The ISC Int'l Journal of Information Security

ISeCure

January 2022, Volume 14, Number 1 (pp. 69–82)

http://www.isecure-journal.org

An Efficient Pairing-Free Identity-Based Certificateless Signcryption

Saeed Banaeian Far [1,*] and Maryam Rajabzadeh Asaar [1]

[1] *Department of Electrical and Computer Engineering, Science and Research Branch Islamic Azad University, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: December 12, 2020
Revised: May 20, 2021
Accepted: July 7, 2021
Published Online: August 28, 2021

Keywords:
Certificateless Signcryption, Efficiency, Hyperelliptic Curve, Point Factorization problem, Random Oracle

Type: Research Article

doi: 10.22042/ISECURE.2021.261788.587

dor: 20.1001.1.20082045.2022.14.1.6.6

ABSTRACT

A certificateless (CL) signcryption scheme is a cryptographic primitive that provides user authentication and message confidentiality at the same time. CL signcryption schemes (as a type of certificateless encryption scheme) have solved problems concerning malicious server presentation, and the server who issues users' partial private keys and certificates cannot obtain users' signing keys. Therefore, the CL signcryption scheme is an excellent choice for protecting users' signing keys and providing user authentication and message confidentiality. Moreover, signcryption schemes have lower computational costs than signature and encryption schemes. The present study presents a short and efficient CL signcryption scheme based on the hyperelliptic curve (HC). Applying HC as the calculation base for designing the presented CL signcryption scheme reduces key-length from 160 bits to 80. The presented CL signcryption scheme is shorter than other recently-proposed ones with regard to communication overhead with its less than one-third shorter length compared to the shortest of the others. Moreover, it is more efficient than other recently-proposed CL signcryption schemes in the user-side computational cost, including the *key generation* and *user key generation* phases that have been halved in total. Finally, the security of the presented CL signcryption scheme was analyzed in the random oracle (RO) model based on the hardness of the point factorization problem (PFP) on HC.

1 Introduction

Privacy is a significant challenge on the internet and in computer networks. Various features are needed that are believed to provide privacy, such as message confidentiality, user authentication, and user's signing key protection are three features that can be provided privacy for users. Applying encryption algorithms [1, 2] (symmetric or asymmetric) is the main method for providing message confidentiality, and digital signature schemes are known as one of the most popular primitives that provide user authentication [3]. Creating a certificateless version of cryptosystems (e.g., encryption and signature) is a method which can protect users' signing keys when the signer wants no one (even the server) to be able to obtain its signing key [4, 5]. In this method, a part of the user's private key is generated by the user, while the central authority has no access to it. The user creates a new pair of public-private keys using its certificate and a unique private random number. The *key escrow* feature is defined as a feature in cryptosystems whereby the server who issued users' certificates

* Corresponding author.
Email addresses: saeed.banaeian@srbiau.ac.ir, asaar@srbiau.ac.ir

ISSN: 2008-2045

can obtain users' private signing keys [6]. However, certificateless schemes do not provide the mentioned feature and solve the key escrow problem.
The signcryption scheme [7] and a certificateless (CL) signcryption scheme [5] are two primitives that provide user authentication and message confidentiality at the same time. However, the server cannot break the message confidentiality or forge users' signatures if a secure CL signcryption scheme is applied. The main difference between "signcryption" and "signature and encryption" (signature then encryption or encryption then signature) is in the computational overhead where in signcryptions, a type of symmetric encryption (i.e., AES) is applied.

Conributions: In this paper, an efficient and short CL signcryption scheme is presented based on the hyperelliptic curve (HC). In the communication overhead, we show that the presented CL signcryption scheme has a lower communication overhead than other recently-proposed CL signcryption schemes, among which it has about one-third communication overhead of the shortest. Additionally, the presented CL signcryption scheme is more efficient in the computational overhead than other recently-proposed CL signcryption schemes in all phases where it is halved in total. In the presented CL signcryption scheme, the signcryption and unsigncryption phases are similar to the basic signcryption scheme [7], but the partial private key extract and user key generation phases are quite different and more efficient. One of the efficiency aspects is to choose HC as the base of calculations in such a way that the 80-bit key length on HC provides equal security to the 160-bit key length on an elliptic curve (EC).
We analyze the security of the presented CL signcryption scheme in the RO model and reduce the security of the presented CL signcryption scheme to the hardness of the point factorization problem (PFP) [1] on HC. We finally compare the presented CL signcryption scheme with other existing schemes and show its efficiency.

Organization: We present an overview of several CL signcryption schemes in Section 2. In Section 3, we describe the paper preliminaries. In Section 4, we present our CL signcryption scheme and analyze it. Finally, in Section 5, we compare the presented CL signcryption scheme with recently-proposed schemes.

2 Related Works

In 2016, Zhou *et al.* presented a CL signcryption scheme that provides unforgeability and confidentiality against adversary type I and type II in the standard model [9]. Their CL signcryption scheme was designed based on the hardness of the modified decisional bilinear Diffie- Hellman (M-DBDH) and the square computational Diffie- Hellman (Squ-CDH).
In 2017, Yu *et al.* presented a CL signcryption scheme with no pairing [11]. Their presented CL signcryption scheme's security is based on the difficulty of computational Diffie-Hellman (CDH) and the discrete logarithm (DL) problems. In the same year, Luo *et al.* presented another pairing-based CL signcryption scheme [12] which is an enhanced model of the Zhou *et al.*'s scheme [9]. However, the Luo *et al.*'s scheme was designed on the hardnesses of the CDH and the decisional bilinear Diffie-Hellman (DBDH) problems while their scheme was made more efficient than [9].
Rastegari and Berenjkoub presented yet another CL signcryption scheme [10]. They presented an improvement of the Liu *et al.*' scheme [13].
In 2018, Caixue presented a CL signcryption scheme with no random oracle [16] that was designed on the complexity of four hard problems including the truncated decision q-augmented bilinear Diffie-Hellman exponent (TD-q-ABDHE), the M-DBDH, the q-Strong Diffie-Hellman (q-SDH), and the Squ-CDH.
In 2019, Shan proposed a CL signcryption scheme that was designed based on the Modified-PS' hardness [17]. No random oracle was applied in Shan's CL signcryption scheme. However, he used a bilinear pairing map that made much computation overhead in the signcryption and unsigncryption phases. In the same year, Gao *et al.* presented a pairing-free CL signcryption scheme that could be adopted to access control in wireless body area networks (WBAN) [19]. They then applied their presented CL signcryption scheme in a protocol designed for WBAN.
In 2020, Lin *et al.* analyzed the Rastegari *et al.*'s CL signcryption scheme and showed that [10] is vulnerable to known session temporary information [20]; But Lin *et al.* presented no improvement. In the same year, Liu *et al.* presented another CL signcryption scheme for use in WBAN systems [21] (similar to [19]). The Liu *et al.*'s scheme was designed based on the hardness of DLP while their scheme was heavier than [19] (see the comparison in Section 5).
As another CL signcryption scheme, Kasyoka *et al.* analyzed the Wei and Ma's scheme [23], which was proposed in 2019 for cloud storage, and improved its vulnerability to unforgeability against adversary types I and II [22]. Additionally, Kasyoka *et al.* claimed that their proposed CL signcryption scheme was more efficient than [23]. Mandal *et al.* in 2020 presented an access control scheme for the internet of things that applied a CL signcryption scheme to provide message confidentiality and user (node) authentication simultaneously [24]. A lattice-based CL signcryption scheme was designed based on the

[1] The PFP on HC can be assumed equivalent to the hyperelliptic curve discrete logarithm problem (HCDLP).

