electronics

Article

Mobile Sensoring Data Verification via a Pairing-Free Certificateless Signature Secure Approach against Novel Public Key Replacement Attacks

Guilin Wang [1], Hua Shen [2], Liquan Chen [1], Jinguang Han [1] and Ge Wu [1,*]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China; wglin@seu.edu.cn (G.W.)
[2] School of Computer Science, Hubei University of Technology, Wuhan 430068, China
* Correspondence: gewu@seu.edu.cn

Abstract: To achieve flexible sensing coverage with low deployment costs, mobile users need to contribute their equipment as sensors. Data integrity is one of the most fundamental security requirements and can be verified by digital signature techniques. In the mobile crowdsensing (MCS) environment, most sensors, such as smartphones, are resource-limited. Therefore, many traditional cryptographic algorithms that require complex computations cannot be efficiently implemented on these sensors. In this paper, we study the security of certificateless signatures, in particular, some constructions without pairing. We notice that there is no secure pairing-free certificateless signature scheme against the super adversary. We also find a potential attack that has not been fully addressed in previous studies. To handle these two issues, we propose a concrete secure construction that can withstand this attack. Our scheme does not rely on pairing operations and can be applied in scenarios where the devices' resources are limited.

Keywords: mobile sensors; data integrity; certificateless signature; public key replacement attack; pairing-free

check for updates

Citation: Wang, G.; Shen, H.; Chen, L.; Han, J.; Wu, G. Mobile Sensoring Data Verification via a Pairing-Free Certificateless Signature Secure Approach against Novel Public Key Replacement Attacks. *Electronics* **2023**, *12*, 4865. https://doi.org/10.3390/electronics12234865

Academic Editors: Chuan Zhang, Tong Wu and Weiting Zhang

Received: 30 October 2023
Revised: 27 November 2023
Accepted: 27 November 2023
Published: 2 December 2023

1. Introduction

Various mobile sensors are utilized in IoT devices to perform real-time data detection. These sensors capture sensitive information such as vehicle status, power system data, and personal health information, among others. Once collected, the data are transmitted to a central server for processing, making data security a critical consideration. To ensure data credibility and reliability, the use of digital signatures for integrity verification and message tracing is imperative for these sensing devices. Given the limited hardware resources of these devices, signature schemes with less complex pairing computations are preferred. Over the last few decades, the public key infrastructure/certificate authority (PKI/CA) system has been extensively employed. Within this system, upper layers issue certificates for lower layers, constructing a chain of trust from the trusted root to individual entities. However, many signature schemes reliant on the PKI/CA system introduce complex certificate management challenges, including distribution, update, and revocation, which are often financially burdensome for sensor devices.

Shamir [1] proposed identity-based cryptography (IBC) as a solution to eliminate the need for certificates. This approach allows users to directly generate their public keys from identity information, such as the IP. The private key generator (PKG) is responsible for holding the system master key and using it to generate all user's private keys. By bypassing the need for certificates, IBC ensures the correctness of public key generation directly from identity information. Despite this advantage, the system's security heavily relies on the PKG. Consequently, a key escrow problem arises, as every private key is generated by the PKG, who then has the capability to arbitrarily compromise the security of the scheme.

Consequently, if the PKG is breached or lacks full trust, the safety of the entire system is compromised, leaving no user immune to potential security breaches.

Al-Riyami and Paterson [2] introduced certificateless public key cryptography (CLPKC) as a solution to the shortcomings of existing systems. In CLPKC, the key generation center (KGC) is responsible for controlling the master private key and differs from traditional PKI/CA systems in that it only generates a portion of the private key for users. Users must independently select and safeguard a secret value, using it to calculate both the complete private key and public key. As a result, the explicit binding of public keys and identity information through certificates is eliminated. Instead, the implicit binding of identity and public key occurs through the use of partial private keys, ensuring that only a valid user can generate a valid private key. Although KGC has ownership of the master private key, the secret values remain unknown. Consequently, CLPKC resolves the issue of key escrow in the IBC and eliminates the need for certificates in PKI/CA systems. Yet, the complexity and power of adversaries increase, posing new challenges. Consequently, there is ongoing research to comprehensively evaluate adversary capabilities and develop a fully secure CLPKC scheme.

1.1. Related Works

In 2003, Al-Riyami and Paterson [2] introduced the CLPKC system, which was based on the IBE scheme proposed by Boneh and Franklin [3] in 2001, and included an adversary model and security definition. However, their signature scheme was compromised by Huang et al. in 2005 [4]. Meanwhile, Yum and Lee developed general secure constructions for signature schemes (CLS) [5] and encryption schemes (CLE) [6] in 2004, which were constructed on a PKI/CA scheme and an IBC scheme. Despite this, subsequent work by Hu et al. [7] and Libert et al. [8] in 2006 demonstrated the insecurity of Yum and Lee's general construction. In response to the threat posed by malicious KGC, Au et al. further fortified the security model of CLPKC in 2006 and determined that a class of schemes with the same key structure may be vulnerable under malicious KGC and unable to address key escrow issues [9]. Building on this, Huang et al. revisited the CLPKC security model in 2007, categorizing adversaries into three levels: Normal, Strong, and Super adversaries. In addition, they proposed a secure CLS scheme specifically designed to withstand super adversaries [10].

A number of CLPKC schemes have been proposed; they aim to address the limitations of pairing operations, which can be expensive and inefficient in lightweight equipment like mobile sensors. Baek et al. [11] introduced the first CLPKC scheme without pairing operations, using the Schnorr signature [12]. However, Sun et al. [13] identified some drawbacks in Baek's approach and subsequently developed a new CLE scheme. Notably, Zhang and Mao [14] also devised a CLS scheme using the RSA signature. Despite this, Xu et al. [15] highlighted a flaw in the CLS scheme proposed by Gowri et al. [16], revealing that their signatures were susceptible to forgery. In response, Xu et al. [15] proposed a secure CLS scheme designed to withstand normal adversaries. Additionally, Karati et al. [17] developed a highly efficient CLS scheme by eliminating the map-to-point hash function, although Zhang et al. [18] later discovered that this scheme was vulnerable to breach through the replacement of the public key. Several other CLS schemes [19–24] were also proposed but were ultimately proven to be weak. More recently, Du et al. [25] and Xiang et al. [26] put forth two super-secure CLS schemes, but their security proofs were found to be incorrect, particularly with regard to the divisor always being calculated as zero when addressing underlying difficulties.

