Ad Hoc Networks 124 (2022) 102702

Contents lists available at ScienceDirect

Ad Hoc Networks

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

A secure and efficient certificateless signature scheme for Internet of Things☆

Dengmei Xiang [a], Xuelian Li [a,*], Juntao Gao [b,c], Xiachuan Zhang [a]

[a] School of Mathematics and Statistics, Xidian University, Xi'an, Shaanxi, 710071, China
[b] School of Telecommunication and Engineering, Xidian University, Xi'an, Shaanxi, 710071, China
[c] Guangxi Key Laboratory of Cryptography and Information Security, China

ARTICLE INFO

Keywords:
Authentication
Certificateless signatures
Internet of Things (IoT)
Provably secure

ABSTRACT

The Internet of Things (IoT) is a new technological innovation, which makes things intelligent and our life more convenient. To ensure secure communication between smart objects in the IoT, certificateless signature is a feasible cryptographic tool to provide data integrity and identity authentication, which eliminates the cumbersome certificate management in the certificate-based signature system, as well as the key escrow problem in the identity-based cryptosystem. However, most of the existing certificateless signature schemes are not all secure to resist various attacks, such as public key replacement attacks or malicious-but-passive key-generation-center attacks. Besides, due to the limited storage and processing capabilities of these smart things, they are unable to meet the real-time demands of the IoT completely. This paper first analyzes Jia's scheme. We prove that the claimed solution is not resistant to the Type II strong adversaries. Then, we propose a novel certificateless signature scheme and prove its existentially unforgeable under the elliptic curve discrete logarithm problem assumption. Finally, the comprehensive performance evaluations indicate that, at the same security level, our scheme is more efficient than other certificateless signature schemes and is well suitable for the resource-constrained IoT environment.

1. Introduction

The Internet of Things (IoT) is a more universal network architecture that combines wireless communication and sensor technology. It further realizes the information exchange and communication between people and things, things and things based on the Internet, and expands its applications in industry, manufacturing, transportation, medical treatment, agriculture, personal life scenarios and so on [1,2].

Numerous and various types of smart devices embedded with sensors, chips, etc, have been deployed on the IoT. However, firstly, most of these devices are exposed to public networks, the security issues such as eavesdropping, tampering, and forgery of the collected massive data in the transmission process are becoming increasingly severe [3]. In IoT, data security is of vital importance, it will bring catastrophic consequences if the data are not reliable. A study from *Juniper Research* reports that spending on IoT cybersecurity solutions is set to reach over $6 billion globally by 2023. Secondly, these smart devices designed for specific application environments usually have weak computing power, small storage space, narrow communication range, and poor processing power. Lightweight cryptographic modules are thus considered in the IoT. Therefore, how to develop a secure and lightweight authentication mechanism for IoT networks has become a crucial security component of the system [4]. The Internet of Vehicles (IoV) is the most potential application in the IoT, but it also has the problems mentioned above. For instance, as shown in Fig. 1, vehicles equipped with smart devices (e.g., On Board Units, OBUs) communicate with other vehicles e.g., Vehicles to Vehicles (V2V) or roadside infrastructure including Roadside Units (RSUs) et al. e.g., Vehicles to Infrastructure (V2I) through wireless networks (e.g., IEEE802.11p or Cellular Based V2X, C-V2X) to exchange information, such as road information ahead, current location and driving speed, etc. At first, owing to the open and complex communication environment, attackers can launch various attacks, such as data eavesdropping, tampering, forgery, and denial of service. Hence, it is particularly important to verify the legitimacy of the vehicle's identity and the validity of the broadcast message to prevent malicious attacks [5,6]. Besides, the surge in the number of vehicle nodes, high-speed movement, frequent information interaction, and high real-time requirements in the IoV, combined with the limitations of the vehicle's power supply, space, and computing capabilities, to provide a secure authentication protocol for this system with low communication delays is of extreme urgency.

☆ The work was supported in part by the Key Research and Development Program of Shaanxi, China (No. 2021ZDLGY06-04) and the Guangxi Key Laboratory of Cryptography and Information Security, China (No. GCIS201802).

* Corresponding author.
E-mail addresses: dengmei1093@163.com (D. Xiang), xuelian202@163.com (X. Li), jtgao@mail.xidian.edu.cn (J. Gao), xiachuan666@gmail.com (X. Zhang).

https://doi.org/10.1016/j.adhoc.2021.102702
Received 24 June 2021; Received in revised form 25 August 2021; Accepted 6 September 2021
Available online 8 October 2021

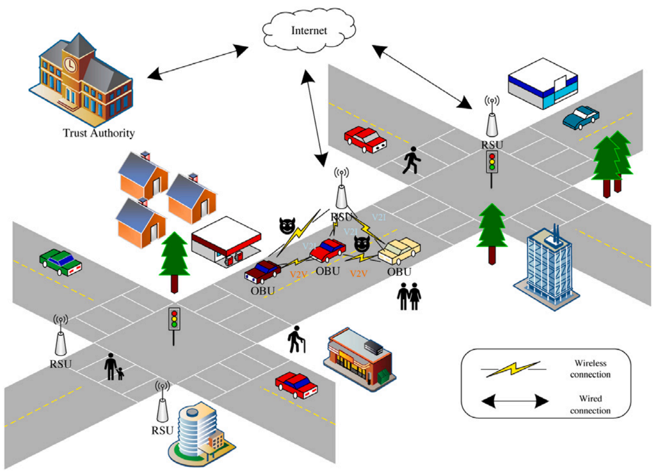

Fig. 1. Challenges in IoT.

Digital signature technology is widely concerned because of its data integrity, identity authentication, and non-repudiation. However, traditional digital signatures are no longer suitable for resource-constrained IoT environments. For example, the early certificate-based authentication system, which requires performing operations such as issuing, storing, revoking, and updating certificates when specific implementation. Not only are there complex certificate management issues, but the overhead of issuing and maintaining certificates for each IoT device that continues to grow is too expensive. While the identity-based cryptosystem proposed by Shamir [7] gets rid of certificates, it brings the defects of key escrow as well. In other words, the private key of the user is merely generated by the key generation center (KGC), the malicious KGC can forge any user's signature at will. Also, once KGC is targeted by an adversary, all users will face the risk of their private keys being leaked, at this time, the user has no privacy at all. In 2003, the idea of the certificateless public key cryptosystem (CL-PKC) proposed by Al-Riyami [8] combines the advantages of the above two cryptosystems. It does not require the centralized certificate and key management, meanwhile, the complete private key is composed of a partial private key provided by KGC and a secret value chosen by the user, respectively. As a result, certificateless signature (CLS) is more in line with the distributed, flexible and variable characteristics of IoT, and has fewer restrictions on the IoT implementation. Nevertheless, it was discovered that Al-Riyami's scheme was insecure. Furthermore, the currently presented CLS schemes are either easy to be broken or inefficient. Consequently, designing a secure and efficient CLS for IoT is still a prominent challenge.

1.1. Related work

Al-Riyami et al. [8] first proposed CL-PKC and defined the security model. Huang et al. [9] pointed out Al-Riyami et al.'s scheme insecure against public key replacement attacks (Type I adversaries), while it also cannot resist malicious-but-passive KGC attacks (Type II adversaries) in Au et al. [10]. Later, a generic construction for CLS was put forward by Yum and Lee [11]. Zhang et al. [12] revised the security model of the CLS scheme and constructed a secure CLS scheme from pairings.

Liu et al. [13] designed the first CLS scheme in the standard model, but the scheme is vulnerable to the attacks of Type II adversaries explained in Huang et al. [14]. Yuan et al. [15] proved their scheme secure under the computational Diffie–Hellman assumption in the standard model. After that, a large number of CLS schemes in the standard model were proposed [16–19]. And other CLS schemes have also been presented in recent years [20–22]. However, the above schemes are all based on expensive pairing operations, which makes the scheme inefficient in reality.

