
4: Check the MAC value of the failed group
5: Add invalid signatures into Ig
6: Add valid signatures into Vg
7: Return Ig and Vg

After finding the invalid signature node, APC notifies AS_i to process it. AS_i removes members of the invalid signature group from the revocable list and updates its own key.

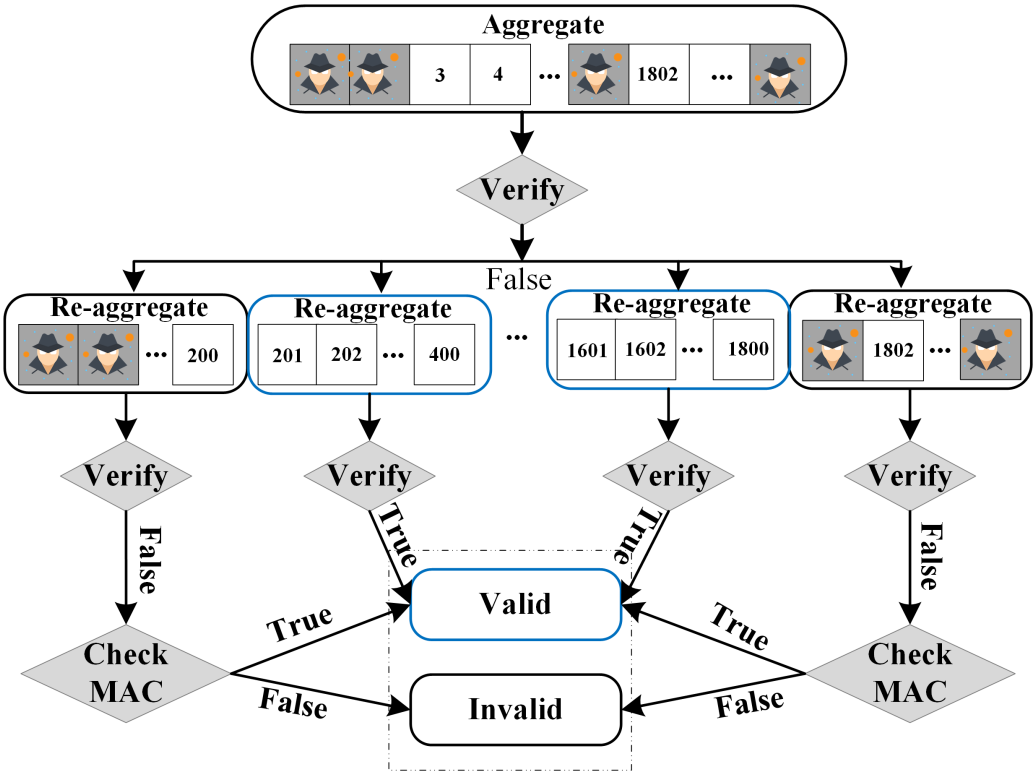

Figure 3. Invalid signature detection.

4.1.3. Complexity and Correctness Analysis

To theoretically evaluate the efficiency of the proposed invalid signature detection algorithm, we analyze its computational and communication complexity and compare it with several existing detection methods.

(1) Computational complexity
Let n be the total number of signatures and m be the number of groups. The proposed algorithm first performs re-aggregate verification on m groups, which costs $C_m = m \cdot C_{agg}(n/m)$, where $C_{agg}(k)$ denotes the cost of verifying k aggregated signatures. Then, for each failed group, it checks MAC values for each signature, costing $O(k)$ per group. In the worst case, all groups fail, leading to $O(n)$ MAC checks. However, since m is configurable and typically small, the overall complexity remains practical. Under parallel execution, the MAC checks for different groups can be performed simultaneously, reducing the effective time complexity to $O(n/m)$.

(2) Communication overhead
The algorithm introduces an additional MAC value per signature, which adds a fixed overhead of $|\mathbb{Z}_q^*|$ bytes per client. This is negligible compared to the overall signcryption size and does not affect the aggregation efficiency.

(3) Theoretical comparison
We compare our method with five representative detection schemes. For detailed experimental analysis, please refer to Section 6.2.

(a) *Stepwise verification:* $O(n)$ time, no parallelism.
(b) *Binary search:* $O(\log n)$ rounds, each requiring aggregate verification.
(c) *Elementary symmetric polynomial:* Efficient for small batches but scales poorly.
(d) *Fault-Tolerant verification:* Require constructing multiple sub-families, leading to high overhead.
(e) *2-D matrix method:* Supports parallelism but requires constructing a matrix and incurs a higher baseline cost.

(4) Correctness
The correctness of the detection algorithm relies on the unforgeability of the MAC values and the correctness of the aggregate verification. Since the MAC is derived from the signer's private key and the message, any invalid signature will cause a mismatch in the MAC verification phase, ensuring that malicious nodes are accurately identified.

4.2. Revocable Mechanism

There is a revocable list $\mathcal{L}_{si}$ in the i-th area server to manage client revocations. When registering, KGC sends client's anonymity PID_C and requirement RT_C to the corresponding AS.

During C2A, APC alerts the corresponding AS when it discovers that C is a malicious client. AS receives the warning and removes C's identity from $\mathcal{L}_{si}$. Then, AS updates its private and public keys, and encrypts its private key using legitimate clients' public keys and sends it to the clients.

When a malicious client mixes into a secure session group, the legitimate group member immediately reports it to the APC, then APC notifies the AS to handle it.

5. Security Proof

5.1. Formal Proof

In this subsection, we give a proof of the security of the proposed signcryption scheme under the ROM. Theorems 1 and 2 prove the confidentiality of the signcryption scheme.

Theorem 1. *The signcryption scheme proposed in this paper is capable of resisting attacks from Type I adversaries in the ROM. Specifically, given that $\mathcal{A}_1$ can win the following game with non-negligible probability (NNP) ϵ within polynomial-time (PT), the challenger $\mathcal{C}_1$ can solve the CDH problem with probability $(1-\frac{1}{q_1})^{q_{ppk}}(1-\frac{q_{uq}}{2^k})\frac{1}{q_1 q_2}\epsilon$.*

Proof. In Type I attacks, $\mathcal{A}_1$ has the ability to replace users' public key, but cannot access the system private key. There is an instance of the CDH problem (P, aP, bP), $\mathcal{C}_1$ interacts with $\mathcal{A}_1$ through the following game and utilizes $\mathcal{A}_1$ ability to solve abP. And we only consider the case of single signcryption.

Initialization Phase: $\mathcal{C}_1$ runs the *System Setup*(1^κ) to initialize the system, randomly selects $s_C, s_A \in \mathbb{Z}_q^*$, sets up $P_{C-pub} = s_C p, P_{A-pub} = s_A P = aP$, broadcasts $params = \{G, q, P, P_{C-pub}, P_{A-pub}, H, \tilde{H}_1, H_1 \sim H_6\}$, and keeps (s_A, s_C) in secret. $\mathcal{C}_1$ randomly selects $s_{si} \in \mathbb{Z}_q^*$ as the private key of AS$_i$ and calculates $Q_{si} = x_{si}P$ as the public key of AS$_i$.

Query Phase: $\mathcal{C}_1$ maintains the initially empty list $\mathcal{L}_1^C$, $\mathcal{L}_1^A$, $\mathcal{L}_2$, $\mathcal{L}_3$, $\mathcal{L}_{user}^1$, and $\mathcal{L}_{user}^2$ for answering and storing a series of queries from $\mathcal{A}_1$.