1.2. Motivations

In the CLS secure model, adversaries are categorized into two types. Type I adversaries have the capability to replace the public key with any string. In the security proof, the simulator must provide the correct signature in response to a signature inquiry, irrespective of whether the public key has been replaced. Not only that, the question arises of whether this signature should be valid before or after the replacement. Different levels of adversaries are defined by Huang based on this distinction, namely normal, strong, and super. The primary differentiator among these levels is the validity of the signatures they are able to obtain. A normal adversary may obtain a signature that is valid before the replacement, while a strong adversary may obtain a signature that is valid after the replacement, only if it supplies the corresponding secret values. On the other hand, a super adversary has the ability to replace the public key with a new key and receive a valid signature under the new key. In 2011, Huang introduced the first super security certificateless signature scheme using pairing. Subsequent works attempted to propose a secure CLS scheme without pairing, but the majority failed to achieve security against the super adversary.

Among the proposed pairing-free CLS schemes, the partial private key is typically calculated through Schnorr signature [12], which includes a random number R. It is important to note that this random number should be publicly available in the public key. Consequently, a $TypeI$ adversary has the ability to query for a partial private key and replace the public key, and the order of these two operations is not limited. Furthermore, the presence of super adversaries introduces the potential for them to substitute the private key without providing the new secret value. This vulnerability becomes even more pronounced when the adversary first replaces the random number with a new one and then requests the new partial private key under the new number, rendering existing schemes unable to respond correctly without a new secret value. It is essential to recognize that this vulnerability has been previously overlooked in CLS schemes that do not involve pairing.

1.3. Contributions

- Under the ECDLP assumption, this paper proposes a secure CLS scheme without pairing. Our work includes completing the security proof against super adversaries in the ROM, as shown by [10].
- We fix the weakness that the simulator of the CLS scheme using Schnorr signatures could not answer partial private key queries after replacing the public key. Specifically, we adjusted the structure of the public key to partially restrict these queries
- Our signature scheme breaks away from pairing operations and the signature length is only two group elements, achieving a balance between computational efficiency and transmission costs.

1.4. Structure

In Section 2, we present the outline of CLS schemes and the security model. In Section 3, we introduce our secure CLS scheme without pairing, and in Section 4, we demonstrate its security. Section 5 analyzes the efficiency of our scheme, while Section 6 provides a summary of this paper.

2. Certificateless Signature Schemes

2.1. Construction

A CLS scheme usually involves three parties: the KGC, one user who signs a message, and another user who verifies the signature and consists of six algorithms:

- **Setup(λ).** KGC runs this algorithm with inputting security parameter λ. The final output is the system public parameters PP and the system master secret key msk. KGC publishes PP and keeps msk private.
- **PartialPrivateKey(PP, msk, ID).** KGC runs this algorithm with inputting PP, msk and a user identity ID. Then KGC must distribute the output as user partial private key D_{ID} securely.
- **SecretValue(PP, ID).** A user runs this algorithm by inputting PP and ID. The final output serves as the secret value x_{ID}.
- **PublicKey(PP, ID, x_{ID}, D_{ID}).** A user runs this algorithm with inputting PP, ID, x_{ID} and D_{ID}. The output serves as its public key PK_{ID} and should be published.

- **Sign($PP, m, ID, x_{ID}, D_{ID}$).** A user runs this algorithm with inputting PP, a message m, ID, x_{ID} and D_{ID}. The output serves as the signature σ.
- **Verify($PP, \sigma, m, ID, PK_{ID}$).** A user runs this algorithm with inputting PP, ID, PK_{ID}, m and σ. Then it outputs "1" when validation is successful and otherwise outputs 0.

2.2. Security Models

We consider two types of super adversaries. The $TypeI$ adversaries simulate external attackers who are allowed to replace public keys arbitrarily and get partial private keys and secret values by corrupting some users. The $TypeII$ adversaries simulate the malicious KGC. They own the system master key but are not allowed to replace public keys. In this paper, we prove the security through two games, and the attack ability of adversaries is described by the access to the oracles. Specifically, the following five oracles will be considered.

- ***CreateUser(ID).*** This oracle will reply with a public key. When the ID is queried for the first time, the oracle generates a partial private key, a secret value, and a public key and records all information. It will reply according to records.
- ***PartialPrivateKeyExtract(ID).*** This oracle will reply with a partial private key. When the ID is queried for the first time, the oracle call ***Createuser(ID).*** It will reply according to the records.
- ***SecretValueExtract(ID).*** This oracle will reply with a secret value. When the ID is queried for the first time, the oracle calls ***Createuser(ID).*** It will reply according to the records.
- ***ReplacePublicKey(ID,PK').*** This oracle will change the public key of ID in records. When the ID is queried for the first time, the oracle calls ***Createuser(ID)***. Then it changes the public key to PK' in records.
- ***SuperSign(ID,m).*** The oracle will reply with a legal signature of a message m under the PK and ID in records. Note that the PK may have been replaced and there may be no secret value in records.

Game I : A challenger C interacts with a super $TypeI$ adversary A_1 through Game I. C controls all the oracle and records the interactive information. The complete game processes are as follows:

Init. C runs $Setup$ and transmits PP to A_1.

Query. A_1 can query for the above five oracles adaptively and C must respond correctly.

Forgery. A_1 finally outputs a signature $\sigma*$, a message $m*$, $PK*$ and $ID*$.

If the following equations hold, A_1 wins in Game I.

1. A_1 has not asked for the partial private key of $ID*$,
2. A_1 has not asked for a signature of the message $m*$ under $ID*$ and $PK*$,
3. The signature $\sigma*$ is valid, i.e.,

$$Verify(PP, ID*, PK*, m*, \sigma*) = 1 \tag{1}$$

Game II: The challenger C interacts with a super $TypeII$ adversary A_2 through game II. C controls all the oracle and records the interactive information. The complete game processes are as follows:

Init. C runs the $Setup$ algorithm and transmits both PP and msk to A_2.

