Efficient Pairing-Free Certificateless Signature Scheme for Secure Communication in Resource-Constrained Devices

Gowri Thumbur, *Senior Member, IEEE*, G. Srinivasa Rao, P. Vasudeva Reddy, N. B. Gayathri, and D. V. Rama Koti Reddy

***Abstract*— With the development of wireless communication technology, many network systems are interconnected with large number of smart devices and internet to gather and share electronic data. Due to its open nature, the data transmitted over public networks. Thus ensuring privacy and data security are of great importance. Also computing power, storage and bandwidth requirements are the main constraints in the development of many applications. To resolve these security and efficiency issues, this letter presents an efficient pairing free certificateless signature scheme. This scheme is proven secure and unforgeable. Finally, the comparative analysis shows the efficiency of our scheme.**

***Index Terms*— Elliptic curve public key cryptography, digital signatures, efficient computation and low bandwidth, secure communication, resource constrained devices.**

I. Introduction

WITH the developments of wireless communication technology, many network systems such as WSNs, VANETs, IoT etc., are interconnected with smart devices and are connected via internet to gather and share electronic data. In many open networks, the data transmitted over public networks; thus ensuring privacy and data security are of particular importance in many applications [1], [2]. The cryptographic primitive called digital signature assures the integrity and authentication of the data transmitted over the public channels. Traditional Public Key Cryptography (PKC), by Diffie and Hellman [3], and Identity-based cryptography (ID-PKC), by Shamir [4], are two different cryptographic frameworks which provides authentication and non- repudiation for digital communications. The ID-PKC eliminates the key management problems in traditional PKC. However, key escrow problem is inherent problem in ID-PKC. In 2003, Al-Riyami and Paterson [5] proposed Certificateless Public Key Cryptography (CL-PKC) in which the user's private key is a combination of a partial private key generated by the Key Generation Centre (KGC) and user's secret value. Thus CL-PKC solves the key escrow problem. Following the work of [5], many CLS schemes [6], [7] and security models for CL-based schemes have been devised [7], [6].

Manuscript received February 10, 2020; revised March 13, 2020; accepted April 14, 2020. Date of publication April 22, 2020; date of current version August 12, 2020. The associate editor coordinating the review of this letter and approving it for publication was T. Han. *(Corresponding author: P. Vasudeva Reddy.)*

Gowri Thumbur is with the Department of Electronics and Communication Engineering, Gandhi Institute of Technology and Management, Visakhapatnam 530045, India.

G. Srinivasa Rao, P. Vasudeva Reddy, and N. B. Gayathri are with the Department of Engineering Mathematics, Andhra University, Visakhapatnam 530003, India (e-mail: vasucrypto@andhrauniversity.edu.in).

D. V. Rama Koti Reddy is with the Department of Instrument Technology, Andhra University, Visakhapatnam 530003, India.

Digital Object Identifier 10.1109/LCOMM.2020.2988818

With the advancement of wireless communication technology including those for sensors and more usage of recourse constrained devices, the above mentioned schemes are not much efficient because of computing power, storage space and bandwidth capacity constraints [1], [2]. Since Elliptic Curve Cryptography (ECC) provides high security with shorter keys and hence to implement cryptographic primitives in resource constrained devices, ECC is an ideal choice [8]. However, the computation of pairing operations in ECC and map to point hash functions are very expensive. Therefore, it is necessary to propose an efficient and secure pairing free CLS scheme for resource constrained devices.

The first Pairing free CLS scheme was proposed by He *et al.* [9] in 2012. Since then many CLS schemes are designed without using pairings [10]–[18]. Tsai *et al.* [10] and Tian and Huang [11] shows that the scheme [9] is not secure against malicious KGC attack and also presented an improved version of [9]. In 2012, Gong and Li [12] noticed that the Tsai *et al.* [10] is insecure and they proposed a real CLS scheme. In 2014, Yeh *et al.* [13] shows that the Gong and Li [12] scheme is not secure and proposed an efficient pairing free CLS scheme based on DLP. In 2015, Wang *et al.* [14] presented a modified Yeh *et al.* [13] to achieve more efficiency. In 2016, Wang *et al.* [15] presented a CLS scheme for resource limited systems. In 2017, Yeh *et al.* [16] presented a CLS scheme. Recently, in 2018, Jia *et al.* [17] proved that the scheme Yeh *et al.* [16] scheme is not secure against Super Type-I and Type-II adversaries and presented an improved scheme. In 2018, Karati *et al.* [18] presented a new CLS scheme using ECC without bilinear pairings in the ROM model. Nasrollah and Vanda [19] showed that the scheme Karati *et al.* (2018) is not secure against Type-I adversary.