- H_1 *query*: When $\mathcal{A}_1$ launches a hash query on H_1, if there exits a tuple $(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, RT_{C_i}, \alpha_i^1)$ in $\mathcal{L}_1^C$, $\mathcal{C}_1$ returns α_i^1 to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $\alpha_i^1 \in \mathbb{Z}_q^*$ and returns it to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds the tuple $(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, RT_{C_i}, \alpha_i^1)$ to $\mathcal{L}_1^C$.
- $\tilde{H}_1$ *query*: When $\mathcal{A}_1$ launches a hash query on $\tilde{H}_1$, if there exits a tuple $(ID_{A_i}, R_{A_i}, X_{A_i}, P_{A-pub}, \alpha_i^2)$ in $\mathcal{L}_C^A$, $\mathcal{C}_1$ returns α_i^A to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $\alpha_i^A \in \mathbb{Z}_q^*$ and returns it to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds the tuple $(ID_{A_i}, R_{A_i}, X_{A_i}, P_{A-pub}, \alpha_i^2)$ to $\mathcal{L}_1^A$.
- H_2 *query*: When $\mathcal{A}_1$ launches a hash query on H_2, if there exits a tuple $(ID_{A_i}, V_i, F_i, t_i, \beta_i)$ in $\mathcal{L}_2$, $\mathcal{C}_1$ returns β_i to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects β_i and returns it to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds the tuple $(ID_{A_i}, V_i, F_i, t_i, \beta_i)$ to $\mathcal{L}_2$.
- H_3 *query*: When $\mathcal{A}_1$ launches a hash query on H_3, if there exits a tuple $(PID_{C_i}, M_i, F_i, X_{C_i}, R_{C_i}, t_i, \gamma_i)$ in $\mathcal{L}_3$, $\mathcal{C}_1$ returns γ_i to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $\gamma_i \in \mathbb{Z}_q^*$ and returns it to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds $(PID_{C_i}, M_i, F_i, X_{C_i}, R_{C_i}, t_i, \gamma_i)$ to $\mathcal{A}_1$.
- *Public key query*:

  (1) When $\mathcal{C}_1$ receives a public key query on PID_{C_i} initiated from $\mathcal{A}_1$, if there exits a tuple $(PID_{C_i}, x_{C_i}, ppk_{C_i}, X_{C_i}, R_{C_i})$ in $\mathcal{L}_{user}^1$, $\mathcal{C}_1$ returns (X_{C_i}, R_{C_i}) to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $x_{C_i}, \alpha_i^1 \in \mathbb{Z}_q^*$ and calculates $X_{C_i} = x_{C_i}P$.

  – Case 1: If $PID_{C_i} \neq PID_C^*$, where PID_C^* is the target identity, $\mathcal{C}_1$ randomly selects $ppk_{C_i} \in \mathbb{Z}_q^*$, calculates $R_{C_i} = ppk_{C_i}P - \alpha_i^1 P_{C-pub}$, and returns (X_{C_i}, R_{C_i}) to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds $(PID_{C_i}, x_{C_i}, ppk_{C_i}, X_{C_i}, R_{C_i})$ and $(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, RT_{C_i}, \alpha_i^1)$ to $\mathcal{L}_{user}^1$ and $\mathcal{L}_1^C$, respectively.

  – Case 2: If $PID_{C_i} = PID_C^*$, $\mathcal{C}_1$ randomly selects $r_{C_i} \in \mathbb{Z}_q^*$, calculates $R_{C_i} = r_{C_i}P$, $\mathcal{C}_1$ returns (X_{C_i}, R_{C_i}) to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds $(PID_{C_i}, x_{C_i}, \perp, X_{C_i}, R_{C_i})$ and $(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, RT_{C_i}, \alpha_i^1)$ to $\mathcal{L}_{user}^1$ and $\mathcal{L}_1^C$, respectively.

  (2) When $\mathcal{C}_1$ receives a public key query on ID_{A_i} initiated from $\mathcal{A}_1$, if there exits a tuple $(ID_{A_i}, x_{A_i}, ppk_{A_i}, X_{A_i}, R_{A_i})$ in $\mathcal{L}_{user}^2$, $\mathcal{C}_1$ returns (X_{A_i}, R_{A_i}) to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $x_{A_i}, \alpha_i^2 \in \mathbb{Z}_q^*$ and calculates $X_{A_i} = x_{A_i}P$.

  – Case 1: If $ID_{A_i} \neq ID_A^*$, where ID_A^* is the target identity, $\mathcal{C}_1$ randomly selects $ppk_{A_i} \in \mathbb{Z}_q^*$, calculates $R_{A_i} = ppk_{A_i}P - \alpha_i^2 P_{C-pub}$, and returns (X_{A_i}, R_{A_i}) to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds $(ID_{A_i}, x_{A_i}, ppk_{A_i}, X_{A_i}, R_{A_i})$ and $(ID_{A_i}, R_{A_i}, X_{A_i}, P_{A-pub}, \alpha_i^2)$ to $\mathcal{L}_{user}^2$ and $\mathcal{L}_1^A$, respectively.

– Case 2: If $ID_{A_i} = ID_A^*$, $\mathcal{C}_1$ randomly selects $r_{A_i} \in \mathbb{Z}_q^*$, calculates $R_{A_i} = r_{C_i}P$, $\mathcal{C}_1$ returns (X_{A_i}, R_{A_i}) to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds $(ID_{A_i}, x_{A_i}, \perp, X_{A_i}, R_{A_i})$ and $(ID_{A_i}, R_{A_i}, X_{A_i}, P_{A-pub}, \alpha_i^2)$ to $\mathcal{L}_{user}^2$ and $\mathcal{L}_1^A$, respectively.

- *Partial private key query*: When $\mathcal{C}_1$ receives a partial private key on PID_{C_i} (or ID_{A_i}), if $PID_{C_i} \neq PIDC^*$ (or $ID_{A_i} \neq ID_A^*$), $\mathcal{C}_1$ retrieves ppk_{C_i} (or ppk_{A_i}) from $\mathcal{L}_{user}^1$ (or $\mathcal{L}_{user}^2$) and returns ppk_{C_i} (or ppk_{A_i}) to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $\alpha \in \mathbb{Z}_q^*$ to $\mathcal{A}_1$, and terminates the interaction.
- *Private key query*: When $\mathcal{C}_1$ receives a private key query on PID_{C_i} (or PID_{A_i}), $\mathcal{C}_1$ retrieves s_{C_i} (or s_{A_i}) from $\mathcal{L}_{user}^1$ (or $\mathcal{L}_{user}^2$) and returns s_{C_i} (or s_{A_i}) to $\mathcal{A}_1$.
- *Public key replacement query*:

  (1) $\mathcal{A}_1$ randomly selects $(X_{C_i}^{'}, R_{C_i}^{'})$ as the replacement public key, send it to $\mathcal{C}_1$ and launches a public key replacement query on PID_{C_i}. $\mathcal{C}_1$ performs public key substitution and adds $(PID_{C_i}, \perp, \perp, X_{C_i}^{'}, R_{C_i}^{'})$ to $\mathcal{L}_{user}^1$.

