sensors

MDPI

Article

An Efficient Certificateless Anonymous Signcryption Scheme for WBAN

Weifeng Long [1,2,*], Lunzhi Deng [2], Jiwen Zeng [1], Yan Gao [2] and Tianxiu Lu [2]

[1] School of Mathematical Sciences, Xiamen University, Xiamen 361005, China; jwzeng@xmu.edu.cn
[2] Guizhou Provincial Specialized Key Laboratory of Information Security Technology in Higher Education Institutions, School of Mathematical Sciences, Guizhou Normal University, Gui'an New District, Guiyang 550025, China; denglunzhi@163.com (L.D.); 19010060164@gznu.edu.cn (Y.G.); 232200061298@gznu.edu.cn (T.L.)
* Correspondence: 460143769@gznu.edu.cn

Abstract: A Wireless Body Area Network (WBAN), introduced into the healthcare sector to improve patient care and enhance the efficiency of medical services, also brings the risk of the leakage of patients' privacy. Therefore, maintaining the communication security of patients' data has never been more important. However, WBAN faces issues such as open medium channels, resource constraints, and lack of infrastructure, which makes the task of designing a secure and economical communication scheme suitable for WBAN particularly challenging. Signcryption has garnered attention as a solution suitable for resource-constrained devices, offering a combination of authentication and confidentiality with low computational demands. Although the advantages offered by existing certificateless signcryption schemes are notable, most of them only have proven security within the random oracle model (ROM), lack public ciphertext authenticity, and have high computational overheads. To overcome these issues, we propose a certificateless anonymous signcryption (CL-ASC) scheme suitable for WBAN, featuring anonymity of the signcrypter, public verifiability, and public ciphertext authenticity. We prove its security in the standard model, including indistinguishability, unforgeability, anonymity of the signcrypter, and identity identifiability, and demonstrate its superiority over relevant schemes in terms of security, computational overheads, and storage costs.

Keywords: signcryption; certificateless cryptography; Wireless Body Area Networks (WBANs); standard model; efficient; security

Citation: Long, W.; Deng, L.; Zeng, J.; Gao, Y.; Lu, T. An Efficient Certificateless Anonymous Signcryption Scheme for WBAN. *Sensors* **2024**, *24*, 4899. https://doi.org/10.3390/s24154899

Academic Editor: Jiankun Hu

Received: 5 June 2024
Revised: 19 July 2024
Accepted: 26 July 2024
Published: 28 July 2024

1. Introduction

The latest data by the World Health Organization (WHO) reveals that the average global life expectancy has reached 73.4 years. Furthermore, demographic projections estimate that by 2050, the number of individuals aged 60 and above will surge to 2.1 billion. This demographic shift towards an older population is exacerbating the shortage of medical resources, making healthcare for the elderly a critical issue for nations worldwide. The escalating costs of healthcare have driven medical systems to embrace new technologies to enhance current practices. To capitalize on the benefits of wireless technology in the realms of telemedicine and mobile health, a novel type of wireless network has emerged: the Wireless Body Area Network (WBAN) [1]. The WBAN is a specialized sensor network that facilitates the exchange of vital health information between patients and healthcare providers via the internet.

A standard WBAN encompasses an array of either implantable [2,3] or wearable sensor nodes and control units [4]. The role of these sensor nodes is to diligently monitor the critical physiological parameters of individuals, covering a range of critical health indicators such as blood pressure, oxygen saturation levels, respiratory rate, heart rate, skin temperature, and various other essential signs of life. In addition to these vital signs, they also measure environmental factors, such as ambient temperature, humidity levels, and light intensities.

The sensor nodes engage in communication with a central controller, which acts as a conduit for relaying the aggregated health data to medical personnel and servers within the network. The WBAN framework is shown in Figure 1. The implementation of WBAN has significantly enhanced the efficiency of healthcare delivery, as it reduces the frequency with which patients need to visit hospitals. Furthermore, the system is capable of facilitating clinical diagnoses and providing some emergency medical responses. Given the significant role that WBAN will play in the healthcare system, it is projected that the WBAN market will exceed 19 trillion US dollars in the next few years [5]. It is expected that there will be 100 billion Internet of Things (IoT) devices in operation globally by the year 2025, with an expected economic impact that will exceed 11 trillion US dollars [6].

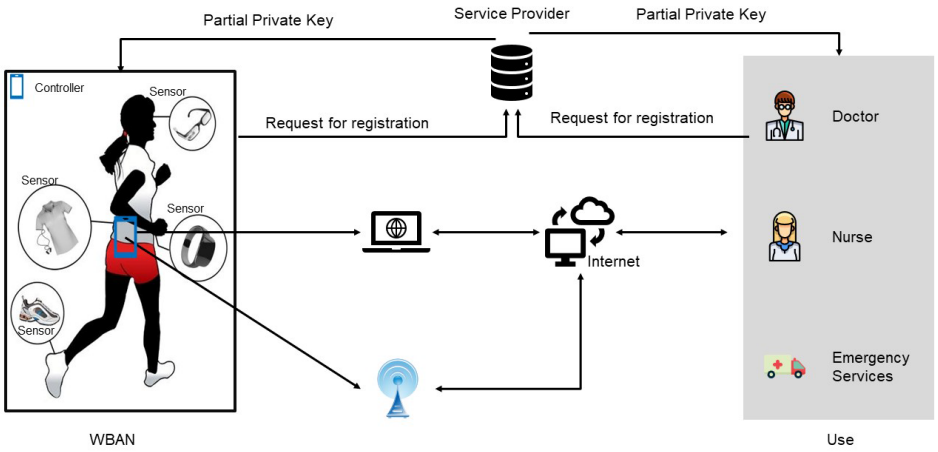

Figure 1. WBAN Framework.

Garnering enormous economic interest, WBAN may be confronted with the risks of data misuse and infringement of user privacy. Although various countries are continuously improving their regulatory systems, their strategies focus on effectiveness and security. For example, the EU's General Data Protection Regulation (GDPR), which came into effect in May 2018, granted privacy regulatory authorities the right to impose fines or file lawsuits against individual companies. It drives societal attention to privacy and security. Whitefield Diffie and Susan Landau concluded that we can best protect our communications through encryption in their book Privacy on the Line. Cryptography has long been a tool for securing communications and protecting privacy. The fundamental goal of cryptography is to achieve secure communication, and it has been observed that the privacy of ordinary people may be infringed upon in communications, which has led to questions being raised about the field of cryptography. The cryptography community has begun to focus on the social impact of its work. For instance, the Association for Computing Machinery (ACM) upholds detailed codes of ethics and professional conduct, including directives on honesty, privacy, and societal contribution [7]. The American Mathematical Society (AMS) and Mathematical Association of America (MAA) provide more generalized guidance on ethical conduct: The MAA requires Directors, Officers, Members, those compensated by the MAA and those donating their time, and all employees, to observe high standards of business and personal ethics in the conduct of their duties and responsibilities [8]. When mathematical work may affect the public health, safety or general welfare, it is the responsibility of mathematicians to disclose the implications of their work to their employers and to the public, if necessary [9]. Yet, the International Association for Cryptologic Research (IACR), despite its focus on cryptography, lacks a comprehensive ethical statement. Phillip Rogaway [10] emphasized the ethical responsibilities of cryptography work, not only to focus on technical and mathematical challenges but also to recognize the impact of their work in society, and to be driven by ethics to make more meaningful contributions to society. Designing cryptography schemes requires a reflective approach that navigates complex

ethical terrains, considering how cryptography tools and techniques affect social norms and values, and is capable of both protecting individual privacy and enabling surveillance.

The core element of ensuring the security of WBAN systems lies in the establishment of an efficient security framework. Within this framework, the two major security challenges of authentication and confidentiality are particularly crucial and require urgent solutions. In response to these challenges, encryption technology and digital signatures have been widely adopted as effective means to enhance security and verification mechanisms. In practice, when both encryption and signature functions are required concurrently, a common approach is to prioritize signature processing followed by encryption, in order to ensure the integrity and confidentiality of information. However, given the stringent constraints of low-power sensor devices in WBANs, such as limited onboard energy and central processing unit (CPU) processing capabilities, executing complex encryption programs appears impractical. To overcome this technical barrier, an innovative "signcryption" technology [11] has emerged, which ingeniously combines the functions of signing and encryption. This not only simplifies the operational process but also adapts to resource-constrained environments. Most importantly, compared to the traditional method of signing first and then encrypting, signcryption technology exhibits greater applicability in resource-constrained application scenarios such as WBANs due to its higher cost-effectiveness.

Currently, in response to the security challenges faced by WBANs, scholars have conducted extensive research from multiple angles and designed a series of signcryption schemes to tackle the security challenges faced by WBANs [12–16]. The core of these schemes lies in the establishment of three major cryptographic systems: the Public Key Cryptography (PKC), Identity-based Public Key Cryptography (ID-PKC), and Certificateless Public Key Cryptography (CL-PKC). During the process of system deployment, PKC often confronts intricate challenges related to certificate management. While ID-PKC can effectively bypass the difficulties of certificate management encountered in PKC, its drawback lies in the necessity of implementing a key escrow mechanism. Although a lightweight ID-PKC is highly suitable for resource-constrained WBANs, the security is compromised when the Private Key Generator (PKG) is compromised, as the PKG learns the private keys of all users. In other words, the PKG can decrypt ciphertexts in Identity-Based Encryption (IBE) schemes and can forge signatures for messages in Identity-Based Signature (IBS) schemes. Therefore, ID-PKC is only suitable for small-scale networks like WBANs, rather than large-scale networks such as the Internet. In this context, where the communication between Internet users and WBANs is being considered, CL-PKC emerges as an ideal choice compared to ID-PKC.

However, WBAN utilizes public communication channels, making the transmitted data highly vulnerable to eavesdropping, interception, replay, forgery, and tampering by adversaries. Therefore, it is very important to design an efficient and secure CLSC scheme to realize secure communication in WBAN. In order to achieve security, we must overcome a series of technical challenges [17–19]. The scope of these challenges covers a wide range of issues, including confidentiality, integrity, authentication, non-repudiation [5], anonymity, public verifiability, and public ciphertext authenticity. To tackle the aforementioned challenges, we present a certificateless anonymous signcryption (CL-ASC) scheme specifically for WBAN. Under the standard model, we have demonstrated that the scheme satisfies the requirements of anonymity of the signcrypter, and identity identifiability.

1.1. Related Work

The following related work can be focused on from two aspects: firstly, research regarding the CLSC Scheme itself; secondly, the application and exploration of the CLSC scheme within WBANs.

To eliminate key escrow in ID-PKC and simplify the certificate management in traditional PKC, Al-Riyami and Paterson [20] introduced the concept of CL-PKC. In CL-PKC, a user's complete private key comprises two parts: one is a partial private key generated by KGC, and the other is a secret value generated by the user themselves. Additionally, public

keys do not require certificates. Therefore, the certificateless public key cryptosystem boasts significant advantages and has garnered widespread attention since its inception [21–24]. In 2008, Barbosa and Farshim [23] combined the certificateless public key system with signcryption to introduce the Certificateless Signcryption (CLSC) scheme, while also defining the formal security concepts of CLSC schemes. The certificateless signcryption has the advantages of both the certificateless public key cryptographic system and signcryption. Building on this foundational work, numerous CLSC schemes have been proposed [25–32], but most of them have been proven secure in the ROM. It is well known that proofs in the ROM serve only as heuristic evidence and do not necessarily imply security in practical implementations [33]. Therefore, it is imperative to consider how to construct provably secure schemes without relying on random oracles. In 2010, Liu et al. [24] first proposed a certificateless signcryption scheme in the standard model; unfortunately, this model is insecure in the face of a malicious but passive Key Generation Center (KGC) and a public key substitution attack [34–36]. Subsequently, Jin et al. [37] adopted a new method to optimize and improve Liu's scheme and proved that their improved scheme is secure in the standard model. However, Xiong [38] demonstrated that Jin's scheme is not resistant to chosen ciphertext attacks and is vulnerable to malicious but passive KGC attacks. In 2017, Luo et al. [28] constructed a CLSC scheme and claimed to achieve unforgeability against adaptive chosen message attacks and ciphertext indistinguishability against adaptive chosen ciphertext attacks in the standard model. However, Yuan [39] pointed out that the scheme [28] failed to fulfil its purported security claims. Subsequently, Rastegari et al. [40] discovered a critical flaw in the scheme and proposed a revised CLSC scheme, but Lin [41] analyzed it and concluded that the scheme [40] was insecure. Therefore, how to propose a secure certificateless signcryption scheme under the standard model remains an open question.

There are two types of adversaries in certificateless cryptosystems. The Type I adversary, $\mathcal{A}_1$, mimics an "external" adversary who does not know the master secret key but can replace anyone's public key. The Type II adversary, $\mathcal{A}_2$, mimics an "internal" adversary who knows the master secret key but cannot replace anyone's public key. It should be noted that $\mathcal{A}_2$ only encompasses the "honest-but-curious" KGC, but a malicious and passive KGC may attempt to decrypt ciphertexts or forge signatures by embedding additional trapdoors in the public parameters [29]. Therefore, a stronger security model is needed to capture the operations of a malicious yet passive KGC. In 2007, Au et al. [42] introduced the concept of a malicious yet passive KGC as a Type II adversary. This type of attacker is malicious during the initial setup phase of the system, thereby allowing the Type II adversary to generate all public parameters and the master secret key. For adversary $\mathcal{A}_2$, a malicious yet passive $\mathcal{A}_2$ attack is more realistic and powerful than an honest-but-curious $\mathcal{A}_2$ attack. To resist attacks by a malicious but passive KGC and public key substitution attacks, we consider the malicious but passive KGC as a Type II adversary $\mathcal{A}_2$ in our security model and grant $\mathcal{A}_2$ the ability to replace public keys.

In 2016, Li et al. [43] debuted a CLSC scheme aimed at WBAN access control, claiming it met various security criteria, such as authentication, confidentiality, and non-repudiation, indicating its broad applicability. Unfortunately, the scheme was still vulnerable to replay attacks and lacked public verifiability [44]. In 2018, Li et al. [45] proposed a new CLSC scheme within an economical and anonymous access control mechanism for WBAN, claiming it encompassed security features like anonymity, confidentiality, authentication, integrity, and non-repudiation. However it is noteworthy that their security proofs were conducted in the ROM, and it lacked consideration for public verifiability and publicly ciphertext authenticity. In the same year, Lu et al. [46] developed a traceable threshold attribute signature scheme, aimed at providing better security for mobile healthcare social networks (MHSN). The article claims that the scheme has correctness, unforgeability, traceability, and privacy. However, the security proof of the scheme is also implemented in the ROM and lacks public verifiability and public ciphertext authenticity. In 2018, Liu [47] proposed a lightweight CLSC scheme based on RSA, and designed a lightweight and

efficient WBAN data access control scheme. The article claims that the scheme can meet more security requirements in WBAN. However, the scheme's security is only proven in the ROM. The existing certificateless signcryption-based data access control schemes have the following two weaknesses: (1) most of the security proofs are implemented in ROM. (2) most of the schemes lack anonymity, public verifiability, and publicly ciphertext authenticity. Subsequently, the use of our proposed CLSC scheme to design an efficient and secure WBAN data access control scheme can be considered.

1.2. Motivations and Contributions

Wireless Body Area Networks (WBANs) play a significant role in monitoring health information and creating efficient healthcare systems. The task of designing a secure and economical communication scheme suitable for WBANs is made particularly challenging due to the inherent characteristics of WBANs, such as the open medium channel and the limited resources of sensor nodes. Signcryption is an encryption technology that can simultaneously achieve the functions of public key encryption and digital signatures, which can authenticate users and protect query messages at the same time. It can achieve confidentiality, authentication, integrity, and non-repudiation at a low cost, which is suitable for WBANs. The CLSC schemes proposed in recent years have the following weakness:

- The security proofs of most schemes are implemented in ROM. However, the CLSC schemes with provable security in ROM may have vulnerabilities in practical applications.
- A Type II adversary in the security models of most schemes is considered a "honest but curious" KGC, but in reality, this may be a "malicious but passive" KGC.
- The schemes lack public verifiability and public ciphertext authenticity. This leads to the receiver having to decrypt the ciphertext first and then verify its validity. If the ciphertext is invalid, the decryption work will be wasted.
- Most schemes do not have anonymity of the signcrypter. This is not conducive to protecting the privacy of the sender.
- High computational cost. In order to complete a signcryption–unsigncryption algorithm, the scheme requires multiple pairing operations, which is not suitable for low-power devices.