Query. A_2 can query for four oracles except for $PartialPrivateKeyExtract(ID)$ adaptively and C must respond correctly. A_2 does not need to ask $PartialPrivateKeyExtract(ID)$ as it knows msk.

Forgery. A_2 finally outputs a signature $\sigma*$, a message $m*$, $PK*$ and $ID*$.

If the following equations hold, A_2 wins in Game II.

1. A_2 has not asked for the secret value of $ID*$,
2. A_2 has not replaced the public key of $ID*$,
3. A_2 has not asked for a signature of $m*$ under $ID*$ and $PK*$,

4. The signature $\sigma*$ is valid, i.e.,

$$Verify(PP, ID*, PK*, m*, \sigma*) = 1 \tag{2}$$

3. Our CLS Scheme

3.1. Security Assumptions

Given an elliptic curve group G of a prime order q, a point P is a generator and another point Q is a random element. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is to calculate $a \in Z_q*$ which satisfies the equation $Q = aP$. Our scheme is secure if the probability of solving the ECDLP is negligible for any probabilistic polynomial-time adversary.

3.2. Scheme Construction

There are six algorithms in our construction.

1. **Setup(λ)**: Inputting a security parameters λ, KGC generates public parameters PP and master secret key msk. First, it randomly generates a prime number q of λ-bits and an elliptic curve group G of order q. It randomly picks a generator $P \in G$, a number $s \in Z_q$ and sets $P_{pub} = sP$. It also selects the cryptography hash functions $< H_0, H_1, H_2, H_3, H_4 >$: $\{0,1\}^* \to Z_q$. Finally, KGC publishes $PP = \{G, P_{pub}, H_0, H_1, H_2, H_3, H_4\}$ and sets $msk = s$.

2. **PartialPrivateKey(PP, msk, ID)**: When generating the partial private key for ID, KGC inputs PP, msk and ID. Then KGC randomly selects $r, y_{ID} \in Z_q$ and calculates

$$R_{ID} = rP \tag{3}$$

$$d_{ID} = r + sH_1(ID, R_{ID}, P_{pub}) \tag{4}$$

$$Y_{ID} = y_{ID}P \tag{5}$$

$$\pi_{ID} = y_{ID} + sH_0(ID, Y_{ID}, R_{ID}, P_{pub}) \tag{6}$$

The partial private key $D_{ID} =< R_{ID}, d_{ID}, Y_{ID}, \pi_{ID} >$ must be securely transmitted to the user, and its legality can be verified by calculating $h_0 = H_0(ID, Y_{ID}, R_{ID}, P_{pub})$, $h_1 = H_1(ID, R_{ID}, P_{pub})$ and checking whether the equations $d_{ID}P = R_{ID} + h_1P_{pub}$, $\pi_{ID}P = Y_{ID} + h_0P_{pub}$ hold.

3. **SecretValue(PP, ID)**: With inputting PP and ID, the user randomly selects $x_{ID} \in Z_q$ as the secret value.

4. **PublicKey(PP, ID, x_{ID}, D_{ID})**: When generating the public key, the user inputs PP, ID, x_{ID} and D_{ID}. Then it calculates $X_{ID} = x_{ID}P$ and sets public key $PK_{ID} =< R_{ID}, Y_{ID}, X_{ID}, \pi_{ID} >$.

5. **Sign($PP, m, ID, x_{ID}, D_{ID}, PK_{ID}$)**: When signing a message m, the user inputs PP, m, ID, x_{ID} and $D_{ID} =< R_{ID}, d_{ID}, Y_{ID}, \pi_{ID} >$. Then it selects random $t \in Z_q$ and calculates

$$T = tP \tag{7}$$

$$h_2 = H_2(ID, m, PK_{ID}, T) \tag{8}$$

$$h_3 = H_3(ID, m, PK_{ID}, T) \tag{9}$$

$$h_4 = H_4(ID, m, T, PK_{ID}, P_{pub}) \tag{10}$$

$$\tau = t \cdot h_2 + x \cdot h_3 + d_{ID} \cdot h_4 \tag{11}$$

The user sets $\sigma =< T, \tau >$ as the signature.

6. **Verify($PP, m, \sigma, ID, PK_{ID}$)**: When verifying the legitimacy of a message-signature pair, the user inputs PP, m, σ, ID and PK_{ID}. Then it calculates

$$h_1 = H_1(ID, R_{ID}, P_{pub}) \quad (12)$$

$$h_2 = H_2(ID, m, PK_{ID}, T) \quad (13)$$

$$h_3 = H_3(ID, m, PK_{ID}, T) \quad (14)$$

$$h_4 = H_4(ID, m, T, PK_{ID}, P_{pub}) \quad (15)$$

and checks $\tau P = h_2 T + h_3 X_{ID} + h_4(R_{ID} + h_1 P_{pub})$. Finally, it outputs "1" when validation is successful and otherwise outputs "0".

4. Security Proof

Next, we demonstrate the security of our scheme against two super adversaries.

Theorem 1. *In the Random Oracle Model, assuming that ECDLP is difficult in the selected group G, our scheme is existentially unforgeable against super adversaries. This theorem can be obtained from the Lemmas 1 and 2.*

Lemma 1. *Assuming that there exists a super Type-I adversary A_1 who can (ϵ, t)-win GameI, then the ECDLP in G must be (ϵ', t')-solved.*

Proof. Given a ECDLP instance $< G, P, Q >$, we construct an algorithm C_1 to (ϵ', t')-calculate a solution by interacting with the adversary A_1. □

H_i are simulated as random oracle and C_1 maintains the tables L_i to record the input *val* and output *res* corresponding to H_i. The *GameI* runs as follows.

Setup. C_1 randomly selects $ID*$ as the challenge identity, sets $P_{pub} = Q$ and publishes $PP = \{G, P_{pub}, H_0, H_1, H_2, H_3, H_4\}$.

Query. A_1 can adaptively query to C_1 at any time and C_1 will response as follows.