  (2) $\mathcal{A}_1$ randomly selects $(X_{A_i}^{'}, R_{A_i}^{'})$ as the replacement public key, send it to $\mathcal{C}_1$ and launches a public key replacement query on ID_{A_i}. $\mathcal{C}_1$ performs public key substitution and adds $(ID_{A_i}, \perp, \perp, X_{A_i}^{'}, R_{A_i}^{'})$ to $\mathcal{L}_{user}^2$.
- *Signcryption query*: When $\mathcal{C}_1$ receives a signcryption query from $\mathcal{A}_1$ on $(PID_{C_i}, ID_{A_i}, M_i, t_i)$, $\mathcal{C}_1$ obtains the information of PID_{C_i} and ID_{A_i} from $\mathcal{L}_{user}^1$ and $\mathcal{L}_{user}^2$, respectively. If $PID_{C_i} \neq PID_C^*$, $\mathcal{C}_1$ executes signcryption algorithm to generate $\sigma_i = (F_i, u_i, C_i, t_i)$ and return to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $u_i, \gamma_i \in \mathbb{Z}_q^*$, retrieves α_i^1 and β_i from $\mathcal{L}_1^C$ and $\mathcal{L}_2$, respectively, calculates $F_i = u_iP - (\gamma_i(X_{C_i} + Q_{si})) + R_{C_i} + \alpha_i^1 P_{C-pub}$, $V_i = (x_{A_i} + ppk_{A_i})F_i s$, $C_i \oplus \beta_i$, and returns (F_i, u_i, C_i, t_i) to $\mathcal{A}_1$. $\mathcal{C}_1$ adds $(ID_{A_i}, V_i, F_i, t_i, \beta_i)$ and $(PID_{C_i}, M_i, F_i, X_{C_i}, R_{C_i}, t_i, \gamma_i)$ to $\mathcal{L}_2$ and $\mathcal{L}_3$, respectively.
- *Unsigncryption query*: When $\mathcal{C}_1$ receives a unsigncryption query initiated from $\mathcal{A}_1$ on $(PID_{C_i}, ID_{A_i}, \sigma_i, t_i)$, $\mathcal{C}_1$ obtains information of PID_{C_i} and ID_{A_i} from $\mathcal{L}_{user}^1$ and $\mathcal{L}_{user}^1$, respectively. If $ID_{A_i} \neq ID_A^*$, $\mathcal{C}_1$ performs the unsigncryption algorithm to solve M_i and returns M_i to $\mathcal{A}_i$. If $ID_{A_i} = ID_A^*$, $\mathcal{C}_1$ retrieves $(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, \alpha_i^1)$ and $(PID_{C_i}, M_i, F_i, X_{C_i}, R_{X_i}, t_i, \gamma_i)$ from $\mathcal{L}_1^C$ and $\mathcal{L}_3$, respectively, if $u_iP = F_i + \gamma_i(X_{C_i} + Q_{si} + RC_i + \alpha_i^1 P_{C-pub})$ holds, $\mathcal{C}_1$ returns M_i to $\mathcal{A}_1$, otherwise, $\mathcal{C}_1$ rejects σ_i.

  Challenge Phase: After a finite number of queries and answers, A sends $(PID_C, ID_C, M_i^0, M_i^1)$ to $\mathcal{C}_1$. Where M_i^0, M_i^1 are plaintext messages of equal length. If $ID_A \neq ID_A^*$, $\mathcal{C}_1$ randomly selects $\alpha \in \mathbb{Z}_q^*$, returns it to $\mathcal{A}_1$, and terminates the interaction. Otherwise, $\mathcal{C}_1$ randomly selects $\rho \leftarrow \{0, 1\}$, sets up $F_i = bP$, randomly selects $v_i \in \mathbb{Z}_q^*$ such that $V_i = v_iP = F_i(x_A + ppk_A)$, $C_i^\rho = M_i^\rho \oplus H_2(ID_A, V_i, F_i, t_i)$, $\alpha_i^\rho = H_1(PID_C, R_C, X_C, RT_C, P_{C-pub})$, $\gamma_i^\rho = H_3(PID_C, M_i^\rho, F_i, X_C, R_C, t_i)$, randomly selects $u_i^\rho \in \mathbb{Z}_q^*$ such that $u_i^\rho P = F_i + \gamma_i^\rho(X_C, Q_{si}) + R_C + \alpha_i^\rho P_{C-pub}$, and returns $\sigma_i^\rho = (F_i, u_i^\rho, C_i^\rho, t_i)$ to $\mathcal{A}_1$.

  Guessing Phase: $\mathcal{A}_1$ continues to have a finite number of interactions with $\mathcal{C}_1$, but $\mathcal{A}_1$ cannot initiate a partial private key query and unsigncryption query on ID_A^*. After the interaction, $\mathcal{A}_1$ outputs $\rho^{'} \leftarrow \{0, 1\}$. If $\rho^{'} = \rho$, $\mathcal{A}_1$ wins the game. When $ID_A = ID_A^*$, $\mathcal{A}_1$ considers σ_i^* returned by $\mathcal{C}_1$ to be valid unless $\mathcal{A}_1$ initiates a query on H_2. Finally, $\mathcal{C}_1$ outputs the solution of CDH problem $abP = h_{A^*}^{-1}(V_i^* - (x_A^* + r_A^*)F_i)$. The solution procedure for abP is shown in Equation (6).

$$\begin{aligned} V_i^* &= f_i(X_A^* + R_A^* + h_{A^*}P_{A-pub}) \\ &= b(x_A^* + r_A^* + h_{A^*}a)P \\ &= abPh_{A^*} + (x_A^* + r_A^*)F_i \end{aligned} \tag{6}$$

Probability: To calculate the probability that $\mathcal{C}_1$ can successfully solve the CDH problem, let q_1,q_2,q_{ppk}, and q_{uq} denote the number of queries for $H_1 \& \tilde{H}_1$, H_2, *partial private key*, and *unsigncryption*, respectively. $\mathcal{A}_1$ must satisfy the following events to win the game.

(1) $\mathbb{E}_1$: $\mathcal{C}_1$ does not terminate the game during the query phase, i.e., $\mathcal{A}_1$ cannot initiate the query of *partial private key* and *unsigncryption* on target identity. $P[\mathbb{E}_1] = (1 - \frac{1}{q_1})^{q_{ppk}}(1 - \frac{q_{uq}}{2^\kappa})$

(2) $\mathbb{E}_2$: $\mathcal{C}_1$ does not terminate the game during the challenge phase, i.e., $ID_A = ID_A^*$. $P[\mathbb{E}_2] = \frac{1}{q_1}$.

(3) $\mathbb{E}_3$: $\mathcal{C}_1$ does not initiate H_2 query during the guessing phase. $P[\mathbb{E}_1 | \mathbb{E}_2 \bigcap \mathbb{E}_3] = \frac{1}{q_2}$.

Thus, $\mathcal{C}_1$ can solve the CDH problem with probability $((1 - \frac{1}{q_1})^{q_{ppk}}(1 - \frac{q_{uq}}{2^\kappa})\frac{1}{q_1 q_2})\epsilon$, and $\mathcal{A}_1$ can win the game with probability ϵ. However, there is no efficient algorithm to solve the CDH problem in polynomial time. □

Theorem 2. *The signcryption scheme proposed in this paper is capable of resisting attacks from Type II adversaries in the ROM. Specifically, given that $\mathcal{A}_2$ can win the following game with NNP ϵ within PT, the challenger $\mathcal{C}_2$ can solve the CDH problem with probability* $(1 - \frac{1}{q_1})^{q_{pk}}(1 - \frac{q_{uq}}{2^k})\frac{1}{q_1 q_2}\epsilon$.

Proof. In Type II attacks, $\mathcal{A}_2$ can obtain the system private key, but does not has the ability to replace users' public key. There is an instance of the CDH problem (P, aP, bP), $\mathcal{C}_2$ interacts with $\mathcal{A}_2$ through the following game and utilizes $\mathcal{A}_2$ ability to solve abP.