In order to improve the computation and communication efficiency in CL-based signatures, in this letter, a new and efficient pairing free signature scheme in CL-based setting is proposed. This scheme is proven secure and unforgeable in random oracle model (ROM) under the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). We evaluate the performance of our PF-CLS scheme for different metrics. Efficiency analysis with other existing schemes is presented and it shows that the proposed scheme is much efficient.

Since most of the Internet of Things (IoT) devices possess limited computational power and communication bandwidth, one of the goals of our CLS scheme is to reduce the computation and communication overhead of resource constrained devices. Also, The Internet of Vehicles is one of the most potential areas in IoT and has wide application prospects in the field of intelligent transportation. Compared with ordinary sensors, the vehicle terminal equipment has a more com-

TABLE I
LIST OF ABBREVIATIONS AND NOTATIONS

Notation	Abbreviation
CLS	Certificateless Signature
KGC	Key Generation Center
params	System Parameters
msk	Master secret key
PPK	Partial Private key
(PK_{ID}, SK_{ID})	Public, Private key for the user ID
σ	Signature
m	Message
Adv_1, Adv_2	Type-I and Type-II adversaries

puting power and storage space. Due to the aforementioned functionalities, our CLS scheme is able to be implemented and deployed in IoT environments and Internet of Vehicles etc., where the communication devices have limited computing power, storage space, and communication bandwidth.

The remaining part of the letter is arranged as follows. Section II presents ECDLP and Syntax of our scheme. Section III presents our PF-CLS scheme. Section IV presents security analysis. Efficiency analysis is discussed in section V. Finally, Section VI concludes the letter.

II. PRELIMINARIES

A. Elliptic Curve Discrete Logarithm Problem

For a given elliptic curve group G, and $P, Q \in G$, find a scalar point $a \in Z_q^*$ such that $Q = aP$, where P is the generator of group G and Q is an element in G.

B. Syntax of PF-CLS Scheme

The proposed PF-CLS scheme consists of the following six algorithms.

Setup. KGC gives the security parameter $k \in Z^+$ as input to this algorithm and it produces *params* and msk. KGC publishes *params* and keeps msk secretly.

Set Partial Private Key. KGC gives $params$, msk, $ID \in \{0,1\}^*$ as input to this algorithm and outputs PPK. KGC gives PPK to the user securely.

Set Secret value. User $ID \in \{0,1\}^*$ selects $x_{ID} \in Z_q^*$ at random as secret value and computes $X_{ID} = x_{ID}P$.

Set Public/Private Key. User takes *params,* identity $ID \in \{0,1\}^*$ and PPK as input to this algorithm and it outputs users public and private key pair (PK_{ID}, SK_{ID}).

Signature Generation. Signer gives *params*, signers $ID \in \{0,1\}^*$, message $m \in \{0,1\}^*$ and SK_{ID} as input to this algorithm and it outputs a signature σ_{ID}.

Signature Verification. Any verifier gives message and signature tuple i.e. (m, σ_{ID}), signers $ID \in \{0,1\}^*, PK_{ID}$ and $params$ as input to this algorithm and it outputs 'Accept' if σ_{ID} is a valid; 'Reject' otherwise.

III. PROPOSED PF-CLS SCHEME

Now we will present our concrete PF-CLS scheme. As defined in Section II, our scheme consists of the following six algorithms.

Setup. On the input of a security parameter $k \in Z^+$, KGC performs the following:

1) Chooses an additive group G of elliptic curve points, prime order *q,* and P as a generator of G.
2) Chooses $s \in Z_q^*$ and computes $P_{pub} = sP$, and three secure hash functions H_i: $\{0,1\}^* \rightarrow Z_q^*$ for $i = 1, 2, 3$.
3) Finally, KGC publishes the system parameters a $params = \{q, G, P, P_{pub}, H_i\}$ keeps the master secret key $msk = s$ secretly.