- $Hash_i(val)$. C_1 first checks whether val exists in L_i. If there is a record, C_1 returns $< val, res >$. Otherwise C_1 randomly selects $h_i \in Z_q$, returns $res = h_i$ and insert $< val, res >$ into L_i.
- $CreateUser(ID_i)$. Suppose C_1 queries $CreateUser(ID_i)$ for at most q_u times. It maintains a list L_u and sets a tag in L_u to record whether the $< R, Y, \sigma >$ in the public key has been replaced. C_1 returns the public key according to the record if ID_i is found in the list L_u. Otherwise,
  - If $ID_i = ID*$, C_1 randomly selects $r, x, h_1, \pi, h_0 \in Z_q$, calculates $R = rP$, $X = xP$ and sets $H_1(ID, R, P_{pub}) = h_1$, calculates $Y = \pi P - h_0 P_{pub}$ and sets $H_0(ID, Y_{ID}, P_{pub}, R) = h_0$. Then it returns $PK =< R, X, Y, \pi >$ and inserts $< ID*, r, x, h_1, \pi, h_0, R, X, Y, tag = 0 >$ into the table L_u.
  - If $ID_i \neq ID*$, C_1 randomly selects $d, x, h_1, \pi, h_0 \in Z_q$, calculates $R = dP - h_1 P_{pub}, X = xP, Y = \pi P - h_0 P_{pub}$ and sets $H_1(ID, R, P_{pub}) = h_1, H_0(ID, Y, P_{pub}, R) = h_0$. Then return $PK =< R, X, Y, \pi >$ and insert $(ID_i, d, x, y, h_1, \pi, h_0, R, X, Y, tag = 0)$ into the table L_u.
- $PartialPrivateKeyExtract(ID_i)$. Suppose C_1 queries this oracle for at most q_{ppk} times.
  - If $ID_i = ID*$, abort the game.
  - Otherwise, C_1 searches the table L_u for ID_i. If ID_i is found and $tag = 0$, return d according to the record directly. If ID_i is found while the $tag = 1$, C_1 checks whether the public key $PK =< R, X, Y, \pi >$ is legal by $h_0 = H_0(ID_i, Y, R, P_{pub})$, $\pi P = Y + h_0 P_{pub}$. If the public key is still valid, we use the forking lemma on $h'_0 = H_0(ID_i, Y, R, P_{pub})$ to get a new $< R_1, Y_1, \pi_1 >$ that satisfies $\pi_1 P = Y_1 + h'_0 P_{pub}$. Then we can get $\pi = y + h_0 s$, $\pi_1 = y + h'_0 s$ and $s = \frac{\pi - \pi_1}{h_0 - h'_0}$ is the solution

to the ECDLP instance. If the public key is invalid, we return nothing. In addition, if ID_i is not found, call $CreateUser(ID_i)$ and then return d_{ID}.

- $SecretValueExtract(ID_i)$.
  - If $ID_i = ID*$, abort the game.
  - Otherwise, C_1 searches the table L_u for ID_i. If ID_i is found $tag = 0$, C_1 returns x_{ID_i} according to the record directly. If ID_i is found while the public key has been replaced without providing x_{ID_i}, C_1 returns nothing. If ID_i is not found, C_1 calls $CreateUser(ID_i)$ and returns x_{ID_i}.
- $ReplacePublicKey(ID_i, PK')$. C_1 searches the table L_u to find ID_i. If ID_i is found, it replaces $< R, Y, X, \pi >$ with PK'. Otherwise, C_1 calls $CreateUser(ID_i)$ and replaces $< R, Y, X, \pi >$ with PK'. C_1 sets $tag = 1$.
- $SuperSign(ID_i, m)$.
  - If $ID = ID^*$ or $tag = 1$, C_1 randomly selects $\tau, h_3, h_4, h_2 \in Z_q$ and calculates $T = h_2^{-1}(\tau P - h_3 X - h_4 R - h_4 h_1 P_{pub})$. Then C_1 set $h_2 = H_2(ID_i, m, PK, T)$, $h_3 = H_3(ID_i, m, PK, T)$, $h_4 = H_4(ID_i, m, T, PK, P_{pub})$ in L_2, L_3, L_4. $< T, \tau >$ is valid signature for

$$h_2 T + h_3 X + h_4(R + h_1 P_{pub}) = \tau P \tag{16}$$

    and note that C_1 does not need to know x,
  - If $ID \neq ID^*$ and $tag = 0$, C_1 searches the table L_u to find ID_i. If ID_i is found, C_1 get $< d, x >$. Then C_1 randomly selects $t, h_2, h_3, h_4 \in Z_q$ and sets $h_2 = H_2(ID_i, m, PK, T)$, $h_3 = H_3(ID_i, m, PK, T)$, $h_4 = H_4(ID_i, m, T, PK, P_{pub})$ in L_i. Finally C_1 calculates $\tau = h_2 t + h_3 x + h_4 d$. $< T, \tau >$ is a valid signature.

Forgery. In the end, A_1 outputs $< T, \tau, m, ID >$. If $ID \neq ID*$, aborts. Otherwise, C_1 searches the table L_u to find ID and verifies the signature:

$$h_1 = H_1(ID, R, P_{pub}) \tag{17}$$

$$h_2 = H_2(ID, m, PK, T) \tag{18}$$

$$h_3 = H_3(ID, m, PK, T) \tag{19}$$

$$h_4 = H_4(ID, m, T, PK, P_{pub}) \tag{20}$$

$$\tau P = h_2 T + h_3 X + h_4(R + h_0 P_{pub}) \tag{21}$$

If $tag = 0$, we use the forking lemma on H_4 to get a new output $< T, \tau', m, ID >$. These outputs satisfy $\tau = h_2 t + h_3 x + h_4 d, \tau' = h_2 t + h_3 x + h'_4 d$ so that C_1 can calculate $d = \frac{\tau - \tau'}{h_4 - h'_4}$. If R is not replaced, C_1 owns r and calculates $s = (d - r)/h_1$. s is the solution to the ECDLP instance. If $tag = 1$, we do the same as in $PartialPrivateKeyExtract(ID_i)$ to get s.