Therefore, the purpose of this paper was to introduce a scheme that is both efficient and secure, addressing the aforementioned concerns. The contributions of this paper are as follows:

- We introduce a CL-ASC scheme which is suitable for WBAN, with anonymity of the signcrypter, public verifiability, public ciphertext authenticity, and identifiable identity. There are very few CL-ASC schemes that have all these special features. Compared to other schemes, our scheme has very powerful functions and shows some degree of innovation.
- We provided a stronger security model for the CL-ASC scheme. Our security model considers a malicious but passive KGC as a Type II adversary, which can generate all public parameters and the master secret key during the initial system setup stage, and is endowed with stronger capabilities. In addition, both Type I and Type II adversaries can directly compute hash functions to obtain results. This significantly enhances the capabilities of the adversaries, making the scheme more secure and more aligned with real-world scenarios.
- We demonstrate that our scheme possesses indistinguishability, unforgeability, anonymity of the signcrypter, and identity identifiability in the standard model.
- Compared to the related schemes, our scheme offers superior security performance, along with reduced computational overheads and storage costs, and offers better security, making it more suitable for WBAN.

1.3. Organization

The subsequent sections are structured as follows: in Section 2, we introduce the fundamental concepts; Section 3 elaborates on the system's architecture; Section 4 specifies

the security framework; the proposed scheme is thoroughly described in Section 5; and its security analysis is presented in Section 6. A comparative analysis of performance is presented in Section 7; and Section 8 summarizes the conclusions of our study.

2. Preliminaries

The structure is distinguished by the presence of an additive cyclic group G_1 and a multiplicative cyclic group G_2, each possessing an order of q, with q being a prime number. A bilinear map, denoted by $e : G_1 \times G_1 \to G_2$, is defined by the following properties:

Non-degeneracy: There exist $P, Q \in G_1$ such that $e(P, Q) \neq 1_{G_1}$.

Computability: An efficient computational method exists for determining $e(P, Q)$ for any given P and Q from their respective groups.

Bilinearity: For every pair of elements $P, Q \in G_1$ and integers $a, b \in Z_q$, the map satisfies $e(aP, bQ) = e(P, Q)^{ab}$.

This mapping is referred to as bilinear, as described in [48].

The mathematical problems and assumptions about bilinear mapping used in this paper are as follows:

Definition 1. *Decisional Diffie–Hellman Problem (DDHP): When presented with elements $P, aP, bP, X \in G$, verify if X is indeed abP. Here, $P \in G_1$ and $a, b \in Z_q^*$.*

Definition 2. *Decisional Diffie–Hellman Assumption (DDHA): Under the DDHA, it is assumed that the likelihood of any algorithm capable of operating within polynomial time successfully resolving the DDHP is minimal.*

Definition 3. *Computational Attack Algorithm Problem(CAAP) [31]: Given a tuple (P, aP) for $a \in Z_q^*, P \in G_1$, output a tuple $(c, \frac{1}{a+c}P)$.*

Definition 4. *Computational Attack Algorithm Assumption (CAAA): Under the CAAA, it is assumed that the likelihood of any algorithm capable of running in polynomial time successfully resolving the CAAP is minimal.*

3. System Model

The fundamental security prerequisites for the deployment of a signcryption scheme within WBAN are outlined as follows:

(1) Confidentiality: this means that any unauthorized party, other than the authorized individual or entity, cannot access the data content. Even if an unauthorized user obtains the encrypted data, they cannot decipher the true content of the data.

(2) Authentication: this refers to the authentication of data sources or entities.

(3) Integrity: guaranteeing the integrity of data transmitted in the network, preventing illegal entities from tampering with or deleting query messages.

(4) Non-repudiation: ensuring that the sender of data cannot deny previous commitments or actions.

(5) Unforgeability: if the attacker can forge the patient's signature, the doctor will face obstacles in diagnosis and treatment, which may endanger the patient's life. Therefore, we need the signcryption scheme to be unforgeable under the adaptive chosen message attack.

(6) Anonymity of the signcryptor: in order to protect user privacy, no other entity apart from KGC can indeed ascertain the true identity of the signcryptor.

(7) Identity identifiability: while ensuring user privacy, KGC is capable of verifying and tracing the identity of the signcryptor to ensure the security and credibility of data transmission and usage. Meanwhile, other unauthorized entities are prevented from accessing this sensitive information.

(8) Public verifiability [18]: a third party is registered to affirm the legitimacy of the encrypted message, independent of the access to the sender's private key.

(9) Public ciphertext authenticity [18]: a third party can confirm the authenticity of the ciphertext without the need for decryption, allowing the receiver to discard invalid ciphertexts in advance, saving energy consumption and computation time, which is crucial for small devices.

The system model proposed in this paper is shown in Figure 2, which includes three entities: KGC, sender C, receiver U.

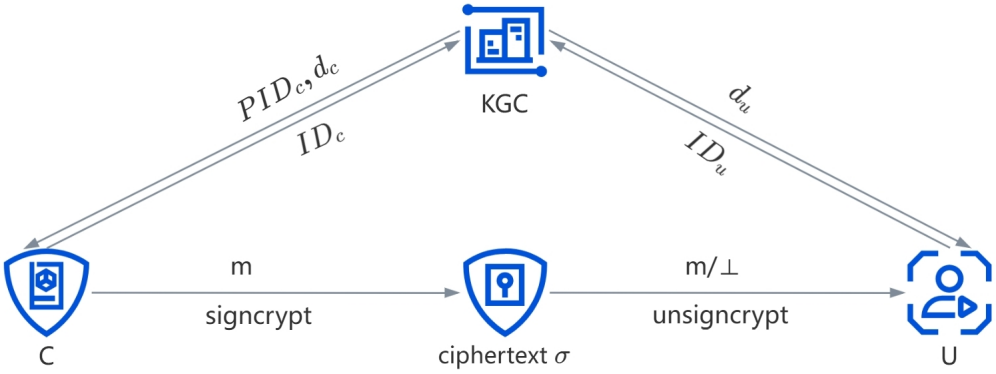

Figure 2. Schematic of system model.

- KGC: Responsible for setting system parameters and publishing them publicly. Additionally, it is also responsible for generating pseudo-identities for sender C and generating partial private keys for both sender C and receiver U.
- C: Uses their own private key to perform signcryption on the data m, generates the ciphertext σ of m, and sends the ciphertext σ to B.
- U: Decrypts the ciphertext Upon receiving it, using their own private key to obtain the data m.

The CL-ASC scheme consists of eight distinct algorithms, each of which is delineated as follows:

- Setup(μ): Input parameter μ for security; KGC generates the system parameters $params$ and master secret key msk. Then KGC has the public $params$, and secretly holds msk.
- PIDG(ID_c, $params$): Input the real identity ID_c of C; KGC generates a pseudo-identity PID_c of C, and sends it to C.
- PPKG(ID_u/PID_c, $params$, msk): Upon receiving the identity ID_u of U (or the pseudo-identity PID_c of C), KGC generates the partial private key d_u (or d_c) of U (of C) and transmits it securely to U (or C).
- SVS(ID_u/PID_c, $params$): U (or C) sets x_u (or x_c) as its secret value.
- FSKS(ID_u/PID_c, $params$, x_u/x_c, d_u/d_c): U (or C) sets its full private key SK_u (or SK_c) as $SK_u = (x_u, d_u)$ (or $SK_c = (x_c, d_c)$).
- UPKG(ID_u/PID_c, $params$, X_u/X_c, R_u/R_c): U (or C) sets its public key as $PK_u = (X_u, R_u)$ (or $PK_c = (X_c, R_c)$).
- Signcrypt($m, ID_u, PK_u, PID_c, SK_c, params$): Takes $params$, message m, U's identity ID_u, U's public key PK_u, C's pseudo-identity PID_c and C's full private key SK_c as input; C returns the ciphertext σ and transmits it to U.
- Unsigncrypt($\sigma, ID_u, SK_u, PID_c, PK_c, params$): Takes $params$, ciphertext σ, U's identity ID_u, U's full private key SK_u, C's pseudo-identity PID_c and C's public key PK_c as input; U returns the corresponding plaintext m or $\perp$.

4. Security Model

Based on the security models proposed by Barbosa et al. [23], Zhou et al. [30] and Deng [49], we present a security model for CL-ASC, and give the following explanations. Against a Type I adversary $\mathcal{A}_1$, we have adopted the original security model proposed by Barbosa and Farshim, based on its notable advantage over another security model,

namely that in the latter, in the public key replacement oracle, $\mathcal{A}_1$ needs to provide the corresponding secret value when replacing the user's public key. Barbosa and Farshim's model demonstrates greater defensive capabilities. For a Type II adversary $\mathcal{A}_2$, our security model takes into account a malicious yet passive KGC as $\mathcal{A}_2$. At this time, $\mathcal{A}_2$ can generate all public parameters and the master key during the initialization phase of the system, given that in practical scenarios, Type 2 adversaries also possess the capability to perform public key replacement attacks. Therefore, we allow $\mathcal{A}_2$ to execute the public key replacement query in our security model, ensuring that the security model effectively defends against such threats. Furthermore, both Type I and Type II adversaries are capable of directly computing hash functions to obtain results.

Based on the above analysis, against indistinguishability under an adaptive chosen ciphertext attack and unforgeability under an adaptive chosen message attack, we present two types of adversaries.

$\mathcal{A}_1$: $\mathcal{A}_1$ is a dishonest user who can replace the public key of any entity with a value of their own choice, but they do not have access to the secret master key.

$\mathcal{A}_2$: $\mathcal{A}_2$ represents a malicious but passive KGC that generates all public parameters and master secret key and can perform public key replacement.

In addition, We introduce a super adversary, $\mathcal{A}$, specifically targeting the anonymity of the signcryptor. $\mathcal{A}$ is a super adversary who possesses the capabilities of both $\mathcal{A}_1$ and $\mathcal{A}_2$, meaning that $\mathcal{A}$ is endowed with the capacity to replace users' public keys and also has access to the master secret key, and can perform secret value queries. However, $\mathcal{A}$ is unable to access the list FI and cannot query the pseudo-identity of the target user.

Definition 5. *If the adversary cannot win the following game with a non-negligible probability in any polynomial time, then the security property of the CLSC scheme is said to satisfy indistinguishability under an adaptive chosen ciphertext attack ($IND-CCA_2$).*

Game 1: The game between the adversary $\mathcal{A}_1$ and the challenger $\mathcal{B}$ unfolds as follows:

- **Initialization phase**: $\mathcal{B}$ obtains *msk* and *params* by executing the setup algorithm, then sends *params* to $\mathcal{A}_1$ and maintains the secrecy of *msk*.
- **Query phase**: For C, the queries $Q_{pid}(ID_c)$ and $Q_{upk}(PID_c)$ are executed before any other queries. For U, the query $Q_{upk}(ID_u)$ should be executed before any other queries. $\mathcal{A}_1$ performs the following types of queries:
  $Q_{pid}(ID_c)$: $\mathcal{A}_1$ sends a user identity ID_c to $\mathcal{B}$, $\mathcal{B}$ returns the pseudo-identity PID_c to $\mathcal{A}_1$.
  $Q_{upk}(PID_c/ID_u)$: $\mathcal{A}_1$ sends a user identity PID_c/ID_u to $\mathcal{B}$, $\mathcal{B}$ returns the corresponding public key PK_c/PK_u to $\mathcal{A}_1$.
  $R_{upk}(PID_c, PK_c')/(ID_u, PK_u')$: $\mathcal{A}_1$ sends a tuple $(PID_c, PK_c')/(ID_u, PK_u')$ to $\mathcal{B}$, $\mathcal{B}$ replaces PK_c/PK_u with PK_c'/PK_u'.
  $Q_{ppk}(PID_c/ID_u)$: $\mathcal{A}_1$ sends a user identity PID_c/ID_u to $\mathcal{B}$, $\mathcal{B}$ returns the partial private key d_c/d_u to $\mathcal{A}_1$. When R_c/R_u is replaced, $\mathcal{A}_1$ cannot perform this query. The reason for imposing this restriction is that it is unreasonable to expect the challenger to provide a partial private key for users who do not know a partial private key.
  $Q_{sv}(PID_c/ID_u)$: $\mathcal{A}_1$ sends a user identity PID_c/ID_u to $\mathcal{B}$, $\mathcal{B}$ returns the secret value x_c/x_u to $\mathcal{A}_1$. When X_c/X_u is replaced, $\mathcal{A}_1$ cannot perform this query. The reason for imposing this restriction is that it is unreasonable to expect the challenger to provide a secret value for users who do not know a secret value.
  $Q_{sc}(m, ID_u, PK_u, PID_c, PK_c)$: $\mathcal{A}_1$ sends tuple $(m, ID_u, PK_u, PID_c, PK_c)$ to $\mathcal{B}$, where m is the plaintext intended for signcryption, ID_u is the identity of U, PK_u is the public key of U whose identity is ID_u, PID_c is the identity of C and PK_c is the public key of C whose identity is PID_c. $\mathcal{B}$ first executes the PPKG algorithm, SVS algorithm and FSKS algorithm using identity PID_c to obtain SK_c, and then executes the signcrypt algorithm using the tuple $(\sigma, ID_u, PK_u, PID_c, SK_c)$ to output the ciphertext σ as the reply to $\mathcal{A}_1$'s query. When the PID_c's public key is replaced, $\mathcal{B}$ may not be able to

access the full private key of PID_c. In this case, the $\mathcal{A}_1$ needs to provide the relevant information of the PID_c.
$Q_{un}(\sigma, ID_u, PK_u, PID_c, PK_c)$: $\mathcal{A}_1$ sends tuple $(\sigma, ID_u, PK_u, PID_c, PK_c)$ to $\mathcal{B}$, where σ is the ciphertext intended for unsigncryption, ID_u is the identity of U, PK_u is the public key of U whose identity is ID_u, PID_c is the identity of C and PK_c is the public key of C whose identity is PID_c. $\mathcal{B}$ first executes the PPKG algorithm, SVS algorithm and FSKS algorithm using identity ID_u to obtain SK_u, and then executes the unsigncrypt algorithm using the tuple $(\sigma, ID_u, SK_u, PID_c, PK_c)$ to obtain the plaintext m or $\perp$ as the reply to $\mathcal{A}_1$'s query. When the ID_u's public key is replaced, $\mathcal{B}$ may not be able to access the full private key of ID_u. In this case, the $\mathcal{A}_1$ needs to provide the relevant information of the ID_u.