The proof process is the same as in Theorem 1, except that the public key replacement attack cannot be launched.

Probability: To calculate the probability that $\mathcal{C}_2$ can successfully solve the CDH problem, let q_1,q_2,q_{pk}, and q_{uq} denote the number of queries for $H_1\&\tilde{H}_1$, H_2, *private key*, and *unsigncryption*, respectively. $\mathcal{A}_2$ must satisfy the following events to win the game.

(1) $\mathbb{E}_1$: $\mathcal{C}_2$ does not terminate the game during the query phase, i.e., $\mathcal{A}_2$ cannot initiate the query of *private key* and *unsigncryption* on target identity. $P[\mathbb{E}_1] = (1 - \frac{1}{q_1})^{q_{pk}}(1 - \frac{q_{uq}}{2^\kappa})$

(2) $\mathbb{E}_2$: $\mathcal{C}_2$ does not terminate the game during the challenge phase, i.e., $ID_A = ID_A^*$. $P[\mathbb{E}_2] = \frac{1}{q_1}$.

(3) $\mathbb{E}_3$: $\mathcal{C}_2$ does not initiate H_2 query during the guessing phase. $P[\mathbb{E}_1|\mathbb{E}_2 \bigcap \mathbb{E}_3] = \frac{1}{q_2}$.

Thus, $\mathcal{C}_2$ can solve the CDH problem with probability $((1 - \frac{1}{q_1})^{q_{pk}}(1 - \frac{q_{uq}}{2^\kappa})\frac{1}{q_1 q_2})\epsilon$, and $\mathcal{A}_2$ can win the game with probability ϵ. However, there is no efficient algorithm to solve the CDH problem in polynomial time. □

Theorems 3 and 4 prove the unforgeability of the signcryption scheme.

Theorem 3. *The signcryption scheme proposed in this paper is capable of resisting attacks from Type I adversaries in the ROM. Specifically, given that $\mathcal{A}_1$ can win the following game with NNP ϵ PPT, the challenger $\mathcal{C}_1$ can solve the ECDL problem with probability* $(1 - \frac{1}{q_1})^{q_{ppk}}\frac{1}{q_1}\epsilon$.

Proof. There is an instance of the ECDL problem (P, aP), $\mathcal{C}_1$ interacts with $\mathcal{A}_1$ through the following game and utilizes $\mathcal{A}_1$ ability to solve a.

Initialization Phase: $\mathcal{C}_1$ runs the *System Setup* (1^κ) to initialize the system, randomly selects $s_C, s_A \in \mathbb{Z}_q^*$, sets up $P_{C-pub} = s_C p = aP, P_{A-pub} = s_A P$, broadcasts $params = \{G, q, P, P_{C-pub}, P_{A-pub}, H, \tilde{H}_1, H_1 \sim H_6\}$, and keeps (s_A, s_C) in secret. $\mathcal{C}_1$ randomly selects $s_{si} \in \mathbb{Z}_q^*$ as the private key of AS_i and calculates $Q_{si} = x_{si}P$ as the public key of AS_i.

Query Phase: $\mathcal{C}_1$ maintains the initially empty list $\mathcal{L}_1$, $\mathcal{L}_3$, and $\mathcal{L}_{user}$ for answering and storing a series of queries from $\mathcal{A}_1$.

- H_1 *query*, H_3 *query*, *Partial private key*, *Private key*, *Signcryption*, and *Unsignscryption query*: The query content and the answer content are the same as in Theorem 1.
- *Public Key query*: When $\mathcal{C}_1$ receives a public key query on PID_{C_i} initiated from $\mathcal{A}_1$, if there exits a tuple $(PID_{C_i}, x_{C_i}, ppk_{C_i}, X_{C_i}, R_{C_i})$ in $\mathcal{L}_{user}$, $\mathcal{C}_1$ returns (X_{C_i}, R_{C_i}) to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ randomly selects $x_{C_i}, \alpha_i \in \mathbb{Z}_q^*$ and calculates $X_{C_i} = x_{C_i}P$.

(1) If $PID_{C_i} \neq PID_C^*$, $\mathcal{C}_1$ randomly selects $ppk_{C_i} \in \mathbb{Z}_q^*$, calculates $R_{C_i} = ppk_{C_i}P - \alpha_i P_{C-pub}$, and returns (X_{C_i}, R_{C_i}) to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds $(PID_{C_i}, x_{C_i}, ppk_{C_i}, X_{C_i}, R_{C_i})$ and $(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, RT_{C_i}, \alpha_i)$ to $\mathcal{L}_{user}$ and $\mathcal{L}_1$, respectively.

(2) If $PID_{C_i} = PID_C^*$, $\mathcal{C}_1$ randomly selects $r_{C_i} \in \mathbb{Z}_q^*$, calculates $R_{C_i} = r_{C_i}P$, $\mathcal{C}_1$ returns (X_{C_i}, R_{C_i}) to $\mathcal{A}_1$. Then, $\mathcal{C}_1$ adds $(PID_{C_i}, x_{C_i}, \perp, X_{C_i}, R_{C_i})$ and $(PID_{C_i}, R_{C_i}, X_{C_i}, P_{C-pub}, RT_{C_i}, \alpha_i)$ to $\mathcal{L}_{user}$ and $\mathcal{L}_1$, respectively.

Forgery Phase: After bounded interactions with $\mathcal{C}_1$, if $\mathcal{A}_1$ does not initiate a partial private key on PID_C^* or unsignscryption query on (PID_C^*, ID_A^*, M_i^*), and then $\mathcal{A}_1$ outputs the signature $\sigma_C^* = (F_C, u_C^*)$ on (PID_C^*, M_i^*). According to the forking lemma [35], $\mathcal{C}_1$ can select a different hash function H_1 yields another signature $\sigma_C^{'} = (F_C, u_C^{'})$ for the (PID_C^*, M_i^*). For this we get the following equations.

$$\begin{aligned} u_C^* P &= (f_C + \gamma_C(x_C + s_{si}) + ppk_C)P \\ &= F_C + \gamma_C(X_C + Q_{si}) + R_C + \alpha_C^* P_{C-pub} \end{aligned} \tag{7}$$

$$\begin{aligned} u_C^{'} P &= (f_C + \gamma_C(x_C + s_{si}) + ppk_C)P \\ &= F_C + \gamma_C(X_C + Q_{si}) + R_C + \alpha_C^{'} P_{C-pub} \end{aligned} \tag{8}$$

$\mathcal{C}_1$ can solve the ECLD problem by Equations (7) and (8),

$$\begin{aligned} (u_C^* - u_C^{'})P &= (\alpha_C^* - \alpha_C^{'})aP \\ a &= (\alpha_C^* - \alpha_C^{'})^{-1}(u_C^* - u_C^{'}) \end{aligned} \tag{9}$$

Probability: To calculate the probability that $\mathcal{C}_1$ can successfully solve the ECDL problem, let q_1 and q_{ppk}, denote the number of queries for H_1 and *partial private key*, respectively. $\mathcal{A}_1$ must satisfy the following events to win the game.

(1) $\mathbb{E}_1$: $\mathcal{C}_1$ does not terminate the game during the query phase, i.e., $\mathcal{A}_1$ cannot initiate the query of *partial private key* on target identity. $P[\mathbb{E}_1] = (1 - \frac{1}{q_1})^{q_{ppk}}$

(2) $\mathbb{E}_2$: $\mathcal{C}_1$ does not terminate the game during the forgery phase. $P[\mathbb{E}_1|\mathbb{E}_2] = \frac{1}{q_1}$.