Table 1. The Summary of Section 2

Ref	Year	Basic scheme	Motivations/Problems	Main idea
Zhou [9]	2016	Yuan [8]	They claimed CL signcryption schemes are vulnerable to Type I security.	Applying bilinear pairing, which is Type I secure. Reducing the scheme's security to M-DBDH and Squ-CDH.
Yu [11]	2017	-	They claimed most CL signcryption schemes have a lot computational overhead due because of pairing.	Proposing a pairing-free and secure CL signcryption scheme.
Luo [12]	2017	Zhou [9]	They claimed that when security-proved CL signcryptions schemes under the RO model are applied in practical situations, the RO model causes many security problems due to its own defects.	Presenting a CL signcryption scheme in the standard model based on the hardness of DBDH and CDH problems.
Rastegari [10]	2017	Liu [13]	Security weaknesses of [13], including attacks against semantic security and existential unforgeability [14, 15].	Improving [13] and presenting a more efficient scheme in the standard model.
Caixue [16]	2018	Zhou [9]	He claimed that the proposed CL signcryption schemes are insecure or inefficient. They need to have long system public parameters. It caused to have a limited storage environment.	Using identity-based encryption to reduce the numbers of stored parameters.
Shan [17]	2019	Canard [18]	No clear motivations were defined.	Proposing an efficient CL signcryption scheme secure in the standard model.
Gao [19]	2019	-	WBAN systems problems and especially the efficiency since most of the users are sensors (low-power devices).	Providing message confidentiality and unforgeability in WBAN systems.
Lin [20]	2020	Rastegari [10]	Cryptanalysis of the known session-specific temporary information attack in the standard model-based CL signcryption schemes.	Implementing the known session-specific temporary information attack on [10] (no improvement was proposed).
Liu [21]	2020	Similar to Gao [19]	Sensitive data confidentiality and efficiency.	Designing an efficient RSA-based CL signcryption scheme based to apply in a data access control protocol for WBAN systems.
Kasyoka [22]	2020	Wei [23]	Weaknesses of [23] to both types of EUF-CMA I and II.	Modifying [23] in such a way that the modified scheme is secure to EUF-CMA I and II, and it is more efficient than.
Yu [26]	2020	Chen [25]	The existing CL signcryption schemes are not secure to quantum computers.	Presenting a quantum-secure CL signcryption scheme that has higher computation efficiency lower communication costs than the existing schemes.
Yuan [27]	2020	Luo [12]	Cryptanalysis of weaknesses of [12] when a malicious KGC wants to implement attacks.	No Improvement of [12] was presented.

hardness of the two problems learning with errors (LWE) and small integer solution (SIS) by Yu *et al.* [26]. The presented lattice-based CL signcryption scheme provides security against quantum computers in addition to providing the required security for CL signcryption schemes. The cryptanalysis of the Luo *et al.*'s scheme [12] was presented by Yuan [27]. Yuan showed that the Luo *et al.*'s scheme did not provide signcryption unforgeability if a curious key generator center (KGC) was present; it also did not provide message confidentiality if a malicious KGC wanted to implement attacks.

The summary of the discussed studies is provided in Table 1, and we will compare and analyze the above-discussed schemes in Section 5 in more detail.

3 Preliminaries

In this section, we review the preliminaries of this paper. The list of used notations is shown in Table 2.

3.1 Hyperelliptic Curve

The curve C is defined as a HC over a finite field $\mathbb{F}_q$ with an equation of C: $y^2 + h(x)y = f(x)$ where $h(x) \in \mathbb{F}_q(x)$ is a polynomial of degree at most g, and $f(x) \in \mathbb{F}_q(x)$ is a monic polynomial of degree $2g + 1$ [28, 29]. The objective is to form a jacobian group $JC(\mathbb{F}_q)$, and select a divisor D where D is a generator

Table 2. The list of used notations

Notation	Description
C	Hyperelliptic curve over prime field F_q
D	Devisor of large prime order n in $JC(F_q)$
$Dec_k(.)$	Symmetric decryption algorithm
$Enc_k(.)$	Symmetric encryption algorithm
g	Generator of G
$h(.)$	Secure one-way map-to-point function
ID_i	Identity of ith user
$JC(F_q)$	Jacobian group
λ	Security parameter
$params$	Set of CL signcryption public parameters
pr_{U_i}	Private key of ith user
$pr_{U_{ix}}$	The CL private key of ith user
pu_{U_i}	Public key of ith user
$pu_{U_{ix}}$	The CL public key of ith user
q	Large prime number ($q \geq 2^{80}$)
s	Master secret key of KGC
U_i	The ith user
V	Verifier (receiver)
x_i	Secret parameter selected by ith user
$\|$	Concatenate operation

of JC group. The D is set as: $D = (a(x), b(x)) = (\Sigma_{i=0}^{g} a_i x^i, \Sigma_{i=0}^{g-1} b_i x^i) \in JC(\mathbb{F}_q)$. There is no solution if $(a, b) \in \mathbb{F}_q \times \mathbb{F}_q$ which simultaneously satisfy a equation $b^2 + h(a)b = f(a)$ and a partial derivative equations.
The HC is more efficient than the elliptic curve (EC), and the 80 bits of a key on the HC provides equal security to the 160 bits of a key on the EC [30]. We use the HC to provide the same security using the 80 bits of keys. It is one of the items that provide more efficiency for the presented CL signcryption scheme in addition to making lower communication overhead.

Complexity assumption: It is assumed that there is no probabilistic polynomial-time algorithm (PPT) to find $r \in \mathbb{F}_q$ by having a point rD on the HC C where D as a generator of JC group. But forgiven r, there is a PPT algorithm to calculate rD.

We define an advantage of a PPT adversary ($\mathcal{A}$) in solving the PFP on HC as: $ADV_{\mathcal{A}}^{PFP} = Pr[\mathcal{A}(rD) = r|rD] < \varepsilon$.

3.2 Certificateless Signcryption

A digital signature is a cryptographic primitive that provides user authentication and message integrity [3]. In the following, we will briefly describe the general certificateless (CL) signcryption scheme.
There is a need to define a feature named *key escrow*. In summary, *key escrow* means that a central authority (CA) who issued users' certificates can obtain their private keys and recover their encrypted messages or signs instead of them [6]. However, in CL signatures, CA cannot obtain users' signature keys [4].
The concept of signcryption was first proposed by Y. Zheng in 1997 [7]. This type of signature provides user authentication and message confidentiality at the same time, where it is only the verifier V that can obtain the plain message since the message is encrypted by a symmetric encryption algorithm. Therefore, the cost of signcryption is much lower than those of sign and encryption ($cost_{signature\_and\_encryption} \ll cost_{signature} + cost_{encryption}$).
As previously mentioned, we can define the CL signcryption scheme as a cryptographic primitive that provides message confidentiality the user authentication simultaneously; CA cannot obtain a users' private keys either [5]. Therefore, CA cannot create a valid signcryption instead of the real signer, and it cannot recover the encrypted message. In the following, the general definition of the generic CL signcryption scheme and its security model will be presented.