- **Challenge phase**: $\mathcal{A}_1$ selects two distinct messages m_0, m_1 of the same length and subsequently transmits the tuple $(m_0, m_1, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ to $\mathcal{B}$, where ID_u^* is the identity of U, PK_u^* is the public key of U whose identity is ID_u^*, PID_c^* is the identity of C and PK_c^* is the public key of C whose identity is PID_c^*. $\mathcal{B}$ randomly chooses a bit $\xi \in \{0,1\}$ and executes the signcryption algorithm using the tuple $(m_\xi, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ to obtain the ciphertext σ^* of m_ξ. Then, $\mathcal{B}$ sends $\sigma^*$to $\mathcal{A}_1$. In this process, $\mathcal{A}_1$ must meet the following conditions:
  (1) ID_u^* is an identity whose partial private key has not been queried by $\mathcal{A}_1$.
  (2) $\mathcal{A}_1$ cannot replace the value of R_u^*.
- **Guess phase**: After receiving σ^*, $\mathcal{A}_1$ performs a series of queries, but there are the following constraints:
  (1) $\mathcal{A}_1$ is not allowed to operate $Q_{ppk}(ID_u^*)$.
  (2) $\mathcal{A}_1$ is not allowed to replace the value of R_u^*.
  (3) $\mathcal{A}_1$ is not allowed to operate $Q_{un}(\sigma^*, ID_u^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$.
  $\mathcal{A}_1$ guesses ξ'. If $\xi' = \xi$, $\mathcal{A}_1$ wins Game 1. The advantage of $\mathcal{A}_1$ is defined as follows:

$$Adv_{\mathcal{A}_1}^{IND-CCA_2} = |P_r[\xi' = \xi] - \tfrac{1}{2}|.$$

Game 2: The game between the adversary $\mathcal{A}_2$ and the challenger $\mathcal{B}$ unfolds as follows:

- **Initialization phase**: $\mathcal{A}_2$ obtains *msk* and *params* by executing the setup algorithm, then sends them to $\mathcal{B}$.
- **Query phase**: $\mathcal{A}_2$ performs various queries similar to Game 1.
- **Challenge phase**: $\mathcal{A}_2$ selects two distinct messages m_0, m_1 of the same length and subsequently transmits the tuple $(m_0, m_1, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ to $\mathcal{B}$, where ID_u^* is the identity of U, PK_u^* is the public key of U whose identity is ID_u^*, PID_c^* is the identity of C and PK_c^* is the public key of C whose identity is PID_c^*. $\mathcal{B}$ randomly chooses a bit $\xi \in \{0,1\}$ and executes the signcryption algorithm using the tuple $(m_\xi, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ to obtain the ciphertext σ^* of m_ξ. Then, $\mathcal{B}$ sends $\sigma^*$to $\mathcal{A}_2$. In this process, $\mathcal{A}_2$ must meet the following conditions:
  (1) ID_u^* is an identity whose secret value has not been queried by $\mathcal{A}_2$.
  (2) $\mathcal{A}_2$ is not allowed to replace the value of X_u^*.
- **Guess phase**: After receiving σ^*, $\mathcal{A}_2$ performs a series of queries, but there are the following constraints:
  (1) $\mathcal{A}_2$ is not allowed to operate $Q_{sv}(ID_u^*)$.
  (2) $\mathcal{A}_2$ is not allowed to replace the value of X_u^*.
  (3) $\mathcal{A}_2$ is not allowed to to operate $Q_{un}(\sigma^*, ID_u^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$.
  $\mathcal{A}_2$ guesses ξ'. If $\xi' = \xi$, $\mathcal{A}_2$ wins Game 2. The advantage of $\mathcal{A}_2$ is defined as follows:

$$Adv_{\mathcal{A}_2}^{IND-CCA_2} = |P_r[\xi' = \xi] - \tfrac{1}{2}|.$$

Definition 6. *If the adversary cannot win the following game with a non-negligible probability in any polynomial time, then the security property of the CLSC scheme is said to satisfy unforgeability under an adaptive chosen message attack (UF-CMA).*

Game 3: The game between the adversary $\mathcal{A}_1$ and the challenger $\mathcal{B}$ unfolds as follows:

- **Initialization phase:** Same as the initialization phase in Game 1.
- **Query phase:** $\mathcal{A}_1$ performs various queries similar to Game 1.
- **Forgery phase:** $\mathcal{A}_1$ outputs a new tuple $(\sigma^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, where σ^* is a ciphertext, ID_u^* is the identity of U, PK_u^* is the public key of U whose identity is ID_u^*, PID_c^* is the identity of C and PK_c^* is the public key of C whose identity is PID_c^*. $\mathcal{A}_1$ wins Game 3 if the subsequent conditions are met:
  (1) In the process of running the unsigncryption algorithm with the tuple $(\sigma^*, ID_u^*, SK_u^*, PID_c^*, PK_c^*)$, $\mathcal{B}$ does not output $\perp$.
  (2) $\mathcal{A}_1$ was not allowed to operate $Q_{ppk}(PID_c^*)$.
  (3) $\mathcal{A}_1$ was not allowed to replace the value of R_c^*.
  (4) $\mathcal{A}_1$ was not allowed to acquire σ^* through running $Q_{sc}(m^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, where m^* represents the plaintext that corresponds to σ^*.
  The advantage of $\mathcal{A}_1$ is defined as follows:

$$Adv_{\mathcal{A}_1}^{UF-CMA} = |P_r[A_1 wins]|.$$

Game 4: The game between the adversary $\mathcal{A}_2$ and the challenger $\mathcal{B}$ unfolds as follows:

- **Initialization phase:** Same as the initialization phase in Game 2.
- **Query phase:** $\mathcal{A}_2$ performs various queries similar to Game 1.
- **Forgery phase:** $\mathcal{A}_2$ outputs a new tuple $(\sigma^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, where σ^* is a ciphertext, ID_u^* is the identity of U, PK_u^* is the public key of U whose identity is ID_u^*, PID_c^* is the identity of C and PK_c^* is the public key of C whose identity is PID_c^*. $\mathcal{A}_2$ wins Game 4 if the subsequent conditions are met:
  (1) In the process of running the unsigncryption algorithm with the tuple $(\sigma^*, ID_u^*, SK_u^*, PID_c^*, PK_c^*)$, $\mathcal{B}$ does not output $\perp$.
  (2) $\mathcal{A}_2$ was not allowed to operate $Q_{sv}(PID_c^*)$.
  (3) $\mathcal{A}_2$ was not allowed to replace the value of X_c^*.
  (4) $\mathcal{A}_2$ was not allowed to acquire σ^* through running $Q_{sc}(m^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, where m^* represents the plaintext that corresponds to σ^*.
  The advantage of $\mathcal{A}_2$ is defined as follows:

$$Adv_{\mathcal{A}_2}^{UF-CMA} = |P_r[A_2 wins]|.$$

Definition 7. *If the adversary cannot win the following game with a non-negligible probability in any polynomial time, then the CLSC scheme is said to be anonymous to the signcrypter.*

Game 5: The game between the super adversary $\mathcal{A}$ and the challenger $\mathcal{B}$ unfolds as follows:

- **Initialization phase:** Same as the initialization phase in Game 2.
- **Query phase:**$\mathcal{A}$ inputs various queries, and $\mathcal{B}$ executes the corresponding algorithm to output the answer.
- **Challenge phase:** $\mathcal{A}$ selects a message m^* and two distinct real identities ID_0^* and ID_1^* of C, where $\mathcal{A}$ has not performed Q_{pid} for ID_0^* and ID_1^*. $\mathcal{A}$ subsequently sends tuple $(m^*, ID_0^*, ID_1^*, ID_u^*, PK_u^*)$ to $\mathcal{B}$, where ID_u^* is the identity of U, PK_u^* is the public key of U whose identity is ID_u^*. $\mathcal{B}$ performs the subsequent steps:
  (1) Randomly selects $\xi \in \{0,1\}$ and invokes the PIDG algorithm with ID_ξ^* to acquire PID_ξ^*.
  (2) Invokes the PPKG algorithm, SVS algorithm and FSKS algorithm with PID_ξ^* to acquire the full private key SK_ξ^* of PID_ξ^*.
  (3) Acquires the ciphertext σ^* by running the signcryption algorithm with the tuple $(m^*, ID_u^*, PK_u^*, PID_\xi^*, SK_\xi^*)$.
  (4) Outputs the tuple $(\sigma^*, ID_u^*, PK_u^*, PID_\xi^*, PK_\xi^*)$ to $\mathcal{A}$.
- **Guess phase:** $\mathcal{A}$ can make a series of queries, but cannot perform Q_{pid} for ID_1^* and ID_0^*. $\mathcal{A}$ makes a guess ξ'. If $\xi' = \xi$, $\mathcal{A}$ will win Game 5.

The advantage of $\mathcal{A}$ is as follows:

$$Adv_A^{ANO-CLSC} = |P_r[\xi' = \xi] - 1|.$$

Definition 8. *If KGC can recognize the true identity of C in any ciphertext, then the CLSC scheme is identifiable.*

5. New Scheme

- Setup: Given a security parameter μ, SP performs the subsequent steps:
  (1) Sets up a bilinear mapping $e : G_1 \times G_1 \to G_2$, where G_1 is an additive cyclic group, G_2 is a multiplicative cyclic group and $|G_1| = |G_2| = q(q > 2^\mu)$.
  (2) Selects a generator P of G_1, and computes $N = e(P, P)$.
  (3) Sets an identity space $\Omega = \{0,1\}^{l_1}$ and a message space $M = \{0,1\}^{l_2}$.
  (4) Selects the following secure hash functions (where $G_1^2 = G_1 \times G_1$).
  H_1: $G_1 \times G_1 \times G_1 \to \{0,1\}^{l_1}$;
  H_2: $\{0,1\}^{l_1} \times G_1 \to Z_q^*$;
  H_3: $G_1 \times \{0,1\}^{l_1} \times G_1^2 \times \{0,1\}^{l_1} \times G_1^2 \to Z_q^*$;
  H_4: $\{0,1\}^{l_2} \times G_1 \times \{0,1\}^{l_1} \times G_1^2 \times \{0,1\}^{l_1} \times G_1^2 \to Z_q^*$;
  H_5: $G_1 \times G_1 \times \{0,1\}^{l_1} \times G_1^2 \times \{0,1\}^{l_1} \times G_1^2 \to Z_q^*$.
  (5) Randomly chooses a number $\delta \in Z_q^*$ and computes $P_{pub} = \delta P$; let master secret key $msk = \{\delta\}$.
  (6) Publishes the params $params = \{G_1, G_2, q, e, P, P_{pub}, N, H_1 \sim H_5\}$.
- PIDG: KGC sets up a list FI, which contains the tuple $(PID_c, e_c, E_c, f_c, F_c)$. Upon receiving an actual identity $ID_c \in \Omega$, KGC performs the subsequent steps:
  (1) Randomly chooses $e_c, f_c \in_R Z_q^*$, and calculates $E_c = e_c P$, $F_c = f_c P$.
  (2) Computes $\triangle C = e_c f_c \delta P$, $PID_c = ID_c \oplus H_1(\triangle C, E_c, F_c)$.
  (3) Sends the pseudo-identity PID_c to C.
  (4) Adds the tuple $(PID_c, e_c, E_c, f_c, F_c)$ to the list FI.
- PPKG:
  (1) After receiving the pseudo-identity PID_c of C, KGC performs the subsequent steps:
    (a) Randomly chooses $r_c \in Z_q^*$, and calculates $R_c = r_c P$.
    (b) Calculates $l_c = H_2(PID_c, R_c), d_c = r_c + l_c \delta$.
    (c) Sends (R_c, d_c) to C via a secure channel.
  (2) After receiving the identity ID_u of U, KGC performs the subsequent steps:
    (a) Randomly chooses $r_u \in Z_q^*$, and calculates $R_u = r_u P$.
    (b) Calculates $l_u = H_2(ID_u, R_u), d_u = r_u + l_u \delta$.
    (c) Sends (R_u, d_u) to U via a secure channel.
  (3) C can confirm d_c's validity by verifying whether the equation $d_c P = R_c + l_c P_{pub}$ holds. If the equation holds, then the partial private key is valid. Otherwise, the partial private key is invalid.
  (4) U can confirm d_u's validity by verifying whether the equation $d_u P = R_u + l_u P_{pub}$ holds. If the equation holds, then the partial private key is valid. Otherwise, the partial private key is invalid.
- SVS:
  (1) C randomly chooses $x_c \in Z_q^*$, sets x_c as its secret value.
  (2) U randomly chooses $x_u \in Z_q^*$, sets x_u as its secret value.
- FSKS:
  (1) C sets the full private key $SK_c = (x_c, d_c)$.
  (2) U sets the full private key $SK_u = (x_u, d_u)$.
- UPKG:
  (1) C computes $X_c = x_c P$, and sets the public key $PK_c = (X_c, R_c)$.
  (2) U computes $X_u = x_u P$, and sets the public key $PK_u = (X_u, R_u)$.
- Signcrypt: Upon receiving a plaintext message $m \in M$, C performs the subsequent steps:
  (1) Calculates $l_u = H_2(ID_u, R_u)$.

(2) Randomly chooses $k \in Z_q^*$, and calculates $K = kP$.
(3) Calculates $\lambda = H_3(K, ID_u, PK_u, PID_c, PK_c)$.
(4) Calculates $\tau = k(R_u + l_u P_{pub} + \lambda X_u)$.
(5) Calculates $\theta = H_5(K, \tau, ID_u, PK_u, PID_c, PK_c) \oplus m$.
(6) Calculates $\rho = H_4(\theta, K, ID_u, PK_u, PID_c, PK_c)$.
(7) Calculates $\omega = \frac{1}{d_c + \rho x_c} P$.
(8) Generates $\sigma = (\theta, K, \omega)$ as the ciphertext.
(9) Transmits σ to U.

- Unsigncrypt: Upon receiving the tuple $\sigma = (\theta, K, \omega)$, U performs the subsequent steps:
(1) Calculates $l_c = H_2(PID_c, R_c)$.
(2) Calculates $\rho = H_4(\theta, K, ID_u, PK_u, PID_c, PK_c)$.
(3) Verifies whether the equation $e(\omega, R_c + l_c P_{pub} + \rho X_c) = N$ holds. If the equation is valid, proceed to step 4. Otherwise, the signature is invalid; output $\perp$.
(4) Calculates $\lambda = H_3(K, ID_u, PK_u, PID_c, PK_c)$.
(5) Calculates $\tau = (d_u + \lambda x_u)K$.
(6) Calculates $m = H_5(K, \tau, ID_u, PK_u, PID_c, PK_c) \oplus \theta$.

Correctness:

$$\begin{aligned}\tau &= (d_u + \lambda x_u)K \\ &= (d_u + \lambda x_u)kP \\ &= k(d_u + \lambda x_u)P \\ &= k(R_u + l_u P_{pub} + \lambda X_u)\end{aligned}$$

$$\begin{aligned}e(\omega, R_c + l_c P_{pub} + \rho X_c) &= e(\frac{1}{d_c + \rho x_c} P, r_c P + l_c \delta P + \rho x_c P) \\ &= e(\frac{1}{r_c + l_c \delta + \rho x_c} P, (r_c + l_c \delta + \rho x_c)P) \\ &= e(\frac{1}{r_c + l_c \delta + \rho x_c} P, (r_c + l_c \delta + \rho x_c)P) \\ &= e(P, P) \\ &= N\end{aligned}$$

Additionally, our scheme offers public verifiability and public ciphertext authenticity. During the initial three steps of Unsigncrypt algorithm, any third party can ascertain the legitimacy of the ciphertext σ without needing C's full private key or the message m. If the ciphertext σ is proven invalid, the receiver can immediately disregard it, thus avoiding further decryption steps. This method conserves computational resources and reduces energy consumption, which is particularly advantageous for small-scale devices by saving both energy and processing time.

From an ethical perspective, we have conducted an analysis of the ethical risks associated with the proposed scheme and its security model. This analytical framework primarily encompasses three core aspects: technical ethics, individual ethics, and social ethics. In terms of technical ethics, we have provided a more robust security model for the CL-ASC scheme. Our security model considers a malicious yet passive KGC as a Type II adversary and allows for such adversaries to replace public keys. Both Type I and Type II adversaries are capable of directly computing hash functions to obtain results. This significantly enhances the adversaries' capabilities, thereby making the scheme more secure. Under our enhanced security model, we will demonstrate that the CL-ASC scheme possesses indistinguishability and unforgeability. Consequently, applying our CL-ASC scheme for communication in WBNA will not result in message leakage. Furthermore, our CL-ASC scheme ensures the anonymity of the signcrypter, effectively safeguarding users' privacy. In individual ethics, the ciphertext of signcryption is encrypted with the sender's private key and the recipient's public key. To unsigncrypt, the recipient's private

key and the sender's public key are required. This ensures that even if a participant is subjected to malicious attacks during data transmission, the transmitted data will not be leaked, thus avoiding the risk of individual ethics. In terms of social ethics: encryption measures are taken for users' private data during the communication process. When strictly implemented, our scheme can maximize the prevention of data leakage during transmission.

6. Security of the Scheme

In the security proofs below, the adversary is capable of directly computing the values of the hash function without necessitating a query to the challenger.

Lemma 1. *If the DDH problem is hard, our scheme is proven to be* $IND - CCA_2$ *against the adversary* $\mathcal{A}_1$ *in the SM.*

Proof. Given the tuple $(P, \alpha P, \beta P, T)$, where $\alpha, \beta \in Z_q^*$ and α, β are unknown. The goal of $\mathcal{B}$ is to determine whether T is equal to $\alpha\beta P$.

Initialization phase: $\mathcal{B}$ obtains *msk* and $params = \{G_1, G_2, q, e, P, P_{pub} = \delta P, N = e(P, P), H_1 \sim H_5\}$ by executing the setup algorithm, then sends *params* to $\mathcal{A}_1$ and maintains the secrecy of *msk*. After the process above, $\mathcal{A}_1$ and $\mathcal{B}$ are both unaware of α and β, but $\mathcal{B}$ is aware of δ, while $\mathcal{A}_1$ is not.

