Analysis and Improvement of a Certificateless Signature Scheme for Resource-Constrained Scenarios

Zhiyan Xu, Min Luo, Muhammad Khurram Khan, *Senior Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, and Debiao He

***Abstract*—Recently, Thumbur *et al*. introduced a paring-free certifcateless signature scheme to resolve security and efficiency issues in resource-constrained devices and claimed it was secure. However, we find that their scheme is vulnerable to signature forgery attack. To solve the above security challenges, we present a new pairing-free certificateless signature scheme and then formally prove its security under the ECDLP assumption, and finally we conduct a performance analysis and comparison. Security analysis and performance evaluation demonstrate that our new proposal can enjoy a higher level of security with a lower computation cost and communication cost, it is more practical in resource-constrained scenarios.**

***Index Terms*—Certificateless signature, forgery attack, secure and efficient, resource-constrained.**

I. Introduction

IN MANY practical application scenarios (such as electronic medical systems, healthcare wireless sensor networks, vehicular ad hoc networks etc.), smart devices are connected to different network systems via Internet to achieve the purpose of data collecting and sharing. However, when data is transmitted through public network channels, it is vulnerable to malicious attacks, so data integrity and privacy have become a very concerned issue in all walks of life. Digital signatures can ensure the integrity and authenticity of data [1].

In traditional certificate-based public key cryptography (TC-PKC) system [2], the user will first generate his own key pair, and then the public key is sent to a trusted certificate authority(CA) to apply for a certificate. However, TC-PKC system faces various certificate management problems. Identity-based public key cryptography(ID-PKC) system [3] can eliminate the certificate management issue in TC-PKC system, but since the user's private key is produced independently by key generation center (KGC), there is an inherent key escrow issue in ID-PKC system. To solve the above problems, the concept of certificateless public key cryptography (CL-PKC) was proposed [4]. In CL-PKC system, the user's private key is jointly produced by the user and KGC, which can eliminate the issues of certificate management in TC-PKC and key escrow in ID-PKC.

Following the work of [4], many researchers have began to conduct theoretical research on certificateless signature(CLS) [5]–[7]. With the popularity of wireless networks and the use of smart devices(such devices are limited in terms of computing power, storage capacity, bandwidth and other resources), the schemes mentioned above cannot meet the application requirements of resource-constrained scenarios due to its low efficiency (using high-cost operations such as pairing operations or mapping to point hash functions) [8], [9]. Therefore, it is urgent to design a secure and efficient CLS scheme that can meet the actual application requirements.

To reduce the computation cost, many pairing-firee CLS schemes have been proposed [10]–[13]. Gong and Li [10] introduced a new CLS scheme. However, Yeh *et al.* [11] demonstrated that the scheme in [10] was not secure. Subsequently, Wang *et al.* [12] also proposed a CLS scheme, but its performance is not good. In 2018, Jia *et al.* [13] presented a novel CLS scheme, but it was soon discovered by Du *et al.* [14] that the proposal can not resist the attacks of Type-I adversaries. Recently, Thumbur *et al.* [15] presented a new CLS scheme and claimd it was secure, unfortunately, we find their scheme is vulnerable to signature forgery attack and cannot achieve its stated purpose.

A. Our Research Contributions

To solve user's data security and efficiency issues in resource-constrained scenarios, we design a new CLS scheme and the main contributions are summarized as follows.

- we first conduct a cryptographic analysis on Thumbur *et al.*'s scheme [15], and point out that the scheme cannot resist signature forgery attacks from Type-I adversaries.
- We propose a new CLS scheme without pairing operations and formally prove its security.