3.2.1 Definition

The generic CL signcryption scheme consists of six PPT algorithms, namely: setup, partial private key extraction, user key generation, private key generation, signcryption, and unsigncryption [9, 10]. These will all be described in the following paragraphs.

(1) ($params$, s) $\leftarrow$ $Setup(\lambda)$: On the security parameter λ, $Setup(\lambda)$ returns the set of system public parameters $params$ and the master secret key s.
(2) (pu_{U_i}, pr_{U_i}) $\leftarrow$ *Partial private key extraction*($params$, s): The *Partial private key extraction* algorithm takes $params$ and s and returns U_i's partial pairs of public-private keys (pu_{U_i}, pr_{U_i}).
(3) $pr_{U_{ix}} \leftarrow$ *User key generation*($params, x_i, pr_{U_i}$): Using $params$, pr_{U_i}, and the selected random value $x_i \in_R [1, n-1]$, the *User key generation* algorithm computes the CL private key $pr_{U_{ix}}$.
(4) $pu_{U_{ix}} \leftarrow$ *key generation*(x_i, $pr_{U_{ix}}$, $params$): On x_i, $params$, and $pr_{U_{ix}}$, the *key generation* algorithm is executed to generate user's CL public key $pu_{U_{ix}}$.
(5) $\sigma \leftarrow$ *Signcryption*($params$, $pr_{U_{ix}}$, pu_V, m): To generate a CL signcryption on the message m, the *Signcryption* algorithm takes $params$, m, verifier's (V) public key pu_V, and $pr_{U_{ix}}$ and returns σ as the CL signcrypt on the message m.
(6) $\{m, \perp\} \leftarrow$ *Unsigncryption*($params$, pr_V, $pu_{U_{ix}}$, σ): The *Unsigncryption* algorithm takes $params$, pr_V, $pu_{U_{ix}}$, and σ and returns m if σ

Table 3. The list of security notations

Notation	Description
$\mathcal{O}_{PK}$	Private key oracle
$\mathcal{O}_{RPK}$	Replace private key oracle
$\mathcal{O}_{SC}$	Signcryption oracle
$\mathcal{O}_{USC}$	Unsigncryption oracle
$\mathcal{O}_I$	$\{\mathcal{O}_{PK}, \mathcal{O}_{RPK}, \mathcal{O}_{SC}, \mathcal{O}_{USC}\}$
$\mathcal{O}_{II}$	$\{\mathcal{O}_{PK}, \mathcal{O}_{SC}, \mathcal{O}_{USC}\}$
$\mathcal{A}_I$	An adversary who has $\mathcal{O}_I$
$\mathcal{A}_{II}$	An adversary who has $\mathcal{O}_{II}$
$\mathcal{C}$	The Challenger

is valid. Else, it returns $\perp$.

3.2.2 Security Model

To describe the security model of the general CL signcryption schemes, there is a need to define some oracles and two types of adversaries, as will follow. Regarding the definition of a CL signcryption scheme, the needed oracles are defined below.

- The $\mathcal{O}_{PK}$ is the private key oracle that takes user's identity ID and returns the user's private key pr'_{U_i}.
- The $\mathcal{O}_{RPK}$ is the replace private key oracle, that takes the new user's private key $pr'_{U_{ix}}$ and replace it by $pr_{U_{ix}}$.
- The $\mathcal{O}_{SC}$ is the signcryption oracle that takes message m, the private key of the ith user U_i, and the public key of V and returns a valid CL signcryption σ'.
- The $\mathcal{O}_{USC}$ is the unsigncryption oracle that takes CL signcryption σ, public key of U_i, and private key of V and returns the valid message m'.

For simplifying the security model, the two types of adversaries are defined below.

$\mathcal{A}_I$: The adversary type I who has no access to the master secret key s, but it can replace the public key of each user (key replacement attack) with a random value it wants. It means, $\mathcal{A}_I$ has only the set of $\mathcal{O}_I = \{\mathcal{O}_{PK}, \mathcal{O}_{RPK}, \mathcal{O}_{SC}, \mathcal{O}_{USC}\}$. This type of adversary is assumed as an outsider adversary (real adversary) who wants to implement attacks such as breaking message confidentiality or forging the CL signcryption scheme. In fact, this type of adversary is defined to show that an outsider adversary can implement no attack (breaking message confidentiality or forging the CL signcryption scheme) to the CL signcryption scheme.

$\mathcal{A}_{II}$: The adversary type II has access to the master secret key s, but cannot replace the user's keys. It means, $\mathcal{A}_{II}$ has the set of $\mathcal{O}_{II} = \{\mathcal{O}_{PK}, \mathcal{O}_{SC}, \mathcal{O}_{USC}\}$ and the master secret key s.
This type of adversary is assumed as an insider adversary (malicious KGC) who wants to forge the CL signcryption scheme or break message confidentiality (malicious-but-passive KGC attack). It is also defined to show that an insider adversary (e.g., malicious KGC) cannot implement the two mentioned attacks on the CL signcryption scheme.

According to defined oracles and two types of adversaries ($\mathcal{A}_I$ and $\mathcal{A}_{II}$ or outsider and insider adversaries), four games are designed to analyze the security of CL signcryption schemes [16, 31, 32]. In the following, we present the four mentioned games and show the used security notations in Table 3. In the presented security model, the defined games are played between $\mathcal{A}_I$ or $\mathcal{A}_{II}$ and the challenger $\mathcal{C}$. As a result, the games for $\mathcal{A}_I$ and $\mathcal{A}_{II}$ are executed separately and $\mathcal{A}_I$ and $\mathcal{A}_{II}$ cannot cooperate.

(1) **Confidentiality against $\mathcal{A}_I$:** This feature means that an outsider adversary (named $\mathcal{A}_I$) who has the ability to implement the key replacement attack cannot break the message confidentiality of a secure CL signcryption scheme. As a formal definition, we can say that a secure CL signcryption scheme is confidential to $\mathcal{A}_I$ if $\mathcal{A}_I$ cannot find a PPT algorithm to obtain the message m which a CL signcryption scheme is created over. To show this, the following game (*Game 1*) is designed.

Game 1: If $\mathcal{A}_I$ can find no PPT algorithm to win in *Game 1* named IND-CLSC-CCA2-I[2], the CL signcryption scheme provides the IND-CLSC-CCA2-I security. The *Game 1* is defined in four phases as follows.

- *Setup:* The security parameter λ is given and a master secret key s is generated by $\mathcal{C}$. Then, $\mathcal{C}$ executes a PPT alghorithm to generate the set of system public parameters *params*. The $\mathcal{C}$ gives the set of system public parameters *params* and the set $\mathcal{O}_I$ to $\mathcal{A}_I$ and keeps its master secret key s secure.
- *Find stage:* The $\mathcal{A}_I$ submits polynomially bounded numbers of queries to all oracles in the set $\mathcal{O}_I$ and stores all responses.
- *Challenge stage:* The $\mathcal{C}$ creates two messages m_i where $i \in \{1, 2\}$ with an equal length, and creates two CL signcrypts σ_1 and σ_2 on m_1 and m_2, respectively. Then,

[2] Indistinguishability-certificateless signcryption-adaptive chosen ciphertext attack-type I

$\mathcal{C}$ gives σ_i where $i \in_R \{1,2\}$, m_1 and m_2 to $\mathcal{A}_I$.