The first pairing-free CLS scheme was presented by He et al. [23]. They claimed that the scheme is secure in the random oracle model. However, the authors [24,25] independently showed that it fail to withstand a strong Type II adversary. Zhang et al. [26] designed a CLS scheme based on RSA. A secure CLS scheme with the discrete logarithm problem was introduced by Wang et al. [27]. Gong et al. [28] and Wang et al. [29] designed a CLS scheme without bilinear pairings, respectively. Then, Yeh et al. [30,31] illustrated that Gong et al.'s scheme [28] cannot resist the attack of a super Type I adversary. Meanwhile, Wang et al.'s scheme [29] can only resist the strong Type I adversary described in [32]. In 2017, Yeh et al. [32] constructed a CLS scheme for IoT. Nevertheless, Jia et al. [33] argued that Yeh's [32] unable to achieve unforgeable against both kinds of adversaries and put forward a novel CLS scheme. Besides, Du et al. [34] provided the evidence that the proposal [33] is insecure for Type I adversaries. Later, in 2018, Karati et al. [35] proposed a new CLS scheme using elliptic curve cryptography (ECC) in the random oracle model. Nasrollah et al. [36] pointed out that Karati et al.'s scheme [35] suffers from Type I adversaries attacks. Recently, Thumbur et al. [37] came up with a new CLS scheme, but it was soon discovered by Xu et al. [38] that is vulnerable to signature forgery attack as Jia's. There exists also several CLS schemes applied to specific scenarios [39–42].

1.2. Our contributions

(1) We review the CLS scheme constructed by Jia et al. and find that their scheme suffers from the attack of malicious-but-passive strong KGC, which means that it cannot resist the strong Type II adversaries.
(2) Under the elliptic curve discrete logarithm problem (ECDLP) assumption, we propose an improved CLS scheme with elliptic curve cryptography (ECC). In the random oracle, we prove that the scheme is unforgeable against adaptively chosen message and identity attacks (EUF-CMA) for two kinds of super adversaries.
(3) Our scheme is not based on the expensive pairing but uses the ECC system which is relatively fast. Compared with the typical RSA signature, the CLS that is based on ECC can achieve the same security function with better performance (e.g., shorter key length, less storage, ect.), so it is suitable for the IoT environment. Moreover, the detailed analysis shows that our signature scheme with reliable security and relatively high execution efficiency.

2. Preliminaries

2.1. Complexity assumption

Elliptic curve cryptography (ECC): First, a nonsingular elliptic curve defined over a finite field F_q is denoted as $E(F_q)$, where q is a large prime number. $E(F_q)$ consists of points satisfying the equation $y^2 = x^3+ax+b \bmod q$ and an infinite point O, where $a, b \in F_q$ and $4a^3+27b^2 \neq 0 \bmod q$. Let all points on the elliptic curve and the infinite point O be an additive cyclic group G under the additive operation of points.

$$G = \{(x, y) : y^2 = x^3 + ax + b \bmod q\} \cup O$$

Let n be the order of group G. P is a generator of G with $nP = O$, $P \in G$. Scalar multiplication in group G is denoted as $kP = P + \cdots + P$ (k times), where $k \in Z_n^*$.

Complexity Assumption: Elliptic curve discrete logarithm problem (ECDLP).

G is an additive cyclic group of order q, $P \in G$ is a generator of group G. Given a point Q, for any probabilistic polynomial time (PPT) algorithm, it is infeasible to calculate an integer $k \in Z_q^*$ such that $Q = kP$ with non-negligible probability.

Compared with other public key cryptosystems, ECC achieves the same security level with a smaller key size and storage amount, less computation and bandwidth. It is universally recognized that the 224-bit elliptic curve enables the same security level as 2048-bit RSA.

2.2. Certificateless signature

A CLS scheme contains three entities: the key generation center (KGC), the signer, and the verifier. There are seven algorithms in a CLS scheme as follows.

- *Setup:* The KGC initializes the system by running this algorithm. On inputting a security parameter λ, the algorithm outputs a system master key *msk* and the system public parameters *PP*.
- *Extract-Partial-Private-Key:* On inputting the master key *msk*, public parameters *PP*, the user's identity *ID*, the KGC extracts the user's partial private key *d* and sends it to the user through a secure channel.
- *Set-Secret-Value:* The user takes as input *PP*, identity *ID*, and outputs a secret value *x*.
- *Set-Private-Key:* Input *PP*, *d*, and the secret value *x*, the algorithm returns the private key *SK*.
- *Set-Public-Key:* The user inputs *PP*, *x*, and outputs the public key *PK*.
- *Sign:* Taking as input a message *m*, *PP*, the signer's *ID* and private key *SK*, the signer calls this algorithm to generate a signature σ.
- *Verify:* Receiving the signature σ, *PP*, the signer's *ID*, *PK* and the message *m*, the verifier runs this algorithm and returns a "1" or "0" to indicate whether the signature is valid or not.

2.3. Security model

In the first security model defined by Al-Riyami et al. [8], it is possible for adversaries to replace any entity's public key because the certificate is not required for authentication. As such, a Type I adversary $\mathcal{A}_1$ is allowed to replace the user's public key with their chosen value, and does not get the system master key and user's partial private key. They act as an external attacker. On the contrary, a Type II adversary $\mathcal{A}_2$ who models a malicious-but-passive KGC has the system master key and can learn the user's partial private key, but cannot replace the public key of the target user.

Later, depending on the adversary's ability, Huang et al. [14] further divided the adversary into three security levels: normal, strong, and super. We still adopt their adversary model to evaluate the security of the CLS scheme. When making *Sign* queries, a normal adversary only learns the user's valid signature with an original public key. In short, once the user's public key has been replaced, the normal adversary cannot get a valid signature. For a strong adversary, if the public key has been replaced, a valid signature is available for the strong adversary only after providing the corresponding secret value of the new public key. The super adversary does not need to submit a new secret value when he/she obtain a valid signature by using the replaced public key. By defining the security model, we enhance the adversary's ability to be higher than that in the real world, making the scheme more reasonable and acceptable in the real world, so as to ensure the effective implementation of the scheme.

If a CLS scheme is able to resist a super adversary, which means that it can also withstand the attack from the strong and normal adversary. The formalized security model via games between challengers C_1 (or C_2) and super adversaries $\mathcal{A}_1$ (or $\mathcal{A}_2$).

Definition 1. A CLS scheme is said to be existentially unforgeable against adaptively chosen message and identity attacks (EUF-CMA), if for any polynomial-time super adversary $\mathcal{A}_1$ and $\mathcal{A}_2$, their advantage $Adv_{\mathcal{A}_i}(\lambda)$ is negligible in the following two games, i=1, 2.

Game I. The game is interactive between a challenger C_1 and a super Type I adversary $\mathcal{A}_1$. C_1 maintains a user list L_u and two hash lists L_{H_2}, L_{H_3}. The game proceeds three phases as below.

- *Initialization.* C_1 runs the $Setup(1^\lambda)$ algorithm to generate the system master key *msk* and public parameters *PP* and sends *PP* to $\mathcal{A}_1$. C_1 keeps *msk* secretly.
- *Queries.* $\mathcal{A}_1$ is allowed to issue polynomial queries to the challenger C_1.

  (1) *Create-User(ID).* This oracle generates all the required parameters for a user *ID*. C_1 first looks up the list L_u to confirm whether the user has been created or not. If it has, C_1 returns *PK*. Otherwise, C_1 respectively executes the following algorithms *Extract-Partial-Private-Key, Set-Secret-Value, Set-Private-Key, Set-Public-Key* and outputs (d, x, PK). Then C_1 sends *PK* to $\mathcal{A}_1$ and adds (ID, d, x, PK) to the list L_u. We suppose that *Create-User* has always been queried precedes other oracles.

  (2) *Replace-Public-Key*(ID, x', PK'). On receiving such a query, C_1 replaces (x, PK) with (x', PK') and updates the list L_u. Here, the adversary may not provide the secret value corresponding to PK'. In this case, $x' = \perp$.

  (3) *Extract-Secret-Value(ID).* For such a query, C_1 checks the list L_u and returns *x*. Note that if the *Replace-Public-Key* oracle has been queried on input (ID, x', PK') and $\mathcal{A}_1$ does not provide the secret value x', C_1 will return a "$\perp$".

  (4) *Extract-Partial-Private-Key(ID).* C_1 searches the list L_u and returns *d* to $\mathcal{A}_1$.

  (5) *Super-Sign(ID,m).* C_1 calls the *Sign* algorithm and outputs a signature σ such that $Verify(ID, m, \sigma, PP, PK) = 1$, where *PK* is the latest public key stored in L_u. If *PK* has been replaced, the public key is the one submitted by the adversary.

- *Forgery.* After polynomial queries, $\mathcal{A}_1$ outputs a forged message-signature pair (m^*, σ^*) for the target identity ID^*. $\mathcal{A}_1$ wins in Game I when the following conditions hold:

  1. $Verify(ID^*, m^*, \sigma^*, PP, PK^*) = 1$
  2. $\mathcal{A}_1$ did not ask *Super-Sign* with input (ID^*, m^*);
  3. *Extract-Partial-Private-Key* has never been queried with input ID^*.