Set Partial Private key. KGC generates Partial private key of a user $ID \in \{0,1\}^*$, as follows.

1) Chooses a random $r_i \in Z_q^*$ and computes $R_i = r_iP$.
2) Computes $d_i = (r_i + sh_{1i}) \mod q$, where

$$h_{1i} = H_1(ID_i, R_i, P_{pub}).$$

1) KGC gives $D_i = (d_i, R_i)$ as Partial private key (PPK) to the user through secure channel.
2) The user can validate the *PPK* by verifying the equation $d_iP = R_i + h_{1i}P_{pub}$.

Set Secret Value. The user ID_i randomly picks a number $x_i \in Z_q^*$ and set x_i as his secret value. Also the user computes $X_i = x_iP$.

Set Public / Private key. The user ID_i generates his public key PK_i and private key SK_i as follows:

1) User ID_i computes $h_{2i} = H_2(ID_i, X_i)$ and computes $Q_i = R_i + h_{2i}X_i$.
2) User sets his $PK_i = (Q_i, R_i)$ and $SK_i = (d_i, x_i)$.

Signature Generation. Signer ID_i generates a signature on a message $m \in \{0,1\}^*$, as follows.

1) Choose a random $u_i \in Z_q^*$ and computes $U_i = u_iP$.
2) Compute $h_{2i} = H_2(ID_i, X_i)$, where $X_i = x_iP$ and $h_{3i} = H_3(ID_i, m_i, PK_i, U_i), v_i = u_i + h_{3i}(d_i + h_{2i}x_i) \mod q$. The signer outputs the signature $\sigma_i = (U_i, v_i)$.

Signature Verification. On the input of $params, ID_i$, PK_i signature $\sigma_i = (U_i, v_i)$ and message m_i, any verifier can verifies the signature σ_i on m_i as follows:

1) Compute $h_{1i} = H_1(ID_i, R_i, P_{pub}), h_{3i} = H_3(ID_i, m_i, PK_i, U_i)$.
2) Verify the equation $v_iP = U_i + h_{3i}(Q_i + h_{1i}P_{pub})$. If yes, the verifier outputs Accept; else it outputs Reject.

IV. ANALYSIS OF THE PROPOSED SCHEME

A. Correctness

The correctness of the proposed scheme can be justified by verifying the above equation as follows.

$$\begin{aligned} v_iP &= (u_i + h_{3i}(d_i + h_{2i}x_i))P \\ &= (u_i + h_{3i}((r_i + sh_{1i}) + h_{2i}x_i))P \\ &\times U_i + h_{3i}(R_i + h_{1i}P_{pub} + h_{2i}X_i) \\ &= U_i + h_{3i}(Q_i + h_{1i}P_{pub}). \end{aligned}$$

B. Security Analysis

We prove that the proposed PF-CLS scheme is existential unforgeable against Type-I and Type-II adversaries as defined in the security model [6].

Theorem 1: In the Random oracle model, PF- CLS Scheme is secure and is existentially unforgeable against Type–I adversary Adv_1 under the ECDLP assumption.

Proof: Let Adv_1 is a Type–I adversary who can forge a valid signature with help of ξ. Now we construct an algorithm ξ which can solve the ECDLP using Adv_1. For $(P, Q = sP)$ of ECDLP,ξ's goal is to find *s*. Let ξ takes ID^* target identity of Adv_1 on a message m^*.

Setup Phase. Algorithm ξ sets $P_{pub} = Q = sP$, and executes the setup algorithm to generate the system parameters.

Queries Phase. In this phase Adv_1 asks a series of queries and these are answered by ξ adaptively. ξ maintains an initially empty lists $L_1, L_2, L_3, L_{Cuser}, L_{psk}$.