C_1 will solve the ECDLP if the following events occur:

- ϵ_1: C_1 never aborts in $GameI$,
- ϵ_2: A_1 generates a valid forgery $< T, \tau, m, ID >$,
- ϵ_3: In the forgery, $ID = ID^*$

So the probability of C_1 is $Pr[\epsilon_1 \wedge \epsilon_2 \wedge \epsilon_3] = Pr[\epsilon_1] \cdot Pr[\epsilon_2|\epsilon_1] \cdot Pr[\epsilon_3|\epsilon_1 \wedge \epsilon_2]$.

C_1 will abort in the $GameI$ if A_1 extracts the partial private key for any userID^*. So $Pr[\epsilon_1] = (1 - 1/q_u)^{q_{ppk}}$. If C_1 does not abort in the $GameI$, A_1 generates a valid forgery with ϵ. So $Pr[\epsilon_2|\epsilon_1] = \epsilon$. As the ID^* is selected randomly, $Pr[\epsilon_3|\epsilon_1 \wedge \epsilon_2] = 1/q_u$. So the probability is $\epsilon' = Pr[\epsilon_1 \wedge \epsilon_2 \wedge \epsilon_3] = Pr[\epsilon_1] \cdot Pr[\epsilon_2|\epsilon_1] \cdot Pr[\epsilon_3|\epsilon_1 \wedge \epsilon_2] = (1 - 1/q_u)^{q_{ppk}} \cdot 1/q_u \cdot \epsilon$.

Lemma 2. *Assuming that there exists a super Type-II adversary A_2 who can (ϵ, t)-win $GameII$, then the ECDLP must be (ϵ, t)-solved.*

Proof. Given a ECDLP instance $< G, P, Q >$, we construct an algorithm C_2 to (ϵ', t')-calculate a solution by interacting with the adversary A_2 . □

H_i are simulated as random oracle and C_2 maintains the tables L_i to record the input *val* and output *res* corresponding to H_i. The *GameII* runs as follows.

Setup. C_2 randomly selects $ID*$ as the challenge identity and $s \in Z_q$ as the *msk*. Then calculate $P_{pub} = sP$ and public $PP = \{G, P_{pub}, H_0, H_1, H_2, H_3, H_4\}$.

Query. A_2 can adaptively query to C_2 at any time and C_2 will response as follows.

- $Hash_i(val)$. C_2 first checks whether *val* exists in L_i. If there is a record, C_2 returns $< val, res >$. Otherwise C_2 randomly selects $h_i \in Z_q$, returns $res = h_i$ and inserts $< val, res >$ into L_i
- $CreateUser(ID_i)$. Suppose it queries $CreateUser(ID_i)$ for at most q_u times. C_2 maintains a list L_u and sets a *tag* in L_u to record whether the public key has been replaced. C_2 returns the public key if ID_i is in the list. Otherwise,
  - If $ID_i = ID^*$, C_2 randomly selects $r, y_{ID}, h_1, h_0 \in Z_q$, calculates $R = rP, Y = yP, d = r + sh_1, \sigma = y_{ID} + d_{ID}h_0$ and sets $H_1(ID, R, P_{pub}) = h_1, H_0(ID, Y_{ID}, mp, R) = h_0, X = Q$. Then it publishes the public key $PK_{ID_i} =< R, X, Y, \sigma >$ and inserts $< ID*, d, r, y_{ID}, 0, h_1, \sigma, h_0, R, X, Y, tag = 0 >$ into the table L_u.
  - If $ID_i \neq ID^*$, C_2 randomly selects $r, x_{ID}, y_{ID}, h_1, h_0 \in Z_q$, calculates $R = rP, Y = yP, X = xP, d_{ID} = r + sH_1, \sigma = y_{ID} + d_{ID}H_0$ and sets $H_1(ID, R, P_{pub}) = h_1, H_0(ID, Y, P_{pub}, R) = h_0$. Then it publishes the public key $PK_{ID_i} =< R, X, Y, \sigma >$ and inserts $< ID_i, d, x, y, h_1, \sigma, h_0, X, Y, R, tag = 0 >$ into the table L_u.
- $PartialPrivateKeyExtract(ID_i)$. Owning the *msk*, A_2 can arbitrarily finish this query for any ID_i.
- $SecretValueExtract(ID_i)$. Suppose it queries $ExtractSecretValue(ID_i)$ for at most q_{sv} times.
  - If $ID_i = ID*$, abort the game.
  - Otherwise, C_2 searches the table L_u for ID_i.If ID_i is found, it returns x directly. Otherwise, it calls $CreateUser(ID_i)$ and returns x.
- $ReplacePublicKey(ID_i, PK')$. Suppose it queries $ReplacePublicKey(ID_i, PK')$ for at most q_{rp} times.
  - If $ID_i = ID*$, abort the game.
  - Otherwise, C_2 searches the table L_u for ID_i. If ID_i is found, it replaces $< R, Y, X, \sigma >$ with PK'. Otherwise, C_2 calls $CreateUser(ID_i)$, replaces $< R, Y, X, \sigma >$ with PK' and sets $tag = 1$.
- $SuperSign(ID_i, m)$.
  - If $ID = ID^*$ or $tag = 1$, C_2 randomly selects $\tau, h_3, h_4, h_2 \in Z_q$ and calculates $T = (\tau P - h_3X - h_4R - h_4h_1P_{pub})h_2^{-1}$. Then C_2 sets $h_2 = H_2(ID_i, m, PK, T)$, $h_3 = H_3(ID_i, m, PK, T)$, $h_4 = H_4(ID_i, m, T, PK, P_{pub})$ in L_i. $< T, \tau >$ is a valid signature and note that C_2 does not need to know x.
  - If $ID \neq ID^*$ and $tag = 0$, C_2 searches the table L_u to find ID_i. If ID_i is found, C_2 knows $< d, x >$. Otherwise, C_2 calls $CreateUser(ID_i)$ and gets $< d, x >$ for ID_i. Then C_2 randomly selects $t, h_2, h_3, h_4 \in Z_q$ and sets $h_2 = H_2(ID_i, m, PK, T)$, $h_3 = H_3(ID_i, m, PK, T)$, $h_4 = H_4(ID_i, m, T, PK, P_{pub})$ in L_i. Finally C_2 calculates $\tau = h_2t + h_3x + h_4d$. $< T, \tau >$ is valid signature.