  The probability of $\mathcal{A}_1$ winning the game is denoted as

$$Adv_{\mathcal{A}_1}(\lambda) = \left|[Verify(ID^*, m^*, \sigma^*, PP, PK^*) = 1] - \frac{1}{2}\right|$$

Game II. This game executes between a Type II adversary $\mathcal{A}_2$ and a challenger C_2. Similar to Game I, Game II also goes through three phases.

- *Initialization.* C_2 performs $Setup(1^\lambda)$ and then the master key *msk* is returned along with the public parameters *PP* to $\mathcal{A}_2$.
- *Queries.* $\mathcal{A}_2$ adaptively issues queries to *Create-User, Replace-Public-Key, Extract-Secret-Value, Extract-Partial-Private-Key* and *Super-Sign* as in Game I, and the challenger C_2 gets back the required parameters similar to Game I.
- *Forgery.* $\mathcal{A}_2$ outputs a forgery (ID^*, m^*, σ^*). $\mathcal{A}_2$ succeeds in Game II when the following four conditions are satisfied:

  1. $Verify(ID^*, m^*, \sigma^*, PP, PK^*) = 1$
  2. $\mathcal{A}_2$ did not query *Super-Sign* with input (ID^*, m^*);
  3. $\mathcal{A}_2$ has never queried *Extract-Secret-Value* with input ID^*;
  4. $\mathcal{A}_2$ has never queried *Replace-Public-Key* with input ID^*.

We denote the probability of $\mathcal{A}_2$ winning the game as

$$Adv_{\mathcal{A}_2}(\lambda) = \left|[Verify(ID^*, m^*, \sigma^*, PP, PK^*) = 1] - \frac{1}{2}\right|$$

3. Review and analysis of Jia et al.'s scheme

3.1. Jia et al.'s scheme

Jia et al.'s CLS scheme [33] involves seven algorithms as follows.

- *Setup:* KGC generates an elliptic additive group G of order q, where q is a prime number with the length of λ-bit and λ is a secure parameter. Let $P \in G$ be a generator of group G. KGC randomly picks $s \in Z_q^*$ as the system master key *msk* and calculates $P_{pub} = sP$. KGC chooses three hash functions $H_1 : \{0,1\}^* \times G \to Z_q^*, H_2 : \{0,1\}^* \times G \times G \to Z_q^*, H_3 : \{0,1\}^* \times Z_q^* \times G \times G \to Z_q^*$. And then KGC publishes system public parameters $PP = \{G, P, P_{pub}, H_1, H_2, H_3\}$.
- *Extract-Partial-Private-Key:* For the user's identity *ID*, the KGC randomly chooses $r_{ID} \in Z_q^*$ and computes $R_{ID} = r_{ID}P, h_1 = H_1(ID, R_{ID}), d = (r_{ID}+h_1 s) \bmod q$. KGC sends (R_{ID}, d) to the user through a secure channel.
- *Set-Secret-Value:* The user randomly chooses a secret value $x_{ID} \in Z_q^*$.
- *Set-Private-Key:* The user sets the private key $SK = (d, x_{ID})$.
- *Set-Public-Key:* The user first computes $X_{ID} = x_{ID}P$, $h_2 = H_2(ID, X_{ID})$, $Q_{ID} = R_{ID} + h_2 X_{ID}$ and sets the public key $PK = (R_{ID}, Q_{ID})$.
- *Sign:* Given message *m*, public parameters *PP*, identity *ID* and private key *SK*, the signer picks $t \in Z_q^*$ and computes $T = tP = (T_x, T_y), r = T_x \bmod q$, sets $h_3 = H_3(ID, m, T, PK, h_1)$ and calculates $\tau = t^{-1}(h_3 + r(d + h_2 x_{ID})) \bmod q$. The signer outputs the signature $\sigma = (T, \tau)$ on message *m*.
- *Verify:* When receiving (m, ID, PP, PK, σ), the verifier first calculates: $h_1 = H_1(ID, R_{ID}), h_3 = H_3(ID, m, T, PK, h_1), r = T_x \bmod q$. Then it verifies $\tau T = h_3 P + r(Q_{ID} + h_1 P_{pub}) \bmod q$ whether holds. If yes, the verifier output "1", otherwise, outputs "0".

3.2. Cryptanalysis of Jia et al.'s scheme

In this section, we point out the weaknesses of the scheme described above. Their scheme is vulnerable to attacks launched by malicious-but-passive KGC. It cannot resist the attack of a strong Type II adversary, let alone super Type II adversaries. We illustrate how to forge a signature that can pass the verification by replacing the system public key. The concrete construction is as follows.

Attack: A strong adversary $\mathcal{A}_2$ performs the following actions to forge a valid signature on a chosen message m^* for target identity ID^*. Since the adversary $\mathcal{A}_2$ is a Type II adversary who has the ability to replace the system public key P_{pub} with a new key P'_{pub}. The detailed attack is shown as follows:

1. $\mathcal{A}_2$ can eavesdrop a valid signature $\sigma = (T, \tau)$, learns user's public key $PK = (R, Q)$ and some extra parameters (e.g., hash value) from the previous session, where $h_1 = H_1(ID, R)$;
2. The adversary first chooses two random numbers $t', z \in Z_q^*$, and then computes $T' = t'P = (T'_x, T'_y), r' = T'_x \bmod q, P'_{pub} = \frac{1}{h_1}(zP - Q), z \in Z_q^*, h'_3 = H_3(ID^*, m^*, T', PK, h_1), \tau' = (t')^{-1}(h'_3 + r'z) \bmod q$. At last, outputs $\sigma' = (\tau', T')$ as the forged signature.

The following equation indicates that the forged signature can be easily verified:

$$\begin{aligned}\tau'T' &= (t')^{-1}(h'_3 + r'z)(t'P) \bmod q = (h'_3 + r'z)P = h'_3 P + r'zP \\ &= h'_3 P + r'(Q + h_1 P'_{pub})\end{aligned}$$

Remarks. Because the malicious KGC cannot obtain the user's secret value but can replace the system public key P_{pub}. In Jia's scheme, the Type II adversary uses the relationship between P_{pub} and h_1 to bypass the requirement of the secret value X_{ID} in verification, thus successfully forges the valid signature of any message. To overcome this drawback, we modify the input of hash function, signature and verification phases to make the scheme more robust.

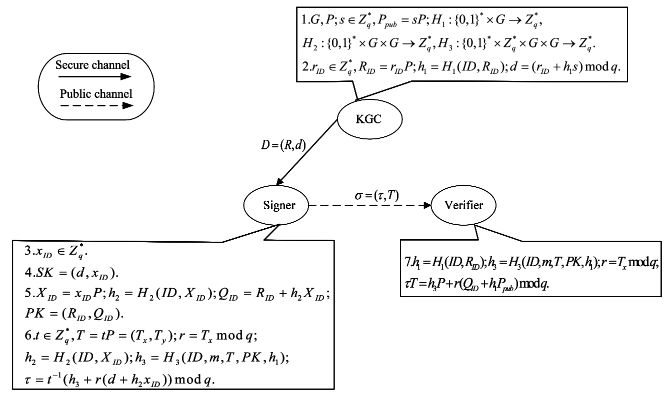

Fig. 2. Jia et al.'s CLS scheme.

4. The proposed CLS scheme

4.1. Our construction

We propose a provably secure CLS scheme that has the following seven algorithms.

- *Setup:* KGC selects a prime number q with a length of λ-bit. λ is a secure parameter. G is an elliptic additive group of order q over the finite field F_q. $P \in G$ is a generator. KGC randomly picks $s \in Z_q^*$ and calculates $P_{pub} = sP$, where system master secret key *msk*=*s* keeps secretly. Select three distinct secure hash functions $H_i : \{0,1\}^* \to Z_q^* (i = 1, 2, 3)$. KGC publishes system public parameters $PP = \{q, G, P, P_{pub}, H_1, H_2, H_3\}$.
- *Extract-Partial-Private-Key:* Given an identity *ID*, the KGC randomly chooses $r \in Z_q^*$ and computes $R = rP$, $h_1 = H_1(ID, R, P_{pub}), d = (r + h_1 s) \bmod q$. KGC secretly sends partial private key pairs (R, d) to the user. Its validity can be verified by the equation $dP = R + h_1 P_{pub} \bmod q$.
- *Set-Secret-Value:* The user randomly picks a number $x \in Z_q^*$ and sets x as his/her secret value, computes $X = xP$.
- *Set-Public-Key:* The user sets the public key $PK = (R, X)$.
- *Set-Private-Key:* The user sets the private key $SK = (d, x)$.
- *Sign:* On inputting message *m*, system public parameters *PP*, the identity *ID*, and the private key *SK*, the signer randomly picks $t \in Z_q^*$ and computes $T = tP$ and $h_2 = H_2(ID, T, PK)$, sets $h_3 = H_3(ID, m, T, PK, P_{pub})$ and calculates $\tau = x^{-1}(h_2 t + h_3 d) \bmod q$. The signer outputs the signature $\sigma = (T, \tau)$.
- *Verify:* After receiving a message-signature tuple (m, ID, PP, PK, σ), the verifier first calculates: $h_1 = H_1(ID, R, P_{pub})$, $h_2 = H_2(ID, T, PK), h_3 = H_3(ID, m, T, PK, P_{pub})$. Then it checks $\tau X = h_2 T + h_3(R + h_1 P_{pub}) \bmod q$. If yes, the verifier output "1", else it outputs "0".