- *Guess stage:* The $\mathcal{A}_I$ wins if guesses a valid value for i with a probability more than $\frac{1}{2}$ in a polynomial-time such that $Unsigncrypt(\sigma_i) = m_i$.

The $\mathcal{A}_I$'s advantage in *Game 1* is defined as $ADV_{\mathcal{A}_I}^{Game1} = |Pr[\mathcal{A}_I(\sigma_i, m_1, m_2) = i \ |\sigma_i, m_1, m_2] - \frac{1}{2}|$, and $\mathcal{A}_I$ wins *Game 1* if it has a non-negligible advantage against $\mathcal{C}$.

(2) **Confidentiality against $\mathcal{A}_{II}$:** Confidentiality against $\mathcal{A}_{II}$ says that a secure CL signcryption scheme keeps the content of a signcrypted message for an insider adversary (malicious KGC). As a formal definition, we can say that a secure CL signcryption scheme is confidential for $\mathcal{C}_{II}$ if $\mathcal{A}_{II}$ cannot find a PPT algorithm to obtain the message m that a CL signcryption scheme is created over. To show this, *Game 2* has been designed and written below.

Game 2: If $\mathcal{A}_{II}$ can find no PPT algorithm to win in *Game 2* named IND-CLSC-CCA2-II[3], the CL signcryption scheme provides the IND-CLSC-CCA2-II security. The *Game 2* is defined in four phases as follows.

- *Setup:* The security parameter λ is given, and a master secret key s is generated by $\mathcal{C}$. Then, $\mathcal{C}$ executes a PPT alghorithm to generate the set of system public parameters $params$. The $\mathcal{C}$ gives the set of system public parameters $params$, the master secret key s, and the set $\mathcal{O}_{II}$ to $\mathcal{A}_{II}$.
- *Find stage:* The $\mathcal{A}_{II}$ submits polynomially bounded numbers of queries to all oracles in $\mathcal{O}_{II}$ and stores all responses.
- *Challenge stage:* This phase is the same *Challenge stage* phase described in *Game 1*.
- *Guess stage:* This phase is the same *Guess stage* phase described in *Game 1*.

The $\mathcal{A}_{II}$'s advantage is defined as $ADV_{\mathcal{A}_{II}}^{Game2} = |Pr[\mathcal{A}_{II}(\sigma_i, m_1, m_2) = i \ |\sigma_i, m_1, m_2] - \frac{1}{2}|$, and $\mathcal{A}_{II}$ wins *Game 2* if it has a non-negligible advantage against $\mathcal{C}$.

(3) **Unforgeability against $\mathcal{A}_I$:** This feature says that an outsider adversary should not be able to forge a secure CL signcryption scheme by implementing the key replacement attack. As a formal definition, we can say the CL signcryption scheme is unforgeable for $\mathcal{A}_I$ if $\mathcal{A}_I$ can find no PPT algorithm to create a valid CL signcryption on a message m. *Game 3* has been written below to show this feature in more detail.

Game 3: If $\mathcal{A}_I$ can find a PPT algorithm to win in *Game 3* named of EUF-CLSC-CMA-I[4] with a negligible probability, the CL signcryption scheme provides the EUF-CLSC-CMA-I security. The *game 3* includes three phases as follows.

- *Setup:* This phase is the same *Setup* phase described in *Game 1*.
- *Queries:* This phase is the same *Find stage* phase described in *Game 1*.
- *Forgery:* The $\mathcal{A}$ selects two identities ID_1 and ID_2 and a message m. Then, it creates a CL signcrypt on m instead of the user by a selected identity ID_1 as $signcrypt(params, k, pr_{U_1}, pu_{U_2}, m) = \sigma_1$ and $\mathcal{A}_I$ sends σ_1 to the user U_2 who has the identity ID_2.

The $\mathcal{A}$ wins *Game 3* if the user U_2 by the identity ID_2 unsigncrypts the received CL signcryption σ_1 successfully using its private key and the U_1's public key. The $\mathcal{A}_I$'s advantage is defined as $ADV_{\mathcal{A}_I}^{Game3} = Pr[m \leftarrow Unsigncrypt(params, pr_{U_2}, pu_{U_1}, \sigma_1)]$, and $\mathcal{A}_I$ wins *Game 3* if it has a non-negligible advantage against $\mathcal{C}$.

(4) **Unforgeability against $\mathcal{A}_{II}$:** This feature says that an insider adversary should not be able to forge a secure CL signcryption scheme. As a formal definition, it can be said that the CL signcryption scheme is unforgeable for $\mathcal{A}_{II}$ if $\mathcal{A}_{II}$ can find no PPT algorithm to create a valid CL signcryption on a message m. To show this feature in a game, *Game 4* has been written in the following.

Game 4: If $\mathcal{A}_{II}$ can find a PPT algorithm to win in *Game 4* named of EUF-CLSC-CMA-II[5] with a negligible probability, the CL signcryption scheme provides EUF-CLSC-CMA-II security. The *Game 4* includes three phases as follows.

- *Setup:* This phase is the same *Setup* phase described in *Game 2*.
- *Queries:* This phase is the same *Find stage* phase described in *Game 2*.
- *Forgery:* This phase is the same *Forgery* phase described in *Game 3*.

[3] Indistinguishability-certificateless signcryption-adaptive chosen ciphertext attack-type II

[4] Existential unforgeability-certificateless signcryption-chosen message attack-type I

[5] Existential unforgeability-certificateless signcryption-chosen message attack-type II

The $\mathcal{A}_{II}$'s advantage is defined as $ADV^{Game4}_{\mathcal{A}_{II}} = Pr[m \leftarrow Unsigncrypt(params, pr_{U_2}, pu_{U_1}, \sigma_1)]$, and $\mathcal{A}_{II}$ wins *Game 4* if it has a non-negligible advantage against $\mathcal{C}$.

4 The Efficient Certificateless Signcryption

In this section, we describe our CL signcryption scheme and analyze it in the RO model based on the hardness of the PFP on HC.

4.1 The Scheme

Below we describe the details of our CL signcryption scheme.

4.1.1 Setup Phase

In this phase, KGC executes a PPT algorithm to setup the system. This algorithm takes security parameter λ and the KGC's master secret key s as input parameters and returns the set of system's public parameters $params = \{C, D, Enc_k/Dec_k, h(.), q, JC(F_q), G, g\}$ where C is a hyperelliptic curve on a finite field F_q, D is a devisor of the large prime order n in $JC(F_q)$ ($n \geq 2^{80}$), Enc_k/Dec_k is a secure symmetric encryption/decryption algorithm (e.g., AES), $h(.)$ is a secure one way map-to-point (MtP) function that is defined as a map $h : \{0,1\}^* \rightarrow \mathbb{F}_q \times \mathbb{F}_q$, q is a large prime number ($q > 2^{80}$), $JC(F_q)$ is a jacobian group, and g is the generator of the group G. Then, KGC publishes the set of system's public parameters $params$ and keeps s secure.

4.1.2 Partial Private Key Extraction Phase

The KGC computes the U_i's partial private key as $pr_{U_i} = sh(ID_i)$ and sends it to U_i through a secure channel. The user U_i keeps its partial private key pr_{U_i} secure.

4.1.3 User Key Generation Phase

The user U_i selects a random number $x_i \in_R [1, n-1]$ and computes its CL private key $pr_{U_{ix}} = x_i h(pr_{U_i})$ and keeps it secure.