- *Forgery*. In the end, A_2 outputs $< T, \tau, m, ID >$. If $ID \neq ID*$, aborts. Otherwise, C_2 searches the table L_u to find ID and verifies the signature as follows:

$$h_1 = H_1(ID, R, P_{pub}) \tag{22}$$
$$h_2 = H_2(ID, m, PK, T) \tag{23}$$
$$h_3 = H_3(ID, m, PK, T) \tag{24}$$
$$h_4 = H_4(ID, m, T, PK, P_{pub}) \tag{25}$$
$$\tau P = h_2 T + h_3 X + h_4(R + h_0 P_{pub}) \tag{26}$$

Use forking lemma on H_3 to get a new $< T, \tau' >$ so that $\tau' = t \cdot h_2 + x \cdot h_3' + d_{ID} \cdot h_4$. Then calculate $x = \frac{\tau - \tau'}{h_3 - h_3'}$ is the solution to the ECDLP.

C_2 will solve the ECDLP if the following events occur:

1. ϵ_1: C_2 never aborts in the $GameI$,
2. ϵ_2: A_2 generates a valid forgery $< T, \tau, m, ID >$,
3. ϵ_3: In the forgery, $ID = ID^*$

So the probability of C_2 is $Pr[\epsilon_1 \wedge \epsilon_2 \wedge \epsilon_3] = Pr[\epsilon_1] \cdot Pr[\epsilon_2|\epsilon_1] \cdot Pr[\epsilon_3|\epsilon_1 \wedge \epsilon_2]$.

C_2 will abort in the $GameII$ if A_2 extracts the secret value or replaces the public key for the user ID^*. So $Pr[\epsilon_1] = (1 - 1/q_u)^{q_{sv}}(1 - 1/q_u)^{q_{rp}}$. If C_2 does not abort in the $GameII$, A_2 generates a valid forgery with ϵ. So $Pr[\epsilon_2|\epsilon_1] = \epsilon$. As the ID^* is selected randomly, $Pr[\epsilon_3|\epsilon_1 \wedge \epsilon_2] = 1/q_u$. So the probability is $\epsilon' = Pr[\epsilon_1 \wedge \epsilon_2 \wedge \epsilon_3] = Pr[\epsilon_1] \cdot Pr[\epsilon_2|\epsilon_1] \cdot Pr[\epsilon_3|\epsilon_1 \wedge \epsilon_2] = (1 - 1/q_u)^{q_{ppk}+q_{rp}} \cdot 1/q_u \epsilon$.

5. Efficiency Analysis

We analyze the efficiency and security of our CLS scheme and compare it with a series of schemes. Among these schemes, Huang et al. [10] designed a secure CLS scheme against super adversaries but relies on pairing. All other solutions do not require pairing and can not be proven to be safe against the super adversary. We conduct simulation experiments in the environment in Table 1 and choose a type-D pairing which is discovered by [27] and constructed on the curve $y^2 = x^3 + ax + b$ over the field F_q for a 160-bit prime q. So the length of a point x-coordinate in G_1 is roughly the same as 160-bit. The embedding degree is 6 so that the size of finite field in G_2 and G_t is 960-bit. The notations and time of different operations are shown in Table 2. The theoretical analysis of all schemes is shown in Table 3. Here $|G_1|, |G_2|$ and $|Z_q|$ denote the element size in G_1, G_2 and Z_q. To make Table 3 clearer, we ignore the insignificant time of A_1, M_t, I_q, A_q and M_q. The time of different schemes is shown in the Figure 1.

It has been observed that several secure certificateless signature schemes have been introduced without utilizing pairing, yet none of them were able to be proven secure against super adversaries. Some of the schemes, which are based on the Schnorr signature, are unable to respond to the super adversary's query when requesting specific private keys after the replacement of the public key. Our proposed solution not only attains security against super adversaries but also rectifies this minor issue, all the while maintaining a reasonable level of efficiency in signing and verifying. While Huang's scheme also achieves security against super adversaries, it relies on pairing operations, leading to increased computational time and signature size compared to our scheme. Consequently, our scheme effectively enhances both security and efficiency, while also addressing a slight deficiency in the security model.

Table 1. Experiment Environment.

CPU	OS	RAM	Compiler&Library
Inter i7-12700 @4.9 GHz	Ubuntu 20.04.1	32GB DDR5	PBC 0.5.14 & GCC 9.4.0

Table 2. Notation and time of the group operation.

Notation	Operation	Time (ms)
A_1	a point addition in G_1	0.0029
M_1	a scalar multiplication in G_1	0.3552
A_2	a point addition in G_2	0.0145
M_2	a scalar multiplication in G_2	2.8250
M_t	a multiplication in G_t	0.0045
Ex_t	a exponential operation in G_t	0.6497
P	a pairing operation : $G_1 \times G_2 \rightarrow G_t$	2.2532
I_q	a inversion operation in Z_q	0.0028
A_q	a addition in Z_q	0.0007
M_q	a multiplication in Z_q1	0.0006

Table 3. Theoretical Analysis.

Scheme	Sign	Verify	PPK	∣Sign∣	∣PK∣	∣PPK∣	Security
[10]	$M_2 + P + 2M_1$	$2M_2 + A_2 + 2P + Ex_t$	M_1	$\|G_1\| + 2\|Zq\|$	$\|G_2\|$	$\|G_1\|$	Super typeI&II
[26]	M_1	$4M_1$	M_1	$\|G_1\| + \|Zq\|$	$2\|G_1\|$	$\|G_1\| + \|Zq\|$	Strong typeI&II
[24]	M_1	$3M_1$	$2M_1$	$\|G_1\| + \|Zq\|$	$\|G_1\|$	$\|G_1\| + \|Zq\|$	Insecure
[28]	M_1	$3M_1$	M_1	$\|G_1\| + \|Zq\|$	$\|G_1\|$	$\|G_1\| + \|Zq\|$	Insecure
[17]	$2M_1 + M_2$	$2Ex_t + P$	M_1	$\|G_2\| + \|G_1\|$	$\|G_2\| + \|G_1\|$	$2\|G_1\|$	Insecure
[16]	M_1	$3M_1$	M_1	$\|G_1\| + \|Zq\|$	$2\|G_1\|$	$\|G_1\| + \|Zq\|$	Insecure
[25]	M_1	$4M_1$	M_1	$\|G_1\| + \|Zq\|$	$2\|G_1\|$	$\|G_1\| + \|Zq\|$	Strong typeI&II
Ours	M_1	$5M_1$	$2M_1$	$\|G_1\| + \|Zq\|$	$3\|G_1\| + \|Zq\|$	$2\|G_1\| + 2\|Zq\|$	Super typeI&II

* The Sign, Verify and PPK denote the operations in *Sign*,*Verify* and *PartialPrivateKey* algorithms. ∣Sign∣, ∣PK∣, and ∣PPK∣ represent the length of the signature, public key, and partial private key. The Security represents the level of adversary that these schemes can resist.