As can be seen from Figs. 2 and 3, we first modify the input of h_1. If the Type II attacker replaces the system public key with P'_{pub}, then it will become $h'_1 = H_1(ID, R, P'_{pub})$, and the corresponding $h'_1 P'_{pub}$ will appear in the verification equation. However, P'_{pub} as the input of h'_1, so it is infeasible to forge through the equation $P'_{pub} = \frac{1}{h'_1}(zP - Q)$. In addition, the design of our scheme is also relatively simple, especially in the *Set-Public-Key* and *Sign* phase.

4.2. Security proof

Theorem 1. *In the random oracle, the proposed CLS scheme is EUF-CMA secure against super Type I and super Type II adversaries if the ECDLP is intractable.*

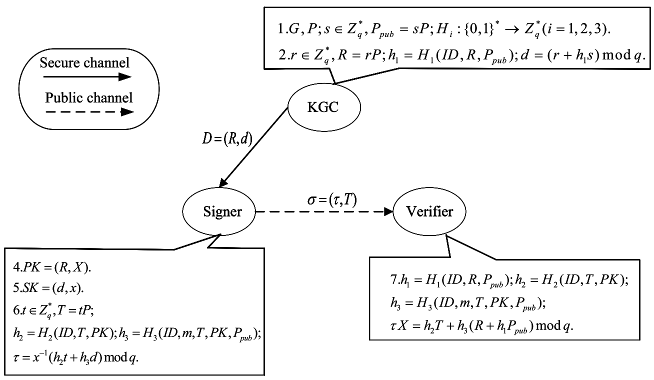

Fig. 3. Our CLS scheme.

Theorem 1 is deduced from the security model described in Section 2.3 and Lemmas 1 and 2 as below.

Lemma 1. *Supposed that a polynomial-time super Type I adversary $\mathcal{A}_1$ who wins in Game I with non-negligible probability ε, where q_{cu}, q_{eppk} denotes the maximum number of Create-User and Extract-Partial-Private-Key queries by the adversary $\mathcal{A}_1$, respectively. There must be an algorithm C_1 can solve the ECDLP problem with advantage $\varepsilon_1 \geq \left(1 - \frac{1}{q_{cu}}\right)^{q_{eppk}} \frac{1}{q_{cu}} \varepsilon$.*

Proof. Let $\mathcal{A}_1$ be a PPT super Type I adversary who breaks the unforgeability of our CLS scheme with probability ε in Game I. We construct an algorithm C_1 that calls $\mathcal{A}_1$ as a subroutine and aims at solving ECDLP. That is, C_1 is given an ECDLP instance $(G, P, Q = sP)$ and tries to work out s with the help of the adversary $\mathcal{A}_1$.

H_1, H_2, H_3 are simulated as random oracles. C_1 maintains a user list L_u, and two hash lists L_{H_2}, L_{H_3} separately records the queries to H_2 and H_3. The three lists are initially empty. The game between $\mathcal{A}_1$ and C_1 proceeds as follows.

Initialization. C_1 randomly chooses an identity ID_t as the target identity, sets $PP = (G, P, Q = P_{pub} = sP)$ and sends PP to $\mathcal{A}_1$, where s is unknown to C_1.

Queries. $\mathcal{A}_1$ is allowed to issues polynomial queries to C_1.

Create-User(ID). As for querying with identity *ID*, C_1 first looks up the list L_u and returns *PK*, if the corresponding entry in the list L_u. Otherwise, if $ID \neq ID_t$, C_1 randomly chooses $d, x, h_1 \in Z_q^*$, computes $R = dP - h_1 P_{pub} \bmod q$, $X = xP$ and sets $h_1 = H_1(ID, R, P_{pub})$. If $ID = ID_t$, C_1 selects $r, x, h_1 \in Z_q^*$ and lets $R = rP, X = xP, h_1 = H_1(ID, R, P_{pub}), d = \perp$. At last, C_1 outputs $PK = (R, X)$ to $\mathcal{A}_1$ and adds (ID, R, X, d, x, h_1) to the list L_u.

Hash query. C_1 responds to $\mathcal{A}_1$'s hash queries as follows.

H_1-query. $\mathcal{A}_1$ issues (ID, R, P_{pub}) to this oracle. C_1 first checks the list L_u and returns h_1 to $\mathcal{A}_1$ when it exists. Otherwise, C_1 asks *Create-User* oracle and extracts h_1 from L_u and returns it to $\mathcal{A}_1$.

H_2-query. When $\mathcal{A}_1$ queries H_2 on (ID, T, PK), C_1 looks for the entry in the list L_{H_2}. If it has, C_1 outputs h_2. Otherwise, C_1 randomly chooses $h_2 \in Z_q^*$ and sets $h_2 = H_2(ID, T, PK)$. C_1 returns h_2 to $\mathcal{A}_1$ and adds (ID, T, PK, h_2) to L_{H_2}.

H_3-query. On inputting with (ID, m, T, PK, P_{pub}), C_1 first checks the list L_{H_3}. If there is an entry, C_1 outputs h_3. Otherwise, C_1 randomly picks $h_3 \in Z_q^*$ and sets $h_3 = H_3(ID, m, T, PK, P_{pub})$. C_1 returns $h_3 = H_3(ID, m, T, PK, P_{pub})$ to $\mathcal{A}_1$ and adds $(ID, m, T, PK, P_{pub}, h_3)$ to L_{H_3}.

Extract-Partial-Private-Key(ID). If $ID = ID_t$, C_1 returns "$\perp$" and aborts the game. Otherwise, C_1 searches the list L_u and returns d to $\mathcal{A}_1$.

Extract-Secret-Value(ID). For this query, C_1 searches the list L_u. If the entry exists, C_1 returns x to $\mathcal{A}_1$. Otherwise, C_1 makes a *Create-User(ID)* query with identity *ID* and gets back x. Here, the *Extract-Secret-Value* oracle does not output the secret value when the user's public key has been replaced and $\mathcal{A}_1$ does not provide a corresponding x.

Replace-Public-Key(ID, x', PK'). If $\mathcal{A}_1$ asks such a query with (ID, PK'), where $PK' = (R', Q')$. C_1 looks up the list L_u and updates the entry (ID, R, X, d, x, h_1) with (ID, R', X', d, x, h_1). Here x sets "$\perp$".

Super-Sign(ID,m). Upon receiving this query, C_1 check whether the three tuples (ID, R, X, d, x, h_1), (ID, T, PK, h_2) and $(ID, m, T, PK, P_{pub}, h_3)$ are contained in the three lists L_u, L_{H_2} and L_{H_3}, respectively.

- If $ID \neq ID_t$ and $x \neq \perp$ (the public key has not been replaced), C_1 randomly selects $t, h_2, h_3 \in Z_q^*$, sets $h_2 = H_2(ID, T, PK), h_3 = H_3(ID, m, T, PK, P_{pub})$, calculates $T = tP$ and $\tau = x^{-1}(h_2 t + h_3 d) \bmod q$.
- If $ID = ID_t$ or $x = \perp$, C_1 randomly selects $\tau, h_2, h_3 \in Z_q^*$ and computes $T = h_2^{-1}[\tau X - h_3(R + h_1 P_{pub})] \bmod q$. C_1 outputs $\sigma = (T, \tau)$ and adds (ID, R, X, d, x, h_1), (ID, T, PK, h_2) and $(ID, m, T, PK, P_{pub}, h_3)$ to the list L_u, L_{H_2} and L_{H_3}, respectively.

C_1 outputs $\sigma = (T, \tau)$ and adds (ID, R, X, d, x, h_1), (ID, T, PK, h_2) and $(ID, m, T, PK, P_{pub}, h_3)$ to the list L_u, L_{H_2} and L_{H_3}, respectively.