Query phase: $\mathcal{B}$ sets $ID^{\diamondsuit}$ as the challenge target identity. For C, $\mathcal{A}_1$ must first execute $Q_{pid}(ID_c)$ and $Q_{upk}(PID_c)$ before any other queries. For U, $\mathcal{A}_1$ must first execute $Q_{upk}(ID_u)$ before any other queries. There are eight empty tables, L_{UC}, L_{UU}, L_{RC}, L_{RU}, L_{KC}, L_{KU}, L_{VC} and L_{VU}, maintained by $\mathcal{B}$. $\mathcal{A}_1$ can conduct the following types of queries, and $\mathcal{B}$ simulates $\mathcal{A}_1$'s queries as follows:

$Q_{pid}(ID_c)$: When $\mathcal{A}_1$ provides an identity ID_c for a query, $\mathcal{B}$ executes the PIDG algorithm to output the PID_c as $\mathcal{A}_1$'s response.

$Q_{upk}(PID_c)$: $\mathcal{B}$ maintains a list L_{UC}, which contains the tuple $(PID_c, X_c, x_c, R_c, r_c)$. When $\mathcal{A}_1$ provides an identity PID_c for a query, if the PID_c is on the the list L_{UC}, $\mathcal{B}$ returns PK_c as $\mathcal{A}_1$'s response. Otherwise, PID_c is queried as a new identity, $\mathcal{B}$ randomly chooses $x_c, r_c \in Z_q^*$, sets $PK_c = (x_cP, r_cP)$ as $\mathcal{A}_1$'s response, and adds $(PID_c, x_cP, x_c, r_cP, r_c)$ to the list L_{UC}.

$Q_{upk}(ID_u)$: $\mathcal{B}$ maintains a list L_{UU}, which contains the tuple $(ID_u, X_u, x_u, R_u, r_u)$. When $\mathcal{A}_1$ provides an identity ID_u for a query, if the ID_u is on the the list L_{UU}, $\mathcal{B}$ returns PK_u as $\mathcal{A}_1$'s response. Otherwise, ID_u is queried as a new identity, and $\mathcal{B}$ performs the subsequent steps:

(1) If $ID_u = ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $x^{\diamondsuit} \in Z_q^*$, sets $PK_u = PK^{\diamondsuit} = (x^{\diamondsuit}P, \alpha P)$ as $\mathcal{A}_1$'s response, and adds $(ID_u, x^{\diamondsuit}P, x^{\diamondsuit}, \alpha P, \nabla)$ to the list L_{UU}(where ∇ represents a null value).

(2) If $ID_u \neq ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $x_u, r_u \in Z_q^*$, computes $PK_u = (x_uP, r_uP)$ as $\mathcal{A}_1$'s response, and adds $(ID_u, x_uP, x_u, r_uP, r_u)$ to the list L_{UU}.

$R_{upk}(PID_c, PK_c, PK_c')$: $\mathcal{B}$ maintains a list L_{RC}, which contains the tuple (PID_c, PK_c, PK_c'). When $\mathcal{A}_1$ requests to replace the PID_C's public key PK_C with PK_c', $\mathcal{B}$ updates PK_c to PK_c', and adds (PID_c, PK_c, PK_c') to the list L_{RC}.

$R_{upk}(ID_u, PK_u, PK_u')$: $\mathcal{B}$ maintains a list L_{RU}, which contains the tuple (ID_u, PK_u, PK_u'). When $\mathcal{A}_1$ requests to replace the ID_u's public key PK_u with PK_u', $\mathcal{B}$ updates PK_u to PK_u', and adds (ID_u, PK_u, PK_u') to the list L_{RU}.

$Q_{ppk}(PID_c)$: $\mathcal{B}$ maintains a list L_{KC}, which contains the tuple (PID_c, d_c). When $\mathcal{A}_1$ provides an identity PID_c for a query, $\mathcal{B}$ searches for $(PID_c, x_cP, x_c, r_cP, r_c)$ in the list L_{UC}, executes the PPKG algorithm, and outputs d_c as $\mathcal{A}_1$'s response, then adds (PID_c, d_c) to the list L_{KC}.

$Q_{ppk}(ID_u)$: $\mathcal{B}$ maintains a list L_{KU}, which contains the tuple (ID_u, d_u). When $\mathcal{A}_1$ provides an identity PID_c for a query, $\mathcal{B}$ performs the subsequent steps:

(1) If $ID_u = ID^{\diamondsuit}$, then $\mathcal{B}$ fails and terminates the process.

(2) If $ID_u \neq ID^{\diamondsuit}$, $\mathcal{B}$ searches for $(ID_u, x_uP, x_u, r_uP, r_u)$ in the list L_{UU}, executes the PPKG algorithm to output d_u as $\mathcal{A}_1$'s response, and then adds (ID_u, d_u) to the list L_{KU}.

$Q_{sv}(PID_c)$: $\mathcal{B}$ maintains a list L_{VC}, which contains the tuple (PID_c, x_c). When $\mathcal{A}_1$ provides an identity PID_c for a query, $\mathcal{B}$ searches for $(PID_c, x_cP, x_c, r_cP, r_c)$ in the list L_{UC}, outputs x_c as $\mathcal{A}_1$'s response, and then adds (PID_c, x_c) to the list L_{VC}.

$Q_{sv}(ID_u)$: $\mathcal{B}$ maintains a list L_{VU}, which contains the tuple (ID_u, x_u). When $\mathcal{A}_1$ provides an identity ID_u for a query, $\mathcal{B}$ searches for $(ID_u, x_uP, x_u, r_uP, r_u)$ in the list L_{UU}, outputs x_u as $\mathcal{A}_1$'s response, and then adds (ID_u, x_u) to the list L_{VU}.

$Q_{sc}(m, ID_u, PK_u, PID_c, PK_c)$: When $\mathcal{A}_1$ provides tuple $(m, ID_u, PK_u, PID_c, PK_c)$ for a query, B performs as follows:

(1) If $PID_c \in L_{RC}$, then $PK_c = (x_cP, r_cP)$ is replaced by $PK_c' = (x_c'P, r_c'P)$. If $x_u' \neq x_u$ (or $r_u' \neq r_u$), $\mathcal{A}_1$ must send x_c' (or r_c') to $\mathcal{B}$. $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity PID_c to obtain SK_c, and then executes the signcrypt algorithm with tuple $(m, ID_u, PK_u, PID_c, SK_c)$ to output the ciphertext σ as $\mathcal{A}_1$'s response.

(2) If $PID_c \notin L_{RC}$, $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity PID_c to obtain SK_c, and then executes the signcrypt algorithm with tuple $(m, ID_u, PK_u, PID_c, SK_c)$ to output the ciphertext σ as $\mathcal{A}_1$'s response.

$Q_{un}(\sigma, ID_u, PK_u, PID_c, PK_c)$: When $\mathcal{A}_1$ provides tuple $(\sigma, ID_u, PK_u, PID_c, PK_c)$ for a query, $\mathcal{B}$ performs as follows:

(1) If $ID_u \in L_{RU}$, then $PK_u = (x_uP, r_uP)$ is replaced by $PK_u' = (x_u'P, r_u'P)$. If $x_u' \neq x_u$ (or $r_u' \neq r_u$), $\mathcal{A}_1$ must send x_u' (or r_u') to $\mathcal{B}$. $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity ID_u to obtain SK_u, and then executes the unsigncryption algorithm with tuple $(\sigma, ID_u, SK_u, PID_c, PK_c)$ to output the plaintext m or $\perp$ as $\mathcal{A}_1$'s response.

(2) If $ID_u \notin L_{RU}$ and $ID_u \neq ID^{\diamondsuit}$, $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity ID_u to obtain SK_u, and then executes the unsigncryption algorithm with tuple $(\sigma, ID_u, SK_u, PID_c, PK_c)$ to output the plaintext m or $\perp$ as $\mathcal{A}_1$'s response.

(3) If $ID_u \notin L_{RU}$ and $ID_u = ID^{\diamondsuit}$, $\mathcal{B}$ fails and terminates the process.

Challenge phase: $\mathcal{A}_1$ selects two distinct messages m_0, m_1 of the same length and subsequently transmits the tuple $(m_0, m_1, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ to $\mathcal{B}$. $\mathcal{B}$ performs the subsequent steps:

In Situation I, if $ID_u^* \neq ID^{\diamondsuit}$, then $\mathcal{B}$ randomly chooses $\xi \in \{0,1\}$ and performs $Q_{sc}(m_\xi, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ and outputs the ciphertext σ^* to $\mathcal{A}_1$.

In Situation II, if $ID_u^* = ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $\xi \in \{0,1\}$ and performs the subsequent steps:

(1) Searches for $(PID_c^*, x_c^*P, x_c^*, r_c^*P, r_c^*)$ in the list L_{UC}.

(2) Sets $PK_c^* = (x_c^*P, r_c^*P)$.

(3) Calculates $l_c^* = H_2(PID_c^*, R_c^*)$, $d_c^* = r_c^* + l_c^*\delta$.

(4) Searches for $(ID^{\diamondsuit}, x^{\diamondsuit}P, x^{\diamondsuit}, \alpha P, \nabla)$ in the list L_{UU}.

(5) Sets $PK_u^* = PK^{\diamondsuit} = (x^{\diamondsuit}P, \alpha P)$.

(6) Calculates $l_u^* = H_2(ID^{\diamondsuit}, \alpha P)$.

(7) Sets $K^* = \beta P (k^* = \beta)$.

(8) Calculates $\lambda^* = H_3(K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, $\tau^* = T + K^*(l_u^*\delta + \lambda^* x_u^*)$, $\theta^* = H_5(K^*, \tau^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*) \oplus m_\xi$, $\rho^* = H_4(\theta^*, K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, and $\omega^* = \frac{1}{d_c^* + \rho^* x_c^*} P$.

(9) Outputs $\sigma^* = (\theta^*, K^*, \omega^*)$ to $\mathcal{A}_1$.

Guess phase: $\mathcal{A}_1$ performs various queries adaptively as in the query phase and follows the rules of Game 1. After that, $\mathcal{A}_1$ outputs its guess $\xi' \in \{0,1\}$.

Solving the DDH problem: $\mathcal{B}$ returns "1" , if $\xi' = \xi$. Otherwise, $\mathcal{B}$ outputs "0". If $T = \alpha\beta P$, then

$$\begin{aligned} \tau^* &= \alpha\beta P + K^*(l_u^*\delta + \lambda^* x_u^*) \\ &= \beta(R_u^* + l_u^* P_{pub} + \lambda^* X_u^*) \end{aligned}$$

This means that σ^* is a true ciphertext. Therefore, the advantage of $\mathcal{A}_1$ in distinguishing symbol ξ is ε, that is to say:

$$\Pr[\mathcal{B} \to 1 | T = \alpha\beta P] = \Pr[\xi' = \xi | T = \alpha\beta P] = \tfrac{1}{2} + \varepsilon.$$

If $T \neq \alpha\beta P$, then σ^* is not a true ciphertext. This implies that for this σ^*, the distribution of $\xi = 0$ and $\xi = 1$ is the same. Therefore, $\mathcal{A}_1$ cannot have any advantage in identifying symbol ξ, that is to say:

$$\Pr[\mathcal{B} \to 1 | T \neq \alpha\beta P] = \Pr[\xi' = \xi | T \neq \alpha\beta P] = \tfrac{1}{2}.$$

Probability: Let q_{UU}, q_{RU}, q_{KU} and q_{UN} represent the number of $\mathcal{A}_1$ executes $Q_{upk}(ID_u)$, $R_{upk}(ID_u)$, $Q_{ppk}(ID_u)$ and $Q_{un}(\sigma, ID_u, PK_u, PID_c, PK_c)$, respectively. Next, we will calculate the probability of $\mathcal{B}$ successfully solving a given DDH problem. To facilitate understanding, we defined the following three events:

π_1: $\mathcal{A}_1$ has neither operated $Q_{ppk}(ID^{\diamondsuit})$ nor replaced the value of $R_u^{\diamondsuit}(\alpha P)$.

π_2: $\mathcal{A}_1$ has not failed in the $Q_{un}()$.

π_3: $ID_u^* = ID^{\diamondsuit}$.

Because if $\mathcal{A}_1$ replaces the public key of ID_u, it cannot perform $Q_{ppk}()$ for ID_u, therefore $L_{RU} \cap L_{KU} = \emptyset$. Based on the analysis, we can obtain the following results:

$$\begin{aligned}
\Pr[\pi_1] &= \frac{q_{UU} - q_{RU} - q_{KU}}{q_{UU}} \\
\Pr[\pi_2 | \pi_1] &= (1 - \frac{1}{q_{UU}})^{q_{UN}} \approx e^{-\frac{q_{UN}}{q_{UU}}} \\
\Pr[\pi_3 | \pi_1 \wedge \pi_2] &= \frac{1}{q_{UN} - q_{RU} - q_{KU}}
\end{aligned}$$

Then, the following results can be derived:

$$\begin{aligned}
\Pr[\mathcal{B}_{success}] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\
&= \Pr[\pi_1] \Pr[\pi_2 | \pi_1] \Pr[\pi_3 | \pi_1 \wedge \pi_2] \\
&\approx \frac{q_{UU} - q_{RU} - q_{KU}}{q_{UU}} e^{-\frac{q_{UN}}{q_{UU}}} \cdot \frac{1}{q_{UN} - q_{RU} - q_{KU}} \\
&\approx \frac{1}{q_{UU}} e^{-\frac{q_{UN}}{q_{UU}}}
\end{aligned}$$

Consequently, if $\mathcal{A}_1$ can distinguish symbol ξ with the advantage ε , then $\mathcal{B}$ can resolve the DDH problem with a probability of $\frac{\varepsilon}{q_{UU}} e^{-\frac{q_{UN}}{q_{UU}}}$. □

Lemma 2. *If the DDH problem is hard, our scheme is proven to be $IND - CCA_2$ against the adversary $\mathcal{A}_2$ in the SM.*

Proof. Given the tuple $(P, \alpha P, \beta P, T)$, where $\alpha, \beta \in Z_q^*$ and α, β are unknown. The goal of $\mathcal{B}$ is to determine whether T is equal to $a\beta P$.

Initialization phase: $\mathcal{A}_2$ obtains msk and $params = \{G_1, G_2, q, e, P, P_{pub} = \delta P, N = e(P, P), H_1 \sim H_5\}$ by executing the setup algorithm, then sends them to $\mathcal{B}$. After the process above, neither $\mathcal{A}_2$ nor $\mathcal{B}$ knows α and β, but $\mathcal{A}_2$ and $\mathcal{B}$ know δ.

Query phase: $\mathcal{B}$ sets $ID^{\diamondsuit}$ as the challenge target identity. For C, $\mathcal{A}_2$ must first execute $Q_{pid}(ID_c)$ and $Q_{upk}(PID_c)$ before any other queries. For U, $\mathcal{A}_2$ must first execute $Q_{upk}(ID_u)$ before any other queries. There are eight empty tables, $L_{UC}, L_{UU}, L_{RC}, L_{RU}, L_{KC}$, L_{KU}, L_{VC} and L_{VU}, maintained by $\mathcal{B}$. $\mathcal{A}_2$ can conduct the following types of queries, and $\mathcal{B}$ simulates $\mathcal{A}_2$'s queries as follows:

$Q_{pid}(ID_c)$: Similar to Lemma 1.

$Q_{upk}(PID_c)$: Similar to Lemma 1.

$Q_{upk}(ID_u)$: $\mathcal{B}$ maintains a list L_{UU}, which includes the the tuple $(ID_u, X_u, x_u, R_u, r_u)$. When $\mathcal{A}_2$ provides an identity ID_u for a query, if the ID_u is on the the list L_{UU}, $\mathcal{B}$ returns PK_u as $\mathcal{A}_2$'s response. Otherwise, ID_u is queried as a new identity, $\mathcal{B}$ performs the subsequent steps:

(1) If $ID_u = ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $r^{\diamondsuit} \in Z_q^*$, sets $PK_u = PK^{\diamondsuit} = (\alpha P, r^{\diamondsuit} P)$ as $\mathcal{A}_2$'s response, and adds $(ID^{\diamondsuit}, \alpha P, \nabla, r^{\diamondsuit} P, r^{\diamondsuit})$ to the list L_{UU}(where ∇ represents a null value).

(2) If $ID_u \neq ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $x_u, r_u \in Z_q^*$, computes $PK_u = (x_u P, r_u P)$ as $\mathcal{A}_2$'s response, and adds $(ID_u, x_u P, x_u, r_u P, r_u)$ to the list L_{UU}.

$R_{upk}(PID_c, PK_c')$: Similar to Lemma 1.

$R_{upk}(ID_u, PK_u')$: Similar to Lemma 1.

