electronics

MDPI

Article

Improving Throughput of Mobile Sensors via Certificateless Signature Supporting Batch Verification

Chuan He[1,2,*], Bo Zhang [1,2], Liang Zhang [3], Zesheng Xi [1,2], Yuan Fang [3] and Yunfan Wang [1,2]

[1] State Grid Smart Grid Research Institute Co., Ltd., Beijing 102200, China; zhangbo@geiri.sgcc.com.cn (B.Z.); xizesheng@geiri.sgcc.com.cn (Z.X.); wangyunfan@geiri.sgcc.com.cn (Y.W.)
[2] State Grid Key Laboratory of Information & Network, Nanjing 211100, China
[3] State Grid Anhui Electric Power Co., Ltd., Information & Telecommunication Branch, Hefei 230061, China; zhangliang@geiri.sgcc.com.cn (L.Z.); fangxuan@geiri.sgcc.com.cn (Y.F.)
* Correspondence: hechuan@geiri.sgcc.com.cn

Abstract: Mobile sensors enjoy the advantages of easy installation and low consumption, which have been widely adopted in many information systems. In those systems where data are generated rapidly, the throughput of the sensors is one of the most fundamental factors that determine the system functionality. For example, to guarantee data integrity, digital signature techniques can be applied. In many practical scenarios, such as the smart grid system, data are generated rapidly and, hence, the signature together with the data must also be transmitted and verified in time. This requires the mobile sensors to support a high-throughput data processing ability. In this setting, how to achieve efficient signature schemes supporting batch verification must be considered. Many signatures, such as the original national cryptographic standard, namely, the SM2 algorithm, do not support batch verification and are in a public-key infrastructure setting. In this paper, we propose a SM2-based certificateless signature scheme with batch verification, which is suitable for the aforementioned environment. The scheme extends the Chinese cryptographic standard SM2 algorithm to the certificateless setting and multiple signatures can be verified simultaneously. Another advantage of this scheme is that its signing phase does not involve any pairing operation. The verification phase only requires a constant pairing operation, which is not related to the number of signatures to be verified. The construction is generic and can be instantiated using any traditional signature scheme.

Keywords: mobile sensor; data integrity; certificateless signature; batch verification; pairing free

Citation: He, C.; Zhang, B.; Zhang, L.; Xi, Z.; Fang, Y.; Wang, Y. Improving Throughput of Mobile Sensors via Certificateless Signature Supporting Batch Verification. *Electronics* **2023**, *12*, 4700. https://doi.org/10.3390/electronics12224700

Academic Editor: Antonio Brogi

Received: 12 September 2023
Revised: 30 September 2023
Accepted: 13 November 2023
Published: 19 November 2023

1. Introduction

A smart grid is a new type of network based on multiple power devices, which integrates a power data transmission function and power transmission function. These terminal devices often have limited hardware resources while generating mass data. How to increase throughput while ensuring data validation with limited hardware resources is an urgent problem that smart grids face. The validation of power data ensures, on the one hand, the integrity of data. Integrity directly affects the statistical, regulatory, and distribution of the electricity resources. In addition, considering the uncertain work environment of these devices, it is inevitable that sensor equipment malfunctions lead to data anomalies. At this point, it is necessary to quickly locate the device sending the abnormal data. Therefore, tracing the source of the error data is also a function that needs to be implemented. Digital signature technology can protect data from tampering and repudiation, which is sufficient for the data requirements of smart grids.

The earliest digital signature schemes mostly relied on the public-key infrastructure (PKI) setting, which bound user identity and public key information through the issuance of certificates by a CA (a certificate authority). However, this certificate can bring complex certificate management issues to the system and have high requirements for communication

bandwidth and storage resources for devices so that it is unsuitable for grid devices. To address this issue, cryptography researchers have proposed identity-based cryptosystems (IBCs) [1,2] that directly use some identifiable information as public keys, such as phone numbers and email addresses. However, the private key comes entirely from the private key generator (PKG) in IBCs. This centralized trust dependency brings serious key escrow problems. Once the PKG center is attacked, it brings security issues to all subordinate devices. The certificateless public-key cryptosystem (CLPKC) inherits the advantages of the previous systems. In the CLPKC, there is neither certificates nor the problem of key escrow. Therefore, the CLPKC is more suitable for smart grid equipment in resource constrained scenarios.

In detail, many smart grid sensors are embedded with sequential numbers in the equipment. The sequential numbers can be treated as identities of the users in the systems and can be used to verify signatures or trace the origin of the message. On the other hand, the identity-based setting is not enough for the smart grid environment since there are large numbers of nodes in the system and it is not easy to select a widely adopted and fully trusted third party as the PKG. Therefore, certificateless cryptography is a prominent candidate for such a system. As is known to all, the pairing operations are comparatively complex and take much more time for computation than other group structures like the elliptic curve setting. However, many certificateless cryptographic schemes share similar algebraic structures to the identity-based constructions and are built upon pairing-friendly groups. Pairing-free certificateless signature schemes have not been widely developed, especially the scheme derived from the national cryptographic standard. To sum up, signature schemes that satisfy the pairing-free, certificateless setting, based on the published standard have many applications in smart grid system. Unfortunately, few constructions have been studied in the literature.