Queries on H_1: $H_1(ID_i, R_i, P_{pub})$. When Adv_1 asks a query on $H_1(ID_i, R_i, P_{pub}), \xi$ returns h_{1i} if such tuple exists in L_1. If not, ξ selects $h_{1i} \in Z_q^*$ and sets $H_1(ID_i, R_i, P_{pub}) = h_{1i}.\xi$ returns h_{1i} to Adv_1 and inserts $(ID_i, R_i, P_{pub}, h_{1i})$ to the list L_1.

Queries on H_2:$H_2(ID_i, X_i)$. When Adv_1 asks a H_2 query on $(ID_i, X_i), \xi$ returns h_{2i} if such tuple already exists in L_2. If not, ξ selects a random $h_{2i} \in Z_q^*$ and sets $H_2(ID_i, X_i) = h_{2i}$. ξ returns h_{2i} to Adv_1 and inserts (ID_i, X_i, h_{2i}) to the list L_2.

Queries on H_3:$H_3(ID_i, m_i, PK_i, U_i)$. When Adv_1 asks H_3 query on $(ID_i, m_i, PK_i, U_i), \xi$ returns h_{3i} if it exists in L_3. Otherwise, ξ sets $H_3(ID_i, m_i, PK_i, U_i) = h_{3i}.\xi$ returns h_{3i} to Adv_1 and inserts $(ID_i, m_i, PK_i, U_i, h_{3i})$ to the list L_3.

Reveal Partial Secret key Oracle $PSK(ID_i)$. When Adv_1 asks a query on $PSK(ID_i), \xi$ returns $D_i = (d_i, R_i)$, if it already exists in L_{psk}. If $ID_i = ID^*, \xi$ aborts. Otherwise ξ chooses $a_i, b_i \in Z_q^*$ and sets $d_i = a_i, H_1(ID_i, R_i, P_{pub}) = b_i$ and $R_i = a_iP - b_iP_{pub}.\xi$ adds $(ID_i, R_i, P_{pub}, b_i)$ to L_1 and (ID_i, R_i, d_i) to L_{psk} list.

Create User Oracle $Cuser(ID_i)$. When Adv_1 asks a query on $Cuser(ID_i), \xi$ returns the current public key as $PK_i = (Q_i, R_i)$, if it already exists in L_{Cuser}. Otherwise, ξ does as follows.

(i) If $ID_i = ID^*, \xi$ chooses $a_i, b_i, c_i, x_i \in Z_q^*$ and sets $R_i = a_iP, H_1(ID_i, R_i, P_{pub}) = b_i$ and $X_i = x_iP$ and $H_2(ID_i, X_i) = c_i$. Now ξ sets $Q_i = R_i + h_{2i}X_i = a_iP + c_i(x_iP)$ and adds $(ID_i, R_i, P_{pub}, b_i)$ to $L_1, (ID_i, X_i, c_i)$ to L_2 and $(ID_i, Q_i, R_i, x_i, \perp)$ to the list L_{Cuser}. Finally, ξ returns the Public key $PK_i = (Q_i, R_i)$ to Adv_1.

(ii) If $ID_i \neq ID^*, \xi$ recovers (ID_i, R_i, d_i) from $L_{psk}.\xi$ sets $X_i = x_iP, H_2(ID_i, X_i) = c_i, c_i, x_i \in Z_q^*$ and$Q_i = R_i + c_iX_i = R_i + h_{2i}X_i.\xi$ outputs $PK_i = (Q_i, R_i)$ as public key. ξ adds (ID_i, X_i, c_i) to the list L_2 and $(ID_i, Q_i, R_i, x_i, d_i)$ to L_{Cuser}.

Reveal Secret value Oracle $RSK(ID_i)$. When Adv_1 asks a query on $RSK(ID_i), \xi$ does as follows. If $ID_i = ID^*, \xi$ aborts. Otherwise, ξ retrieve the tuple $(ID_i, Q_i, R_i, x_i, d_i)$ from L_{Cuser} and sends x_i to Adv_1. If not exists in L_{Cuser} list then ξ performs a query on $Cuser(ID_i)$ to produce (x_i, Q_i) and inserts to $L_{Cuser}.\xi$ returns x_i as a secret value.