4.1.4 Key Generation Phase

The user U_i uses the selected random number x_i and $params$ to generate its CL pair of public-private key as $(pu_{U_{ix}}, pr_{U_{ix}})$ where $pu_{U_{ix}} = pr_{U_{ix}} D$, and publishes its public key $pu_{U_{ix}}$.

4.1.5 Signcryption Phase

To create the CL signcryption on the message m, the user U_i selects a random number $k \in_R [1, n-1]$ and computes $K_1 = h(kD)$, $K_2 = h(kpu_V)$, $C = Enc_{K_2}(m)$, $r = h(K_1 \| m)$, $S = \frac{k}{r + pr_{U_{ix}}}$, and $R = rD$. Then, U_i sends $\sigma = \{C, R, S\}$ as the CL signcryption on the message m to the verifier V.

4.1.6 Unsigncryption Phase

On receiving the CL signcryption set $\sigma = \{C, R, S\}$, V unsigncrypts the CL signcryption σ using its private key pr_V by calculating $K_1 = h(S(pu_{U_{ix}} + R))$, $K_2 = h(S(pr_V(pu_{U_{ix}} + R)))$, $m = Dec_{K_2}(C)$, and $r = h(K_1 \| m)$. Then, V verifies the decrypted message m if $rD = R$.

Correctness: The proposed identity-based CL signcryption scheme works since $K_1 = h(S(pu_{U_{ix}} + R)) = h(\frac{k}{r+pr_{U_{ix}}}(pr_{U_{ix}} D + rD)) = h(\frac{k}{r+pr_{U_{ix}}}(pr_{U_{ix}} + r)D) = h(kD)$, and $K_2 = h(S(pr_V(pu_{U_{ix}} + R))) = h(\frac{k}{r+pr_{U_{ix}}} pr_V(pr_{U_{ix}} D + rD)) = h(\frac{k}{r+pr_{U_{ix}}} pr_V(pr_{U_{ix}} + r)D) = kpr_V D = kpu_V$.

4.2 Security Analysis

In this section, we show, in the RO model, the presented CL signcryption scheme provides the security features described in Section 3.2.2.

4.2.1 Confidentiality Against $\mathcal{A}_I$

The $\mathcal{A}_I$ wants to find the message m that $\mathcal{C}$ encrypted it (symmetric) using the presented method in CL signcryption scheme.

Theorem 1. *The presented CL signcryption scheme provides IND-CLSC-CCA2-I security in the RO model, and $\mathcal{A}_I$ can implement the key replacement attack on the presented CL signcryption scheme with a negligible advantage against $\mathcal{C}$ if the PFP on HC is hard.*

Proof. The CL signcryption scheme is IND-CLSC-CCA2-I-secure if $\mathcal{A}_I$ can find no PPT algorithm to win in *Game 1* with a probability more than $\frac{1}{2}$. The *Game 1* is written in the next.

- *Setup:* The $\mathcal{C}$, who has the security parameters λ and the master secret key s, executes the *setup phase* (or it calls $Setup(\lambda)$) of the presented CL signcryption scheme. Then, $\mathcal{C}$ gives $params$, and the set $\mathcal{O}_I = \{\mathcal{O}_{PK}, \mathcal{O}_{RPK}, \mathcal{O}_{SC}, \mathcal{O}_{USC}\}$ to $\mathcal{A}_I$ and keeps the master secret key s secure.
- *Find stage:* The $\mathcal{A}_I$ generates polynomially bounded numbers of queries and submits them to all existing oracles in $\mathcal{O}_I$ as follows:
  - *Private key query:* The $\mathcal{A}$ submits polynomially bounded numbers of ID_i to $\mathcal{O}_{PK}$ to get the partial private key pr'_{U_i} from $\mathcal{O}_{PK}$. The $\mathcal{O}_{PK}$, with the help of random oracle, returns pr'_{U_i} to $\mathcal{A}$ ($\mathcal{A}$ takes help from $\mathcal{O}_{PK}$

since it, as an outsider adversary, has not access to s). The $\mathcal{A}$ then calculates the public key as $pu'_{U_i} = pr'_{U_i}D$ and generates a table including all sent queries and returned responses ($\{ID_i, pr'_{U_i}, pu'_{U_i}\}$). The generated table will help to $\mathcal{A}$ in the guess phase.

- *Replace public key query:* The $\mathcal{A}$ submits polynomially bounded numbers of ID_i as queries to $\mathcal{O}_{RPK}$ to get the full private key related to the sent query from $\mathcal{O}_{RPK}$. The $\mathcal{O}_{RPK}$ returns $pr'_{U_{ix}}$ to $\mathcal{A}$ (in the background of $\mathcal{O}_{RPK}$, the random oracle calculates $pr'_{U_{ix}}$ so that satisfies the equation of $pr'_{U_{ix}} = x_i h(pr_{U_i})$). The $\mathcal{A}$ collects all responses and calculates the public key as $pu'_{U_{ix}} = pr'_{U_{ix}}D$.
  // In the following, it should be sent queries to the both of $\mathcal{O}_{SC}$ and $\mathcal{O}_{USC}$ oracles since the both of signcryption and unsigncryption algorithms use secret values.
- *Signcryption query:* The $\mathcal{A}$ generates polynomially bounded numbers of messages m_i as queries. It then submits the generated messages m_i, the received $pr'_{U_{ix}}$, and $pu_{\mathcal{C}}$ as queries to $\mathcal{O}_{SC}$ to get valid signcryptions $\sigma'_i = \{C'_i, R'_i, S'_i\}$ from $\mathcal{O}_{SC}$ as responses. The CL signcryption set $\sigma'_i = \{C'_i, R'_i, S'_i\}$ is generated due to presented CL signcryption algorithm described inSection 4.1.5. The $\mathcal{A}$ stores the queries-responses table $\{m_i, \sigma'_i\}$ to use it in the guess phase.
- *Unsigncryption query:* The $\mathcal{A}$ submits polynomially bounded numbers of random sets $\sigma'_i = \{C'_i, R'_i, S'_i\}$ as cipher queries to $\mathcal{O}_{USC}$ to get the decrypted messages m'_i as responses (the queries sent to $\mathcal{O}_{USC}$ are equivalent to queries sent to a symmetric decryption oracle to get m'_i from C'_i). The $\mathcal{O}_{USC}$ calculates (according to the unsigncryption algorithm described in Section 4.1.6) and returns m'_i to $\mathcal{A}$ as responses (the returned m'_i to $\mathcal{A}$ can be assumed as the returned response from the symmetric decryption oracle). The $\mathcal{A}$ collects all sets $\{\sigma'_i, m'_i\}$.