In addition, due to the high real-time requirements of data in the power system, the signature algorithm used must be able to calculate quickly [3]. The terminal node may generate electricity data at any time, and the server will receive multiple data streams from multiple nodes at the same time. The server must be able to quickly process signature verification, which requires the signature algorithm to preferably support batch verification. In these systems, both efficiency [4] and privacy-preserving properties [5] need to be taken into consideration. Signatures that support batch verification can solve this problem. Namely, the signatures on the data collected from various sensors and other equipment can be aggregated in a certain node before being transmitted to the center and can later be verified together. The framework is shown in Figure 1.

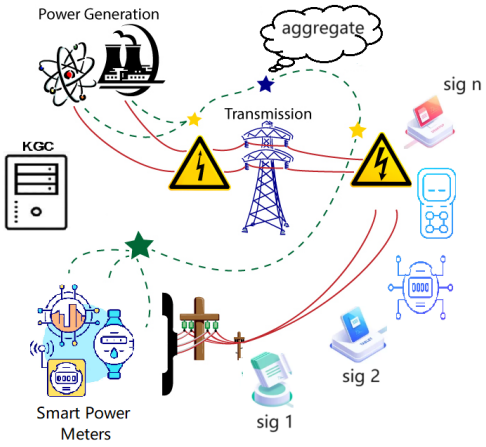

Figure 1. Framework of certificateless signature with batch verification.

1.1. Related Work

The CLPKC was first proposed by Al-Riyami and Paterson [6] to deal with the key escrow problem in the identity-based encryption (IBE) system [2]. They provided the construction of three schemes including encryption, signature, and key agreement. In

addition, two basic adversary models in the certificateless cryptosystem were identified, namely, Type I adversary and Type II adversary. Due to the excellent properties of no-certificate, many researchers were attracted and many follow-up work was proposed. Yum and Lee summarized a general secure construction method of the certificateless signature (CLS) scheme [7] and certificateless encryption (CLE) scheme [8]. However, later these constructions were proved to be unsafe by Hu et al. [9] and Libert et al. [10]. In 2005, Huang et al. [11] proved that there was a security risk in the original Al-Riyami and Paterson scheme [6]. Au et al. [12] re-examined the security model of CLPKC and proposed the concept of a new adversary model called the malicious key generation center (KGC). Huang et al. [13] further subdivided each type opponent into three levels based on their attack capabilities and provided a super secure certificateless signature scheme. Among the known models, security against the super-type adversary achieves the most secure level. Nevertheless, the signature length was slightly long and contained three group elements. In recent years, many new shorter certificateless signature schemes and certificateless aggregate signatures [14,15] were proposed. There are also some schemes that have been proven to be insecure. For example, Shim [16] analyzed five recent articles and found that they can all be forged by adversaries. Therefore, how to construct secure certificateless signatures still requires a very rigorous approach. For a comparative survey of certificateless signature, ref. [17] is a good reference for the related studies until 2022. Two other related but earlier surveys can be found in [18,19].

In addition to solving the key escrow problem, compared to IBCs, another major advantage of certificateless cryptosystem is that they can be implemented without pairing. Baek et al. [20] explored the first certificateless encryption scheme without pairing using the Schnorr signature. However, Sun et al. [21] showed that the scheme in [20] did not consider public key attacks. They fixed the problem using a new scheme with a more stringent security model. The certificateless signature scheme without pairing was finished by He et al. [22] in 2010. For the IoT scenario, Gong et al. [23] and Yang et al. [24] designed a certificateless aggregation signature without pairing and Dai et al. [25] proposed a certificateless aggregation signcryption without pairing. Moreover, many certificateless schemes based on other PKI signatures have been studied. Using the RSA signature, Zhang et al. [26] also constructed a CLS scheme. Another study point is constructing CLS schemes based on already-published cryptographic standards. In 2022, Tang et al. [27] proposed a CLS scheme (in Chinese) based on the Chinese national cryptographic standard. The scheme is built upon the identity-based standard, namely, the SM9 (SM stands for the Chinese pinyi "shangmi", which means a commercial cryptography application) algorithm. As a result, it must rely on the pairing operation. Recently, He et al. [28] proposed a new CLS scheme using the SM2 algorithm without pairing. But their scheme requires zero-knowledge proof to verify the user public keys and how to support batch verification remains unknown. For batch verification, the certificateless aggregate signature (CLAS) [15,29] technique can be considered.

1.2. Motivation and Contributions

From the above analysis, we can see the enormous advantages of the certificateless cryptosystem and the feasibility of constructing a certificateless scheme based on the traditional signature scheme. However, current research is mostly limited to the implementation of the most basic signature schemes, while some signature algorithms with special functions have not yet emerged. For example, in systems with high throughput and low latency requirements, batch verification of signatures is also a crucial attribute that directly affects the availability of the entire system. Currently, there is no batch verifiable certificateless signature algorithm based on the national security algorithm. The primary contributions of this study include:

1. We propose a certificateless signature algorithm with batch verification based on the Chinese national cryptographic standards, in particular with the SM2 algorithm;
2. Our scheme supports batch verification of multiple signatures, thereby accelerating the algorithm in high throughput scenarios.