Manuscript received October 3, 2020; revised November 22, 2020; accepted December 1, 2020. Date of publication December 4, 2020; date of current version April 9, 2021. The work was supported in part by the National Key Research and Development Program of China (No. 2018YFC1604000), the National Natural Science Foundation of China (Nos.61902115, 61972294, 61932016), the Special Project on Science and Technology Program of Hubei Provience (No. 2020AEA013), the Natural Science Foundation of Hubei Province (No. 2020CFA052) and the Wuhan Municipal Science and Technology Project (No. 2020010601012187). Muhammad Khurram Khan was supported by the Researchers Supporting Project of King Saud University (No. RSP-2020/12). Kim-Kwang Raymond Choo was supported by the Cloud Technology Endowed Professorship. The associate editor coordinating the review of this letter and approving it for publication was T. Han. *(Corresponding author: Min Luo.)*

Zhiyan Xu is with the Hubei Education Cloud Service Engineering Technology Research Center, College of Computer, Hubei University of Education, 430205 Wuhan, China (e-mail: csxzy@whu.edu.cn).

Min Luo is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: mluo@whu.edu.cn).

Muhammad Khurram Khan is with the Center of Excellence in Information Assurance (CoEIA), College of Computer and Information Sciences, King Saud University, Riyadh 11653, Saudi Arabia (e-mail: mkhurram@ksu.edu.sa).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249-1644 USA (e-mail: raymond.choo@fulbrightmail.org).

Debiao He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/LCOMM.2020.3042648

- We conduct a performance analysis and comparison with several other pairing-free CLS schemes.

B. Organization of the Letter

The remainder of the letter is arranged as below. Sections II presents the preliminaries. Sections III revisits Thumbur *et al.*'s scheme [15], and Sections IV presents a cryptographic analysis on their scheme. Our new CLS scheme is demonstrated in Section V. Security proof and performance evaluation are described in Sections VI and VII. Finally, we give a summary of this letter in the last section.

II. Preliminaries

Elliptic curve discrete logarithm problem and syntax of CLS scheme are the same as described in Section 2 of Thumbur *et al.*'s scheme [15].

III. Revisiting Thumbur *et al.*'s Scheme

Thumbur *et al.*'s scheme [15] composes six algorithms described as follows.

Setup. Performed by KGC to complete system initialization.

1. Input the security parameter $k \in Z^+$, select q order additive group G, where P is a generator of G.
2. Select $s \in Z_q^*$ as the system master key, calculate $P_{pub} = sP$ as the system public key, and define three hash functions: H_i:$\{0,1\}^* \to Z_q^*$, where $i = 1, 2, 3$.
3. Publish system parameter list $params = (k, q, G, P, P_{pub}, H_i)$ and keep s in secret.

Set Partial Private key. Performed by KGC to produce the user's partial private key with identity $ID_i \in \{0,1\}^*$.

1. Select a random value $r_i \in Z_q^*$ and calculate $R_i = r_i P$.
2. Calculate $h_{1i} = H_1(ID_i, R_i, P_{pub})$ and $d_i = r_i + sh_{1i}$ mod q.
3. Secretly send the partial private key (d_i, R_i) to the user, and it can be verified by verifying $d_i P = R_i + h_{1i} P_{pub}$.

Set Secret Value. Performed by user to produce the secret value.

1. Randomly choose $x_i \in Z_q^*$ as his secret value.
2. Compute $X_i = x_i P$.

Set Public/Private key. Performed by user to produce the public-private key pair.

1. Compute $h_{2i} = H_2(ID_i, X_i)$ and $Q_i = R_i + h_{2i} X_i$.
2. Set $PK_i = (Q_i, R_i)$ as his public key and $SK_i = (d_i, x_i)$ as his private key.

Sign. Performed by signer to produce the signature of the message $m_i \in \{0,1\}^*$.

1. Randomly choose $u_i \in Z_q^*$ and compute $U_i = u_i P$.
2. Compute $X_i = x_i P$, $h_{2i} = H_2(ID_i, X_i)$, $h_{3i} = H_3(ID_i, m_i, PK_i, U_i)$ and $v_i = u_i + h_{3i}(d_i + h_{2i} x_i)$ mod q.
3. Output the signature $\sigma_i = (U_i, v_i)$.

Verify. Performed by the verifier to determine the validity of the signature.