Forgery. Finally, $\mathcal{A}_1$ provides a valid message-signature tuple $(m^*, \sigma^* = (T^*, \tau^{(1)}), h_3^*)$ for ID^* with PK^* that may be replaced by $\mathcal{A}_1$. That is, the equation $\tau^{(1)} X^* = h_2^* T^* + h_3^*(R^* + h_1^* P_{pub}) \bmod q$ holds. Meanwhile, $\mathcal{A}_1$ is not allowed to submit ID^* to *Extract-Partial-Private-Key* and (m^*, ID^*) has never been queried to *Super-Sign*. If $ID^* \neq ID_t$, C_1 aborts the game. Otherwise, C_1 looks up the list L_u, and L_{H_2}, L_{H_3} for the tuple $(ID^*, R^*, X^*, d^*, x^*, h_1^*), (ID^*, T^*, PK^*, h_2^*)$ and $(ID^*, m^*, T^*, PK^*, P_{pub}, h_3^*)$. Due to the forking lemma [43], C_1 replays $\mathcal{A}_1$ with the same random tape, but provides two different values of $h_3(h_3^{(2)}, h_3^{(3)})$, $\mathcal{A}_1$ would output another two valid forgeries $(T^*, \tau^{(2)})$ and $(T^*, \tau^{(3)})$ which satisfy:

$$\tau^{(2)} X^* = h_2^* T^* + h_3^{(2)}(R^* + h_1^* P_{pub}) \bmod q$$
$$\tau^{(3)} X^* = h_2^* T^* + h_3^{(3)}(R^* + h_1^* P_{pub}) \bmod q$$

For convenience, we set $D = (\tau^{(1)} - \tau^{(2)})(h_3^{(1)} - h_3^{(3)}) - (\tau^{(1)} - \tau^{(3)})(h_3^{(1)} - h_3^{(2)})$.

Here, $h_3^{(1)} = h_3^*$. Besides, $T^* = t^* P, R^* = r^* P, P_{pub} = sP, X^* = x^* P$. And then we have the following three linear independent equalities.

$$\tau^{(1)} = (x^*)^{-1}[h_2^* t^* + h_3^{(1)}(r^* + h_1^* s)] \bmod q$$
$$\tau^{(2)} = (x^*)^{-1}[h_2^* t^* + h_3^{(2)}(r^* + h_1^* s)] \bmod q$$
$$\tau^{(3)} = (x^*)^{-1}[h_2^* t^* + h_3^{(3)}(r^* + h_1^* s)] \bmod q$$

Where t^*, x^* and s are unknown for C_1. C_1 is capable of figuring out the value s.

$$(\tau^{(1)} - \tau^{(2)}) x^* = (h_3^{(1)} - h_3^{(2)})(r^* + h_1^* s)$$

$$(\tau^{(1)} - \tau^{(3)}) x^* = (h_3^{(1)} - h_3^{(3)})(r^* + h_1^* s) \tag{1}$$

so

$$x^* = \frac{h_3^{(1)} - h_3^{(2)}}{\tau^{(1)} - \tau^{(2)}}(r^* + h_1^* s)$$

Taking x^* into Eq. (1), we have $D \cdot (r^* + h_1^* s) = 0 \bmod q$, and $s = -\frac{r^*}{h_1^*} \bmod q$, which is the solution to the ECDLP instance.

Then, we discuss the probability of C_1 winning probability in Game I. C_1 succeeds if the following three events occur:

- E_1: When $\mathcal{A}_1$ queries *Extract-Partial-Private-Key* oracle, C_1 does not abort the game.
- E_2: σ^* is a valid forgery on (m^*, ID^*).
- E_3: For the forged signature (m^*, σ^*) submitted by $\mathcal{A}_1$ in the *Forgery* phase, we have $ID^* = ID_t$.

Firstly, we have $Pr[E_2|E_1] \geq \varepsilon$.

Besides, the probability that the event E_1 happens satisfies $Pr[E_1] \geq \left(1 - \frac{1}{q_{cu}}\right)^{q_{eppk}}$.

In the submitted forgery, when $ID^* = ID_t$, the probability is $Pr[E_3|E_1E_2] \geq \frac{1}{q_{cu}}$.

Therefore, C_1's advantage is

$$\varepsilon_1 \geq Pr[E_1E_2E_3] = Pr[E_1]Pr[E_2|E_1]Pr[E_3|E_1E_2]$$
$$\geq \left(1 - \frac{1}{q_{cu}}\right)^{q_{eppk}} \frac{1}{q_{cu}}\varepsilon$$

Lemma 2. *Supposed that a polynomial-time super Type II adversary $\mathcal{A}_2$ who wins in Game II with non-negligible probability ε, then there exists a polynomial-time algorithm C_2 who can solve the ECDLP problem with advantage $\varepsilon_2 \geq \left(1 - \frac{1}{q_{cu}}\right)^{q_{esv}+q_{rpk}} \frac{1}{q_{cu}}\varepsilon$.*

Note that q_{cu}, q_{esv} and q_{rpk} denote the maximum number of *Create-User, Extract-Secret-Value* and *Replace-Public-Key* oracle queried by adversaries $\mathcal{A}_2$, respectively.

Proof.

$\mathcal{A}_2$ is a PPT super Type II adversary who breaks the unforgeability of our CLS scheme with probability ε in Game II. We can construct an algorithm C_2, which calls $\mathcal{A}_2$ as a subroutine and aims at solving ECDLP. Given an ECDLP instance $(G, P, Q = xP)$, C_2 tries to find x under $\mathcal{A}_2$'s forgery. H_1, H_2, H_3 are simulated as random oracles. C_2 maintains three empty lists L_u, L_{H_2} and L_{H_3}. The game is interactive between $\mathcal{A}_2$ and C_2 as follows.

Initialization. C_2 randomly selects an identity ID_t as the target identity, picks $s \in Z_q^*$, sets $P_{pub} = sP$ and $PP = (G, P, P_{pub})$, C_2 sends (PP, s) to $\mathcal{A}_2$.

Queries. $\mathcal{A}_2$ issues polynomial queries to C_2.

Create-User(ID). For this query with identity *ID*, C_2 first checks the list L_u. If *ID* is in L_u, C_2 returns *PK*. Otherwise, C_2 randomly picks $r, h_1 \in Z_q^*$, calculates $R = rP, d = (r + h_1 s) \bmod q$ and sets $h_1 = H_1(ID, R, P_{pub})$. If $ID \neq ID_t$, C_2 randomly chooses $x \in Z_q^*$, computes $X = xP$. If $ID = ID_t$, C_2 sets $X = xP, x = \perp$. C_2 returns $PK = (R, X)$ and adds (ID, R, X, d, x, h_1) to the list L_u.

Hash query. The answers to H_1-query, H_2-query and H_3-query are similar to do in Game I.

Extract-Partial-Private-Key(ID). C_2 searches the list L_u. If the entry exists, returns d to $\mathcal{A}_2$. Otherwise, C_2 makes a *Create-User* query with identity *ID* and returns *d*.

Extract-Secret-Value(ID). On receiving this query, if $ID = ID_t$, C_2 returns "$\perp$" and aborts the game. Otherwise, C_2 checks the list L_u and returns x to $\mathcal{A}_2$.

Replace-Public-Key(ID, x', PK'). When $\mathcal{A}_2$ issues query with (ID, PK'), where $PK' = (R', Q')$. If $ID = ID_t$, C_2 returns "$\perp$" and aborts the game. Otherwise, C_2 looks for the list L_u and updates (ID, R, X, d, x, h_1) with (ID, R', X', d, x, h_1). Here x is "$\perp$".

Super-Sign(ID,m). C_2 responds to $\mathcal{A}_2$'s sign queries similar to what C_1 does in Game I. On receiving this query, C_2 first checks the list L_u, L_{H_2} and L_{H_3} for the tuple (ID, R, X, d, x, h_1), (ID, T, PK, h_2) and $(ID, m, T, PK, P_{pub}, h_3)$, respectively.

- If $ID \neq ID_t$ and $x \neq \perp$ (the public key has not been replaced), C_2 randomly selects $t, h_2, h_3 \in Z_q^*$, sets $h_2 = H_2(ID, T, PK), h_3 = H_3(ID, m, T, PK, P_{pub})$, computes $T = tP$ and $\tau = x^{-1}(h_2 t + h_3 d) \bmod q$.
- If $ID = ID_t$ or $x = \perp$, C_2 randomly chooses $\tau, h_2, h_3 \in Z_q^*$ and computes $T = h_2^{-1}[\tau X - h_3(R + h_1 P_{pub})] \bmod q$.