$Q_{ppk}(PID_c)$: Similar to Lemma 1.

$Q_{ppk}(ID_u)$: $\mathcal{B}$ maintains a list L_{KU} , which contains the tuple (ID_u, d_u). When $\mathcal{A}_2$ provides an identity ID_u for a query, $\mathcal{B}$ searches for $(ID_u, x_u P, x_u, r_u P, r_u)$ in the list L_{UU}, and then executes PPKG algorithm to output the tuple d_u. After that, $\mathcal{B}$ adds (ID_u, d_u) to the list L_{KU}.

$Q_{sv}(PID_c)$: Similar to Lemma 1.

$Q_{sv}(ID_u)$: $\mathcal{B}$ maintains the list L_{VU}, which contains the tuple (ID_u, x_u). When $\mathcal{A}_2$ provides an identity ID_u for a query, $\mathcal{B}$ performs the subsequent steps:

(1) If $ID_u = ID^{\diamondsuit}$, then $\mathcal{B}$ fails and terminates the process.

(2) If $ID_u \neq ID^{\diamondsuit}$, $\mathcal{B}$ searches for $(ID_u, x_u P, x_u, r_u P, r_u)$ in the list L_{UU}, outputs x_u as $\mathcal{A}_2$'s response, and then adds (ID_u, x_u) to the list L_{VU}.

$Q_{sc}(m, ID_u, PK_u, PID_c, PK_c)$: Similar to Lemma 1.

$Q_{un}(\sigma, ID_u, PK_u, PID_c, PK_c)$: Similar to Lemma 1.

Challenge phase: $\mathcal{A}_2$ selects two distinct messages m_0, m_1 of the same length and subsequently transmits the tuple $(m_0, m_1, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ to $\mathcal{B}$. $\mathcal{B}$ performs the subsequent steps:

In Situation I, if $ID_u^* \neq ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $\xi \in \{0,1\}$ and performs $Q_{sc}(m_\xi, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ to output the ciphertext σ^* to $\mathcal{A}_2$.

In Situation II, if $ID_u^* = ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $\xi \in \{0,1\}$ and performs the subsequent steps:

(1) Searches for $(PID_c^*, x_c^* P, x_c^*, r_c^* P, r_c^*)$ in the list L_{UC}.

(2) Sets $PK_c^* = (x_c^* P, r_c^* P)$.

(3) Calculates $l_c^* = H_2(PID_c^*, R_c^*)$, $d_c^* = r_c^* + l_c^* \delta$.

(4) Searches for $(ID^{\diamondsuit}, \alpha P, \nabla, r^{\diamondsuit} P, r^{\diamondsuit})$ in the list L_{UU}.

(5) Sets $PK_u^* = PK^{\diamondsuit} = (\alpha P, r^{\diamondsuit} P)$.

(6) Calculates $l_u^* = H_2(ID^{\diamondsuit}, r^{\diamondsuit} P)$, $d_u^* = r_u^* + l_u^* \delta$ (where $r_u^* = r^{\diamondsuit}$).

(7) Sets $K^* = \beta P (k^* = \beta)$.

(8) Calculates $\lambda^* = H_3(K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, $\tau^* = \lambda^* T + (r_u^* + l_u^* \delta) K^*$, $\theta^* = H_5(K^*, \tau^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*) \oplus m_\xi$, $\rho^* = H_4(\theta^*, K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, and $\omega^* = \frac{1}{d_c^* + \rho^* x_c^*} P$.

(9) Outputs $\sigma^* = (\theta^*, K^*, \omega^*)$ to $\mathcal{A}_2$.

Guess phase: $\mathcal{A}_2$ performs various queries adaptively, as in the query phase, and follows the rules of Game 2. After that, $\mathcal{A}_2$ outputs its guess $\xi' \in \{0,1\}$.

Solving the DDH problem: $\mathcal{B}$ returns "1", if $\xi' = \xi$. Otherwise, $\mathcal{B}$ outputs "0". If $T = \alpha\beta P$, then

$$\begin{aligned} \tau^* &= \lambda^* \alpha\beta P + (r_u^* + l_u^* \delta) K^* \\ &= \beta(r_u^* + l_u^* \delta + \lambda^* \alpha) P \\ &= \beta(R_u^* + l_u^* P_{pub} + \lambda^* X_u^*) \end{aligned}$$

This means that σ^* is a true ciphertext. Therefore, the advantage of $\mathcal{A}_2$ in distinguishing symbol ξ is ε, that is to say:

$$\Pr[\mathcal{B} \to 1 | T = \alpha\beta P] = \Pr[\xi' = \xi | T = \alpha\beta P] = \tfrac{1}{2} + \varepsilon.$$

If $T \neq \alpha\beta P$, then σ^* is not a true ciphertext. This implies that for this σ^*, the distribution of $\xi = 0$ and $\xi = 1$ is the same. Therefore, $\mathcal{A}_2$ cannot have any advantage in identifying symbol ξ, that is to say:

$$\Pr[\mathcal{B} \to 1 | T \neq \alpha\beta P] = \Pr[\xi' = \xi | T \neq \alpha\beta P] = \frac{1}{2}.$$

Probability: Let q_{UU}, q_{RU}, q_{VU} and q_{UN} represent the number of $\mathcal{A}_2$ executes $Q_{upk}(ID_u)$, $R_{upk}(ID_u)$, $Q_{sv}(ID_u)$ and $Q_{un}(\sigma, ID_u, PK_u, PID_c, PK_c)$, respectively. Next, we will calculate the probability of $\mathcal{B}$ successfully solving a given DDH problem. To facilitate understanding, we defined the following three events:

π_1: $\mathcal{A}_2$ has neither operated $Q_{sv}(ID^{\diamondsuit})$ nor replaced the value of $X_u^{\diamondsuit}(\alpha P)$.

π_2: $\mathcal{A}_2$ has not failed in the $Q_{un}()$.

π_3: $ID_u^* = ID^{\diamondsuit}$.

Because if $\mathcal{A}_2$ replaces the public key of ID_u, it cannot perform $Q_{sv}()$ for ID_u, therefore $L_{RU} \cap L_{VU} = \emptyset$. Based on the analysis, we can obtain the following results:

$$\begin{aligned}
\Pr[\pi_1] &= \frac{q_{UU} - q_{RU} - q_{VU}}{q_{UU}} \\
\Pr[\pi_2 | \pi_1] &= (1 - \frac{1}{q_{UU}})^{q_{UN}} \approx e^{\frac{q_{UN}}{q_{UU}}} \\
\Pr[\pi_3 | \pi_1 \wedge \pi_2] &= \frac{1}{q_{UN} - q_{RU} - q_{VU}}
\end{aligned}$$

Then, the following results can be derived:

$$\begin{aligned}
\Pr[\mathcal{B}_{success}] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\
&= \Pr[\pi_1] \Pr[\pi_2 | \pi_1] \Pr[\pi_3 | \pi_1 \wedge \pi_2] \\
&\approx \frac{q_{UU} - q_{RU} - q_{VU}}{q_{UU}} e^{-\frac{q_{UN}}{q_{UU}}} \cdot \frac{1}{q_{UN} - q_{RU} - q_{VU}} \\
&\approx \frac{1}{q_{UU}} e^{-\frac{q_{UN}}{q_{UU}}}
\end{aligned}$$

Consequently, if $\mathcal{A}_2$ can distinguish symbol ξ with the advantage ε , then $\mathcal{B}$ can resolve the DDH problem with a probability of $\frac{\varepsilon}{q_{UU}} e^{-\frac{q_{UN}}{q_{UU}}}$. □

Theorem 1. *If the DDH problem is hard, our scheme is proven to be $IND - CCA_2$ in the SM.*

Proof. From Lemmas 1 and 2, we can see that the conclusion is correct. □

Lemma 3. *If the CCA problem is hard, our scheme is proven to be UF-CMA against the adversary $\mathcal{A}_1$ in the SM.*

Proof. Given the tuple $(P, \alpha P)$. The goal of $\mathcal{B}$ is to output the tuple $(\gamma, \frac{1}{\alpha+\gamma} P)$.

Initialization phase: Same as the initialization phase in Lemma 1.

Query phase: $\mathcal{B}$ sets $PID^{\diamondsuit}$ as the challenge target identity. $\mathcal{A}_1$ can conduct the following types of queries, and B simulates $\mathcal{A}_1$'s queries as follows:

$Q_{pid}(ID_c)$: Similar to Lemma 1.

$Q_{upk}(PID_c)$: $\mathcal{B}$ maintains a list L_{UC}, which includes the tuple $(PID_c, X_c, x_c, R_c, r_c)$. When $\mathcal{A}_1$ provides an identity PID_c for a query, if the PID_c is on the list L_{UC}, $\mathcal{B}$ returns PK_c as $\mathcal{A}_1$'s response. Otherwise, PID_c is queried as a new identity, $\mathcal{B}$ performs the subsequent steps:

(1) If $PID_c = ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $x^{\diamondsuit} \in Z_q^*$, sets $PK_c = PK^{\diamondsuit} = (x^{\diamondsuit} P, \alpha P)$ as $\mathcal{A}_1$'s response, and adds the tuple $(PID^{\diamondsuit}, x^{\diamondsuit} P, x^{\diamondsuit}, \alpha P, \nabla)$ to the list L_{UC} (where ∇ represents a null value).

(2) If $PID_c \neq ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $x_c, r_c \in Z_q^*$, sets $PK_c = (x_cP, r_cP)$ as $\mathcal{A}_1$'s response, and adds the tuple $(PID_c, x_cP, x_c, r_cP, r_c)$ to the list L_{UC}.

$Q_{upk}(ID_u)$: $\mathcal{B}$ maintains a list L_{UU}, which contains the tuple $(ID_u, X_u, x_u, R_u, r_u)$. When $\mathcal{A}_1$ provides an identity ID_u for a query, if the ID_u is on the the list L_{UU}, $\mathcal{B}$ returns PK_u as $\mathcal{A}_1$'s response. Otherwise, ID_u is queried as a new identity, $\mathcal{B}$ randomly chooses $x_u, r_u \in Z_q^*$, sets $PK_u = (x_uP, r_uP)$, and adds $(ID_u, x_uP, x_u, r_uP, r_u)$ to the list L_{UU}.

$R_{upk}(PID_c, PK_c')$: Similar to Lemma 1.

$R_{upk}(ID_u, PK_u')$: Similar to Lemma 1.

$Q_{ppk}(PID_c)$: $\mathcal{B}$ maintains a list L_{KC}, which includes the tuple (PID_c, d_c). When $\mathcal{A}_1$ provides an identity PID_c for a query, $\mathcal{B}$ performs the subsequent steps:

(1) If $PID_c = PID^{\diamondsuit}$, then $\mathcal{B}$ fails and terminates the process.

(2) If $PID_c \neq PID^{\diamondsuit}$, $\mathcal{B}$ searches for $(PID_c, x_cP, x_c, r_cP, r_c)$ in the list L_{UC}, executes the PPKG algorithm to output d_c as $\mathcal{A}_1$'s response, and then adds (PID_c, d_c) to the list L_{UC}.

$Q_{ppk}(ID_u)$: Similar to Lemma 2.

$Q_{sv}(PID_c)$: Similar to Lemma 1.

$Q_{sv}(ID_u)$: Similar to Lemma 1.

$Q_{sc}(m, ID_u, PK_u, PID_c, PK_c)$: When $\mathcal{A}_1$ provides tuple $(m, ID_u, PK_u, PID_c, PK_c)$ for a query, $\mathcal{B}$ performs as follows:

(1) If $PID_c \in L_{RC}$, then $PK_c = (x_cP, r_cP)$ is replaced by $PK_c' = (x_c'P, R_c'P)$. If $x_c' \neq x_c$ (or $r_c' \neq r_c$), $\mathcal{A}_1$ must send x_c' (or r_c') to $\mathcal{B}$. $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity PID_c to obtain SK_c, and then executes the signcrypt algorithm with tuple $(m, ID_u, PK_u, PID_c, SK_c)$ to output the ciphertext σ as $\mathcal{A}_1$'s response.

(2) If $PID_c \notin L_{RC}$ and $PID_c \neq PID^{\diamondsuit}$, $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity PID_c to obtain SK_c, and then executes the signcrypt algorithm with tuple $(m, ID_u, PK_u, PID_c, SK_c)$ to output the ciphertext σ as $\mathcal{A}_1$'s response.

(3) If $PID_c \notin L_{RC}$ and $PID_c = PID^{\diamondsuit}$, $\mathcal{B}$ fails and terminates the process.

$Q_{un}(\sigma, ID_u, PK_u, PID_c, PK_c)$: When $\mathcal{A}_1$ provides tuple $(\sigma, ID_u, PK_u, PID_c, PK_c)$ for a query, $\mathcal{B}$ performs as follows:

(1) If $ID_u \in L_{RU}$, then $PK_u = (x_uP, r_uP)$ is replaced by $PK_u' = (x_u'P, r_u'P)$. If $x_u' \neq x_u$ (or $r_u' \neq r_u$), $\mathcal{A}_1$ must send x_u' (or r_u') to $\mathcal{B}$. $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity ID_u to obtain SK_u, and then executes the signcrypt algorithm with tuple $(\sigma, ID_u, SK_u, PID_c, PK_c)$ to output the plaintext m or $\perp$ as $\mathcal{A}_1$'s response.

(2) If $PID_c \notin L_{Ru}$, $\mathcal{B}$ first executes the PPKG algorithm and FSKS algorithm using identity ID_u to obtain SK_u, and then executes the signcrypt algorithm with tuple $(\sigma, ID_u, SK_u, PID_c, PK_c)$ to output the plaintext m or $\perp$ as $\mathcal{A}_1$'s response.

Forge phase: $\mathcal{A}_1$ outputs a tuple $(\sigma^* = (\theta^*, K^*, \omega^*), ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ and wins Game 3.

Solving CCA problem: If $PID_c^* \neq PID^{\diamondsuit}$, then $\mathcal{B}$ fails. Otherwise, $PID_c^* = PID^{\diamondsuit}$, then $PK_c^* = PK^{\diamondsuit} = (x^{\diamondsuit}P, \alpha P)$. Since σ^* is a valid ciphertext, it follows that $\omega^* = \frac{1}{d_c^* + \rho^* x_c^*} P$. $\mathcal{B}$ proceeds with the following steps:

(1) Searches for $(ID_u^*, x_u^*P, x_u^*, r_u^*P, r_u^*)$ in the list L_{UU}.

(2) Sets $PK_u^* = (x_u^*P, r_u^*P)$.

(3) Calcuates $l_u^* = H_2(ID_u^*, R_u^*)$, $d_u^* = r_u^* + l_u^*\delta$.

(4) Searches for $(PID^{\diamondsuit}, x^{\diamondsuit}P, x^{\diamondsuit}, \alpha P, \nabla)$ in the list L_{UC}.

(5) Sets $PK_c^* = PK^{\diamondsuit} = (x^{\diamondsuit}P, \alpha P)$.

(6) Calcuates $l_c^* = H_2(PID^{\diamondsuit}, \alpha P)$, $\lambda^* = H_3(K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, $\tau^* = (d_u^* + \lambda^* x_u^*)K^*$, $\rho^* = H_4(\theta^*, K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, $m^* = H_5(K^*, \tau^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*) \oplus \theta^*$, $\gamma = l_c^*\delta + \rho^* x_c^*$ (where $x_c^* = x^{\diamondsuit}$).

(7) Generates (γ, ω^*).

$$
\begin{aligned}
(\gamma, \omega^*) &= (\gamma, \frac{1}{d_c^* + \rho^* x_c^*} P) \\
&= (\gamma, \frac{1}{\alpha + l_c^* \delta + \rho^* x_c^*} P) \\
&= (\gamma, \frac{1}{\alpha + \gamma} P)
\end{aligned}
$$

Therefore, (γ, ω^*) serves as the response to the CCA problem.

Probability: Let q_{UC}, q_{RC}, q_{KC} and q_{SC} represent the number of $\mathcal{A}_1$ executes $Q_{upk}(PID_c)$, $R_{upk}(PID_c)$, $Q_{ppk}(PID_c)$ and $Q_{sc}(\sigma, ID_u, PK_u, PID_c, PK_c)$, respectively. Next, we will calculate the probability of $\mathcal{B}$ successfully solving a given CCA problem. To facilitate understanding, we defined the following three events:

π_1: $\mathcal{A}_1$ has neither operated $Q_{ppk}(PID^{\diamondsuit})$ nor replaced the value of $R_c^{\diamondsuit}(\alpha P)$.