1. Input $params$, ID_i, PK_i, $\sigma_i = (U_i, v_i)$ and m_i.
2. Compute $h_{1i} = H_1(ID_i, R_i, P_{pub})$ and $h_{3i} = H_3(ID_i, m_i, PK_i, U_i)$.
3. Verify the following equation (1)
$$v_i P = U_i + h_{3i}(Q_i + h_{1i} P_{pub}) \tag{1}$$
If equation (1) holds, emit 1; Otherwise, emit 0.

IV. Previously Unknown Vulnerability of Thumbur *et al.*'s CLS Scheme

Thumbur *et al.*'s CLS signature scheme [15] cannot resist the attacker of Type-I in the CLS security model, and the specific description is as below.

Setup. The challenger C performs $Setup$ algorithm to produce the system parameter list $params$ and master key s, then C returns $params$ to the adversary A_1 and keeps s secret.

Replace public key. A_1 completes the public key replacement by performing the following operations.

1. Calculate $h_{1i} = H_1(ID_i, R_i, P_{pub})$, where ID_i, R_i and P_{pub} are public.
2. Randomly choose $t_i \in Z_q^*$.
3. Calculate $Q_i^* = t_i P - h_{1i} P_{pub}$ to replace the original public key Q_i of the user ID_i.

Signature forgery. To forge the signature of the user ID_i on the message m_i, A_1 performs the following operations.

1. Select a random value $u_i^* \in Z_q^*$ and calculate $U_i^* = u_i^* P$.
2. Calculate $h_{3i}^* = H_3(ID_i, m_i, PK_i^*, U_i^*)$, where ID_i, m_i, PK^* and U_i^* are public.
3. Calculate $v_i^* = u_i^* + h_{3i}^* t_i$ mod q.
4. Output the forged signature $\sigma_i^* = (U_i^*, v_i^*)$.

Verify. It is easy to determine that the forged signature σ_i^* is valid, the details are as follows.

1. Compute $h_{1i} = H_1(ID_i, R_i, P_{pub})$ and $h_{3i}^* = H_3(ID_i, m_i, PK_i^*, U_i^*)$
2. Substitute v_i^* into the left side of equation (1), we have
$$\begin{aligned} v_i^* P &= (u_i^* + h_{3i}^* t_i) P \\ &= u_i^* P + h_{3i}^* t_i P \end{aligned}$$
3. Because the adversary A_1 replaced the signer's public key Q_i with $Q_i^* = t_i P - h_{1i} P_{pub}$ in **Replace public key** phase, we can deduce $t_i P = Q_i^* + h_{1i} P_{pub}$, and we can get the following equation (2), that is, the forged signature can successfully pass the equation (1).
$$v_i^* P = U_i^* + h_{3i}^*(Q_i^* + h_{1i} P_{pub}) \tag{2}$$

From the above signature forgery and verification process, we can find that the forged signature is valid and the signature forgery attack of the adversary A_1 against Thumbur *et al.*'s scheme [15] is feasible.

V. Our Proposed CLS Scheme

Our CLS scheme includes six algorithms described as follows, where the list of notations and annotations is shown in table I and the scheme is also illustrated in Fig.1 to enhance readability.

Setup. Performed by KGC to complete system initialization.

1. Input the security parameter $k \in Z^+$, select q order additive group G, where P is a generator of G.

TABLE I

LIST OF NOTATIONS AND ANNOTATIONS

Notations	Annotations
CLS	Certificateless signature
KGC	Key generation center
$params$	System parameter List
k	System security parameter
s	System master key
P_{pub}	System public key
ID_i	Identity of user $\mathscr{U}_i$
PK_i	Public key of user $\mathscr{U}_i$
SK_i	Private key of user $\mathscr{U}_i$
m_i	Message
σ_i	Signature of user $\mathscr{U}_i$ on message m_i
A_1, A_2	Type-I and Type-II adversaries

2. Select $s \in Z_q^*$ as the system master key, caculate $P_{pub} = sP$ as the system public key.
3. Define three hash functions: h_1, h_2, h_3:$\{0,1\}^* \rightarrow Z_q^*$.
4. Publish the system parameter list $params = (k, q, G, P, P_{pub}, h_1, h_2, h_3)$ and keep s in secret.