- *Challenge stage:* The $\mathcal{C}$ selects a random parameter x_i and executes all phases of *Partial private key extraction*, *User key generation*, and *Key generation*. It then selects two challenges m_i with equal length where $i \in \{1,2\}$ ($|m_1| = |m_2|$) and two random parameters $k_i \in_R [1, n-1]$ and calculates for $i \in \{1,2\}$, as $K_{i1} = h(k_i D)$, $K_{i2} = h(k_i pu_{\mathcal{A}})$, $C_i = Enc_{K_{i2}}(m_i)$, $r_i = h(K_{i1}|m_i)$, $S_i = \frac{k_i}{r_i + pr_{\mathcal{C}}}$, and $R_i = r_i D$.
  The $\mathcal{C}$ gives the two m_i where $i \in \{1,2\}$ (the two generated messages) and $\sigma_i = \{C_i, R_i, S_i\}$ where $i \in_R \{1,2\}$ (one of the generated signcryptions) to $\mathcal{A}$ and asks it to guess a valid value for i with probability more than $\frac{1}{2}$ such that $m_i = Dec_{h(S_i(pr_{\mathcal{A}}(pu_{\mathcal{C}}+R_i)))}(C_i)$.
- *Guess stage:* The $\mathcal{A}_I$ wins *Game 1* if it guesses the valid for i with a probability more than $\frac{1}{2}$. According to the collected responses from all oracles in $\mathcal{O}_I$ (the stored tables), $\mathcal{A}_I$ tries to guess the valid answer. To achieve this it tries as follows:
  - The $\mathcal{A}_I$ can calculate $K_{1i} = h(S_i(pu_{\mathcal{C}} + R_i))$ since it gives valid values S_i, R_i, and $pu_{\mathcal{C}}$ as a public parameter.
  - The $\mathcal{A}_I$ cannot find a valid value for $pr_{\mathcal{C}}$ since it cannot factor $pr'_{\mathcal{C}} = sh(ID_{\mathcal{C}})$ into s and $h(ID_{\mathcal{C}})$.
  - The $\mathcal{A}_I$ cannot calculate $K_{2i} = h(S_i(pr_{\mathcal{C}}(pu_{\mathcal{A}_I} + R_i)))$ since it has not valid value for $pr_{\mathcal{C}}$.
  - The $\mathcal{A}_I$ cannot calculate $m_i = Dec_{K_{2i}}(C_i)$ since it cannot calculate the valid value for decryption key K_{2i} (it is the main step in for breaking message confidentiality).
  - The $\mathcal{A}_I$ can learn no distinguish between C_1 and C_2 since the applied symmetric Dec_k algorithm is assumed secure (IND-CCA2-secure).

According to above guesses the advantage of $\mathcal{A}_I$ in *Game 1* is calculated as $ADV^{Game1}_{\mathcal{A}_I} = |Pr[\mathcal{A}_I(C_1, C_2, m_i)_{|_{i \in_R \{1,2\}}} = i | C_1, C_2, m_i] - \frac{1}{2}| = Pr[\mathcal{A}_I(C_i) = m_i | C_i] = Pr[\mathcal{A}_I(C_i, R_i, S_i) = K_{2i} | \sigma_i = \{C_i, R_i, S_i\}] = Pr[\mathcal{A}_I(sh(ID_{\mathcal{C}})) = s\ pr_{\mathcal{C}}]$. Regarding the hardness of the PFP on HC $Pr[\mathcal{A}_I(sh(ID_{\mathcal{C}})) = s\ |pr_{\mathcal{C}}] < \varepsilon$; In fact, the guessing of valid answer is equivalent to find the valid value of s, and $\mathcal{A}_I$ should be able to solve PFP if it wants to find s (the IND-CLSC-CCA2-I security is reduced to the hardness of PFP). Therefore, the advantage of $\mathcal{A}_I$ against $\mathcal{C}$ in *Game 1* is $ADV^{Game1}_{\mathcal{A}_I} < \varepsilon$, and the presented CL signcryption scheme provides confidentiality against $\mathcal{A}_I$. □

4.2.2 Confidentiality Against $\mathcal{A}_{II}$

The $\mathcal{A}_{II}$ wants to find the message m that $\mathcal{C}$ encryped it using the presented method in the presented CL signcryption scheme.

Theorem 2. *The presented CL signcryption scheme provides IND-CLSC-CCA2-II security in the RO model, and $\mathcal{A}_{II}$ can break the message confidentiality of the presented CL signcryption scheme with a negligible advantage against $\mathcal{C}$ on having the master secret key of KGC s if the PFP on HC is hard.*

Proof. The CL signcryption scheme is IND-CLSC-CCA2-II-secure if $\mathcal{A}_{II}$ cannot find a PPT algorithm to win *Game 2* with a probability more than $\frac{1}{2}$. The *Game 2* is written in the following.

- *Setup:* The $\mathcal{C}$ selects a master secret key s. To generate *params*, $\mathcal{C}$ executes *Setup* phase of the presented CL signcryption scheme using the security parameter λ. The $\mathcal{C}$ then gives *params*, $\mathcal{O}_{II} = \{\mathcal{O}_{PK}, \mathcal{O}_{SC}, \mathcal{O}_{USC}\}$, and s to $\mathcal{A}_{II}$ (in this game, $\mathcal{A}$, as an insider adversary or malicious KGC, has access to s).
- *Find stage:* The $\mathcal{A}_{II}$ generates polynomially bounded numbers of queries and submits them to all oracles in $\mathcal{O}_{II}$ as follows:
  // There is no need to send query to $\mathcal{O}_{PK}$ since $\mathcal{A}$ has access to s, and it can calculate the valid partial private key pr_{U_i}.
  - *Private key query:* The $\mathcal{A}$ submits polynomially bounded numbers of ID_i as queries to $\mathcal{O}_{PK}$ to get the partial private key pr'_{U_i} from $\mathcal{O}_{PK}$. On receiving pr'_{U_i}, $\mathcal{A}_{II}$ stores all received responses and calculates the public key $pu'_{U_i} = pr'_{U_i}D$ using the received partial private key pr'_{U_i} and D. The $\mathcal{A}$ then generates a table consist of all sent queries and responses $(\{ID_i, pr'_{U_i}, pu'_{U_i}\})$. This table will be used in the guess phase.
    // In this game, $\mathcal{O}_{USC}$ is assumed as the decryption oracle which returns m'_i for taking C'_i.
  - *Signcryption query:* The $\mathcal{A}$ generates polynomially bounded numbers of messages m_i. Then, it submits all generated messages m_i, the random full private key $pr'_{U_{ix}}$ ($\mathcal{A}$ has to use a random full private key $pr'_{U_{ix}}$ since it has not access to the valid x_i), and $pu_{\mathcal{C}}$ as queries to $\mathcal{O}_{SC}$ to get signature sets $\sigma'_i = \{C'_i, R'_i, S'_i\}$ as responses. The $\mathcal{O}_{SC}$ returns $\sigma'_i = \{C'_i, R'_i, S'_i\}$ to $\mathcal{A}$ as responses. The $\mathcal{A}_{II}$ stores all sets $\{m_i, \sigma'_i\}$ to use in the guess phase.
  - *Unsigncryption query:* The $\mathcal{A}$ generates polynomially bounded numbers of random signature sets σ_i as queries and submits them to $\mathcal{O}_{USC}$ to get messages m'_i as responses (the queries sent to $\mathcal{O}_{USC}$ are equivalent to queries sent to a symmetric decryption oracle to get m'_i from C'_i). The $\mathcal{A}_{II}$ stores all sets $\{m'_i, \sigma_i\}$ for using in the guess phase (the returned m'_i to $\mathcal{A}$ can be assumed as the returned response from the symmetric decryption oracle).
- *Challenge stage:* This phase is the same *Challenge stage* phase described in Section 4.2.1.
- *Guess stage:* The $\mathcal{A}_{II}$ wins this game if it guesses the valid value for i with a probability more than $\frac{1}{2}$. According to the collected responses from all existing oracles in $\mathcal{O}_{II}$, $\mathcal{A}_{II}$ tries to guess the valid answer for i. To achieve this it tries as follows:
  - The $\mathcal{A}_{II}$ can calculate $K_{1i} = h(S_i(pu_{\mathcal{C}} + R_i))$ since it gives valid values S_i, R_i, and $pu_{\mathcal{C}}$ as a public parameter.
  - The $\mathcal{A}_{II}$ can find valid values for $pr_{\mathcal{C}}$ and $pu_{\mathcal{C}}$ by calculating $pr_{\mathcal{C}} = sh(ID_{\mathcal{C}})$ and $pu_{\mathcal{C}} = pr_{\mathcal{C}}D$ since it has the master secret key s.
  - The $\mathcal{A}_{II}$ cannot find a valid values for $pr_{\mathcal{C}x}$ and $pu_{\mathcal{C}x}$ since it has not access to $\mathcal{O}_{RPK}$, and it cannot guess a valid value for the random number x_i.
  - The $\mathcal{A}_{II}$ cannot calculate $K_{2i} = h(S_i(pr_{\mathcal{C}x}(pu_{\mathcal{A}_{II}} + R_i)))$, and it has not a valid value for $pr_{\mathcal{C}x}$ since it cannot factor $pr_{\mathcal{C}x} = x_i pr_{\mathcal{C}x}$ into its factors to find x_i.
  - The $\mathcal{A}_{II}$ cannot calculate $m_i = Dec_{K_{2i}}(C_{ix})$ since it cannot calculate the valid value for decryption key K_{2i}.
  - The $\mathcal{A}_{II}$ can learn no distinguish between C_1 and C_2 since the applied symmetric Dec_k algorithm is assumed secure.