C_2 outputs $\sigma = (T, \tau)$ and adds (ID, R, X, d, x, h_1), (ID, T, PK, h_2) and $(ID, m, T, PK, P_{pub}, h_3)$ to the list L_u, L_{H_2} and L_{H_3}, respectively.

Forgery. Finally, $\mathcal{A}_2$ submits a valid tuple $(m^*, \sigma^* = (T^*, \tau^*), h_2^*)$ for ID^*. Also, $\mathcal{A}_2$ has never issues ID^* to *Extract-Secret-Value, Replace-Public-Key* and (m^*, ID^*) has never been queried to *Super-Sign*. If $ID^* \neq ID_t$, C_2 aborts the game. Otherwise, C_2 searches the list L_u, L_{H_2} and L_{H_3} for the entry $(ID^*, R^*, X^*, d^*, x^*, h_1^*), (ID^*, T^*, PK^*, h_2^*)$ and $(ID^*, m^*, T^*, PK^*, P_{pub}, h_3^*)$. According to the forking lemma [43], C_2 replays $\mathcal{A}_2$ with the same random tape, and provides a distinct value of $h_2(h_2{}')$, $\mathcal{A}_2$ outputs another valid forgery (T^*, τ') which satisfies:

$$\tau' X^* = h_2{}' T^* + h_3{}^*(R^* + h_1{}^* P_{pub}) \bmod q$$

However, $T^* = t^*P, R^* = r^*P, P_{pub} = sP$, $X^* = xP$. And then we have the following linear independent equalities.

$$\tau^* = x^{-1}[h_2^* t^* + h_3^*(r^* + h_1^* s)] \bmod q \tag{2}$$

$$\tau' = x^{-1}[h_2' t^* + h_3^*(r^* + h_1^* s)] \bmod q$$

Where t^* and x are unknown for C_2. C_2 can successfully figure out x from the above equalities.

As for $(\tau^* - \tau')x = (h_2^* - h_2')t^*$, we learn $t^* = \frac{\tau^* - \tau'}{h_2^* - h_2'}x$.

Taking t^* into Eq. (2), we have

$$[\tau^* - \frac{h_2^*(\tau^* - \tau')}{h_2^* - h_2'}]x = h_3^*(r^* + h_1^* s)$$

so

$$x = \frac{h_3^*(r^* + h_1^* s)(h_2^* - h_2')}{\tau^*(h_2^* - h_2') - h_2^*(\tau^* - \tau')},$$

which is the solution to the ECDLP instance.

Next, we analyze the probability that C_2 winning probability in Game II. C_2 succeeds if:

- E_1: When $\mathcal{A}_2$ queries *Extract-Secret-Value* and *Replace-Public-Key* oracles, C_2 does not abort the game.
- E_2: σ^* is a valid forgery on (m^*, ID^*).
- E_3: For the forged signature (m^*, σ^*) submitted by $\mathcal{A}_2$ in the *Forgery* phase, we have identity $ID^* = ID_t$.

Firstly, we have $Pr[E_2|E_1] \geq \varepsilon$.

Besides, the probability that the event E_1 happens, which is $Pr[E_1] \geq \left(1 - \frac{1}{q_{cu}}\right)^{q_{esv}} \left(1 - \frac{1}{q_{cu}}\right)^{q_{rpk}}$.

In the submitted forgery, when $ID^* = ID_t$, the probability is $Pr[E_3|E_1E_2] \geq \frac{1}{q_{cu}}$.

Therefore, C_2's advantage is

$$\varepsilon_2 \geq Pr[E_1E_2E_3] = Pr[E_1]Pr[E_2|E_1]Pr[E_3|E_1E_2]$$
$$\geq \left(1 - \frac{1}{q_{cu}}\right)^{q_{esv}+q_{rpk}} \frac{1}{q_{cu}}\varepsilon$$

5. Performance evaluation

We evaluate the performance of our scheme and other CLS schemes without pairing. We take the experimental results from Xu [38] as support for our simulation. To achieve a reliable security level of 1024-bits RSA algorithm, we choose a non-singular elliptic curve E: $y^2 = x^3 + ax + b \bmod q$, where $a, b \in Z_q^*$, G is an additive group of order q over E, p and q are both prime numbers with the length of 160 bits. The related operations are implemented based on the MIRACL library and the calculation time of several basic operations is shown in Table 1. Note that, in Table 2, $|G|$, $|G_1|$ and $\left|Z_q^*\right|$ denotes the size of the point in the elliptic curve group G, the size of a group element in the multiplicative cyclic group G_1 and the size of a group element in the group Z_q^*, respectively. Besides, super, normal, and insecure indicate the level of security that these schemes can achieve against two types of adversaries.

The computational cost mainly comes from signing and verifying. From Table 2 and Fig. 4, in terms of signature cost, it can be observed that the scheme [27] takes the most time and needs $T_{ex} = 3.8636$ ms. The signature cost is the least in [28,29,32,37,38], which is $T_{sm} = 0.442$ ms. Ours is equivalent to that of these schemes [23,33–36] and costs $T_{sm} + T_{inv} = 0.6308$ ms, which is slightly higher than [28,29,32,37,38],

Table 1
Notation and running time of operation.

Notation	Operation	Time (ms)
T_{sm}	A scalar multiplication on elliptic curve	0.4420
T_{pa}	A point addition on elliptic curve	0.0018
T_m	A modular multiplication operation	0.0011
T_a	A modular addition operation	0.0008
T_{inv}	A modular inversion operation	0.1888
T_h	A general hash operation	0.0001
T_{ex}	A modular exponentiation operation	3.8636

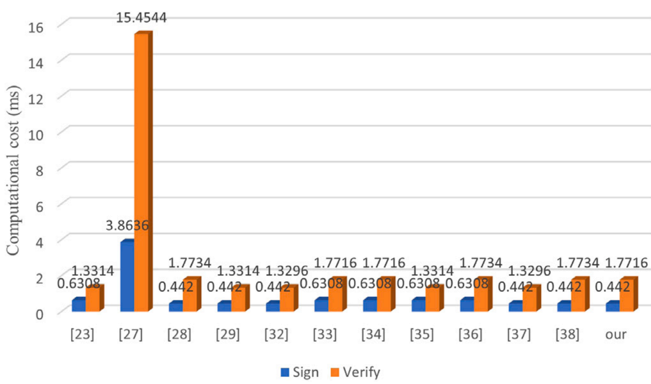

Fig. 4. Comparison of the computation costs.

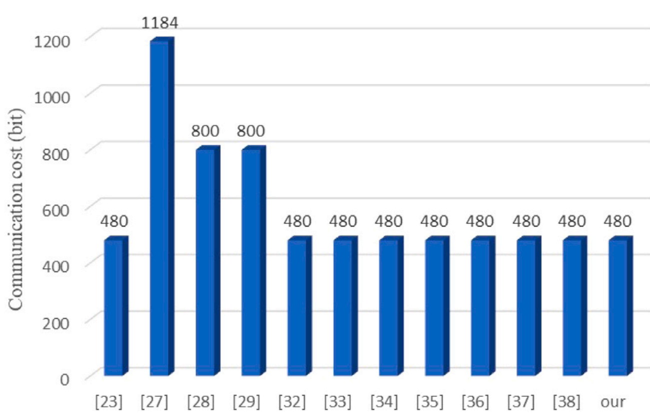

Fig. 5. Comparison of the communication costs.

however, the cost of the signature in our CLS scheme can be reduced to T_{sm} = 0.442 ms by the pre-calculation for a modular inversion operation. For the verification cost, since [27] performs four exponential operations, it requires 15.4544 ms. Obviously, it is also the least efficient among the listed free-pairing CLS schemes. Our scheme requires $4T_{sm} + 2T_{pa}$ = 1.7716 ms, as does in [33,34], and is slightly superior to other schemes [28,36,38] that is $4T_{sm} + 3T_{pa}$ = 1.7734 ms, which may put these schemes in an unfavorable position where an enormous amount of messages need to be verified, e.g., IoV.

Due to IoT devices with limited power and communication bandwidth, the communication costs also need to be taken into account, the most decisive factor affecting the communication overhead is the signature size. As shown in Table 2 and Fig. 5, the size of the signature of the scheme [27] ($|G_1| + \left|Z_q^*\right|$ = 1184 bits) is more than twice that of these schemes [23,32–38] and our scheme ($|G| + \left|Z_q^*\right|$ = 480 bits). Besides, [28,29] require $2\,|G| + \left|Z_q^*\right| = 800$ bits. The signature length of these schemes [23,32–38] is the same as ours, so it is suitable for the scenarios such as restricted communication distance.