1.3. Technical Overview

From the above analysis, we can see that current studies on certificateless signature (CLS) schemes encounter the limitations of either relying on pairing operations like the scheme [27] built on the SM9 algorithm, or the underlying scheme not being selected as the cryptographic standard. The scheme proposed by He et al. [28] is extended from the SM2 algorithm and does not involve any pairing operation. However, it does not support batch verification. We first review the basic idea of He et al.'s construction. The core technical transformation from a traditional signature scheme to a certificate-based signature scheme is show in Figure 2.

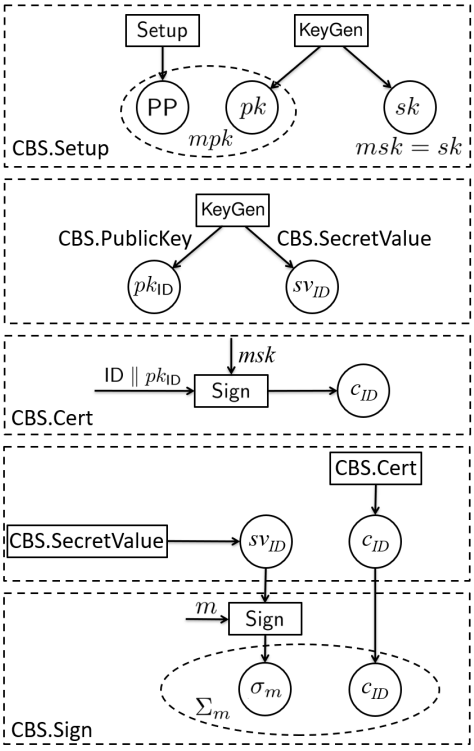

Figure 2. Transformation from traditional signature to certificate-based signature.

A certificateless signature is similar to a certificate-based signature. The main difference is the secret value setting phase. In the CLS scheme, it is not necessary for the user to select a secret value and compute the public key first, before transmitting the public key to the key generation center (KGC) to obtain the partial private key. This means that the user can apply the partial private key from the KGC first; then, generate the secret value and compute the public key later. During the key extraction phase, the user's public key may not yet be generated and, hence, cannot directly use the above transformation. To solve this issue, the user's public key contain two parts: one part is from the key extraction phase, which is similar to the certificate-based setting; the other part is generated by the user itself. These two parts are independently generated but must be used together to sign a message. This paves the way for transforming a traditional signature scheme into a CLS scheme.

As for the zero-knowledge proof part, we use the property of bilinear pairing to replace the complex proof process. Even though this brings the pairing operation into the scheme, it only appears in the verification phase and the signing phase does not involve any pairing operation. For verification, since our scheme supports batch verification, multiple signatures can be verified simultaneously and the number of pairing operations is constant. This means that the additional time cost caused by the pairing operations is a fixed value and, hence, it will not incur too much computational cost during batch verification. The details of batch verification are depicted in Section 4.2.

1.4. Organization

The structure of this article is organized as follows. In Section 2, two preliminaries will be briefly introduced, including CLS and bilinear pairing, and a SM2-based CLS scheme will be reviewed. In Section 3, a new signature scheme with batch verification will be proposed, and in Section 4, the performance of these schemes will be evaluated through simulation experiments. Finally, a conclusion of the entire article is provided in Section 5.

2. Preliminaries

We will describe the definition of two preliminaries, including bilinear pairing and the certificateless signature. We will also review a SM2-based CLS scheme.

2.1. Bilinear Pairing

For three cyclic groups G_1, G_2, G_T of a prime order q, a map $e : G_1 \times G_2 \to G_T$ is a bilinear pairing if and only if three properties hold:

- Computable: given any $g \in G_1, h \in G_2$, calculating $e(g, h) \in G_T$ is efficient;
- Bilinear: for $x, y \in Z_q$, the equation $e(g^x, h^y) = e(g, h)^{xy}$ always holds;
- Nondegenerate: if g is a generator of G_1 and h is a generator of G_2, $e(g, h)$ will also be a generator of G_T.