π_2: $\mathcal{A}_1$ has not failed in $Q_{sc}()$.

π_3: $PID_c^* = PID^{\diamondsuit}$.

Because if $\mathcal{A}_1$ replaces the public key of PID_c, it cannot perform $Q_{ppk}()$ for PID_c, therefore $L_{RC} \cap L_{KC} = \emptyset$. Based on the analysis, we can obtain the following results:

$$
\begin{aligned}
\Pr[\pi_1] &= \frac{q_{UC} - q_{RC} - q_{KC}}{q_{UC}} \\
\Pr[\pi_2|\pi_1] &= (1 - \frac{1}{q_{UC}})^{q_{SC}} \approx e^{-\frac{q_{SC}}{q_{UC}}} \\
\Pr[\pi_3|\pi_1 \wedge \pi_2] &= \frac{1}{q_{UC} - q_{RC} - q_{KC}}
\end{aligned}
$$

Then, the following results can be derived:

$$
\begin{aligned}
\Pr[\mathcal{B}_{success}] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\
&= \Pr[\pi_1] \Pr[\pi_2|\pi_1] \Pr[\pi_3|\pi_1 \wedge \pi_2] \\
&\approx \frac{q_{UC} - q_{RC} - q_{KC}}{q_{UC}} e^{-\frac{q_{SC}}{q_{UC}}} \cdot \frac{1}{q_{UC} - q_{RC} - q_{KC}} \\
&= \frac{1}{q_{UC}} e^{-\frac{q_{SC}}{q_{UC}}}
\end{aligned}
$$

Consequently, if $\mathcal{A}_1$ can forge a real ciphertext with advantage ε, then $\mathcal{B}$ can resolve the DDH problem with a probability of $\frac{\varepsilon}{q_{UC}} e^{-\frac{q_{SC}}{q_{UC}}}$. □

Lemma 4. *If the CCA problem is hard, our scheme is proven to be UF-CMA against the adversary $\mathcal{A}_2$ in the SM.*

Proof. Given the tuple $(P, \alpha P)$. The goal of $\mathcal{B}$ is to output the tuple $(\gamma, \frac{1}{\alpha+\gamma} P)$.

Initialization phase: Same as the initialization phase in Lemma 2.

Query phase: $\mathcal{B}$ sets $PID^{\diamondsuit}$ as the challenge target identity. $\mathcal{A}_2$ can conduct the following types of queries, and B simulates $\mathcal{A}_1$'s queries as follows:

$Q_{pid}(ID_c)$: Similar to Lemma 1.

$Q_{upk}(PID_c)$: $\mathcal{B}$ maintains the list L_{UC}, which includes the tuple $(PID_c, X_c, x_c, R_c, r_c)$. When $\mathcal{A}_2$ provides an identity PID_c for a query, if the PID_c is on the list L_{UC}, $\mathcal{B}$ returns PK_c as $\mathcal{A}_2$'s response. Otherwise, PID_c is queried as a new identity, $\mathcal{B}$ performs the subsequent steps:

(1) If $PID_c = ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $r^{\diamondsuit} \in Z_q^*$, sets $PK_c = PK^{\diamondsuit} = (\alpha P, r^{\diamondsuit} P)$ as $\mathcal{A}_2$'s response, and adds the tuple $(PID^{\diamondsuit}, \alpha P, \nabla, r^{\diamondsuit} P, r^{\diamondsuit})$ to the list L_{UC} (where ∇ represents a null value).

(2) If $PID_c \neq ID^{\diamondsuit}$, $\mathcal{B}$ randomly chooses $x_c, r_c \in Z_q^*$, set $PK_c = (x_cP, r_cP)$ as $\mathcal{A}_2$'s response, and adds the tuple $(PID_c, x_cP, x_c, r_cP, r_c)$ to the list L_{UC}.

$Q_{upk}(ID_u)$: Similar to Lemma 3.

$R_{upk}(PID_c, PK_c')$: Similar to Lemma 1.

$R_{upk}(ID_u, PK_u')$: Similar to Lemma 1.

$Q_{ppk}(PID_c)$: Similar to Lemma 1.

$Q_{ppk}(ID_u)$: Similar to Lemma 3.

$Q_{sv}(PID_c)$: $\mathcal{B}$ maintains the list L_{VC}, which includes the tuple (PID_c, x_c). When $\mathcal{A}_2$ provides an identity PID_c for a query, $\mathcal{B}$ performs the subsequent steps:

(1) If $PID_c = PID^{\diamondsuit}$, then $\mathcal{B}$ fails and terminates the process.

(2) If $PID_c \neq PID^{\diamondsuit}$, $\mathcal{B}$ searches for $(PID_c, x_cP, x_c, r_cP, r_c)$ in the list L_{VC}, outputs x_c as $\mathcal{A}_2$'s response, and then adds (PID_c, x_c) to the list L_{VC}.

$Q_{sv}(ID_u)$: Similar to Lemma 1.

$Q_{sc}(m, ID_u, PK_u, PID_c, PK_c)$: Similar to Lemma 3.

$Q_{un}(\sigma, ID_u, PK_u, PID_c, PK_c)$: Similar to Lemma 3.

Forge phase: $\mathcal{A}_2$ outputs the tuple $(\sigma^* = (\theta^*, K^*, \omega^*), ID_u^*, PK_u^*, PID_c^*, PK_c^*)$ and wins Game 4.

Solving the CCA problem: If $PID_c^* \neq PID^{\diamondsuit}$, then $\mathcal{B}$ fails. Otherwise, $PID_c^* = PID^{\diamondsuit}$, then $PK_c^* = PK^{\diamondsuit} = (\alpha P, r^{\diamondsuit}P)$. Since σ^* is a valid ciphertext, it follows that $\omega^* = \frac{1}{d_c^* + \rho^* x_c^*} P$. $\mathcal{B}$ proceeds with the following steps:

(1) Searches for $(ID_u^*, x_u^*P, x_u^*, r_u^*P, r_u^*)$ in the list L_{UU} .

(2) Sets $PK_u^* = (x_u^*P, r_u^*P)$.

(3) Calculates $l_u^* = H_2(ID_u^*, R_u^*)$, $d_u^* = r_u^* + l_u^*\delta$.

(4) Searches for $(PID^{\diamondsuit}, \alpha P, \Delta, r^{\diamondsuit}P, r^{\diamondsuit})$ in the list L_{UC} .

(5) Sets $PK_c^* = PK^{\diamondsuit} = (\alpha P, r^{\diamondsuit}P)$.

(6) Calculates $l_c^* = H_2(PID^{\diamondsuit}, r^{\diamondsuit}P)$, $\lambda^* = H_3(K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, $\tau^* = (d_u^* + \lambda^* x_u^*)K^*$, $\rho^* = H_4(\theta^*, K^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*)$, $m^* = H_5(K^*, \tau^*, ID_u^*, PK_u^*, PID_c^*, PK_c^*) \oplus \theta^*$, $\gamma = \rho^{*-1}(r_c^* + l_c^*\delta)$ (where $r_c^* = r^{\diamondsuit}$).

(7) Generates $(\gamma, \rho^*\omega^*)$.

$$\begin{aligned}(\gamma, \rho^*\omega^*) &= (\gamma, \rho^* \frac{1}{d_c^* + \rho^* x_c^*} P) \\ &= (\gamma, \rho^* \frac{1}{r_c^* + l_c^*\delta + \alpha\rho^*} P) \\ &= (\gamma, \frac{1}{\rho^{*-1}(r_c^* + l_c^*\delta) + \alpha} P) \\ &= (\gamma, \frac{1}{\alpha + \gamma} P)\end{aligned}$$

Therefore, (γ, ω^*) serves as the response to the CCA problem.

Probability: Let q_{UC}, q_{RC}, q_{VC} and q_{SC} represent the number of $\mathcal{A}_2$ executing $Q_{upk}(PID_c)$, $R_{upk}(PID_c)$, $Q_{sv}(PID_c)$ and $Q_{sc}(\sigma, ID_u, PK_u, PID_c, PK_c)$, respectively. Next, we will calculate the probability of $\mathcal{B}$ successfully solving a given CCA problem. To facilitate understanding, we defined the following three events:

π_1: $\mathcal{A}_2$ has neither operated $Q_{sv}(PID^{\diamondsuit})$ nor replaced the value of $X_c^{\diamondsuit}(\alpha P)$.

π_2: $\mathcal{A}_2$ has not failed in $Q_{sc}()$.

π_3: $PID_c^* = PID^{\diamondsuit}$.

Because if $\mathcal{A}_2$ replaces the public key of PID_c, it cannot perform $Q_{sv}()$ for PID_c, therefore $L_{RC} \cap L_{VC} = \emptyset$. Based on the analysis, we can obtain the following results:

$$\Pr[\pi_1] = \frac{q_{UC} - q_{RC} - q_{VC}}{q_{UC}}$$
$$\Pr[\pi_2|\pi_1] = (1 - \frac{1}{q_{UC}})^{q_{SC}} \approx e^{-\frac{q_{SC}}{q_{UC}}}$$
$$\Pr[\pi_3|\pi_1 \wedge \pi_2] = \frac{1}{q_{UC} - q_{RC} - q_{VC}}$$

Then, the following results can be derived:

$$\begin{aligned}\Pr[\mathcal{B}_{success}] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\ &= \Pr[\pi_1] \Pr[\pi_2|\pi_1] \Pr[\pi_3|\pi_1 \wedge \pi_2] \\ &\approx \frac{q_{UC} - q_{RC} - q_{VC}}{q_{UC}} e^{-\frac{q_{SC}}{q_{UC}}} \cdot \frac{1}{q_{UC} - q_{RC} - q_{VC}} \\ &\approx \frac{1}{q_{UC}} e^{-\frac{q_{SC}}{q_{UC}}}\end{aligned}$$

Consequently, if $\mathcal{A}_2$ can forge a real ciphertext with advantage ε, then $\mathcal{B}$ can resolve the DDH problem with a probability of $\frac{\varepsilon}{q_{UC}} e^{-\frac{q_{SC}}{q_{UC}}}$. □

Theorem 2. *If the CCA problem is hard, our scheme is proven to be UF-CMA in the SM.*

Proof. From Lemmas 3 and 4, we can see that the conclusion is correct. □

Theorem 3. *If the DDH problem is hard, our scheme is proven to be anonymous to the signcrypter against the super adversary $\mathcal{A}$ in the SM.*

Proof. Given the tuple $(P, \alpha P, \beta P, T)$, where $\alpha, \beta \in Z_q^*$ and α, β are unknown. The goal of $\mathcal{B}$ is to determine whether T is equal to $a\beta P$.

Initialization phase: Same as the initialization phase in Lemma 2.

Query phase: $\mathcal{A}$ inputs various queries, and $\mathcal{B}$ executes the corresponding algorithm to generate the answer.

Challenge phase: $\mathcal{A}$ selects a message m^* and two distinct real identities ID_0^* and ID_1^* of C, where $\mathcal{A}$ has not performed Q_{pid} for ID_0^* and ID_1^*. $\mathcal{A}$ subsequently sends tuple $(m^*, ID_0^*, ID_1^*, ID_u^*, PK_u^*)$ to $\mathcal{B}$. $\mathcal{B}$ performs the subsequent steps:

(1) Randomly chooses $\xi \in \{0, 1\}$.

(2) Sets $E_\xi^* = \alpha P, F_\xi^* = \beta P$.

(3) Calculates $PID_\xi^* = ID_\xi^* \oplus H_1(\delta T, E_\xi^*, F_\xi^*)$

(4) Runs the PPKG algorithm, SVS algorithm and FSKS algorithm to acquire the private key SK_ξ^* of PID_ξ^*.

(5) Runs the signcrypt algorithm on the tuple $(m^*, ID_u^*, PK_u^*, PID_\xi^*, SK_\xi^*)$ to acquire the ciphertext σ^*.

(6) Outputs $(\sigma^*, ID_u^*, PK_u^*, PID_\xi^*, PK_\xi^*)$ to $\mathcal{A}$.

Guess phase: $\mathcal{A}$ performs various queries adaptively, as in the query phase. After that, $\mathcal{A}$ outputs its guess $\xi' \in \{0, 1\}$.

Solving the DDH problem: $\mathcal{B}$ returns "1", if $\xi' = \xi$. Otherwise, $\mathcal{B}$ outputs "0". If $T = \alpha\beta P$, then

$$\begin{aligned}PID_\xi^* &= ID_\xi^* \oplus H_1(\delta T, E_\xi^*, F_\xi^*) \\ &= ID_\xi^* \oplus H_1(\delta\alpha\beta P, \alpha P, \beta P)\end{aligned}$$

This means that PID_ξ^* is a true pseudo-identity. Therefore, the advantage of $\mathcal{A}$ in distinguishing symbol ξ is ε, that is to say:

$$\Pr[\mathcal{B} \to 1|T = \alpha\beta P] = \Pr[\xi' = \xi|T = \alpha\beta P] = \tfrac{1}{2} + \varepsilon.$$

If $T \neq \alpha\beta P$, then PID_{ξ}^{*} is not a true pseudo-identity. This implies that for this σ^{*}, the distribution of $\xi = 0$ and $\xi = 1$ is the same. Therefore, $\mathcal{A}$ cannot have any advantage in identifying symbol ξ, that is to say:

$$\Pr[\mathcal{B} \to 1 | T \neq \alpha\beta P] = \Pr[\xi' = \xi | T \neq \alpha\beta P] = \frac{1}{2}.$$

During the proof process, C will not fail.Consequently, if $\mathcal{A}$ can distinguish symbol ξ with the advantage ε , then $\mathcal{B}$ can resolve the DDH problem with a probability of ε. □

Theorem 4. *Our scheme is identifiable.*

Proof. Proof: The KGC can generate the $params = \{G_1, G_2, q, e, P, P_{pub} = \delta P, N = e(P, P), H_1 \sim H_5\}$ and the $msk = \{\delta\}$. Let $(\sigma = (\theta, K, \omega), ID_u, PK_u, PID_c, PK_c)$ be a legitimate ciphertext. The KGC then performs the subsequent steps:

(1) Searches for $(PID_c, e_c, e_c P, f_c, f_c P)$ in the list FI.

(2) Computes $\Delta C = \delta e_c f_c P$, $ID_c = PID_c \oplus H_1(\Delta C, e_c P, f_c P)$.

(3) Outputs the true identity ID_c of C.

Thus, for any ciphertext, the KGC can identify the true identity of C. So our scheme is identifiable. □

7. Performance Analysis

In this section, we delve into a comprehensive evaluation of our scheme's security properties, functionalities, computational expenses, and storage costs. Additionally, we compare its performance with the schemes presented in [28,30,32,40,46,49–51].

7.1. Security Analysis

Firstly, we analyze the security properties and functionalities of our scheme. Theorem 1 indicates that adversaries are unable to obtain valid messages, thus ensuring that our scheme can achieve confidentiality. Theorem 2 demonstrates that no adversary can forge legitimate signatures. Therefore, our scheme can simultaneously satisfy confidentiality, integrity, authentication, and non-repudiation. Theorem 3 indicates that our scheme provides anonymity for the signcryptor. Theorem 4 demonstrates that our scheme is also identity identifiable. Furthermore, our scheme is characterized by public verifiability, public ciphertext authenticity, and is classed as certificateless cryptography.

Secondly, compare the security properties and functional of our scheme with those of the schemes in [28,30,32,40,46,49–51]. The comparison results are shown in Table 1, where SM represents the standard model, ROM represents the random oracle model, $\surd$ represents the scheme compliance attribute, $\times$ represents the scheme non-compliance attribute, and - represents unknown. As shown in Table 1, our scheme satisfies the four security properties of confidentiality, integrity, authentication, non-repudiation. These properties have been proven within the standard model. Since our security model has been enhanced, our scheme stands out as the most secure among all schemes. Furthermore, compared to other schemes, only our scheme concurrently realizes all four functions: anonymity of the signcryptor, identity identifiability, public verifiability, and public ciphertext authenticity, while incorporating a certificateless design. Notably, the anonymity of the signcryptor and identity identifiability can be proven in the standard model. Therefore, our scheme is not only more secure but also has more comprehensive functionalities.