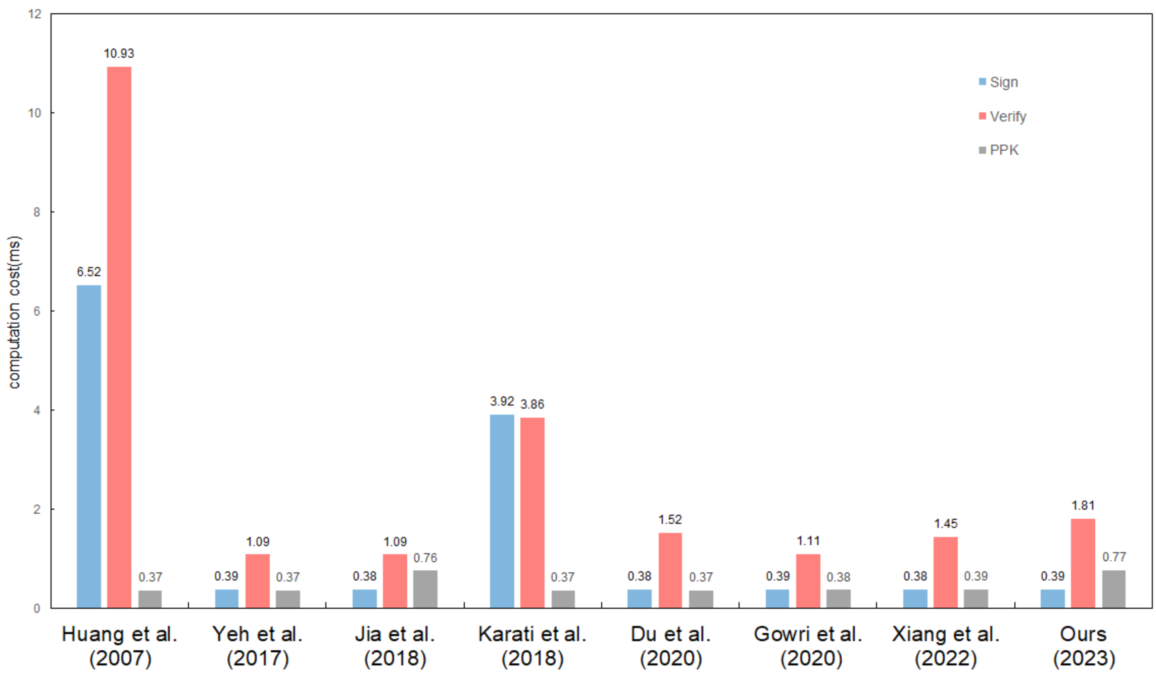

Figure 1. The time of *Sign*,*Verify* and *PartialPrivateKey* algorithms [10,16,17,24–26,28].

6. Conclusions

We find that existing secure CLS schemes against super adversaries often require expensive pairing operations, making them unsuitable for lightweight equipment. Some pairing-free schemes are unable to resist super adversaries and suffer from the issue where the challenger cannot answer partial private key inquiries after replacing the public key. To address these limitations, we have developed a secure CLS scheme against super adversaries without relying on pairing operations, and we have provided comprehensive proof of its security. Experimental testing has demonstrated that our scheme exhibits superior computational efficiency and a smaller signature size compared to schemes offering similar security guarantees.

Author Contributions: Conceptualization, G.W. (Ge Wu); Validation, L.C.; Formal analysis, J.H.; Resources, L.C.; Writing—original draft, G.W. (Guilin Wang); Writing—review & editing, H.S.; Supervision, G.W. (Ge Wu); Project administration, G.W. (Ge Wu). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 62002058), Natural Science Foundation of Jiangsu Province (No. BK20200391), and Fundamental Research Funds for the Central Universities (No. 2242021R40011).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We would like to thank the anonymous reviewers of this paper for their valuable comments and suggestions.

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

References

1. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In Proceedings of the Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, CA, USA, 19–22 August 1984; Blakley, G.R., Chaum, D., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1984; Volume 196, pp. 47–53. [CrossRef]
2. Al-Riyami, S.S.; Paterson, K.G. Certificateless Public Key Cryptography. In Proceedings of the Advances in Cryptology—ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 30 November–4 December 2003; Laih, C., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2894, pp. 452–473. [CrossRef]
3. Boneh, D.; Franklin, M.K. Identity-Based Encryption from the Weil Pairing. In Proceedings of the Advances in Cryptology—CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2001; Kilian, J., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2139, pp. 213–229. [CrossRef]
4. Huang, X.; Susilo, W.; Mu, Y.; Zhang, F. On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In Proceedings of the Cryptology and Network Security, 4th International Conference, CANS 2005, Xiamen, China, 14–16 December 2005; Desmedt, Y., Wang, H., Mu, Y., Li, Y., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3810, pp. 13–25. [CrossRef]
5. Yum, D.H.; Lee, P.J. Generic Construction of Certificateless Signature. In Proceedings of the Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, 13–15 July 2004; Wang, H., Pieprzyk, J., Varadharajan, V., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3108, pp. 200–211. [CrossRef]
6. Yum, D.H.; Lee, P.J. Generic Construction of Certificateless Encryption. In Proceedings of the Computational Science and Its Applications—ICCSA 2004, International Conference, Assisi, Italy, 14–17 May 2004; Proceedings, Part I; Laganà, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Vol. 3043, pp. 802–811. [CrossRef]
7. Hu, B.C.; Wong, D.S.; Zhang, Z.; Deng, X. Key Replacement Attack Against a Generic Construction of Certificateless Signature. In Proceedings of the Information Security and Privacy, 11th Australasian Conference, ACISP 2006, Melbourne, Australia, 3–5 July 2006; Batten, L.M.; Safavi-Naini, R., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4058, pp. 235–246. [CrossRef]
8. Libert, B.; Quisquater, J. On Constructing Certificateless Cryptosystems from Identity Based Encryption. In Proceedings of the Public Key Cryptography—PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, 24–26 April 2006; Yung, M., Dodis, Y., Kiayias, A., Malkin, T., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3958, pp. 474–490. [CrossRef]