Thus, $\mathcal{C}_1$ can solve the ECDL problem with probability $((1 - \frac{1}{q_1})^{q_{ppk}} \frac{1}{q_1})\epsilon$, and $\mathcal{A}_1$ can win the game with probability ϵ. However, there is no efficient algorithm to solve the ECDL problem in polynomial time. □

Theorem 4. *The signcryption scheme proposed in this paper is capable of resisting attacks from Type II adversaries in the ROM. Specifically, given that $\mathcal{A}_2$ can win the following game with NNP ϵ PPT, the challenger $\mathcal{C}_2$ can solve the ECDL problem with probability* $(1 - \frac{1}{q_1})^{q_{pk}} \frac{1}{q_1}\epsilon$.

Proof. There is an instance of the ECDL problem (P, aP), $\mathcal{C}_2$ interacts with $\mathcal{A}_2$ through the following game and utilizes $\mathcal{A}_2$ ability to solve a.

The proof process is the same as in Theorem 3, except that the public key replacement attack cannot be launched.

Probability: To calculate the probability that $\mathcal{C}_2$ can successfully solve the ECDL problem, let q_1 and q_{pk}, denote the number of queries for H_1 and *private key*, respectively. $\mathcal{A}_2$ must satisfy the following events to win the game.

(1) $\mathbb{E}_1$: $\mathcal{C}_2$ does not terminate the game during the query phase, i.e., $\mathcal{A}_2$ cannot initiate the query of *private key* on target identity. $P[\mathbb{E}_1] = (1 - \frac{1}{q_1})^{q_{pk}}$

(2) $\mathbb{E}_2$: $\mathcal{C}_2$ does not terminate the game during the forgery phase. $P[\mathbb{E}_1|\mathbb{E}_2] = \frac{1}{q_1}$.

Thus, $\mathcal{C}_2$ can solve the ECDL problem with probability $((1 - \frac{1}{q_1})^{q_{pk}} \frac{1}{q_1})\epsilon$, and $\mathcal{A}_2$ can win the game with probability ϵ. However, there is no efficient algorithm to solve the ECDL problem in polynomial time. □

5.2. Informal Analysis

Confidentiality, Unforgeability, Integrity, Forgery and Nonrepudiation: It's proven in Theorems 1–4.

Anonymity: The hardness of the ECDL and CDH assumption guarantees that the attacker cannot crack the real identity through the transmitted anonymous identity.

Traceability: In special cases, KGC can calculate the real identity $ID_{C_i} = PID_{C_i} \oplus H(r_{C_i} P_{C-pub}, T_C)$.

Resistance on replay attack: In the scheme, current timestamps are used to keep messages fresh.

Invalid signature traceability: Using MAC values ensures that the aggregated signcryption scheme can quickly find invalid signature nodes.

Forward secrecy: If (x_{C_i}, ppk_{C_i}) is leaked, the attacker cannot accurately calculate $V_i = (f_{i1} x_{C_i} + f_{i2} ppk_{C_i})(X_A + R_A + h_A P_{A-pub})$, because f_{i1} and f_{i2} are unknown unless the CDH assumption can be cracked.

6. Performance Analysis

In this section, the proposed CLASC scheme is comprehensively evaluated in terms of computational cost, communication overhead, and security properties. The experimental platform was implemented in Python, leveraging the Pypbc cryptographic library, and executed within a virtual machine environment configured with 4 GB of memory and running Ubuntu 22.04. Table 2 reports the average execution time of different cryptographic operations, which serves as the baseline data for subsequent performance evaluations. For the experimental setup, in bilinear pairing–based schemes, the element size of group G_T was set to 128 bytes; in ECC-based schemes, the element size of group G was set to 40 bytes, while elements in the modular group $\mathbb{Z}_q^*$ were set to 20 bytes. Furthermore, to ensure both the comparability and generality of the experimental results, the size of each signature data block m was uniformly fixed at 40 bytes.

Table 2. Running time of cryptographic operation.

Notation	Description	Running Time(ms)
$\mathcal{O}_{bp}$	Bilinear pairing operation	3.234053
$\mathcal{O}_{sm}$	Scalar multiplication operation	0.943208
$\mathcal{O}_{pa}$	Point addition operation	0.006740
$\mathcal{O}_{mg}$	Map to G hash operation	2.050080
$\mathcal{O}_{mz}$	Map to $\mathbb{Z}_q^*$ hash operation	0.004499

For comparative evaluation, 8 certificateless signcryption schemes are selected: Basudan et al. [23], Yang et al. [22], Yu et al. [24], Dai et al. [25], Ren et al. [26], Wang et al. [27], Li et al. [34], and Chen et al. [28]

6.1. Certificateless Aggregate Signcryption Scheme

6.1.1. Computational Cost

Table 3 summarizes the computational costs of various signcryption schemes. The "Total cost" comprises two components: (i) the cost for n clients to generate signatures and ciphertexts; and (ii) the cost for the APC to decrypt and verify n ciphertexts. Figure 4 visually compares the computational overhead across schemes during the decryption and verification of n ciphertexts. As illustrated in Table 3 and Figure 4, the overall computational cost of our scheme is lower than those in [22, 23, 28], though marginally higher than those in [24–27, 34]. This modest overhead arises by design: to handle aggregate verification failures efficiently, we introduce an additional authentication value that enables rapid identification and exclusion of invalid signatures. Although this mechanism adds some computational load, it substantially improves system robustness and practical usability in adversarial settings, ultimately achieving a more desirable trade-off between security and efficiency.

Table 3. Comparison of the computational cost of related schemes.