2.2. Certificateless Signature

The CLS scheme usually includes six algorithms:

- Setup (1^λ): The Setup algorithm is usually operated by the KGC to initialize the scheme, which receives a security parameter 1^λ. The system master public and secret key pair (mpk, msk) will be generated;
- KeyExt (mpk, msk, ID): The KeyExt algorithm is usually operated by the KGC, which receives the master key pair mpk, msk and a user identity ID. Finally, a partial private key d_{ID} is generated and transmitted to the user;
- SecretValue (mpk, ID): The SecretValue algorithm is usually completed by a user, which receives the master public key mpk and a user identity ID. Finally, a secret value sv_{ID} is generated and returned to the user;
- PublicKey $(mpk, \mathsf{ID}, sv_{\mathsf{ID}})$: The PublicKey algorithm is usually completed by a user, which receives the master public key mpk, a user identity ID, and a secret value sv_{ID}. A user public key pk_{ID} will be output;
- Sign $(mpk, d_{\mathsf{ID}}, sv_{\mathsf{ID}}, m)$: The Sign algorithm is usually completed by a user signing it. They receive the master public key mpk, a partial private key d_{ID}, a secret value sv_{ID}, and a message m. A signature σ_m on the message m will be output;
- Verify $(mpk, \mathsf{ID}, m, \sigma_m)$: The Verify algorithm is usually completed by a user verifying it. They receive the master public key mpk, a user identity ID, a public key pk_{ID}, a message m, and a signature σ_m. If the output is "1", it means the signature is legal; otherwise, the signature is illegal.

2.3. Review a CLS Scheme Based on SM2

The SM2-based CLS scheme designed by He et al. [28] is made up of six algorithms.

- Setup (1^λ): The Setup algorithm receives the security parameter 1^λ as input and uses the SM2 setup algorithm. It chooses an elliptic curve group $(\mathbb{G}, p, P)$ with parameters a, b and coordinates x_P, y_P. Then, it randomly picks $\alpha \in \mathbb{Z}_p$ and computes $P_{pub} = [\alpha]P$. It also selects a hash function H, such as the SM3 algorithm. Finally, the algorithm returns the master public and secret key pair as

$$mpk = (\mathbb{G}, p, P, P_{pub}, H),\ msk = \alpha.$$

- KeyExt (mpk, msk, ID): The KeyExt algorithm receives the master key pair (mpk, msk) and an identity ID as inputs. Firstly, it randomly selects $x \in \mathbb{Z}_p$ and computes

$ppk_{\mathsf{ID}} = [x]P$. Then, it concatenates the identity ID and the partial public key ppk_{ID}. Finally, it runs the SM2 signature algorithm to produce the partial private key.

1. Compute $e = H(\mathsf{ID} \parallel ppk_{\mathsf{ID}})$;
2. Pick $k \in \mathbb{Z}_p$ randomly and calculate $[k]P = (x_1, y_1), r = (e + x_1) \bmod p$;
3. Compute $s = ((1+\alpha)^{-1} \cdot (k - r \cdot \alpha)) \bmod p$.

The partial private key is $d_{\mathsf{ID}} = (r, s, x, ppk_{\mathsf{ID}})$;

- ScretValue (mpk, ID): The ScretValue algorithm receives the master public key mpk and an identity ID. Then, it runs the SM2 key generation algorithm. It randomly selects a $y \in \mathbb{Z}_p$ and sets $sv_{\mathsf{ID}} = (x, y)$ with the random value x received from the KGC. Next, it outputs the secret value sv_{ID};
- PublicKey $(mpk, \mathsf{ID}, sv_{\mathsf{ID}})$: The PublicKey algorithm receives the master public key mpk, an identity ID, and a secret value $sv_{\mathsf{ID}} = (x, y)$. Then, it computes $[y]P$ and generates a noninteractive zero-knowledge proof (NIZKP) π of holding the unique y with respect to $[y]P$. Next, it sets $pk_{\mathsf{ID}} = (ppk_{\mathsf{ID}}, [y]P, \pi)$ and outputs pk_{ID} as the public key;
- Sign $(mpk, d_{\mathsf{ID}}, sv_{\mathsf{ID}}, m)$: The Sign algorithm receives the master public key mpk, a partial private key $d_{\mathsf{ID}} = (r, s, x, ppk_{\mathsf{ID}})$, a secret value $sv_{\mathsf{ID}} = (x, y)$, and a message m. It first concatenates the identity ID and message m. Then, it computes $(x + y) \bmod p$ and runs the SM2 signing algorithm with $(x + y)$ to generate the part signature. In detail,

  1. Compute $e = H(\mathsf{ID} \parallel m)$;
  2. Pick $k \in \mathbb{Z}_p$ randomly and compute $[k]P = (x_1, y_1), r' = (e + x_1) \bmod p$;
  3. Compute $s' = ((1 + (x + y))^{-1} \cdot (k - r' \cdot (x + y))) \bmod p$.

  Next, it outputs the signature $\sigma_m = (r, s, r', s')$;
- Verify$(mpk, \mathsf{ID}, pk_{\mathsf{ID}}, m, \sigma_m)$: The Verify algorithm receives the master public key mpk, an identity ID, a public key $pk_{\mathsf{ID}} = (ppk_{\mathsf{ID}}, [y]P, \pi)$, a message m, and a signature $\sigma_m = (r, s, r', s')$. Then, it runs the SM2 algorithm to verify (r, s) and (r', s') and checks whether π is valid. In detail,

  1. Compute $e'_1 = H(\mathsf{ID} \parallel ppk_{\mathsf{ID}}), e'_2 = H(\mathsf{ID} \parallel m)$;
  2. Compute $t_1 = (r + s) \bmod p, t_2 = (r' + s') \bmod p$;
  3. Compute $[s]P + [t_1]P_{pub} = (x'_1, y'_1), [s']P + [t_2](ppk_{\mathsf{ID}} + [y]P) = (x'_2, y'_2)$;
  4. Compute $R = (e'_1 + x'_1), R' = (e'_2 + x'_2)$.