Set Partial Private key. Performed by KGC to produce the user $\mathscr{U}_i$'s partial private key with identity $ID_i \in \{0,1\}^*$.

1. Select a random value $r_i \in Z_q^*$, compute $R_i = r_i P$ and keep it public.
2. Compute $\alpha_i = h_1(ID_i, R_i, P_{pub})$, $d_i = r_i + s\alpha_i$ mod q.
3. Secretly send the partial private key d_i to $\mathscr{U}_i$, where $\mathscr{U}_i$ can verify its validity by verifying $d_i P = R_i + \alpha_i P_{pub}$.

Set Secret Value. Performed by user to produce the secret value.

1. Randomly choose $x_i \in Z_q^*$ as his secret value.
2. Compute $X_i = x_i P$.

Set Public/Private key. Performed by user to produce the public-private key pair.

1. Set $PK_i = (R_i, X_i)$ as his public key.
2. Set $SK_i = (d_i, x_i)$ as his private key.

Sign. Performed by signer $\mathscr{U}_i$ to produce the signature of message $m_i \in \{0,1\}^*$.

1. Input system parameters $params$, signer's identity ID_i, signing key pair (PK_i, SK_i) and message m_i.
2. Choose $u_i \in Z_q^*$ and compute $U_i = u_i P$.
3. Compute $\beta_i = h_2(ID_i, X_i, P_{pub})$, $\gamma_i = h_3(ID_i, m_i, PK_i, U_i)$ and $v_i = d_i + \gamma_i u_i + \beta_i x_i$ mod q.
4. Set $\sigma_i = (U_i, v_i)$ as the signature of message m_i.

Verify. Performed by the verifier to determine the validity of the signature.

1. Input system parameters $params$, signer's identity ID_i and his public key PK_i, message m_i and its signature $\sigma_i = (U_i, v_i)$.
2. Compute the values $\alpha_i = h_1(ID_i, R_i)$, $\beta_i = h_2(ID_i, PK_i, P_{pub})$ and $\gamma_i = h_3(ID_i, m_i, PK_i, U_i)$.
3. Verify the following equation (3)

$$v_i P = R_i + \alpha_i P_{pub} + \beta_i X_i + \gamma_i U_i \tag{3}$$

If equation (3) holds, emit 1; Otherwise, emit 0.

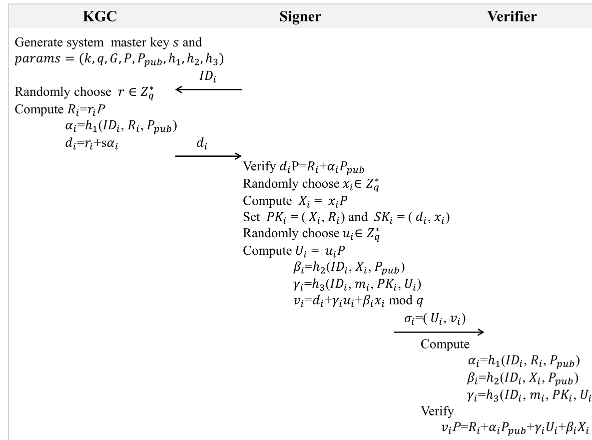

Fig. 1. Our proposed CLS shceme.

VI. ANALYSIS OF OUR CLS SCHEME

A. Correctness

Suppose $\sigma_i = (U_i, v_i)$ is the signature produced by our proposed CLS scheme, it is easy to verify that equation (3) is established, and details are as follows.

$$\begin{aligned} v_i P &= (d_i + \gamma_i u_i + \beta_i x_i)P \\ &= (r_i + s\alpha_i)P + \gamma_i u_i P + \beta_i x_i P \\ &= R_i + \alpha_i P_{pub} + \beta_i X_i + \gamma_i U_i \end{aligned}$$

B. Provable Security

In this section, we demonstrate that our presented CLS scheme is existential unforgeable against Type-I and Type-II adversaries as defined in [5]. The threat model and security proof of our CLS scheme are described as follows.