Scheme	Signcryption Cost (ms)	Unsigncryption Cost (ms)	Total Cost (ms)
Basudan [23]	$5\mathcal{O}_{sm} + 2\mathcal{O}_{pa} + 3\mathcal{O}_{mz}$ $= 4.743017$	$(2n+3)\mathcal{O}_{bp} + (2n-2)\mathcal{O}_{pa} +$ $(2n+1)\mathcal{O}_{mz} = 6.490584n + 9.693178$	$(2n+3)\mathcal{O}_{bp} + 5n\mathcal{O}_{sm} + (4n-2)\mathcal{O}_{pa}$ $+(5n+1)\mathcal{O}_{mz} = 11.233595n + 9.693178$
Yang [22]	$6\mathcal{O}_{sm} + 3\mathcal{O}_{mg} + 2\mathcal{O}_{mz}$ $= 11.818486$	$5\mathcal{O}_{bp} + (3n+3)\mathcal{O}_{sm} + 5n\mathcal{O}_{mg} + (5n-4)$ $\mathcal{O}_{pa} + 2n\mathcal{O}_{mz} = 13.122722n + 18.972929$	$5\mathcal{O}_{bp} + (6n+3)\mathcal{O}_{sm} + 8n\mathcal{O}_{mg} + (5n-4)$ $\mathcal{O}_{pa} + 4n\mathcal{O}_{mz} = 22.111584n + 18.972929$
Yu [24]	$3\mathcal{O}_{sm} + 2\mathcal{O}_{pa} + 3\mathcal{O}_{mz}$ $= 2.856601$	$2n\mathcal{O}_{sm} + (4n-1)\mathcal{O}_{pa} + 3n\mathcal{O}_{mz}$ $= 1.926873n - 0.00674$	$5n\mathcal{O}_{sm} + (6n-1)\mathcal{O}_{pa} + 6n\mathcal{O}_{mz}$ $= 4.783474n - 0.006740$
Dai [25]	$3\mathcal{O}_{sm} + \mathcal{O}_{pa} + 3\mathcal{O}_{mz}$ $= 2.849861$	$(3n+1)\mathcal{O}_{sm} + (3n-1)\mathcal{O}_{pa} + 3n\mathcal{O}_{mz}$ $= 2.863341n + 0.936468$	$(6n+1)\mathcal{O}_{sm} + (4n-1)\mathcal{O}_{pa} + 6n\mathcal{O}_{mz}$ $= 5.713202n + 0.936468$
Ren [26]	$2\mathcal{O}_{sm} + \mathcal{O}_{mz} = 1.890915$	$2n\mathcal{O}_{sm} + (2n-1)\mathcal{O}_{pa} + 2n\mathcal{O}_{mz}$ $= 1.908894n - 0.00674$	$4n\mathcal{O}_{sm} + (2n-1)\mathcal{O}_{pa} + 4n\mathcal{O}_{mz}$ $= 3.804308n - 0.006740$
Wang [27]	$3\mathcal{O}_{sm} + 2\mathcal{O}_{mz} = 2.838622$	$2n\mathcal{O}_{sm} + (2n-1)\mathcal{O}_{pa} + 2n\mathcal{O}_{mz}$ $= 1.908894n - 0.00674$	$5n\mathcal{O}_{sm} + (2n-1)\mathcal{O}_{pa} + 4n\mathcal{O}_{mz}$ $= 4.747516n - 0.006740$
Li [34]	$3\mathcal{O}_{sm} + \mathcal{O}_{pa} + 4\mathcal{O}_{mz}$ $= 2.85436$	$2n\mathcal{O}_{sm} + (2n-1)\mathcal{O}_{pa} + 2n\mathcal{O}_{mz}$ $= 1.908894n - 0.00674$	$5n\mathcal{O}_{sm} + (3n-1)\mathcal{O}_{pa} + 6n\mathcal{O}_{mz}$ $= 4.763254n - 0.006740$
Chen [28]	$3\mathcal{O}_{sm} + 3\mathcal{O}_{pa} + 5\mathcal{O}_{mz}$ $= 2.872339$	$(4n+1)\mathcal{O}_{sm} + (3n-3)\mathcal{O}_{pa} + (3n+2)\mathcal{O}_{mz}$ $= 3.786329n + 0.931986$	$7n\mathcal{O}_{sm} + (6n-1)\mathcal{O}_{pa} + 6n\mathcal{O}_{mz}$ $= 6.66989n + 0.931986$
Our shceme	$4\mathcal{O}_{sm} + 2\mathcal{O}_{pa} + 5\mathcal{O}_{mz}$ $= 3.808807$	$(2n+2)\mathcal{O}_{sm} + (3n+1)\mathcal{O}_{pa} + 3n\mathcal{O}_{mz}$ $= 1.920133n + 1.893156$	$(6n+2)\mathcal{O}_{sm} + (6n-1)\mathcal{O}_{pa} + 8n\mathcal{O}_{mz}$ $= 5.73568n + 0.936468$

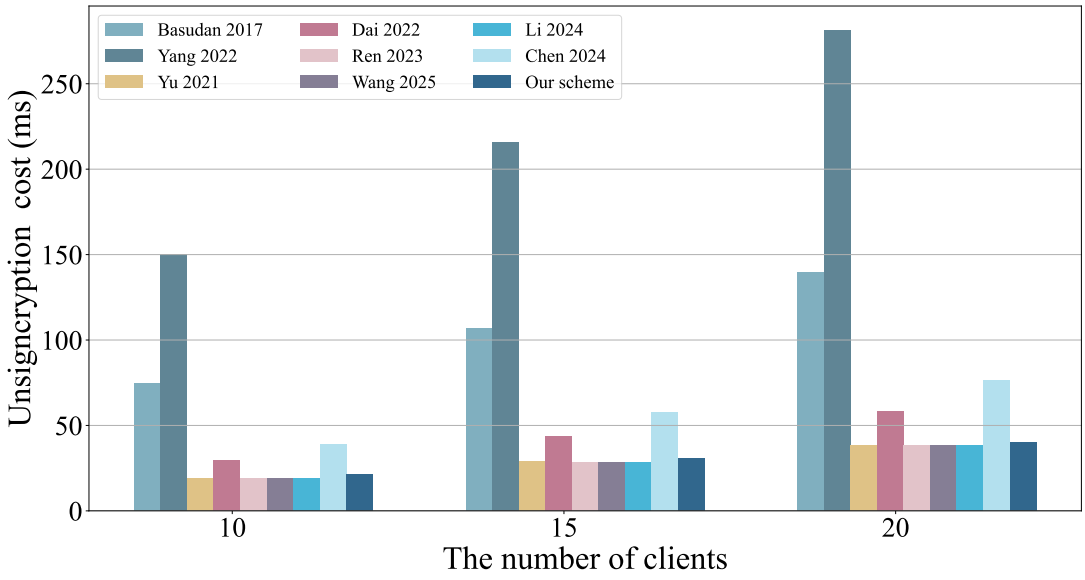

Figure 4. Computation comparison with related schemes of the total cost [22–28, 34].

6.1.2. Communication Overhead & Security Property

Table 4 compares the communication overhead of the proposed scheme with several representative existing schemes during signcryption and aggregation. The results indicate that for signcryption operations, our scheme incurs significantly lower overhead than those in [22,23,26,27] and is comparable to [24,25,28,34]. In aggregation, the proposed scheme again demonstrates lower overhead than [22,23,26,27,34], while performing on par with [24,25,28]. Furthermore, in terms of security attributes, the proposed scheme offers enhanced protection, effectively countering common attacks including replay, forgery, and invalid signature injection. These findings confirm that the proposed design not only achieves superior communication efficiency but also maintains strong security, thereby striking an effective balance between performance and protection.

Table 4. Comparison of the communication cost and security properties of related schemes.

Scheme	Signcryption (Bytes)	Aggregate (Bytes)	①	②	③	④	⑤	⑥
Basudan [23]	$2\|G_T\| + \|m\| = 296$	$(n+1)\|G_T\| + n\|m\| = 168n + 128$	+	+	+	−	+	−
Yang [22]	$\|\mathbb{Z}_q^*\| + 3\|G_T\| + \|m\| = 444$	$(n+1)\|G_T\| + n\|m\| + n\mathbb{Z}_q^* = 188n + 128$	−	+	+	+	+	−
Yu [24]	$2\|\mathbb{Z}_q^*\| + \|G\| + \|m\| = 120$	$n\|G\| + n\|m\| + (n+1)\mathbb{Z}_q^* = 100n + 20$	+	+	+	−	+	−
Dai [25]	$2\|\mathbb{Z}_q^*\| + \|G\| + \|m\| = 120$	$n\|G\| + n\|m\| + (n+1)\mathbb{Z}_q^* = 100n + 20$	+	+	+	+	+	−
Ren [26]	$\|\mathbb{Z}_q^*\| + 2\|G\| + \|m\| = 140$	$2n\|G\| + n\|m\| + \mathbb{Z}_q^* = 120n + 20$	−	+	+	−	+	−
Wang [27]	$\|\mathbb{Z}_q^*\| + 2\|G\| + \|m\| = 140$	$2n\|G\| + n\|m\| + \mathbb{Z}_q^* = 120n + 20$	−	+	+	−	+	−
Li [34]	$2\|\mathbb{Z}_q^*\| + \|G\| + \|m\| = 120$	$(n+2)\|G\| + n\|m\| + (n+2)\mathbb{Z}_q^* = 100n + 120$	+	+	+	+	+	+
Chen [28]	$2\|\mathbb{Z}_q^*\| + \|G\| + \|m\| = 120$	$n\|G\| + n\|m\| + (n+1)\mathbb{Z}_q^* = 100n + 20$	+	+	+	+	+	−
Our scheme	$2\|\mathbb{Z}_q^*\| + \|G\| + \|m\| = 120$	$n\|G\| + n\|m\| + (n+1)\mathbb{Z}_q^* = 100n + 20$	+	+	+	+	+	+