Furthermore, in Table 2, the schemes in [28,29] can only withstand the normal Type I adversary, the schemes in [32,33,37] are even insecure for the Type I adversary, which means that they are vulnerable to forgery attacks. And the schemes in [23,32,35] can merely against the normal Type II adversary. The total computational overhead is the least in the schemes [32,37] that is $T_{sm} + 3T_{sm} + 2T_{pa}$ = 1.7716 ms, and they cannot resist not only public key replacement attack, but also malicious KGC. Only our scheme and the four schemes [27,34,36,38] can resist two types of super adversaries at the same time. We will give a performance analysis on these four schemes.

Since the nature of IoT devices, such as limited computing and processing power, the computational cost should be as small as possible, so the most time-consuming scheme [27] is not the best candidate for IoT. Our scheme only adds some lightweight components to reach the highest security standard as the schemes in [34,36,38]. As shown in Figs. 5 and 6, the total computational cost is $T_{sm} + 4T_{sm} + 2T_{pa} = 2.2136$ ms in our scheme, which slightly outperforms that of the schemes in [34,36,38], and their schemes take $T_{sm} + T_{inv} + 4T_{sm} + 2T_{pa} = 2.4024$ ms, $T_{sm} + T_{inv} + 4T_{sm} + 3T_{pa} = 2.4042$ ms and $T_{sm} + 4T_{sm} + 3T_{pa} = 2.2154$ ms, respectively. Then, the communication overhead of our scheme is the same as that of the schemes in [34,36,38] (480 bits), which is much better than the scheme [27] (1184 bits). In summary, we have reached the highest security level while ensuring low computational costs and have achieved true unforgeability.

According to the above analysis, our scheme achieves better performance without losing security. The proposed protocol has advantages in terms of the total computational time. Meanwhile, this scheme provides a practical approach for the resource-constrained IoT environment.

Table 2
Performance comparison for pairing-free CLS schemes.

Scheme	Sign	Verify	Signature size	Security A_1	Security A_2
He [23]	$T_{sm} + T_{inv}$	$3T_{sm} + 3T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Super	Normal [24,25]
Wang [27]	T_{ex}	$4T_{ex}$	$\|G_1\| + \left\|Z_q^*\right\|$	Super	Super
Gong [28]	T_{sm}	$4T_{sm} + 3T_{pa}$	$2\,\|G\| + \left\|Z_q^*\right\|$	Normal	Super
Wang [29]	T_{sm}	$3T_{sm} + 3T_{pa}$	$2\,\|G\| + \left\|Z_q^*\right\|$	Normal	Super
Yeh [32]	T_{sm}	$3T_{sm} + 2T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Insecure [33]	Normal [33]
Jia [33]	$T_{sm} + T_{inv}$	$4T_{sm} + 2T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Insecure [34]	Normal[our]
Du [34]	$T_{sm} + T_{inv}$	$4T_{sm} + 2T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Super	Super
Karati [35]	$T_{sm} + T_{inv}$	$3T_{sm} + 3T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Super	Normal [36]
Pakniat [36]	$T_{sm} + T_{inv}$	$4T_{sm} + 3T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Super	Super
Thumbur [37]	T_{sm}	$3T_{sm} + 2T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Insecure [36]	Super
Xu [38]	T_{sm}	$4T_{sm} + 3T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Super	Super
Our	$T_{sm} + T_{inv}$	$4T_{sm} + 2T_{pa}$	$\|G\| + \left\|Z_q^*\right\|$	Super	Super

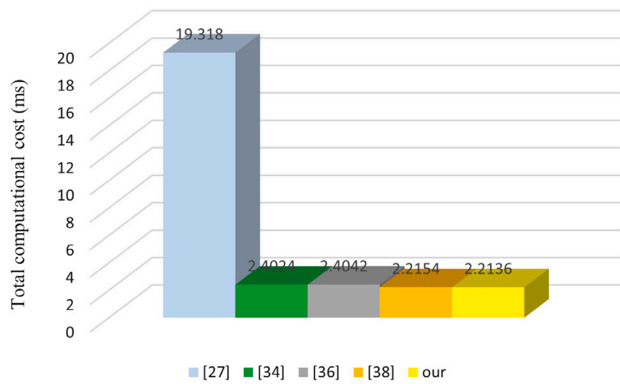

Fig. 6. Comparison of the total computational cost against super adversaries.

6. Conclusions

The IoT has a profound impact on social life, production, and the human way of thinking. To address the essential matters of data integrity and identity authentication in the IoT environment, we propose a new CLS scheme and present a strict security proof, which is unforgeable against adaptive chosen-message attacks under elliptic curve discrete logarithm problem hard assumptions in the random oracle model. In addition, there are not complex and time-consuming pairing or map-to-point hash operations in our scheme, and the result from performance analysis also demonstrates that it is more efficient than other similar schemes. In future work, we will evaluate our proposed scheme in some real IoT scenarios (such as Internet of vehicles and smart home), and further improve and design new solutions.

Declaration of competing interest

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

References

[1] M. Serror, S. Hack, M. Henze, M. Schuba, K. Wehrle, Challenges and opportunities in securing the industrial Internet of Things, IEEE Trans. Ind. Inform. 17 (5) (2021) 2985–2996, http://dx.doi.org/10.1109/TII.2020.3023507.

[2] K.N. Qureshi, S. Din, G. Jeon, F. Piccialli, Internet of vehicles: Key technologies, network model, solutions and challenges with future aspects, IEEE Trans. Intell. Transp. Syst. 22 (3) (2021) 1777–1786, http://dx.doi.org/10.1109/TITS.2020.2994972.

[3] IoT Devices in the Enterprise 2020: Shadow IoT Threat Emerges, [Online], https://www.zscaler.com/press/zscaler-released-second-annual-iot-report-2020.

[4] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, B. Yan, BC-SABE: Blockchain-aided searchable attribute-based encryption for cloud-IoT, IEEE Internet Things J. 7 (9) (2020) 7851–7867, http://dx.doi.org/10.1109/JIOT.2020.299323.

[5] Y. Wang, J. Yu, B. Yan, G. Wang, Z. Shan, BSV-PAGS: Blockchain-based special vehicles priority access guarantee scheme, Comput. Commun. 161 (2020) 28–40, http://dx.doi.org/10.1016/j.comcom.2020.07.012.

[6] F. Li, D. Wang, Y. Wang, X. Yu, N. Wu, J. Yu, H. Zhou, Wireless communications and mobile computing blockchain-based trust management in distributed internet of things, Wirel. Commun. Mob. Comput. 2020 (2020) http://dx.doi.org/10.1155/2020/8864533.

[7] A. Shamir, Identity-based cryptosystems and signature schemes, in: Advances in Cryptology, Springer, 1985, pp. 47–53, http://dx.doi.org/10.1007/3-540-39568-7_5.

[8] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, in: Advances in Cryptology - ASIACRYPT 2003, Springer, 2003, pp. 452–473, http://dx.doi.org/10.1007/978-3-540-40061-5_29.

[9] X. Huang, W. Susilo, Y. Mu, F. Zhang, On the security of certificateless signature schemes from Asiacrypt 2003, in: Cryptology and Network Security, Springer, 2005, pp. 13–25, http://dx.doi.org/10.1007/11599371_2.

[10] M.H. Au, Y. Mu, J. Chen, D.S. Wong, J.K. Liu, G. Yang, Malicious KGC Attacks in Certificateless Cryptography, Association for Computing Machinery, 2007, http://dx.doi.org/10.1145/1229285.1266997.

[11] D.H. Yum, P.J. Lee, Generic construction of certificateless signature, in: Information Security and Privacy, Springer, 2004, pp. 200–211, http://dx.doi.org/10.1007/978-3-540-27800-9_18.

[12] Z. Zhang, D.S. Wong, J. Xu, D. Feng, Certificateless public-key signature: Security model and efficient construction, in: Applied Cryptography and Network Security, Springer, 2006, pp. 293–308, http://dx.doi.org/10.1007/11767480_20.

[13] J.K. Liu, M.H. Au, W. Susilo, Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model: Extended abstract, in: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, Association for Computing Machinery, 2007, pp. 273–283, http://dx.doi.org/10.1145/1229285.1266994.

[14] X. Huang, Y. Mu, W. Susilo, D.S. Wong, W. Wu, Certificateless signature revisited, in: Information Security and Privacy, Springer, 2007, pp. 308–322, http://dx.doi.org/10.1007/978-3-540-73458-1_23.