  If the proof π is valid and the equations $r = R, r' = R'$ hold, it outputs "1". Otherwise, it outputs "0".

3. A Certificateless Signature Scheme Supporting Batch Verification

3.1. Zero-Knowledge Proof with Pairing

In the above scheme, we need to provide a NIZKP of y in the user public key to avoid adversaries bypassing $[x]P$ by setting $[y]P$. However, zero-knowledge proof requires additional overhead and increases the length of the user public key. We provide an extension scheme that uses bilinear pairing tools to verify the binding relationship between $[x]P$ and $[y]P$. A user who verifies the signature can ensure that the signer knows the y corresponding to Y by calculating $e([x]P, [y]P) = e([xy]P, P)$. The extension scheme is depicted in the following.

3.2. Construction

Next, we describe our new certificateless signature scheme with batch verification based on SM2. Our scheme also consists of six algorithms.

- Setup (1^λ): The Setup algorithm receives a security parameter 1^λ. It generates an elliptic curve group $(\mathbb{G}, p, P)$ with parameters a, b and coordinates x_P, y_P. Then, it

picks $\alpha \in \mathbb{Z}_p$ randomly and sets $P_{pub} = [\alpha]P$. Next, it chooses a hash function H, such as the SM3 algorithm. Finally, it outputs the master key pair as

$$mpk = (\mathbb{G}, p, P, P_{pub}, H),\ msk = \alpha.$$

- KeyExt (mpk, msk, ID): The KeyExt algorithm receives the master key pair (mpk, msk) and an identity ID as inputs. It first picks $x \in \mathbb{Z}_p$ randomly and calculates $ppk_{\mathsf{ID}} = [x]P$. Then, it concatenates ID with ppk_{ID}. Next, it runs the SM2 algorithm to generate a partial private key.
  1. Compute $e = H(\mathsf{ID} \parallel ppk_{\mathsf{ID}})$;
  2. Pick $k \in \mathbb{Z}_p$ randomly and compute $[k]P = (x_1, y_1), r = (e + x_1) \bmod p$;
  3. Compute $s = ((1+\alpha)^{-1} \cdot (k - r \cdot \alpha)) \bmod p$.

  It transmits the partial private key $d_{\mathsf{ID}} = (r, s, x, ppk_{\mathsf{ID}})$ to the user safely;
- ScretValue (mpk, ID): The ScretValue algorithm receives the master public key mpk and an identity ID as inputs. Then, it runs the SM2 key generation algorithm. It selects $y \in \mathbb{Z}_p$ randomly and sets $sv_{\mathsf{ID}} = (x, y)$ with the random value x received from KGC. Next, it outputs the secret value sv_{ID};
- PublicKey $(mpk, \mathsf{ID}, sv_{\mathsf{ID}})$: The PublicKey algorithm receives the master public key mpk, a user identity ID, and a secret value $sv_{\mathsf{ID}} = (x, y)$ of the user as inputs. Then, it computes $[y]P$ and $[xy]P$. Next, it sets $pk_{\mathsf{ID}} = (ppk_{\mathsf{ID}}, [y]P, [xy]P)$ and produces the public key pk_{ID};
- Sign $(mpk, d_{\mathsf{ID}}, sv_{\mathsf{ID}}, m)$: The Sign algorithm inputs the master public key mpk, a user partial private key $d_{\mathsf{ID}} = (r, s, x, ppk_{\mathsf{ID}})$, a secret value $sv_{\mathsf{ID}} = (x, y)$, and a message m. It first concatenates the identity ID and the message m. Then, it computes $xy \bmod p$ and runs the SM2 signing algorithm with xy to generate the part signature. In detail,
  1. Compute $e = H(\mathsf{ID} \parallel m)$;
  2. Pick $k' \in \mathbb{Z}_p$ randomly and compute $[k']P = (x_2, y_2), r' = (e + x_2) \bmod p$;
  3. Compute $s' = ((1+xy)^{-1} \cdot (k' - r' \cdot xy) \bmod p$.