Threat model. Adversary cannot obtain the system master private key when it can replace the user public key, and the adversary cannot replace the user public key when it can obtain the system master private key. Furthermore, the adversary can listen to the information on all open channels and can cut off the communication of one party.

Theorem 1: In the random oracle model, if the Type-I adversary A_1 can successfully forge a signature with non-negligible probability, then the challenger C can solve the ECDLP problem with non-negligible probability.

Proof: Suppose A_1 is an adversary of Type-I. Challenger C calls A_1 to solve ECDLP in a polynomial time. Assuming that $(P, P_1 = sP)$ is an instance of ECDLP, the ultimate goal of C is to compute the value of s.

Setup. C executes $Setup$ algorithm to produce the system parameter list $params$, and returns it to A_1. Let $P_{pub} = sP$, randomly select ID_{gt} as the challenged identity. For ease of simulation, during the query, C maintains six lists L_{h_1}, L_{h_2}, L_{h_3}, L_d, L_x and L_{PK} to store the query values related to h_1, h_2, h_3, d_i, x_i, and PK_i. All lists are initialized to empty.

h_1 **queries.** A_1 inputs (ID_i, R_i), if (ID_i, R_i, α_i) exists in list L_{h_1}, then C directly returns α_i; Otherwise C selects a random value $\alpha_i \in Z_q^*$, returns to A_1 and stores (ID_i, R_i, α_i) to the list L_{h_1}.

h_2 **queries.** A_1 inputs (ID_i, PK_i, P_{pub}), if $(ID_i, PK_i, P_{pub}, \beta_i)$ exists in list L_{h_2}, then C directly returns β_i; Otherwise C randomly selects $\beta_i \in Z_q^*$, returns to A_1 and stores $(ID_i, PK_i, P_{pub}, \beta_i)$ to the list L_{h_2}.

h_3 **queries.** A_1 inputs (ID_i, m_i, PK_i, U_i), if $(ID_i, m_i, PK_i, U_i, \gamma_i)$ exists in list L_{h_3}, then C directly returns γ_i; Otherwise, C selects a random value $\gamma_i \in Z_q^*$, returns to A_1 and stores $(ID_i, m_i, PK_i, U_i, \gamma_i)$ to the list L_{h_3}.

Partial private key oracle. When receiving a partial private key query from A_1 regarding user U_i with his identity ID_i, C first determines whether $ID_i = ID_{gt}$ holds, if it holds, then C aborts. Otherwise, C traverses the list L_d, if (ID_i, d_i) is present, it returns d_i; Otherwise, C selects random values d_i, $\alpha_i \in Z_q^*$, computes $R_i = d_i P - \alpha_i P_{pub}$ and sets $\alpha_i = H_1(ID_i, R_i)$, C adds (ID_i, R_i, α_i) to L_{h_1}, (ID_i, d_i) to L_d and returns d_i to A_1.

Reveal secret value oracle. When receiving a sectet value query from A_1 regarding user U_i with his identity ID_i, C first determines whether $ID_i = ID_{gt}$ holds, if it holds, C aborts; Otherwise, C traverses the list L_x, if (ID_i, x_i) is present, it returns x_i; Otherwise, C selects a random value $x_i \in Z_q^*$, stores (ID_i, x_i) to L_x and returns x_i to A_1.

Replace public key oracle. When receiving a replace public key query from A_1 regarding user U_i with the identity ID_i, C first traverses (ID_i, PK_i) from the list L_{PK}, and replace it with (ID_i, PK_i^*).