Like Section 4.2.1, the advantage of $\mathcal{A}_{II}$ in *Game 2* is calculated as $ADV^{Game2}_{\mathcal{A}_{II}} = |Pr[\mathcal{A}_{II}(C_1, C_2, m_i)_{|i \in_R \{1,2\}} = i | C_1, C_2, m_i] - \frac{1}{2}| = Pr[\mathcal{A}_{II}(C_i) = m_i | C_i] = Pr[\mathcal{A}_{II}(C_i, R_i, S_i) = K_{2i} | \sigma_i = \{C_i, R_i, S_i\}]$. The last equation value is lower than ε since $\mathcal{A}_{II}$ has not valid value for $pr_{\mathcal{C}x}$. Therefore, the advantage of $\mathcal{A}_{II}$ against $\mathcal{C}$ in *Game 2* is $ADV^{Game2}_{\mathcal{A}_{II}} < \varepsilon$, and the presented CL signcryption scheme provides confidentiality against $\mathcal{A}_{II}$. In fact, to find a valid value for $pr_{\mathcal{C}x}$, $\mathcal{A}_{II}$ should be able to factor $pu_{\mathcal{C}x} = pr_{\mathcal{C}x}D$ into $pr_{\mathcal{C}x}$ and D. The factoring of $pu_{\mathcal{C}x}$ is equivalent to PFP on HC; and we can say that the presented CL signcryption scheme is IND-CLSC-CCA2-II secure since PFP on HC is hard. □

4.2.3 Unforgeability Against $\mathcal{A}_I$

The $\mathcal{A}_I$ wants to create a valid CL signcryption on the message m.

Theorem 3. *The presented CL signcryption scheme provides EUF-CLSC-CMA-I security in the RO model, and $\mathcal{A}_I$ can forge the presented CL signcryption scheme using the key replacement attack with a negligible advantage against $\mathcal{C}$ if the PFP on HC is hard.*

Proof. The CL signcryption scheme is EUF-CLSC-CMA-I-secure if $\mathcal{A}_I$ can find a PPT algorithm to win *Game 3* with a negligible probability. The *Game 3* is

written in the following.

- *Setup:* This phase is the same *setup* phase described in Section 4.2.1.
- *Queries:* This phase is the same *find stage* phase described in Section 4.2.1.
- *Forgery:* In this phase, $\mathcal{A}_I$ wants to create a valid CL signcryption using the private key $pr_{U_{1x}}$ and the public key pu_{U_2} (generates a valid CL signcryption or forges it) in which the created CL signcryption is verified successfully by the public key $pu_{U_{1x}}$ and the private key pr_{U_2}. // The $\mathcal{A}$ uses the stored query-response tables to forge the CL signcryption.
  - The $\mathcal{A}_I$ cannot calculate valid value for $pr_{U_1} = sh(ID_1)$ since it has not access to the master secret key s, and it cannot factor pr'_{U_1} into its factors.
  - The $\mathcal{A}_I$ cannot compute valid value for $pu_{U_1} = pr_{U_1}D$ since it cannot cannot calculate pr_{U_1}.
  - To compute $pr_{U_{1x}} = x_1 h(pr_{U_1})$, $\mathcal{A}_I$ selects a random value x_1. But it cannot calculate $pr_{U_{1x}}$ successfully since it has not access to the valid value pr_{U_1}.
  - To create the CL signcryption on the message m, $\mathcal{A}_I$ selects a random number k_1 and calculates $K_1 = h(k_1 D)$, $K_2 = h(k_1 pu_{U_2})$, $C = Enc_{K_2}(m)$, and $r = h(K_1 \| m)$. But, it cannot compute $S = \frac{k_1}{r + pr_{U1x}}$ since it has not the valid value for pr_{U1x}.

According to the above $\mathcal{A}_I$'s tries, the advantage of $\mathcal{A}_I$ to forge successfully the presented CL signcryption scheme is calculated as $ADV_{\mathcal{A}_I}^{Game3} = Pr[\mathcal{A}_I(pr'_{U_1}) = s|pr'_{U_1}]$. According to the hardness of the PFP on HC, the mentioned probability is lower than ε, and the advantage of $\mathcal{A}_I$ against $\mathcal{C}$ to forge the presented CL signcryption scheme is $ADV_{\mathcal{A}_I}^{Game3} < \varepsilon$. In fact, forging the presented CL signcryption scheme is reduced to the hardness of the PFP on HC. Therefore, it can be said that the presented CL signcryption scheme provides EUF-CLSC-CMA-I security since the PFP on HC is hard. □

4.2.4 Unforgeability Against $\mathcal{A}_{II}$

The $\mathcal{A}_{II}$ wants to create a valid CL signcryption on the message m.

Theorem 4. *The presented CL signcryption scheme provides EUF-CLSC-CMA-II security in the RO model, and $\mathcal{A}_{II}$ can forge the presented CL signcryption scheme on having the KGC's master secret key s with a negligible advantage against $\mathcal{C}$ if the PFP on HC is hard.*

Proof. The CL signcryption scheme is EUF-CLSC-CMA-II secure if $\mathcal{A}_{II}$ can find PPT algorithm to win in *Game 4* with a negligible probability. In *Game 4*, $\mathcal{A}$, as malicious KGC, has access to the master secret key s. *Game 4* is written below:

- *Setup:* This phase is the same *setup* phase described in Section 4.2.2.
- *Queries:* This phase is the same *find stage* phase described in Section 4.2.2.
- *Forgery:* In this phase, $\mathcal{A}_{II}$ wants to create a valid CL signcryption using the private $pr_{U_{1x}}$ and the public key pu_{U_2} in which the created CL signcryption is verified successfully by the public key $pu_{U_{ix}}$ the private key pr_{U_2}.
  - The $\mathcal{A}_{II}$ can calculate a valid value for $pr_{U_1} = sh(ID_1)$ since it has the master secret key s.
  - The $\mathcal{A}_{II}$ cannot calculate a valid value for $pr_{U_{1x}} = x_1 h(ID_1)$ since it has not access to the valid value for x_1 (or $\mathcal{A}_{II}$ has not access to the replace private key oracle $\mathcal{O}_{RPK}$).
  - Like Section 4.2.3, to create the CL signcryption, $\mathcal{A}_{II}$ cannot compute $S = \frac{k_1}{r + pr_{U1x}}$ since it has not the valid value for pr_{U1x}.