① Anonymity ② Traceability ③ Forward security ④ Resist replay attack ⑤ Type I and Type II attacks ⑥ Invalid signature detection; $|G|$, $|G_T|$, $|\mathbb{Z}_q^*|$, and $|m|$: the length of an element in G, G_T, $\mathbb{Z}_q^*$, and m, respectively; n: the number of clients; +: Resist; −: Not resist.

6.2. Evaluate Invalid Signature Detection Algorithm

Efficient detection of invalid signature nodes is essential when aggregate verification fails. Existing schemes include stepwise verification, binary search, elementary symmetric polynomial-based verification, fault-tolerant aggregate verification, and 2-D matrix batch verification. In our scheme, single-signature verification requires $3\mathcal{O}_{sm} + 4\mathcal{O}_{pa} + 2\mathcal{O}_{mz} = 2.865582\,ms$, while aggregate verification of n signatures takes $C_n = (n+2)\mathcal{O}_{sm} + (3n+1)\mathcal{O}_{pa} + 2n\mathcal{O}_{mz} = 0.972426n + 1.893156\,ms$. For experimental consistency, we assume a batch size of 500 signatures, with one invalid signature, and all ciphertexts decrypted beforehand.

(1) Stepwise verification: In the worst case, this method requires validating all signatures individually, resulting in a total time of $(3\mathcal{O}_{sm} + 4\mathcal{O}_{pa} + 2\mathcal{O}_{mz}) * 500 = 1432.791\,ms$.

(2) Binary search: Upon verification failure, the signature set is recursively split into two subgroups, each verified separately [30]. For 500 signatures, the total cost is $2C_{250} + 2C_{125} + C_{63} + C_{62} + C_{31} + C_{32} + 2C_{16} + 2C_8 + 2C_4 + 2C_2 + 2C_1 = 1006.503\,ms$.

(3) Elementary symmetric polynomial verification: This approach partitions signatures into m batches and leverages elementary symmetric polynomials to identify invalid groups and nodes [31]. With $m = 10$, the total detection time is $10C_{10} + 50C_1 + 60\mathcal{O}_{sm} + 58\mathcal{O}_{pa} = 705.278\,ms$.

(4) Fault-Tolerant verification: Based on the d-cover-free family concept, this method constructs multiple signature sub-families with limited overlap [32,33]. For 500 signatures, 33 sub-families of 32 signatures each are constructed, requiring $33C_{32} = 1089.356\,ms$ to locate the invalid node.

(5) 2-D matrix batch verification: Signatures are arranged in an $a \times b$ matrix where $a \times b = n$. For $n = 500$, a 20×25 matrix is built [34]. Detection requires $20C_{25} + 25C_{20} = 1057.542\,ms$ in sequential mode, reducible to $C_{25} + C_{20} = 47.545\,ms$ under parallel computation.

(6) Our method: We employ a MAC-based mechanism for invalid signature detection. This approach supports parallel processing of m subgroups, significantly improving efficiency. A single MAC check costs $M_1 = \mathcal{O}_{sm} + 2\mathcal{O}_{mz} = 0.952206\,ms$. With $m = 25$ and multi-threading, the total detection time is $C_{20} + 20M_1 = 40.386\,ms$; without multi-threading, it is $500M_1 = 476.103\,ms$.

In summary, the proposed detection algorithm achieves superior efficiency in identifying invalid signatures. Although it introduces modest additional computation, the method enables rapid and precise localization of malicious nodes, greatly enhancing system resilience in the event of aggregate verification failures. Thus, the algorithm effectively balances security and operational efficiency.

6.3. Discussion

The proposed scheme is designed with a strong emphasis on communication efficiency and system throughput. During the encryption phase, each client transmits only 120 bytes of encrypted data, a size comparable to that of other ECC-based schemes. Although a 20-byte MAC value is incorporated to enable invalid signature detection, its impact on overall transmission volume remains negligible. Furthermore, the aggregation mechanism reduces the number of communication rounds, thereby improving bandwidth utilization. These characteristics make the scheme particularly suitable for Enhanced Mobile Broadband (eMBB) applications, which require support for high concurrency, low latency, and large-scale data transmission. The key advantages of the proposed scheme include:

(1) Low communication overhead: Renders the scheme suitable for high-frequency, multi-device communication scenarios.
(2) Batch verification and rapid detection: Significantly cuts verification latency and improves system responsiveness.
(3) Lightweight cryptographic operations: Relies exclusively on ECC, avoiding computationally expensive bilinear pairings, and is thus well-suited for terminal and edge devices.

Therefore, this scheme is not only suitable for wireless medical sensor networks but also has the potential for deployment in eMBB and other high-speed, multi-connection scenarios. In the future, it will be validated and optimized in more IoT environments with high bandwidth demands.

7. Conclusions

This paper introduces an efficient CLASC scheme tailored for WMSNs. By combining elliptic curve cryptography with an anonymity mechanism, the scheme guarantees data confidentiality, integrity, and identity privacy, while substantially reducing the overhead typically associated with traditional certificate management. To counteract performance degradation caused by aggregate verification failures, we further propose an invalid signature detection algorithm that rapidly locates and excludes malicious nodes, thereby strengthening system robustness and practical utility. Security analysis includes rigorous proofs in the random oracle model, confirming the scheme's resilience to adaptive chosen-ciphertext attacks and signature forgery. Experimentally, the scheme achieves a better balance between communication overhead and computational efficiency than existing alternatives, exhibiting superior scalability and performance. Overall, the proposed scheme not only ensures strong theoretical security and privacy but also demonstrates compelling practical performance, underscoring its potential to enable secure, reliable, and scalable data exchange in WMSNs.

Author Contributions

C.H.: conceptualization, methodology, software; X.C.: data curation, writing—original draft preparation; Y.C.: visualization, investigation; X.X.: supervision; B.C.: software, validation; J.Y.: writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grants 62372075 and 62272256; in part by the Natural Science Foundation of Chongqing, China under Grant CSTB2024NSCQLZX0084; Graduate Scientific Research and Innovation Foundation of Chongqing(CYB23054).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

No AI tools were utilized for this paper.