  Next, it outputs the signature $\sigma = (r, s, r', s')$;
- Verify $(mpk, \mathsf{ID}, pk_{\mathsf{ID}}, m, \sigma)$: The Verify algorithm receives the master public key mpk, an identity ID, public key $pk_{\mathsf{ID}} = ([x]P, [y]P, [xy]P)$, a message m, and a signature $\sigma = (r, s, r', s)$. It first checks if $e([x]P, [y]P) = e([xy]P, P)$ holds. Then, it runs the SM2 verification algorithm to check the validity of σ_{ID} and σ_m. In detail,
  1. Compute $e'_1 = H(\mathsf{ID} \parallel ppk_{\mathsf{ID}}), t_1 = (r + s) \bmod p, [s]P + [t_1]P_{pub} = (x'_1, y'_1)$, $R = (e'_1 + x'_1)$. Then check if the equations $R = r$ holds;
  2. Compute $e'_2 = H(\mathsf{ID} \parallel m), t_2 = (r' + s') \bmod p, [s']P + [t_2]([xy]P) = (x'_2, y'_2)$, $R' = (e'_2 + x'_2)$.Then check if the equations $R' = r'$ holds;
  3. Check if the equations $e([x]P, [y]P) = e([xy]P, P)$ holds

  If all three equations hold, it outputs "1". Otherwise, it outputs "0".

4. Performance Analyses

4.1. Computational Costs

The efficiency performance of the scheme was evaluated by comparing it with Huang's CLS [13] through simulation experiments. We use $T_{add}, T_P, T_{mul}, T_e$ to represent the time of a point addition, a pairing operation, a scalar multiplication in the elliptic curve group, and an exponential operation in the G_T group. G and Z_p represent the elliptic curve group and the group of integers that are modular to a prime number p without an explicit statement. The experimental environment and the results are shown as Table 1 and Table 2, respectively:

Table 1. Experimental environment.

CPU	OS	RAM	Compiler and Library
Intel i7-12700z	Ubuntu 14.04	32 GB DDR5 4800 MHz	GNU C/C++ & PBC 0.5.14

Table 2. Efficiency comparison of the CLS schemes.

Scheme	Signature Length	Sign Computation	Verify Computation	Sign Time (ms)	Verify Time (ms)
He	$4\|Z_p\|$	$3T_{add} + 3T_{mul}$	$6T_{mul} + 7T_{add}$	1.01	4.81
Huang	$1\|G\| + 2\|Z_p\|$	$3T_{mul} + T_P + T_e$	$2T_{mul} + 2T_P + T_e$	4.08	3.63
Our	$4\|Z_p\|$	$3T_{add} + 3T_{mul}$	$6T_{mul} + 7T_{add}$	0.99	5.26

In the above table, $|Z_p|$ and $|G|$ denote the binary length of an element in group Z_p and G, respectively.

4.2. Batch Verification

This scheme requires the pairing operations in the verification algorithm, which consumes a lot of resources. To accelerate the algorithm, we can batch process a large number of signatures from the same user. For example, when multiple signatures from the same user are received consecutively, the received r, s, X, Y must all be consistent. Therefore, the verification equations can be performed once. The following is a simplified validation algorithm:

Batch-Verify $(mpk, \mathsf{ID}, pk_{\mathsf{ID}}, \{m_1, m_2, \ldots, m_n\}, \{\sigma_1, \sigma_2, \ldots, \sigma_n\})$: The verification algorithm inputs the master public key mpk, a user's identity ID, a public key pk_{ID}, n messages $\{m_1, m_2, \ldots, m_n\}$, and n signatures $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$. The $\{\sigma_i\}_{i\in\{1\ldots n\}}$ is denoted as $\{r_i, s_i, r'_i, s'_i\}$ and the pk_{ID} is denoted as $\{[x]P, [y]P, [xy]P\}$.

1. Compute $e'_{i,1} = H(\mathsf{ID} \parallel ppk_{\mathsf{ID}})$, $t_{i,1} = (r_i + s_i) \bmod p$, $[s_i]P + [t_{i,1}]P_{pub} = (x'_{i,1}, y'_{i,1})$, $R_i = (e'_{i,1} + x'_{i,1})$. Then, check if the equations $R_i = r_i$ holds. For n signatures coming from the same user, the $\{r_i, s_i, [x]P\}$ are the same so that this step only needs to be calculated once for n signatures;
2. Compute $e'_{i,2} = H(\mathsf{ID} \parallel m_i)$, $t_{i,2} = (r'_i + s'_i) \bmod p$, $[s'_i]P + [t_{i,2}]([xy]P) = (x'_{i,2}, y'_{i,2})$, $R'_i = (e'_{i,2} + x'_{i,2})$. Then, check if the equation $R'_i = r'_i$ holds. This step must to be executed for each signature;
3. Check if the equation $e([x]P, [y]P) = e([xy]P, P)$ holds. This step only needs to be calculated once for n signatures.

If all three equations hold, it outputs "1". Otherwise, it outputs "0".

In this way, when verifying n signatures from the same user, step 1 and 3 only need to be performed once and step 2 needs to be performed n times. Thus, the expensive pairing operation only needs to be performed twice.