7.2. Efficiency Analysis

Moving forward, we proceed to compare the computational expenses associated with the previously discussed schemes. To facilitate the comparison, we adopt the computation time of the scheme by He et al. [52] as the benchmark. The relevant operations were implemented using the well-known cryptographic library (MIRACL) on a smartphone (Samsung Galaxy S5 G9001, Qualcomm Snapdragon 801 Quad-core 2.5 GHz Krait 400, GPU Adreno 330, 16GB 2GB RAM, Android 4.4.2 KitKat, Samsung Electronics, Seoul, Republic

of Korea). The symbols for various operations and their precise running times are detailed in Table 2. The function $e : G_1 \times G_1 \rightarrow G_2$ is defined as a bilinear pairing. In this context, G_1 represents an additive group of prime order q, which is constructed on the basis of a singular elliptic curve group over a finite field F_p of prime order. The bit size associated with p and q are designated as 512 and 160 bits, respectively.

Table 1. Comparison of the security properties and functionalities.

Schemes	Confidentiality	Integrity	Authentication	Non-Repudiation	Anonymity of the Signcrypter	Identity Identifiability	Public Verifiability	Public Ciphertext Authenticity	Certificateless	Security Model
Luo [28]	×	×	×	×	×	×	×	×	√	SM
Rastegari [40]	×	√	√	√	×	×	×	×	√	SM
Zhou [30]	√	√	√	√	×	×	×	×	√	SM
Zhou [53]	√	√	√	√	×	×	×	×	√	SM
Lu [46]	-	√	√	√	×	×	×	×	×	ROM
Deng [49]	√	√	√	√	√	√	×	×	√	SM
Li [32]	√	√	√	√	×	×	×	×	√	ROM
Luo [51]	√	√	√	√	×	×	√	√	√	ROM
Karati [50]	√	√	√	√	×	×	√	√	√	SM
our	√	√	√	√	√	√	√	√	√	SM

Table 2. Symbols.

Symbols	Definition
T_{bp}	The overhead it takes to execute a bilinear pairing operation, $T_{bp} \approx 32.713$ ms
T_{htp}	The overhead it takes to execute a hash-to-point operation, $T_{htp} \approx 33.582$ ms
$T_{sm_{G_1}}$	The overhead it takes to execute a scalar multiplication operation in G_1, $T_{sm_{G_1}} \approx 13.405$ ms
$T_{exp_{G_2}}$	The overhead it takes to execute a exponentiation operation in G_2, $T_{exp_{G_2}} \approx 2.249$ ms

In terms of the computational efficiency of the CLSC scheme, its performance mainly depends on the computational costs of the signcryption and unsigncryption algorithms. For this reason, we focus solely on the computational costs of these two algorithms. To compare computational complexity more effectively, our primary focus lies in comparing the two most time-consuming operations: bilinear pairing operation and hash-to-point operation, so as to more accurately evaluate the performance differences between them. From Table 3, we observe that in the signcryption algorithm, our scheme requires only one bilinear pairing operation and does not necessitate any hash-to-point operations. In contrast, scheme [53] necessitates up to five bilinear pairing operations, while scheme [46] requires one bilinear pairing operation in addition to two hash-to-point operations. Although schemes [32,49–51] employ fewer instances of both operations compared to ours, our overall time consumption remains lower than theirs. In the unsigncryption algorithm, our scheme requires only one bilinear pairing operation and does not necessitate any hash-to-point operations. Other schemes perform bilinear pairing operation at least twice, but do not involve hash-to-point operation. Overall, with the exception of scheme [50] our scheme provably employs a lower number of bilinear pairing operations and hash-to-point operations compared to all other schemes. Moreover, our total time consumption is still lower than all other schemes. Therefore, our scheme boasts the lowest computational complexity. Below is a detailed analysis.

We measured the computational overheads for the various schemes, as illustrated in Table 3 and Figure 3. According to the scheme [28], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $3T_{sm_{G_1}} + T_{exp_{G_2}} + 3T_{bp}$, $2T_{sm_{G_1}} + 6T_{bp}$, and $5T_{sm_{G_1}} + T_{exp_{G_2}} + 9T_{bp} = 363.691$ ms, respectively. In the scheme [40], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and total are $4T_{sm_{G_1}} + 2T_{bp}$, $2T_{sm_{G_1}} + 8T_{bp}$, $6T_{sm_{G_1}} + 10T_{bp} = 407.56$ ms, respectively.

In the scheme [30], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $3T_{sm_{G_1}} + 4T_{exp_{G_2}} + T_{bp}$, $5T_{sm_{G_1}} + 4T_{bp}$, and $8T_{sm_{G_1}} + 4T_{exp_{G_2}} + 5T_{bp} = 279.801$ ms, respectively. In the scheme [53], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $5T_{sm_{G_1}} + 3T_{exp_{G_2}} + 5Tbp$, $3T_{sm_{G_1}} + 2T_{exp_{G_2}} + 4T_{bp}$, and $8T_{sm_{G_1}} + 5T_{exp_{G_2}} + 9T_{bp} = 412.902$ ms, respectively. In the scheme [46], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $8T_{sm_{G_1}} + T_{exp_{G_2}} + T_{bp} + 2T_{htp}$, $T_{sm_{G_1}} + 6T_{bp}$, and $9T_{sm_{G_1}} + T_{exp_{G_2}} + 7T_{bp} + 2T_{htp} = 419.049$ ms, respectively. In the scheme [49], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $6T_{sm_{G_1}} + T_{bp}$, $2T_{sm_{G_1}} + T_{exp_{G_2}} + 2T_{bp}$, and $8T_{sm_{G_1}} + T_{exp_{G_2}} + 3T_{bp} = 207.628$ ms, respectively. In the scheme [32], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $6T_{sm_{G_1}} + T_{exp_{G_2}}$, $5T_{sm_{G_1}} + T_{exp_{G_2}} + 5T_{bp}$, and $11T_{sm_{G_1}} + 2T_{exp_{G_2}} + 5T_{bp} = 315.518$ ms, respectively. In the scheme [51], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $2T_{sm_{G_1}} + T_{exp_{G_2}}$, $4T_{sm_{G_1}} + T_{exp_{G_2}} + 4T_{bp}$, and $6T_{sm_{G_1}} + 2T_{exp_{G_2}} + 4T_{bp} = 215.78$ ms, respectively. In the scheme [50], the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $5T_{sm_{G_1}} + T_{exp_{G_2}}$, $3T_{sm_{G_1}} + 2T_{exp_{G_2}} + 2T_{bp}$, and $8T_{sm_{G_1}} + 3T_{exp_{G_2}} + 2T_{bp} = 179.413$ ms, respectively. In our scheme, the computational overhead of the signcryption algorithm, unsigncryption algorithm, and the total are $5T_{sm_{G_1}} + T_{bp} = 99.738$, $3T_{sm_{G_1}} + T_{bp} = 72.928$, and $8T_{sm_{G_1}} + 2T_{bp} = 172.666$ ms, respectively.

Table 3. Comparison of computational overheads.

Schemes	Signcryption (ms)	Unsigncryption (ms)	Total (ms)
Luo [28]	$3T_{sm_{G_1}} + T_{exp_{G_2}} + 3T_{bp}$	$2T_{sm_{G_1}} + 6T_{bp}$	363.691
Rastegari [40]	$4T_{sm_{G_1}} + 2T_{bp}$	$2T_{sm_{G_1}} + 8T_{bp}$	407.56
Zhou [30]	$3T_{sm_{G_1}} + 4T_{exp_{G_2}} + T_{bp}$	$5T_{sm_{G_1}} + 4T_{bp}$	279.801
Zhou [53]	$5T_{sm_{G_1}} + 3T_{exp_{G_2}} + 5T_{bp}$	$3T_{sm_{G_1}} + 2T_{exp_{G_2}} + 4T_{bp}$	412.902
Lu [46]	$8T_{sm_{G_1}} + T_{exp_{G_2}} + T_{bp} + 2T_{htp}$	$T_{sm_{G_1}} + 6T_{bp}$	419.049
Deng [49]	$6T_{sm_{G_1}} + T_{bp}$	$2T_{sm_{G_1}} + T_{exp_{G_2}} + 2T_{bp}$	207.628
Li [32]	$6T_{sm_{G_1}} + T_{exp_{G_2}}$	$5T_{sm_{G_1}} + T_{exp_{G_2}} + 5T_{bp}$	315.518
Luo [51]	$2T_{sm_{G_1}} + T_{exp_{G_2}}$	$4T_{sm_{G_1}} + T_{exp_{G_2}} + 4T_{bp}$	215.78
Karati [50]	$5T_{sm_{G_1}} + T_{exp_{G_2}}$	$3T_{sm_{G_1}} + 2T_{exp_{G_2}} + 2T_{bp}$	179.413
our	$5T_{sm_{G_1}} + T_{bp}$	$3T_{sm_{G_1}} + T_{bp}$	172.666

Based on Figure 3 and the analysis above, the computational cost for unsigncryption in our scheme is lower than all other schemes. While the schemes [30,32,50,51] have a lower computational cost for signcryption than ours, they suffer from a lack of critical functionalities. Specifically, schemes [30,32] do not provide anonymity of the signcrypter, identity identifiability, public verifiability, and public ciphertext authenticity. Additionally, schemes [50,51] also lack anonymity of the signcrypter and identity identifiability. In contrast, our scheme maintains a balance between computational efficiency, the essential security and critical functionalities. In terms of total cost, the total cost of our scheme is the lowest, and it can be observed that the total computational overhead for our scheme is approximately 47.48% of the scheme [28], 42.37% of the scheme [40], 61.71% of the scheme [30], 41.82% of the scheme [53], 41.20% of the scheme [46], 83.16% of the scheme [49], 54.72% of the scheme [32], 80.02% of the scheme [51], and 96.24% of the scheme [50].

Next, we compare the storage costs of the schemes, as shown in Table 4 and Figure 4. Let $|G_1|$, $|G_2|$, $|Z_q^*|$ denote the size of elements in G_1, G_2 and Z_q^*, respectively. Accordingly, we have $|G_1| = |G_2| = 128$ bytes, and $|Z_q^*| = 20$ bytes. The size of the output produced by the hash function is denoted as $\eta = 40$ bytes, the output size of the identity information is

denoted as $\delta = 8$ bytes, and the attribute size is denoted as τ. Let us assume $\tau = \delta = 8$ bytes.

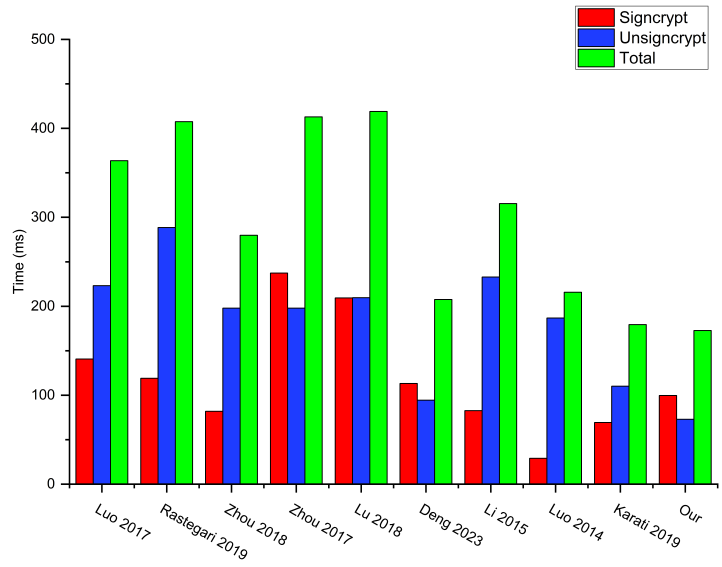

Figure 3. Comparison of computational overheads [28,30,32,40,46,49–51,53].

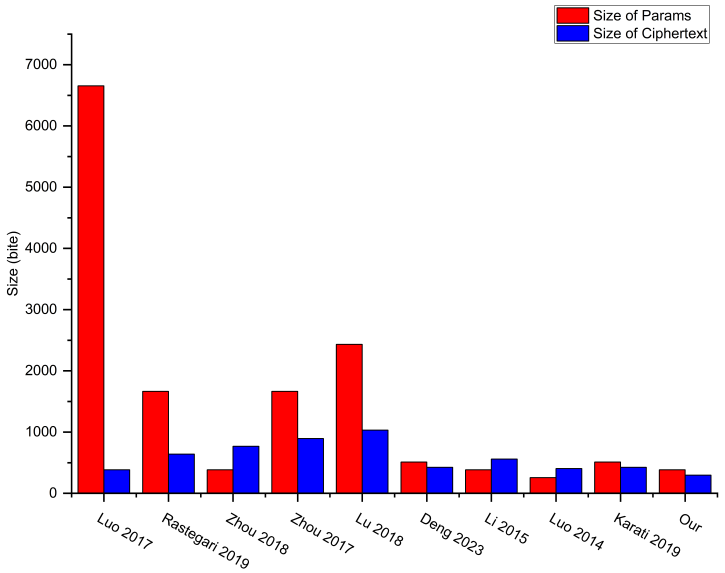

Figure 4. Comparison of storage costs [28,30,32,40,46,49–51,53].

The size of the system parameters in the scheme [28,30,32,40,46,49–51,53], and our scheme are $(\delta + \eta + 4)|G_1| = (8 + 40 + 4) \times 128 = 6656$ bytes, $(\delta + 4)|G_1| + |G_2| = (8 + 4) \times 128 + 128 = 1664$ bytes, $3|G_1| = 3 \times 128 = 384$ bytes, $(\delta + 5)|G_1| = (8 + 5) \times 128 = 1664$ bytes, $(2 + \delta + \tau)|G_1| + |G_2| = (2 + 8 + 8) \times 128 + 128 = 2432$ bytes, $3|G_1| + |G_2| = 3 \times 128 + 128 = 512$ bytes, $|G_1| + 2|G_2| = 128 + 2 \times 128 = 384$ bytes, $2|G_1| = 2 \times 128 = 256$ bytes, $3|G_1| + |G_2| = 3 \times 128 + 128 = 512$ bytes, and $2|G_1| + |G_2| = 2 \times 128 + 128 = 384$ bytes, respectively.

The length of the ciphertext in the scheme [28,30,32,40,46,49–51,53], and our scheme are $2|G_1| + |G_2| = 2 \times 128 + 128 = 384$ bytes, $4|G_1| + |G_2| = 4 \times 128 + 128 = 640$ bytes, $3|G_1| + 3|G_2| = 3 \times 128 + 3 \times 128 = 768$ bytes, $4|G_1| + 3|G_2| = 4 \times 128 + 3 \times 128 = 896$ bytes, $7|G_1| + |G_2| + \delta = 7 \times 128 + 128 + 8 = 1032$ bytes, $3|G_1| + \eta = 3 \times 128 + 40 = 424$ bytes, $4|G_1| + 2|Z_q^*| + \delta = 4 \times 128 + 2 \times 20 + 8 = 560$ bytes, $2|G_1| + |G_2| + |Z_q^*| =$

$2 \times 128 + 128 + 20 = 404$ bytes, $2|G_1| + |G_2| + \eta = 2 \times 128 + 128 + 40 = 424$ bytes, and $2|G_1| + \delta = 2 \times 128 + 8 = 296$ bytes, respectively.

Table 4. Comparison of storage costs.

Scheme	Size of System Parameters (bytes)	Size of Ciphertext (bytes)
Luo [28]	$(\delta + \eta + 4)\|G_1\|$ /(6656)	$2\|G_1\| + \|G_2\|$ /(384)
Rastegari [40]	$(\delta + 4)\|G_1\| + \|G_2\|$/(1664)	$4\|G_1\| + \|G_2\|$ /(640)
Zhou [30]	$3\|G_1\|$/(384)	$3\|G_1\| + 3\|G_2\|$/(768)
Zhou [53]	$(\delta + 5)\|G_1\|$/(1664)	$4\|G_1\| + 3\|G_2\|$/(896)
Lu [46]	$(2 + \delta + \tau)\|G_1\| + \|G_2\|$/(2432)	$7\|G_1\| + \|G_2\| + \delta$/(1032)
Deng [49]	$3\|G_1\| + \|G_2\|$/(512)	$3\|G_1\| + \eta$/(424)
Li [32]	$\|G_1\| + 2\|G_2\|$/(384)	$4\|G_1\| + 2\|Z_q^*\| + \delta$/(560)
Luo [51]	$2\|G_1\|$/(256)	$2\|G_1\| + \|G_2\| + \|Z_q^*\|$ /(404)
Karati [50]	$3\|G_1\| + \|G_2\|$/(512)	$2\|G_1\| + \|G_2\| + \eta$ /(424)
our	$2\|G_1\| + \|G_2\|$/(384)	$2\|G_1\| + \delta$ /(296)

Based on Figure 4 and the previous detailed analysis, our scheme exhibits a notable advantage in ciphertext length, surpassing all other schemes except for scheme [51]. Nevertheless, it must be pointed out that while scheme [51] is relatively close to us in ciphertext length, it fails to offer the two crucial features of anonymity of the signcrypter and identity identifiability. Furthermore, the security proof of scheme [51] relies on the random oracle model, which to some extent undermines its universality and reliability in practical applications. Notably, in terms of the length of system parameters, our scheme achieves the shortest length, which fully demonstrates its superiority in efficiency. It can be observed that the system parameter size of our scheme is approximately 5.77% of the scheme [28], 23.08% of the scheme [40], 100% of the scheme [30], 23.08% of the scheme [53], 15.79% of the scheme [46], 60.76% of the scheme [49], 100% of the scheme [32], 150% of the scheme [51], and 75% of the scheme [50]. The ciphertext size of our scheme is approximately 77.08% of the scheme [28], 46.25% of the scheme [40], 38.54% of the scheme [30], 33.04% of the scheme [53], 28.66% of the scheme [46], 69.81% of the scheme [49], 52.86% of the scheme [32], 73.27% of the scheme [51], and 69.81% of the scheme [50].