9. Au, M.H.; Chen, J.; Liu, J.K.; Mu, Y.; Wong, D.S.; Yang, G. Malicious KGC Attacks in Certificateless Cryptography. *IACR Cryptol. Eprint Arch.* **2006**, 255.
10. Huang, X.; Mu, Y.; Susilo, W.; Wong, D.S.; Wu, W. Certificateless signature revisited. In Proceedings of the Information Security and Privacy: 12th Australasian Conference, ACISP 2007, Townsville, Australia, 2–4 July 2007; Proceedings 12; Springer: Berlin/Heidelberg, Germany, 2007; pp. 308–322.
11. Baek, J.; Safavi-Naini, R.; Susilo, W. Certificateless Public Key Encryption Without Pairing. In Proceedings of the Information Security, 8th International Conference, ISC 2005, Singapore, 20–23 September 2005; Zhou, J.; López, J.; Deng, R.H.; Bao, F., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3650, pp. 134–148. [CrossRef]
12. Schnorr, C.P. Efficient identification and signatures for smart cards. In Proceedings of the Advances in Cryptology—CRYPTO'89 Proceedings 9, Santa Barbara, CA, USA, 11–15 August 1990; Springer: Berlin/Heidelberg, Germany, 1990; pp. 239–252.
13. Sun, Y.; Zhang, F.; Baek, J. Strongly Secure Certificateless Public Key Encryption Without Pairing. In Proceedings of the Cryptology and Network Security, 6th International Conference, CANS 2007, Singapore, 8–10 December 2007; Lecture Notes in Computer Science; Bao, F., Ling, S., Okamoto, T., Wang, H., Xing, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4856, pp. 194–208. [CrossRef]
14. Zhang, J.; Mao, J. An efficient RSA-based certificateless signature scheme. *J. Syst. Softw.* **2012**, *85*, 638–642. [CrossRef]
15. Xu, Z.; Luo, M.; Khan, M.K.; Choo, K.R.; He, D. Analysis and Improvement of a Certificateless Signature Scheme for Resource-Constrained Scenarios. *IEEE Commun. Lett.* **2021**, *25*, 1074–1078. [CrossRef]
16. Gowri, T.; Rao, G.S.; Reddy, P.V.; Gayathri, N.B.; Reddy, D.V.R.K. Efficient Pairing-Free Certificateless Signature Scheme for Secure Communication in Resource-Constrained Devices. *IEEE Commun. Lett.* **2020**, *24*, 1641–1645. [CrossRef]
17. Karati, A.; Islam, S.H.; Biswas, G.P. A pairing-free and provably secure certificateless signature scheme. *Inf. Sci.* **2018**, *450*, 378–391. [CrossRef]
18. Zhang, B.; Zhu, T.; Hu, C.; Zhao, C. Cryptanalysis of a Lightweight Certificateless Signature Scheme for IIOT Environments. *IEEE Access* **2018**, *6*, 73885–73894. [CrossRef]
19. Wang, L.; Chen, K.; Long, Y.; Wang, H. An efficient pairing-free certificateless signature scheme for resource-limited systems. *Sci. China Inf. Sci.* **2017**, *60*, 119102. [CrossRef]
20. Gong, P.; Li, P. Further improvement of a certificateless signature scheme without pairing. *Int. J. Commun. Syst.* **2014**, *27*, 2083–2091. [CrossRef]
21. Wang, L.; Chen, K.; Long, Y.; Mao, X.; Wang, H. A Modified Efficient Certificateless Signature Scheme without Bilinear Pairings. In Proceedings of the 2015 International Conference on Intelligent Networking and Collaborative Systems, INCoS 2015, Taipei, Taiwan, 2–4 September 2015; Xhafa, F., Barolli, L., Eds.; IEEE: Piscataway Township, NJ, USA, 2015; pp. 82–85. [CrossRef]
22. Yeh, K.; Tsai, K.; Kuo, R.; Wu, T. Robust Certificateless Signature Scheme without Bilinear Pairings. In Proceedings of the 2013 International Conference on IT Convergence and Security, ICITCS 2013, Macau, China, 16–18 December 2013; IEEE Computer Society: Piscataway Township, NJ, USA, 2013; pp. 1–4. [CrossRef]
23. Yeh, K.; Tsai, K.; Fan, C. An efficient certificateless signature scheme without bilinear pairings. *Multim. Tools Appl.* **2015**, *74*, 6519–6530. [CrossRef]
24. Jia, X.; He, D.; Liu, Q.; Choo, K.R. An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment. *Hoc Netw.* **2018**, *71*, 78–87. [CrossRef]
25. Du, H.; Wen, Q.; Zhang, S.; Gao, M. A new provably secure certificateless signature scheme for Internet of Things. *Hoc Netw.* **2020**, *100*, 102074. [CrossRef]
26. Xiang, D.; Li, X.; Gao, J.; Zhang, X. A secure and efficient certificateless signature scheme for Internet of Things. *Hoc Netw.* **2022**, *124*, 102702. [CrossRef]
27. Member, A.M.; Nakabayashi, M.; Nonmembers, S.T. New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. *Tech. Rep. Ieice Isec* **2001**, *100*, 1234–1243.
28. Yeh, K.; Su, C.; Choo, K.R.; Chiu, W. A Novel Certificateless Signature Scheme for Smart Objects in the Internet-of-Things. *Sensors* **2017**, *17*, 1001. [CrossRef] [PubMed]