Multi-User-Batch-Verify (mpk, $\{ID_1, ID_2, \ldots, ID_m\}$, $\{pk_{ID_1}, pk_{ID_2}, \ldots, pk_{ID_m}\}$, $\{m_{1,1}, \ldots, m_{1,n}, m_{1,2}, \ldots, m_{m,n}\}$, $\{\sigma_{1,1}, \sigma_{1,2}, \ldots, \sigma_{1,n}, \ldots, \sigma_{m,n}\}$): The verification algorithm inputs the master public key mpk, m identities $\{\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_\mathsf{m}\}$, and m public keys $\{pk_{\mathsf{ID}_1}, pk_{\mathsf{ID}_2}, \ldots, pk_{\mathsf{ID}_\mathsf{m}}\}$. The $m_{i,j}$ and $\sigma_{i,j}$ denote the j-th message and signature for i-th user, respectively. The $\sigma_{i,j}$ is denoted as $(r_{i,j}, s_{i,j}, r'_{i,j}, s'_{i,j})$ and the $pk_{\mathsf{ID}_\mathsf{i}}$ is denoted as $([x_i]P, [y_i]P, [xy_i]P)$.

1. Compute $e'_{i,1} = H(\mathsf{ID}_\mathsf{i} \parallel [x_i]P)$, $t_{i,j,1} = (r_{i,j} + s_{i,j}) \bmod p$, $[s_{i,j}]P + [t_{i,j,1}]P_{pub} = (x'_{i,j,1}, y'_{i,j,1})$, $R_{i,j} = (e'_{i,1} + x'_{i,j,1})$. Then, check if the equation $R_{i,j} = r_{i,j}$ holds. This step needs to be calculated once for each user;
2. Compute $e'_{i,j,2} = H(\mathsf{ID}_\mathsf{i} \parallel m_{i,j})$, $t_{i,j,2} = (r'_{i,j} + s'_{i,j}) \bmod p$, $[s'_{i,j}]P + [t_{i,j,2}]([xy_i]P) = (x'_{i,j,2}, y'_{i,j,2})$, $R'_{i,j} = (e'_{i,j,2} + x'_{i,j,2})$. Then, check if the equation $R'_{i,j} = r'_{i,j}$ holds. This step must to be calculated for each signature;
3. For all m public keys, calculate $\pi_1 = \prod_{i=1,j=1,i!=j}^{m} e([x_i]P, [y_j]P) = \prod_{i=1}^{m}(\prod_{j=1,j!=i}^{m}(e([x_i]P, [y_j]P))$. This calculation can be completed by a third-party assistant and the results

can be sent to the user. Then, the user calculates $\pi_2 = e(\sum_{i=1}^{m}[x_i]P, \sum_{i=1}^{m}[y_i]P)$ and $\pi_3 = e(\sum_{i=1}^{m}[xy_i]P, P)$. Finally, check if the equation $\frac{\pi_2}{\pi_1} = \pi_3$ holds.

If all equations in the three steps hold, it outputs "1". Otherwise, it outputs "0".

In this way, when verifying the signatures from m users, the pairing operation can be completed twice locally rather than increasing with the number of users.

5. Conclusions and Future Work

To accelerate the verification algorithm, we extended the CLS scheme proposed by He et al. [28] and accelerated the algorithm execution through batch verification. The proposed scheme is still based on the Chinese national cryptographic standard (SM2) algorithm and no pairing operation is required during the signing process. This guarantees both efficiency and the requirement of using the standard cryptographic algorithm. In addition, the noninteractive zero-knowledge proof (NIZKP) of the signature is replaced by verifying an equation. This improvement provides efficient batch verification for multiple signatures. The number of pairing operations is constant regardless of the amount of signatures.

In this paper, we propose a basic certificateless signature scheme derived from the SM2 algorithm without resorting to the use of pairing operations. Signing or verifying a single signature does not involve any pairing operation. In addition, we further show how to improve the scheme to support batch verification. Nevertheless, the verification of multiple signatures requires a constant number of pairing operations. Despite the fact that the number is constant and is independent from the number of signatures in a batch verification, how to achieve a fully pairing-free SM2-based certificateless signature scheme that supports batch verification is worth studying. In addition, the security analysis is based on the random oracle model, which treats the hash function as an oracle. How to construct schemes without random oracles would also improve the security to a greater extent.

Author Contributions: Conceptualization, C.H.; formal analysis, B.Z.; methodology, C.H.; software, Z.X.; Writing—original draft, Y.W.; Writing—review and editing, L.Z. and Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the science and technology project of State Grid Corporation of China "Research on lightweight cryptographic technology for power IOT terminal" (Grand No. 5700-202255186A-1-1-ZN).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the related research data are available upon authors request.

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 *Advances in Cryptology: Proceedings of CRYPTO 84 4;* Lecture Notes in Computer Science; Blakley, G.R., Chaum, D., Eds.; Springer: Berlin/Heidelberg, Germany, 1984; Volume 196, pp. 47–53. [CrossRef]
2. 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; Proceedings; Lecture Notes in Computer Science; Kilian, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2139, pp. 213–229. [CrossRef]
3. Zhang, C.; Zhao, M.; Zhu, L.; Zhang, W.; Wu, T.; Ni, J. FRUIT: A Blockchain-Based Efficient and Privacy-Preserving Quality-Aware Incentive Scheme. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3343–3357. [CrossRef]
4. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving Efficient and Privacy-Preserving Neural Network Training and Prediction in Cloud Environments. *IEEE Trans. Dependable Secur. Comput.* **2022**, *early access*. [CrossRef]
5. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving Privacy-Preserving and Verifiable Support Vector Machine Training in the Cloud. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3476–3491. [CrossRef]