References

1. Yu, S.; Park, Y. A Robust Authentication Protocol for Wireless Medical Sensor Networks Using Blockchain and Physically Unclonable Functions. *IEEE Internet Things J.* **2022**, *9*, 20214–20228.
2. Rattal, S.; Badri, A.; Moughit, M.; et al. AI-Driven Optimization of Low-Energy IoT Protocols for Scalable and Efficient Smart Healthcare Systems. *IEEE Access* **2025**, *13*, 48401–48415
3. Oztoprak, A.; Hassanpour, R.; Ozkan, A.; et al. Security Challenges, Mitigation Strategies, and Future Trends in Wireless Sensor Networks: A Review. *ACM Comput. Surv.* **2024**, *57*, 1–29.
4. Kaur, R.; Shahrestani, S.; Ruan, C. Security and Privacy of Wearable Wireless Sensors in Healthcare: A Systematic Review. *Comput. Netw. Commun.* **2024**, *2*, 27–52.
5. Manikandan, R.; Arunprakash, S.; Alsowail, R.A.; et al. A Novel Wireless Sensor Network Deployment for Monitoring and Predicting Abnormal Actions in Medical Environment and Patient Health State. *Alex. Eng. J.* **2025**, *119*, 149–167.
6. He, D.; Cai, Y.; Zhu, S.; et al. A Lightweight Authentication and Key Exchange Protocol With Anonymity for IoT. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 7862–7872.
7. Jiang, Q.; Huang, X.; Zhang, N.; et al. Shake to Communicate: Secure Handshake Acceleration-Based Pairing Mechanism for Wrist Worn Devices. *IEEE Internet Things J.* **2019**, *6*, 5618–5630.
8. Chen, C.M.; Wang, K.H.; Yeh, K.H.; et al. Attacks and Solutions on a Three-Party Password-Based Authenticated Key Exchange Protocol for Wireless Communications. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3133–3142.
9. Hassan, W.H. Current Research on Internet of Things (IoT) Security: A Survey. *Comput. Netw.* **2019**, *148*, 283–294.
10. Sangari, A.S.; Manickam, J.M.L. Public Key Cryptosystem Based Security in Wireless Body Area Network. In Proceedings of the 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], Nagercoil, India, 20–21 March 2014; pp. 1609–1612.
11. Li, J.; Chen, X.; Li, M.; et al. Secure Deduplication With Efficient and Reliable Convergent Key Management. *IEEE Trans.Parallel Distrib. Syst.* **2013**, *25*, 1615–1625.
12. Li, M.; Lou, W.; Ren, K. Data Security and Privacy in Wireless Body Area Networks. *IEEE Wireless Commun.* **2010**, *17*, 51–58.
13. Shen, J.; Miao, T.; Lai, J.; et al. IMS: An Identity-Based Many-to-Many Subscription Scheme With Efficient Key Management for Wireless Broadcast Systems. *IEEE Trans. Serv. Comput.* **2022**, *15*, 1707–1719.
14. Ding, R.; Zhong, H.; Ma, J.; et al. Lightweight Privacy-Preserving Identity-Based Verifiable IoT-Based Health Storage System. *IEEE Internet Things J.* **2019**, *6*, 8393–8405.
15. Xiong, H.; Hou, Y.; Huang, X.; et al. Heterogeneous Signcryption Scheme From IBC to PKI With Equality Test for WBANs. *IEEE Syst. J.* **2021**, *16*, 2391–2400.
16. Al-Riyami, S.S.; Paterson, K.G. Certificateless Public Key Cryptography. In Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 30 November–4 December 2003; Volume 2894, pp. 452–473.
17. Yuanbing, W.; Wanrong, L.; Bin, L. An Improved Authentication Protocol for Smart Healthcare System Using Wireless Medical Sensor Network. *IEEE Access* **2021**, *9*, 105101–105117.
18. Ali, R.; Pal, A.K.; Kumari, S.; et al. An Enhanced Three Factor Based Authentication Protocol Using Wireless Medical Sensor Networks for Healthcare Monitoring. *J. Ambient. Intell. Humaniz. Comput.* **2024**, *15*, 1165–1186.
19. Vimal, V.; Raina, V.; AlGhamdi, A.; et al. Certificate-Less Healthcare Signature Scheme for Secure Consumer Technology-Centric Wireless Medical Sensor Devices. *IEEE Trans. Consum. Electron.* **2025**, *71*, 2.
20. Rabie, O.B.J.; Selvarajan, S.; Hasanin, T.; et al. A Full Privacy-Preserving Distributed Batch-Based Certificate-Less Aggregate Signature Authentication Scheme for Healthcare Wearable Wireless Medical Sensor Networks (HWMSNs). *Int. J. Inf. Secur.* **2024**, *23*, 51–80.
21. Nayak, P.; Swapna, G. Security Issues in IoT Applications Using Certificateless Aggregate Signcryption Schemes: An Overview. *Internet Things* **2023**, *21*, 100641.
22. Yang, Y.; Zhang, L.; Zhao, Y.; et al. Privacy-Preserving Aggregation-Authentication Scheme for Safety Warning System in Fog-Cloud Based VANET. *IEEE Trans. Inf. Forensic Secur.* **2022**, *17*, 317–331.
23. Basudan, S.; Lin, X.; Sankaranarayanan, K. A Privacy-Preserving Vehicular Crowdsensing-Based Road Surface Condition Monitoring System Using Fog Computing. *IEEE Internet Things J.* **2017**, *4*, 772–782.
24. Yu, H.; Ren, R. Certificateless Elliptic Curve Aggregate Signcryption Scheme. *IEEE Syst. J.* **2021**, *16*, 2347–2354.
25. Dai, C.; Xu, Z. Pairing-Free Certificateless Aggregate Signcryption Scheme for Vehicular Sensor Networks. *IEEE Internet Things J.* **2022**, *10*, 5063–5072.
26. Ren, R.; Su, J. A Security-Enhanced and Privacy-Preserving Certificateless Aggregate Signcryption Scheme-Based Artificial Neural Network in Wireless Medical Sensor Network. *IEEE Sens. J.* **2023**, *23*, 7440–7450.

27. Wang, Y.; Peng, C.; Jia, X.; et al. Pairing-Free Blockchain-Assisted Certificateless Aggregation Signcryption Scheme for Vanets. *IEEE Internet Things J.* **2025**, *12*, 15545–15557
28. Chen, D.; Zhou, F.; Liu, Y.; et al. Secure Pairing-Free Certificateless Aggregate Signcryption Scheme for IoT. *J. Syst. Archit.* **2024**, *156*, 103268.
29. Zhang, J.; Shi, C. An Enhanced-Security Certificateless Aggregate Signcryption for Secure Data Transmission in Resource-Constrained Networks. *IEEE Internet Things J.* **2025**, *12*, 34086–34101.
30. Huang, J.L.; Yeh, L.Y.; Chien, H.Y. ABAKA: An Anonymous Batch Authenticated and Key Agreement Scheme for Value-Added Services in Vehicular Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2010**, *60*, 248–262.
31. Xiong, H.; Wu, Y.; Su, C.; et al. A Secure and Efficient Certificateless Batch Verification Scheme With Invalid Signature Identification for the Internet of Things. *J. Inf. Secur. Appl.* **2020**, *53*, 102507.
32. Hartung, G.; Kaidel, B.; Koch, A.; et al. Fault-Tolerant Aggregate Signatures. In *Public-Key Cryptography—PKC 2016, 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, 6–9 March 2016*; Springer: Berlin, Germany, 2016; Volume 9614, pp. 331–356.
33. Wang, G.; Cao, Z.; Dong, X.; et al. Improved Fault-Tolerant Aggregate Signatures. *Comput. J.* **2019**, *62*, 481–489.
34. Li, X.; Zhu, R.; Du, D.; et al. Ecc-Based Certificateless Aggregate Signcryption Scheme in Cyber-Physical Power Systems. *IEEE Syst. J.* **2024**, *18*, 893–904.
35. Pointcheval, D.; Stern, J. Security Arguments for Digital Signatures and Blind Signatures. *J. Cryptol.* **2000**, *13*, 361–396.