Signing oracle. When receiving a sign query from A_1 regarding (ID_i, m_i), C first determines whether $ID_i = ID_{gt}$ holds, if it holds, C randomly selects v_i, α_i, β_i, $\gamma_i \in Z_q^*$, computes $U_i = \gamma_i^{-1}(v_i P - \alpha_i P_{pub} - \beta_i X_i - R_i)$ and returns to A_1; Otherwise, C randomly selects $u_i \in Z_q^*$, traverses lists L_x, L_d, L_{h_2}, L_{h_3} to get x_i, d_i, β_i, γ_i, and computes $U_i = u_i P$ and $v_i = d_i + \gamma_i u_i + \beta_i x_i \bmod q$, and returns σ_i to A_1.

Forgery. Finally, A_1 outputs the forged signature $\sigma_{gt}^* = (U_{gt}^*, v_{gt}^*)$ on pairs (ID_{gt}^*, m_i^*), if σ_{gt}^* is valid, the forged signature should make the verify equation (3) hold, we have,

$$v_{gt}^* P = R_{gt}^* + \alpha_{gt}^* P_{pub} + \beta_{gt}^* X_i^* + \gamma_{gt}^* U_{gt}^* \tag{4}$$

According to the forgery lemma, A_1 can generate another valid signature $\sigma'_{gt} = (U_{gt}^*, v'_{gt})$ in the same way. By selecting a different h_1 and repeating the above process, we have,

$$v'_{gt} P = R_{gt}^* + \alpha'_{gt} P_{pub} + \beta_{gt}^* X_i^* + \gamma_{gt}^* U_{gt}^* \tag{5}$$

Using equation (4) minus (5), we derive the following derivation,

$$\begin{aligned} v_{gt}^* P - v'_{gt} P &= R_{gt}^* + \alpha_{gt}^* P_{pub} + \beta_{gt}^* X_i^* + \gamma_{gt}^* U_{gt}^* \\ &\quad - (R_{gt}^* + \alpha'_{gt} P_{pub} + \beta_{gt}^* X_i^* + \gamma_{gt}^* U_{gt}^*) \\ &= (\alpha_{gt}^* - \alpha'_{gt}) P_{pub} \\ &= (\alpha_{gt}^* - \alpha'_{gt}) sP \end{aligned}$$

Further, according to the above equation we can calculate $s = (v_{gt}^* - v'_{gt})(\alpha_{gt}^* - \alpha'_{gt})^{-1}$.

However, this contradicts the ECDLP assumption. Namely, the signature $\sigma_i = (U_i, v_i)$ cannot be forged by A_1.

Theorem 2: In the random oracle model, if the Type-II adversary A_2 can successfully forge a signature with non-negligible probability, then the challenger C can solve the ECDLP problem with non-negligible probability.

TABLE II
RUNNING TIME OF DIFFERENT OPERATIONS(ms)

Notations	Operations	Running time
T_{ma}	a modular addition operation	0.0008
T_{mm}	a modular multiplication operation	0.0011
T_{inv}	a modular inversion operation	0.1888
T_{hs}	a general hash operation	0.0001
T_{pa}	a point addition operation	0.0018
T_{pm}	a point multiplication operation	0.4421

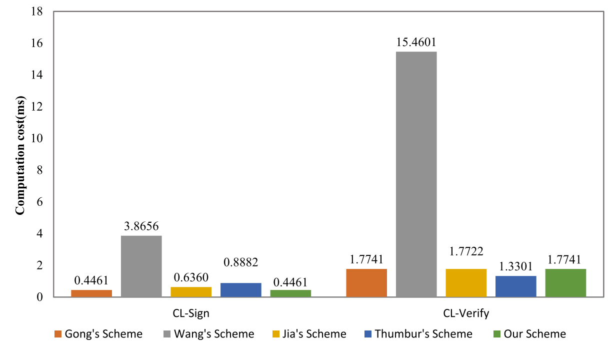

Fig. 2. Comparison of the computation costs.

Proof: The formal proof process is similar to Theorem 1.