6. 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; Lecture Notes in Computer Science; Laih, C., Ed.; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2894, pp. 452–473. [CrossRef]
7. 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; Lecture Notes in Computer Science; Wang, H., Pieprzyk, J., Varadharajan, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3108, pp. 200–211. [CrossRef]
8. 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; Lecture Notes in Computer Science; Laganà, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3043, pp. 802–811. [CrossRef]
9. 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; Lecture Notes in Computer Science; Batten, L.M., Safavi-Naini, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4058, pp. 235–246. [CrossRef]
10. 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; Lecture Notes in Computer Science. Yung, M., Dodis, Y., Kiayias, A., Malkin, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3958, pp. 474–490. [CrossRef]
11. 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; Lecture Notes in Computer Science; Desmedt, Y., Wang, H., Mu, Y., Li, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3810, pp. 13–25. [CrossRef]
12. 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.
13. Huang, X.; Mu, Y.; Susilo, W.; Wong, D.S.; Wu, W. Certificateless Signatures: New Schemes and Security Models. *Comput. J.* **2012**, *55*, 457–474. [CrossRef]
14. Zhang, F.; Shen, L.; Wu, G. Notes on the security of certificateless aggregate signature schemes. *Inf. Sci.* **2014**, *287*, 32–37. [CrossRef]
15. Wu, G.; Zhang, F.; Shen, L.; Guo, F.; Susilo, W. Certificateless aggregate signature scheme secure against fully chosen-key attacks. *Inf. Sci.* **2020**, *514*, 288–301. [CrossRef]
16. Shim, K. Design Principles of Secure Certificateless Signature and Aggregate Signature Schemes for IoT Environments. *IEEE Access* **2022**, *10*, 124848–124857. [CrossRef]
17. Hussain, S.; Ullah, S.S.; Ali, I.; Xie, J.; Inukollu, V.N. Certificateless signature schemes in Industrial Internet of Things: A comparative survey. *Comput. Commun.* **2022**, *181*, 116–131. [CrossRef]
18. Housani, H.A.; Baek, J.; Yeun, C.Y. Survey on certificateless public key cryptography. In Proceedings of the 6th International Conference for Internet Technology and Secured Transactions, ICITST 2011, Abu Dhabi, United Arab Emirates, 11–14 December 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 53–58.
19. Chen, Y.; Tso, R. A survey on security of certificateless signature schemes. *IETE Tech. Rev.* **2016**, *33*, 115–121. [CrossRef]
20. 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; Lecture Notes in Computer Science; Zhou, J., López, J., Deng, R.H., Bao, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3650, pp. 134–148. [CrossRef]
21. 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]
22. He, D.; Chen, J.; Zhang, R. Efficient and provably-secure certificateless signature scheme without bilinear pairings. *IACR Cryptol. ePrint Arch.* **2010**, 632. [CrossRef]
23. Gong, Z.; Gao, T.; Guo, N. PCAS: Cryptanalysis and improvement of pairing-free certificateless aggregate signature scheme with conditional privacy-preserving for VANETs. *Ad. Hoc. Netw.* **2023**, *144*, 103134. [CrossRef]
24. Yang, X.; Wen, H.; Diao, R.; Du, X.; Wang, C. Improved Security of a Pairing-Free Certificateless Aggregate Signature in Healthcare Wireless Medical Sensor Networks. *IEEE Internet Things J.* **2023**, *10*, 10881–10892. [CrossRef]
25. Dai, C.; Xu, Z. Pairing-Free Certificateless Aggregate Signcryption Scheme for Vehicular Sensor Networks. *IEEE Internet Things J.* **2023**, *10*, 5063–5072. [CrossRef]
26. Zhang, J.; Mao, J. An efficient RSA-based certificateless signature scheme. *J. Syst. Softw.* **2012**, *85*, 638–642. [CrossRef]
27. Tang, F.; Gan, N.; Yang, X.; Wang, J. Anti malicious KGC certificateless signature scheme based on blockchain and domestic cryptographic SM9. *Chin. J. Netw. Inf. Secur.* **2022**, *8*, 9–19.

28. He, C.; Zhang, B.; Zhang, L.; Xi, Z.; Fang, Y.; Wang, Y. Pairing-Free Certificateless Signature Scheme based on SM2 Algorithm. In Proceedings of the 2nd International Conference on Network Simulation and Evaluation, NSE 2023, Shenzhen, China, 22–24 November 2023; Springer: Berlin/Heidelberg, Germany, 2023.
29. Gong, Z.; Long, Y.; Hong, X.; Chen, K. Two Certificateless Aggregate Signatures from Bilinear Maps. In *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2007, Qingdao, China, 30 July–1 August 2007*; Feng, W., Gao, F., Eds.; IEEE Computer Society: Washington, DC, USA; pp. 188–193. [CrossRef]