Replace Public key Oracle $RPK(ID_i)$. If Adv_1 wants to replace the Public key $PK_i = (Q_i, R_i)$ of ID_i with $PK_i' = (Q_i', R_i')$, then ξ finds the tuple $(ID_i, Q_i, R_i, x_i, d_i)$ from the list L_{Cuser} and then updates Q_i withQ_i' andR_i withR_i'. Now ξ sets $x_i' = \perp$ and$d_i = \perp$. Hence the replaced tuple is of the form $(ID_i, Q_i', R_i', \perp, \perp)$.

Signing Oracle. When Adv_1 asks a sign query on $(ID_i, m_i), \xi$ does as follows: If $ID_i \neq ID^*$, then ξ recovers the $(ID_i, R_i, P_{pub}, h_{1i}), (ID_i, X_i, h_{2i})$ and $(ID_i, Q_i, R_i, x_i, d_i)$ from the lists L_1, L_2 and L_{Cuser} respectively and generates a valid signature as follows. Choose $u_i, h_{3i} \in Z_q^*$ and compute $v_i = u_i + h_{3i}(d_i + h_{2i}x_i) \bmod q$ and sets $U_i = u_iP.\xi$ returns $\sigma_i = (U_i, v_i)$ to Adv_1 as a valid signature and adds $(ID_i, m_i, PK_i, U_i, h_{3i})$ to L_3. If $ID_i = ID^*$, then ξ recovers the $(ID_i, R_i, P_{pub}, h_{1i})$ from L_1 and $(ID_i, Q_i, R_i, x_i, d_i)$ from L_{cuser}. Here $x_i = \perp$ and $d_i = \perp.\xi$ selects $u_i, h_{3i} \in Z_q^*$ and sets $U_i = v_iP - h_{3i}(Q_i + h_{1i}P_{pub}), v_i = u_i.\xi$ returns $\sigma_i = (U_i, v_i)$ to Adv_1 and adds $(ID_i, m_i, PK_i, U_i, h_{3i})$ to L_3.

Forgery/Output. Finally, Adv_1 returns a valid forged signature tuple $(ID_i^*, m_i^*, \sigma_i^*)$, where $\sigma_i^* = (U_i^*, v_i^*)$. If $ID_i \neq ID^*, \xi$ aborts the simulation. Otherwise, ξ recovers the tuples $(ID_i^*, R_i^*, P_{pub}, h_{1i}^*)\,(ID_i^*, X_i^*, h_{2i}^*)$, $(ID_i^*, m_i^*, PK_i^*, U_i^*, h_{3i}^*), (ID_i^*, Q_i^*, R_i^*, x_i^*, d_i^*)$ from the L_1, L_2, L_3 and L_{Cuser} lists. Since σ_i^* is valid, so $v_i^*P = U_i^* + h_{3i}^*(Q_i^* + h_{1i}^*P_{pub})$. $\Rightarrow \; v_i^* = u_i^* + h_{3i}^*(q_i^* + h_{1i}^*s)$. Here u_i^*, q_i^* and$s$ are unknown values to ξ. By Forking lemma, Adv_1 will output another two valid forged signatures: $\sigma_i^{*(j)} = \left(U_i^*, v_i^{*(j)}\right) for \;\; j = 2, 3.$

$$\Rightarrow v_i^{*(j)} = u_i^* + h_{3i}^{*(j)}(q_i^* + h_{1i}^*s), \quad for \; j = 1, 2, 3,$$

where u_i^*, q_i^* and$s$ are unknown values to ξ. By solving these three linearly independent equations, ξ obtains the value of s, which is the solution of the ECDLP.

Theorem 2: In the Random oracle model, the proposed PF-CLS Scheme is secure and existentially unforgeable against the Type–II adversary Adv_2 under the assumption that ECDLP is intractable.

Proof: The proof is similar to Theorem 1.

V. Efficiency Analysis

In this section, we evaluate the computation and communication cost of our PF-CLS scheme and compare it with other existing schemes. For this, we run a simulation experiment on Intel i7-7700 using Koblitz elliptic curve $y^2 = x^3 + ax + b \bmod p$, where p, q are 160-bit primes. The hardware and software specifications are listed in Table II and Table III lists the run time of few cryptographic operations.