[15] Y. Yuan, C. Wang, Certificateless signature scheme with security enhanced in the standard model, Inf. Process. Lett. 114 (9) (2014) 492–499, http://dx.doi.org/10.1016/j.ipl.2014.04.004.

[16] Y.-H. Hung, S.-S. Huang, Y.-M. Tseng, T.-T. Tsai, Certificateless signature with strong unforgeability in the standard model, Informatica 26 (4) (2015) 663–684.

[17] Q. Mei, Y. Zhao, H. Xiong, A new provably secure certificateless signature with revocation in the standard model, Informatica 30 (4) (2019) 711–728.

[18] Y.-F. Tseng, C.-I. Fan, C.-W. Chen, Top-level secure certificateless signature scheme in the standard model, IEEE Syst. J. 13 (3) (2019) 2763–2774, http://dx.doi.org/10.1109/JSYST.2018.2889780.

[19] C. Wu, H. Huang, K. Zhou, C. Xu, Cryptanalysis and improvement of a new certificateless signature scheme in the standard model, China Commun. 18 (1) (2021) 151–160, http://dx.doi.org/10.23919/JCC.2021.01.013.

[20] L. Zhou, C. Su, K.-H. Yeh, A lightweight cryptographic protocol with certificateless signature for the Internet of Things, ACM Trans. Embed. Comput. Syst. 18 (3) (2019) http://dx.doi.org/10.1145/3301306.

[21] X. Yang, X. Pei, G. Chen, T. Li, M. Wang, C. Wang, A strongly unforgeable certificateless signature scheme and its application in IoT environments, Sensors 19 (12) (2019) http://dx.doi.org/10.3390/s19122692.

[22] A.A. Addobea, J. Hou, Q. Li, H. Li, MHCOOS: An offline-online certificateless signature scheme for M-health devices, Sec. Commun. Netw. 2020 (2020) http://dx.doi.org/10.1155/2020/7085623.

[23] D. He, J. Chen, R. Zhang, An efficient and provably-secure certificateless signature scheme without bilinear pairings, Int. J. Commun. Syst. 25 (11) (2012) 1432–1442, http://dx.doi.org/10.1002/dac.1330.

[24] M. Tian, L. Huang, Cryptanalysis of a certificateless signature scheme without pairings, Int. J. Commun. Syst. 26 (11) (2013) 1375–1381, http://dx.doi.org/10.1002/dac.2310.

[25] J.-L. Tsai, N.-W. Lo, T.-C. Wu, Weaknesses and improvements of an efficient certificateless signature scheme without using bilinear pairings, Int. J. Commun. Syst. 27 (7) (2014) 1083–1090, http://dx.doi.org/10.1002/dac.2388.

[26] J. Zhang, J. Mao, An efficient RSA-based certificateless signature scheme, J. Syst. Softw. 85 (3) (2012) 638–642, http://dx.doi.org/10.1016/j.jss.2011.09.036.

[27] L. Wang, K. Chen, Y. Long, H. Wang, An efficient pairing-free certificateless signature scheme for resource-limited systems, Sci. China Inf. Sci. 60 (11) (2017) 119102, http://dx.doi.org/10.1007/s11432-015-0367-6.

[28] P. Gong, P. Li, Further improvement of a certificateless signature scheme without pairing, Int. J. Commun. Syst. 27 (10) (2014) 2083–2091, http://dx.doi.org/10.1002/dac.2457.

[29] L. Wang, K. Chen, Y. Long, X. Mao, H. Wang, A modified efficient certificateless signature scheme without bilinear pairings, in: 2015 International Conference on Intelligent Networking and Collaborative Systems, 2015, pp. 82–85, http://dx.doi.org/10.1109/INCoS.2015.10.

[30] K.-H. Yeh, K.-Y. Tsai, R.-Z. Kuo, T.-C. Wu, Robust certificateless signature scheme without bilinear pairings, in: 2013 International Conference on IT Convergence and Security (ICITCS), 2013, pp. 1–4, http://dx.doi.org/10.1109/ICITCS.2013.6717878.

[31] K.-H. Yeh, K.-Y. Tsai, C.-Y. Fan, An efficient certificateless signature scheme without bilinear pairings, Multimedia Tools Appl. 74 (16) (2015) 6519–6530, http://dx.doi.org/10.1007/s11042-014-2154-4.

[32] K.-H. Yeh, C. Su, K.-K.R. Choo, W. Chiu, A novel certificateless signature scheme for smart objects in the Internet-of-Things, Sensors 17 (5) (2017) http://dx.doi.org/10.3390/s17051001.

[33] X. Jia, D. He, Q. Liu, K.-K.R. Choo, An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment, Ad Hoc Networks 71 (2018) 78–87, http://dx.doi.org/10.1016/j.adhoc.2018.01.001.

[34] H. Du, Q. Wen, S. Zhang, M. Gao, A new provably secure certificateless signature scheme for Internet of Things, Ad Hoc Netw. 100 (2020) 102074, http://dx.doi.org/10.1016/j.adhoc.2020.102074.

[35] A. Karati, S. Hafizul Islam, G. Biswas, A pairing-free and provably secure certificateless signature scheme, Inform. Sci. 450 (2018) 378–391, http://dx.doi.org/10.1016/j.ins.2018.03.053.

[36] N. Pakniat, B.A. Vanda, Cryptanalysis and improvement of a pairing-free certificateless signature scheme, in: 2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), 2018, pp. 1–5, http://dx.doi.org/10.1109/ISCISC.2018.8546984.

[37] G. Thumbur, G.S. Rao, P.V. Reddy, N. Gayathri, D.R.K. Reddy, Efficient pairing-free certificateless signature scheme for secure communication in resource-constrained devices, IEEE Commun. Lett. 24 (8) (2020) 1641–1645, http://dx.doi.org/10.1109/LCOMM.2020.2988818.
[38] Z. Xu, M. Luo, M.K. Khan, K.-K.R. Choo, D. He, Analysis and improvement of a certificateless signature scheme for resource-constrained scenarios, IEEE Commun. Lett. 25 (4) (2021) 1074–1078, http://dx.doi.org/10.1109/LCOMM.2020.3042648.
[39] H. Xiong, Q. Mei, Y. Zhao, Efficient and provably secure certificateless parallel key-insulated signature without pairing for iIoT environments, IEEE Syst. J. 14 (1) (2020) 310–320, http://dx.doi.org/10.1109/JSYST.2018.2890126.
[40] Y. Zhan, B. Wang, R. Lu, Cryptanalysis and improvement of a pairing-free certificateless aggregate signature in healthcare wireless medical sensor networks, IEEE Internet Things J. 8 (7) (2021) 5973–5984, http://dx.doi.org/10.1109/JIOT.2020.3033337.
[41] L. Deng, Y. Yang, R. Gao, Certificateless designated verifier anonymous aggregate signature scheme for healthcare wireless sensor networks, IEEE Internet Things J. 8 (11) (2021) 8897–8909, http://dx.doi.org/10.1109/JIOT.2021.3056097.
[42] Q. Mei, H. Xiong, J. Chen, M. Yang, S. Kumari, M.K. Khan, Efficient certificateless aggregate signature with conditional privacy preservation in IoV, IEEE Syst. J. 15 (1) (2021) 245–256, http://dx.doi.org/10.1109/JSYST.2020.2966526.
[43] D. Pointcheval, J. Stern, Security arguments for digital signatures and blind signatures, J. Cryptol. 13 (3) (2000) 361–396, http://dx.doi.org/10.1007/s001450010003.

Dengmei Xiang was born in Sichuan, China, in 1997. She is currently pursuing the M.S. degree with Xidian University, Xi'an, China. Her research interests include blockchain, Internet of Things, and digital signature. (Email: dengmei1093@163.com)

Xuelian Li received the Ph.D degree in cryptography in 2010. She is now an associate professor in School of Mathematics and Statistics, Xidian University. Her research interests focus on information security and blockchain. (Email: xuelian202@163.com, xlli@mail.xidian.edu.cn)

Juntao Gao received the Ph.D degree in cryptography in 2006. He is now an associate professor in School of Telecommunication and Engineering, Xidian University. His research interests focus on pseudorandom sequences and blockchain. (Email: jtgao@mail.xidian.edu.cn)

Xiachuan Zhang was born in Shaanxi Province, China, in 1996. She is currently pursuing the M.S.degree with Xidian University, Xi'an, China. Her main research interests include cryptography, Blockchain. (Email: xiachuan666@gmail.com)