VII. PERFORMANCE EVALUATION

We evaluate the performance of our new proposal and other four paring-free CLS schemes [10], [12], [13], [15]. For comparable security with 1024 bits level RSA, we choose a non-singular elliptic curve $E: y^2 = x^3 + ax + b \bmod q$, and a, $b \in Z_q^*$, G is an additive group with order q on E, p and q are both prime numbers with a length of 160 bits. We run the simulation experiment using the MIRACL library [16] on a personal computer(Intel core with I7-4770@3.4GHz CPU, 4GB random memory, and Windows7 operating system). The running time of different operations is shown in table II.

A. Computation Costs

AS shown as results in Fig.2 and Table III, we can observe that Wang *et al.*'s scheme [12] and our scheme are secure, and the other three schemes [10], [13], [15] have security flaws. From the computing performance point of view, our scheme requires to perform one point multiplication operation, two general hash operations, two modular addition operations, and two modular multiplication operations in $Sign$ phase, and it requires to perform four point multiplication operations, three point addition operations, three general hash operations in $Verify$ phase. The total computation cost of our scheme is 2.2202 ms, which is reduced by 88.51% compared with Wang *et al.*'s scheme [12], has the same as that of Gong *et al.*'s scheme [10], reduced by 7.81% compared with Jia *et al.*'s scheme [13], increased by 0.08% compared with Thumbur *et al.*'s scheme [15] in terms of total computation cost.

TABLE III
COMPUTATION COST COMPARISON(ms)

Scheme	Sign (ms)	Verify (ms)	Total (ms)	security
Gong $et\ al.$[10]	$1T_{pm}+2T_{hs}+2T_{mm}+2T_{ma}\approx 0.4461$	$4T_{pm}+3T_{pa}+3T_{hs}\approx 1.7741$	2.2202	No[11]
Wang $et\ al.$[12]	$1T_{exp}+1T_{hs}+1T_{mm}+1T_{ma}\approx 3.8656$	$4T_{exp}+2T_{hs}+5T_{mm}\approx 15.4601$	19.3257	Yes
Jia $et\ al.$[13]	$1T_{pm}+2T_{hs}+3T_{mm}+2T_{ma}+1T_{inv}\approx 0.6360$	$4T_{pm}+2T_{pa}+2T_{hs}\approx 1.7722$	2.4082	No[14]
Thumbur $et\ al.$[15]	$2T_{pm}+2T_{hs}+2T_{mm}+2T_{ma}\approx 0.8882$	$3T_{pm}+2T_{pa}+2T_{hs}\approx 1.3301$	2.2183	No
Our scheme	$1T_{pm}+2T_{hs}+2T_{mm}+2T_{ma}\approx 0.4461$	$4T_{pm}+3T_{pa}+3T_{hs}\approx 1.7741$	2.2202	Yes

TABLE IV
COMMUNICATION COST COMPARISON(bit)

Scheme	Gong $et\ al.$[10]	Wang $et\ al.$[12]	Jia $et\ al.$[13]	Thumbur $et\ al.$[15]	Our scheme
Signature Length	$2\|G\|+\|Z_q^*\|$	$\|Z_p^*\|+\|Z_q^*\|$	$\|G\|+\|Z_q^*\|$	$\|G\|+\|Z_q^*\|$	$\|G\|+\|Z_q^*\|$
Communication Cost	800	1184	480	480	480

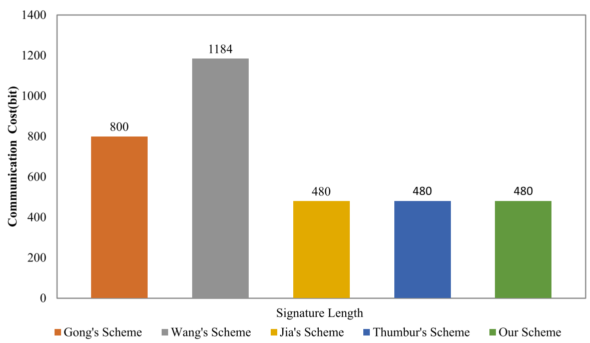

Fig. 3. Comparison of the communication costs.