A. Computation Cost

The computation costs of various CL-based signature schemes are calculated by considering the signing, verification and the total costs. Our scheme requires one scalar multiplication for signing and three scalar multiplications,

TABLE II
HARDWARE AND SOFTWARE SPECIFICATIONS

CPU	Intel core i7-7700@3.40GHz
RAM	4GB DDR3
OS	Windows-7 64-bit
Library	MIRACL, a public C++cryptographic library

TABLE III
LIST OF ABBREVIATIONS AND NOTATIONS

Notations	Time required (in milliseconds)
T_{SM}	Scalar point multiplication : $T_{SM} = 0.442ms$
T_{IN}	Modular inversion operation: $T_{IN} = 0.18879ms$
T_{EX}	Modular exponentiation operation: $T_{EX} = 3.864ms$
T_{EA}	Elliptic curve point addition : $T_{EA} = 0.0018ms$

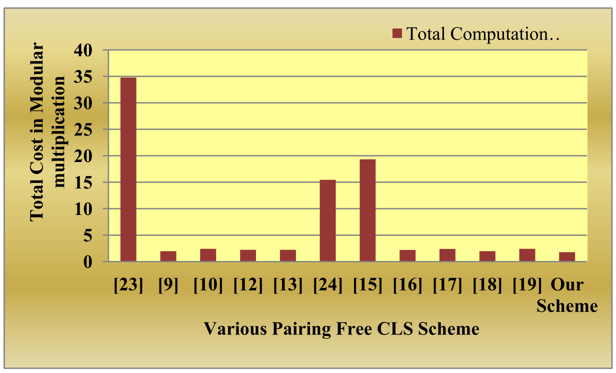

Fig. 1. Comparison of computation cost of pairing-free CLS schemes.

TABLE IV
COMPARISON OF COMPUTATION COSTS FOR PAIRING-FREE CLS SCHEMES

Scheme	Signature generation cost	Signature verification cost	Total Cost (in *ms*)	Security	Improvement in %
Ge et al.[23]	$2T_{EX} = 7.73ms$	$7T_{EX} = 27.05ms$	$34.78ms$	Yes	94.90
He et al.[9]	$1T_{SM} + 1T_{IN} = 0.63ms$	$3T_{SM} + 3T_{EA} = 1.33ms$	$1.96ms$	No [17]	09.71
Tsai et al.[10]	$1T_{SM} + 1T_{IN} = 0.63ms$	$4T_{SM} + 3T_{EA} = 1.78ms$	$2.41ms$	No [12]	26.31
Gong et al.[12]	$1T_{SM} = 0.442ms$	$4T_{SM} + 3T_{EA} = 1.78ms$	$2.22ms$	No [13]	20.03
Yeh et al.[13]	$1T_{SM} = 0.442ms$	$4T_{SM} + 3T_{EA} = 1.78ms$	$2.22ms$	Yes	20.03
Y. L.Wang et al.[24]	$2T_{EX} = 7.73ms$	$2T_{EX} = 7.73ms$	$15.46ms$	Yes	84.44
L.Wang et al.[15]	$1T_{EX} = 3.86ms$	$4T_{EX} = 15.46ms$	$19.32ms$	Yes	87.55
Yeh et al.[16]	$1T_{SM} = 0.442ms$	$4T_{SM} + 2T_{EA} = 1.78ms$	$2.21ms$	No [17]	19.96
Jia et al.[17]	$1T_{SM} + 1T_{IN} = 0.63ms$	$4T_{SM} + 2T_{EA} = 11.78ms$	$2.40ms$	No [25]	26.25
Karati et al. [18]	$1T_{SM} + 1T_{IN} = 0.63ms$	$3T_{SM} + 3T_{EA} = 1.33ms$	$1.96ms$	No [19]	9.71
Nasrollah et al.[19]	$1T_{SM} + 1T_{IN} = 0.63ms$	$4T_{SM} + 3T_{EA} = 1.78ms$	$2.41ms$	Yes	26.31
Our Scheme	$1T_{SM} = 0.442ms$	$3T_{SM} + 2T_{EA} = 1.33ms$	$1.78ms$	Yes	—