In conclusion, our CL-ASC scheme has demonstrated all crucial security properties in the standard model, and it is also more comprehensive in terms of functionality, particularly in offering anonymity of the signcrypter and identity identifiability. With the exception of the scheme [51], which holds a slight advantage in terms of system parameter size, our scheme outperforms all other known schemes in both computational overhead and storage costs. Consequently, compared to existing schemes, our CL-ASC scheme boasts lower computational and storage costs while maintaining a higher level of security. This makes it an ideal and cost-effective communication solution for WBANs.

The ethical and regulatory issues surrounding signcryption schemes primarily manifest in the following aspects: Technical Ethics: Signcryption schemes require ensuring that the technology, in its design and implementation, is not only secure and reliable but also respects user privacy, is transparent and auditable, and adheres to ethical and legal standards.This includes utilizing robust encryption algorithms to safeguard data, adopting decentralized storage to mitigate privacy risks, implementing data minimization principles to reduce the likelihood of breaches, and ensuring transparency to build trust. Social Ethics: Signcryption schemes, which can be employed to protect the communication and transactions of individuals or organizations, must be designed with social interests and public safety in mind. For instance, it should not be permissible for encryption technology to be utilized in support of illegal activities or to evade legal oversight. Individual Ethics: In the design of signcryption schemes, it is imperative to respect and protect the personal privacy of users. This implies that the processes of generating, storing, and utilizing en-

cryption keys must ensure the confidentiality and privacy of user data. Transparency and Accountability: The provider of the signcryption scheme should clarify its responsibilities in data protection and transparently explain its data processing and protection measures to users and interested parties. Legal and Regulatory Compliance: Signcryption schemes must adhere to relevant legal and regulatory requirements, including data protection laws, electronic communication laws, and other pertinent regulations. User Education and Awareness: Users of signcryption technology should be educated about their rights and responsibilities, including how to safely use encryption tools and protect their keys. In summary, the ethical and regulatory issues surrounding signcryption schemes encompass various aspects, such as privacy protection, transparency and accountability, legal and regulatory compliance, and technological neutrality and balance, as well as user education and awareness enhancement. Addressing these issues necessitates concerted efforts and collaboration among technical designers, providers, users, and regulatory bodies.

8. Conclusions

Designing a secure and economical communication scheme specifically for Wireless Body Area Networks (WBANs) is a critical issue that needs urgent attention. Signcryption technology has emerged as an ideal choice for WBAN due to its ability to simultaneously achieve confidentiality, authentication, integrity, and non-repudiation at a relatively low cost. However, while the recently proposed CLSC schemes possess their own advantages, they also suffer from several drawbacks, including reliance on the ROM for security proofs, lack of public verifiability, public ciphertext authenticity and anonymity, and high computational costs. To address these issues, this paper first introduces a novel CL-ASC scheme. Second, it establish an enhanced security model for the CL-ASC scheme. Furthermore, it proves that our CL-ASC scheme possesses indistinguishability, unforgeability, and anonymity of the signcrypter within the standard model. Finally, a comparative analysis of the performance of several CLSC schemes reveals that our CL-ASC scheme has lower computational and storage costs and superior security. Consequently, our CL-ASC scheme offers a more ideal and economical communication solution tailored for WBAN applications.

Author Contributions: Conceptualization, L.D.; Methodology, W.L.; Formal analysis, W.L.; Investigation, W.L.; Data curation, Y.G. and T.L.; Writing—original draft, W.L.; Writing—review & editing, L.D.; Visualization, Y.G. and T.L.; Supervision, L.D. and J.Z.; Project administration, J.Z.; Funding acquisition, L.D. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by the Guiyang City Science and Technology Plan Project under Grant No. [2021]43-8, the Guizhou Province Hundred-level Innovative Talent Project under Grant No. GCC[2022]018-1, the Natural Science Research Project of Guizhou Provincial Department of Education under Grant No. [2023]010, the Guizhou Provincial Science and Technology Plan Project under Grant No. ZK[2024]Key058, and the National Natural Science Foundation of China under Grant No. 61962011, Guizhou Normal University Academic New Seedling Fund Project (QianShiXinMiao[2021]B09).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reichman, A. Standardization of body area networks. In Proceedings of the 2009 IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems, Tel Aviv, Israel, 9–11 November 2009; pp. 1–4.
2. He, J.; Geng, Y.; Wan, Y.; Li, S.; Pahlavan, K. A Cyber Physical Test-Bed for Virtualization of RF Access Environment for Body Sensor Network. *IEEE. Sens. J.* **2013**, *13*, 3826–3836. [CrossRef]
3. Liu, D.; Geng, Y.; Liu, G.; Zhou, M.; Pahlavan, K. WBANs-Spa: An Energy Efficient Relay Algorithm for Wireless Capsule Endoscopy. In Proceedings of the 2015 IEEE 82nd Vehicular Technology Conference, Boston, MA, USA, 6–9 September 2015; pp. 1–5.

4. He, D.; Chan, S.; Tang, S. A Novel and Lightweight System to Secure Wireless Medical Sensor Networks. *IEEE J. Biomed. Health* **2014**, *18*, 316–326. [CrossRef] [PubMed]
5. Jindal, F.; Jamar, R.; Churi, P. Future and Challenges of Internet of Things. *Int. J. Comput. Sci. Inf. Technol.* **2018**, *10*, 13–25. [CrossRef]
6. Limbasiya, T.; Karati, A. Cryptanalysis and improvement of a mutual user authentication scheme for the Internet of Things. In Proceedings of the 2018 International Conference on Information Networking, Chiang Mai, Thailand, 10–12 January 2018; pp. 168–173.
7. ACM Code of Ethics and Professional Conduct. Available online: https://www.acm.org/diversity-inclusion/code-of-ethics (accessed on 19 July 2024).
8. Reporting and Whistleblower Policy. Available online: http://www.maa.org/about-maa/policies-and-procedures (accessed on 19 July 2024).
9. Ethical Guidelines of the American Mathematical Society. Available online: https://www.ams.org/about-us/governance/policy-statements/sec-ethics (accessed on 19 July 2024).
10. Phillip Rogaway. The Moral Character of Cryptographic Work. In Proceedings of the Asiacrypt 2015, Auckland, New Zealand, 2 December 2015.
11. Zheng, Y. Digital signcryption or how to achieve cost(signature & encryption) $\leq$ cost(signature) + cost(encryption). In Proceedings of the 17th Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 1997; pp. 165–179.
12. Cagalaban, G.; Kim, S. Towards a secure patient information access control in ubiquitous healthcare systems using identity-based signcryption. In Proceedings of the 13th International Conference on Advanced Communication Technology, Seoul, Republic of Korea, 13–16 February 2011; pp. 863–867.
13. He, D.B.; Zeadally, S. Authentication Protocol for an Ambient Assisted Living System. *IEEE Commun. Mag.* **2015**, *53*, 71–77. [CrossRef]
14. Liu, J.; Zhang, Z.; Chen, X.; Kwak, K.S. Certificateless remote anonymous authentication schemes for wireless body area networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 332–342. [CrossRef]
15. Ma, C.; Xue, K.; Hong, P. Distributed access control with adaptive privacy preserving property for wireless sensor networks. *Secur. Commun. Netw.* **2014**, *7*, 759–773 [CrossRef]
16. Malone-Lee, J.; Mao, W. Two Birds One Stone: Signcryption Using RSA. In *Lecture Notes in Computer Science, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2612.
17. Li, M.; Lou, W.J.; Ren, K. Data Security and pricacy in wireless body wireless body area networks. *IEEE. Wirel. Commun.* **2010**, *17*, 51–58. [CrossRef]
18. Chow, S.S.M.; Yiu, S.M.; Hui, L.C.K.; Chow, K.P. Efficient Forward and Provably Secure ID-Based Signcryption Scheme with Public Verifiability and Public Ciphertext Authenticity. In Proceedings of the International Conference on Information Security and Cryptology, Chennai, India, 20–22 December 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 352–369.
19. Hu, C.; Zhang, F.; Cheng, X.; Liao, X.; Chen, D. Securing communications between external users and wireless body area networks. In Proceedings of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy, New York, NY, USA, 19 April 2013; pp. 31–36.
20. Al-Riyami, S.S.; Paterson, K.G. Certificateless Public Key Cryptography. In *ASIACRYPT 03*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 452–473.
21. Tsai, T.T.; Lin, H.Y.; Wu, C.Y. Cbeet: Constructing certificate-based encryption with equality test in the cb-pks. *Inf. Technol. Control* **2023**, *52*, 935–951. [CrossRef]
22. Tsai, T.T.; Lin, H.Y.; Tsai, H.C. Revocable certificateless public key encryption with equality test. *Inf. Technol. Control* **2022**, *51*, 638–660. [CrossRef]
23. Barbosa, M.; Farshim, P. Certificateless signcryption. In Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, New York, NY, USA, 18–20 March 2008; pp. 369–372.
24. Liu, Z.; Hu, Y.; Zhang, X.; Ma, H. Certificateless signcryption scheme in the standard model. *Inf. Sci.* **2010**, *180*, 452–464. [CrossRef]
25. Luo, M.; Huang, D.; Hu, J. An Efficient Biometric Certificateless Signcryption Scheme. *J. Comput.* **2013**, *8*, 1853–1860. [CrossRef]
26. Issac, B.; Israr, N. *Case Studies in Secure Computing: Achievements and Trends*, 1st ed.; CRC Press: Boca Raton, FL, USA; Taylor and Francis: Abingdon, UK, 2014.
27. Luo, M.; Wang, S.; Hu, J. A More Efficient and Secure Broadcast Signcryption Scheme Using Certificateless Public-Key Cryptography for Resource-Constrained Networks. *J. Internet Technol.* **2016**, *17*, 81–89.
28. Luo, M.; Wan, Y. An Enhanced Certificateless Signcryption in the Standard Model. *Wireless. Pers. Commun.* **2017**, *98*, 2693–2709. [CrossRef]
29. Hwang, Y.H.; Liu, J.K.; Chow, S.S.M. Certificateless public key encryption secure against malicious KGC attacks in the standard model. *J. Univers. Comput. Sci.* **2008**, *14*, 463–480.
30. Zhou, C.X. Certificateless Signcryption Scheme Without Random Oracles. *Chin. J. Electrons* **2018**, *27*, 1002–1008. [CrossRef]
31. Deng, L.; Yang, Y.; Gao, R.; Chen, Y. Certificateless short signature scheme from pairing in the standard model. *Int. J. Commun. Syst.* **2018**, *31*, e3796. [CrossRef]
32. Li, J.; Zhao, J.; Zhang, Y. Certificateless online/offline signcryption scheme. *Secur. Commun. Netw.* **2015**, *8*, 1979–1990. [CrossRef]

33. Bellare, M.; Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In Proceedings of the 1st ACM conference on Computer and Communications Security, New York, NY, USA, 3–5 November 1993; pp. 62–73.
34. Selvi, S.S.D.; Vivek, S.S.; Rangan, C.P. Security Weaknesses in Two Certificateless Signcryption Schemes. Available online: https://eprint.iacr.org/2010/092 (accessed on 25 July 2024).
35. Miao, S.; Zhang, F.; Li, S.; Mu, Y. On security of a certificateless signcryption scheme. *Inf. Sci.* **2013**, *232*, 475–481. [CrossRef]
36. Weng, J.; Yao, G.; Deng, R.H.; Chen, M.R.; Li, X. Cryptanalysis of a certificateless signcryption scheme in the standard model. *Inf. Sci.* **2011**, *181*, 661–667. [CrossRef]
37. Jin, Z.P.; Wen, Q.Y.; Zhang, H. A Supplement to Liu et al.'s Certificateless Signcryption Scheme in the Standard Model. Available online: https://eprint.iacr.org/2010/252 (accessed on 25 July 2024).
38. Xiong, H. Toward Certificateless Signcryption Scheme Without Random Oracles. Available online: http://eprint.iacr.org/2014/162 (accessed on 25 July 2024).
39. Yuan, Y.M. Security Analysis of an Enhanced Certificateless Signcryption in the Standard Model. *Wireless. Pers. Commun.* **2020**, *112*, 387–394. [CrossRef]
40. Rastegari, P.; Susilo, W.; Dakhlalian, M. Efficient Certificateless Signcryption in the Standard Model: Revisiting Luo and Wan's Scheme from Wireless Personal Communications (2018). *Comput. J.* **2019**, *62*, 1178–1193. [CrossRef]
41. Lin, X.J.; Sun, L.; Yan, Z.; Zhang, X.; Qu, H. On the Security Of A Certificateless Signcryption With Known Session-Specific Temporary Information Security In The Standard Model. *Comput. J.* **2020**, *63*, 1259–1262. [CrossRef]
42. Au, M.H.; Mu, Y.; Chen, J.; Wong, D.S.; Liu, J.K.; Yang, G. Malicious KGC attacks in certificateless cryptography. In Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS 2007). ACM, Singapore, 20–22 March 2007; pp. 302–311. [CrossRef]
43. Li, F.; Hong, J. Efficient Certificateless Access Control for Wireless Body Area Networks. *IEEE. Sens. J.* **2016**, *16*, 5389–5396. [CrossRef]
44. Hussain, S.; Ullah, S.S.; Uddin, M.; Iqbal, J.; Chen, C.-L. A comprehensive survey on signcryption security mechanisms in wireless body area networks. *Sensors* **2022**, *22*, 1072. [CrossRef]
45. Li, F.; Han, Y.; Jin, C. Cost-Effective and Anonymous Access Control for Wireless Body Area Networks. *IEEE. Syst. J.* **2018**, *12*, 747–758. [CrossRef]
46. Lu, Y.; Wang, X.; Hu, C.; Li, H.; Huo, Y. A traceable threshold attribute-based signcryption for mhealthcare social network. *Int. J. Sens. Netw.* **2018**, *26*, 43–53. [CrossRef]
47. Liu, X.; Wang, Z.; Ye, Y.; Li, F. An efficient and practical certificateless signcryption scheme for wireless body area networks. *Comput. Commun.* **2020**, *162*, 169–178. [CrossRef]
48. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. *Siam. J. Comput.* **2003**, *32*, 586–615. [CrossRef]
49. Deng, L.; Wang, B.; Gao, Y.; Chen, Z.; Li, S. Certificateless Anonymous Signcryption Scheme With Provable Security in the Standard Model Suitable for Healthcare Wireless Sensor Networks. *IEEE. Internet Things* **2023**, *10*, 15953–15965. [CrossRef]
50. Karati, A.; Fan, C.I.; Hsu, R.H. Provably secure and generalized signcryption with public verifiability for secure data transmission between resource-constrained iot devices. *IEEE. Internet Things* **2019**, *6*, 10431–10440. [CrossRef]
51. Luo, M.; Tu, M.; Xu, J. A security communication model based on certificateless online/offline signcryption for internet of things. In *Security and Communication Networks* ; Wiley: New York, NY, USA, 2014; Volume 7.
52. He, D.; Zeadally, S.; Kumar, N.; Wu, W. Efficient and Anonymous Mobile User Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-Server Architectures. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2052–2064. [CrossRef]
53. Zhou, C.X.; Gao, G.Y.; Cui, Z.M. Certificateless Signcryption in the Standard Model. *Wireless. Pers. Commun.* **2017**, *92*, 495–513. [CrossRef]