B. Communication Costs

We consider the signature length to evaluate the communication costs of our scheme and the other four schemes [10], [12], [13], [15]. As shown in Fig.3 and Table IV, we can find that the signature length is 480 bits in our scheme, which is reduced by 40.00% compared with Gong *et al.*'s scheme [10], reduced by 59.46% compared with Wang *et al.*'s scheme [12], and has the same as that of Jia *et al.*'s scheme [13] and Thumbur *et al.*'s scheme [15] in terms of communication cost.

VIII. CONCLUSION

CLS has been widely applied in many feilds because of its natural advantages. To design a secure and efficient CLS scheme to meet the application demands of resource-constrained scenarios, we first conduct a security analysis on Thumbur *et al.*'s CLS scheme [15] and demonstrate their CLS scheme to be vulnerable against signature forgery attacks. Then we present a new CLS scheme without pairing to fix security flaw in Thumbur *et al.*'s CLS scheme. Finally, security proof and performance evaluation of new scheme are carried out. Results demonstrate that our new proposal can achieve a higher level of security assurance with lower computation and communication costs. Thus, our scheme is more suitable for real-world deployment, especially for resource-constrained application scenarios.

REFERENCES

[1] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of Internet of Things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.

[2] G. K. Verma, B. B. Singh, N. Kumar, and V. Chamola, "CB-CAS: Certificate-based efficient signature scheme with compact aggregation for industrial Internet of Things environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2563–2572, Apr. 2020.

[3] X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng, "Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage," *Inf. Sci.*, vol. 494, pp. 193–207, Aug. 2019.

[4] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2003, pp. 452–473.

[5] Y.-C. Chen and R. Tso, "A survey on security of certificateless signature schemes," *IETE Tech. Rev.*, vol. 33, no. 2, pp. 115–121, Mar. 2016.

[6] A. Karati, S. H. Islam, and M. Karuppiah, "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3701–3711, Aug. 2018.

[7] N. B. Gayathri, T. Gowri, and P. V. Reddy, "Secure and efficient certificateless aggregate signature scheme from bilinear pairings," *Inf. Secur. J, Global Perspective*, vol. 28, no. 6, pp. 149–163, Nov. 2019.

[8] Y. Zhang, R. H. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5099–5108, Sep. 2019.

[9] Z. Xu, D. He, P. Vijayakumar, K.-K.-R. Choo, and L. Li, "Efficient NTRU lattice-based certificateless signature scheme for medical cyber-physical systems," *J. Med. Syst.*, vol. 44, no. 5, pp. 1–8, May 2020.

[10] P. Gong and P. Li, "Further improvement of a certificateless signature scheme without pairing," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 2083–2091, Oct. 2014.

[11] K.-H. Yeh, K.-Y. Tsai, and C.-Y. Fan, "An efficient certificateless signature scheme without bilinear pairings," *Multimedia Tools Appl.*, vol. 74, no. 16, pp. 6519–6530, Aug. 2015.

[12] L. Wang, K. Chen, Y. Long, and H. Wang, "An efficient pairing-free certificateless signature scheme for resource-limited systems," *Sci. China Inf. Sci.*, vol. 60, no. 11, Nov. 2017, Art. no. 119102.

[13] X. Jia, D. He, Q. Liu, and K.-K.-R. Choo, "An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment," *Ad Hoc Netw.*, vol. 71, pp. 78–87, Mar. 2018.

[14] H. Du, Q. Wen, S. Zhang, and M. Gao, "A new provably secure certificateless signature scheme for Internet of Things," *Ad Hoc Netw.*, vol. 100, Apr. 2020, Art. no. 102074.

[15] G. Thumbur, G. S. Rao, P. V. Reddy, N. B. Gayathri, and D. V. R. O. Reddy, "Efficient pairing-free certificateless signature scheme for secure communication in resource-constrained devices," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1641–1645, Aug. 2020.

[16] M. Scott, "MIRACL—A multiprecision integer and rational arithmetic C/C++ library," Shamus Softw. Ltd., Dublin, Ireland, Tech. Rep., 2003. [Online]. Available: http://www.shamus.ie