TABLE V
COMPARISON OF COMMUNICATION COST

Scheme	[23]	[15]	[24]	[12, 13]	[9,10,15, 17,18,19] Our Scheme
Signature Length	$3\|Z_p^*\|$	$\|Z_q^*\|+\|G_1\|$	$\|Z_q^*\|+2\|G\|$	$2\|Z_q^*\|+\|G\|$	$\|Z_q^*\|+\|G\|$
Communication Cost	3072 bits	1184 bits	800 bits	640 bits	480 bits

two point additions for verification. Thus the total cost of our scheme is $1.78ms$. Similarly, we compute for all other existing PF-CLS schemes [9], [10], [12], [13], [15], [16]–[19], [23], [24] and presented in Table IV. The comparison of computational costs, security, improvement prcentage of our scheme with other PF-CLS schemes are presented in Table-IV. From Table IV, we can observe that the proposed scheme requires the total computation cost as $1.78ms$ and is significantly less than all the existing schemes. The improvement of percentage in computational cost of our scheme is 94.90% over Ge *et al.* [23] scheme, 09.71% over He *et al.* scheme [9], 26.31% over Tsai *et al.* scheme [10], 20.03% over Gong *et al.* scheme [12], 20.03% over Yeh *et al.* scheme [13], 84.44% over Wang and Ye *et al.* scheme [24], 87.55% over Wang *et al.* scheme [15], 19.96% over Yeh *et al.* scheme [16], 26.25% over Jia *et al.* scheme [17], 9.71% over Karati *et al.* scheme [18], 26.31% over Nasrollah and Vanda scheme [19]. Further, most of the existing schemes [9], [10], [12], [16]–[18] in the literature are insecure. The comparison of computational cost of all the Pairing-Free CLS schemes is presented graphically in the Figure 1. From the Table IV and the Figure 1, it is clear that the proposed PF-CLS scheme is significantly more efficient in terms of total computation cost.

B. Communication Cost

To evaluate the communication cost, we consider the signature length. For comparable security with 1024 bit level RSA, we consider the experimental results from [20], [21]. The signature length in our scheme is $|G| + |Z_q^*|(320 + 160 = 480\text{bits})$. Similarly, the communication cost for other PF-CLS schemes are presented in the Table V.

From Table V, we can observe that the communication cost of our scheme is 480 bits. and is less than Ge *et al.* [23], Gong and Li [12], Yeh *et al.* [13], Wang and Ye [24], Wang *et al.* [15] signature schemes; whereas the proposed scheme has equal signature length with He *et al.* [9], Tsai *et al.* [10], Wang *et al.* [15], Yeh *et al.* [16], Jia *et al.* [17], Karati *et al.* [18], Nasrollah and Vanda [19] schemes.

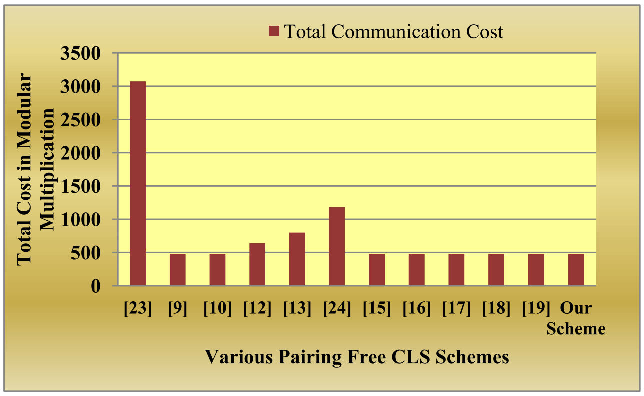

Fig. 2. Comparison of communication cost of pairing-free CLS schemes.

The comparison of communication costs of all the existing schemes are presented graphically in Figure-2. Thus from Table V and Figure 2, the proposed CLS scheme is efficient in terms of Communication point of view.

VI. Conclusion

This letter presents a new and efficient pairing free signature scheme in certificateless based framework. This scheme does not require any complex certificates for authentication of public keys as in traditional PKC and also eliminates key-escrow problem which is inherent in ID-based setting. The proposed scheme is proven secure and is unforgeable with the assumption that ECDL problem is hard. The efficiency analysis shows that the computation cost of our PF-CLS scheme is lower than other existing certificateless based signature schemes and thus the proposed scheme is a good candidate for deployment on resource constrained devices where the devices have limited computing power, storage space and communication bandwidth such as WSNs, VANETs, IoT, sensor devices etc.

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] Y. Zhou, Y. Fang, and Y. Zhang, “Securing wireless sensor networks: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 10, no. 3, pp. 6–28, 3rd Quart., 2008.

[3] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[4] A. Shamir, “Identity-based cryptosystems and signature schemes,” in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 196. Berlin, Germany: Springer-Verlag, 1985, pp. 47–53.

[5] S. S. Al-Riyami and K. G. Paterson, *Certificateless Public Key Cryptography* (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, vol. 2894, 2003, pp. 452–473.

[6] 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.

[7] Z. Zhang, D. S. Wong, J. Xu, and D. Feng, “Certificateless public-key signature: Security model and efficient construction,” in *Proc. Int. Conf. Appl. Cryptography Netw. Secur.*, 2006, pp. 293–308.

[8] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, “Elliptic curve lightweight cryptography: A survey,” *IEEE Access*, vol. 6, pp. 72514–72550, 2018.

[9] D. He, J. Chen, and R. Zhang, “An efficient and provably-secure certificateless signature scheme without bilinear pairings,” *Int. J. Commun. Syst.*, vol. 25, no. 11, pp. 1432–1442, Nov. 2012.

[10] J.-L. Tsai, N.-W. Lo, and T.-C. Wu, “Weaknesses and improvements of an efficient certificateless signature scheme without using bilinear pairings,” *Int. J. Commun. Syst.*, vol. 27, no. 7, pp. 1083–1090, Jul. 2014.

[11] M. Tian and L. Huang, “Cryptanalysis of a certificateless signature scheme without pairings,” *Int. J. Commun. Syst.*, vol. 26, no. 11, pp. 1375–1381, Nov. 2013.

[12] 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.

[13] 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.

[14] L. Wang, K. Chen, Y. Long, X. Mao, and H. Wang, “A modified efficient certificateless signature scheme without bilinear pairings,” in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, Sep. 2015, pp. 82–85, doi: 10.1109/incos.

[15] 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.

[16] K.-H. Yeh, C. Su, K.-K.-R. Choo, and W. Chiu, “A novel certificateless signature scheme for smart objects in the Internet-of-Things,” *Sensors*, vol. 17, no. 5, p. 1001, 2017.

[17] 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.

[18] A. Karati, S. H. Islam, and G. P. Biswas, “A pairing-free and provably secure certificateless signature scheme,” *Inf. Sci.*, vol. 450, pp. 378–391, Jun. 2018, doi: 10.1016/j.ins.2018.03.053.

[19] N. Pakniat and B. A. Vanda, “Cryptanalysis and improvement of a pairing-free certificateless signature scheme,” in *Proc. 15th Int. ISC (Iranian Soc. Cryptol.) Conf. Inf. Secur. Cryptol. (ISCISC)*, Aug. 2018, pp. 1–5, doi: 10.1109/iscisc.2018.8546984.

[20] X. Cao, W. Kou, and X. Du, “A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges,” *Inf. Sci.*, vol. 180, no. 15, pp. 2895–2903, Aug. 2010.

[21] S. H. Tan, S. H. Heng, and B. M. Goi, *Java Implementation for Pairing-Based Cryptosystems* (Lecture Notes in Computer Science), vol. 6019. Berlin, Germany: Springer-Verlag, 2010, pp. 188–198.

[22] *MIRACL Library*. Accessed: Jan. 30, 2020. [Online]. Available: http://certivox.org/display/EXT/MIRACL

[23] A. Ge, S. Chen, and X. Huang, “A concrete certificateless signature scheme without pairings,” in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, vol. 2, 2009, pp. 374–377.

[24] Y. L. Wang and J. Y. Ye, “Applied-information technology in an improved certificateless signature scheme without bilinear pairings,” *Appl. Mech. Mater.*, vol. 685, pp. 528–531, Oct. 2014.

[25] 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.