The advantage of $\mathcal{A}_{II}$ to win *Game 4* is calculated as $ADV_{\mathcal{A}_{II}}^{Game4} = Pr[\mathcal{A}_{II}(pr'_{U_{1x}}) = x_1|pr'_{U_{x1}}]$. According to the hardness of the PFP on HC, the mentioned probability is lower than ε, and the advantage of $\mathcal{A}_{II}$ in *Game 4* is $ADV_{\mathcal{A}_I}^{Game4} < \varepsilon$. In fact, the hardness of the forging of the presented CL signcryption is reduced to the hardness of PFP on HC. Therefore, the presented CL signcryption scheme provides EUF-CLSC-CMA-II security since the PFP on HC is hard. □

5 Comparison

As it was described, the main contribution is to provide an efficient and short CL signcryption in which *i*) it has low communication overhead and low-cost computational cost in signcryption/unsigncryption (Table 5), *ii*) has low-cost in the user key generation and key generation phases (Table 6), and *iii*) has fast execution time (Table 7). Therefore, this section aims to compare the terms of efficiency. Based on Section 4.2, the proposed scheme provided IND-CLSC-CCA2-(I and II) security and EUF-CSLS-CMA-(I and II) security in the RO model; And all compared schemes supported the four mentioned security features in RO model [11, 19, 21, 22, 24, 26] and standard model [9, 10, 12, 16, 17, 20, 27].
The used acronyms and notation for this section are written in Table 4.

Table 4. The list of used acronyms and notations for comparison

Acronym/ Notation	Description
I	Inverse
M	Multiplication
Pa	Pairing
Pow	Power
Sq	Square
$Cost_I$	Cost of the inverse operation
$Cost_M$	Cost of the multiplication operation
$Cost_{Pa}$	Cost of the pairing operation
$Cost_{Pow}$	Cost of the power operation
$Cost_{Sq}$	Cost of the square operation
T_I	Execution time of the inverse operation
T_M	Execution time of the multiplication operation
T_{Pa}	Execution time of the pairing operation
T_{Pow}	Execution time of the power operation
T_{Sq}	Execution time of the square operation

5.1 Communication Overhead and Computational Cost

The presented CL signcryption scheme is compared with some recently-presented CL signcryption schemes, the result of which showed that our presented CL signcryption scheme is more efficient than others on transmitter and verifier sides in such a way that the Luo *et al.*'s scheme [12] is three times bigger than our CL signcryption scheme in terms of communication overhead. According to Table 4, we have shown this comparison in Table 5 in detail (in the mentioned comparison, we gave up the overhead in the KGC side and the cost of MtP function and symmetric encryption/decryption algorithm).
To calculate the communication overhead, the bit-length of all CL signcryptions' tuples should be counted. As an example, in the Zhou *et al.*'s scheme [9], six 1024-bit parameters are returned, and in the Luo *et al.*'s scheme [12], three 1024-bit parameters are returned as the CL signcryption set. Therefore, the Luo *et al.*'s scheme is more efficient than the Zhou *et al.*'s. However, our presented CL signcryption scheme is more efficient than all compared schemes since it returns 416 bits as the CL signcryption, including two 80-bit parameters on the HC and one 256-bit parameter (C) as an output of the used symmetric encryption.

5.2 User Side Cost

In addition to the two phases of *Signcryption* and *Unsigncryption*, the presented CL signcryption scheme is more efficient than recently-presented CL signcryption schemes in the two phases of *User key generation* and *Key generation* since two HC-based multiplication operators are executed by the user in the two mentioned phases in our CL signcryption scheme. However, other recently-presented schemes have higher computational costs than our CL signcryption scheme. That is, in the Caixue *et al.*'s scheme [16] three powers and three multiplications are executed, and in the Luo *et al.*'s scheme [12] two powers and one multiplication are executed (the comparison of other discussed CL signcryption schemes can be found in Table 6). The mentioned comparison is shown in Table 6 in detail.

5.3 Execution Time

According to [33–35], the total execution time on the user side can be estimated with the execution time of multiplication. These estimates are written below, and their calculations are shown in Table 7.

- $T_I \approx 240T_M$
- $T_{Pa} \approx 495T_M$
- $T_{Pow} \approx 240T_M$
- $T_{Sq} \approx 120T_M$

To compare execution time, it is assumed that users have smartphones consisting of Hisilicon Kirin 925 2.45-GHz processor, using OS Google Android 4.4.2, and 3-GB memory [35]. According to this assumption (for more detail, refer to [35]), the execution time of the multiplication operation is assumed to be 0.731 ms. We have also provided the total execution time (ms) in Table 7.

6 Conclusion

In this paper, a short and efficient CL signcryption scheme was presented based on the HC, after which it was proved that its security includes IND-CLSC-CCA2-I and EUF-CLSC-CMA-I against adversary type I, and IND-CLSC-CCA2-II and EUF-CLSC-CMA-II against adversary type II in the RO model. We reduced the mentioned security features to the hardness of the PFP on HC. The main feature of the presented CL signcryption scheme is having a short output as the CL signcryption where the communication overhead is only 416 bits, while recently-presented schemes had overheads of at least three times bigger than our proposed CL signcryption scheme. Moreover, our proposed CL signcryption scheme was proved to be more efficient than other recently-presented CL signcryption schemes in computational cost in all phases. Therefore, the present scheme is shorter and more efficient than others, and it can be applied in low-resource devices in wireless communications as well as in other fields.

Table 5. The Comparison of CL Signcryption Schemes

Feature ⇒ Scheme ⇓	**Hard problem**	**Transmitter side overhead** (signcryption)	**Receiver side overhead** (unsigncryption)	**Overall overhead estimation**	**CL signcryption length** (bit)
Zhou 2016 [9]	M-DBDH, Squ-CDH	$3Cost_{Pa} + 8Cost_{Pow} + 2Cost_{Sq} + 3Cost_M \simeq 3Cost_{Pa}$	$7Cost_{Pa} + 1Cost_{Pow} + 1Cost_I + 1Cost_{Sq} + 5Cost_M \simeq 7Cost_{Pa}$	$10Cost_{Pa}$	$6 \times 1024 = 6144$
Yu 2017 [11]	DLP	$8Cost_{Pow} + 1Cost_I + 2Cost_M \simeq 8Cost_{Pow}$	$6Cost_{Pow} + 2Cost_I + 5Cost_M \simeq 6Cost_{Pow}$	$14Cost_{Pow}$	$4 \times 1024 + 256 = 4352$
Rastegari 2017 [10]	$(K+1)$-CDHE, BDHE,	$3Cost_{Pa} + 6Cost_{Pow} + 2Cost_{Sq} + 6Cost_M \simeq 3Cost_{Pa}$	$5Cost_{Pa} + 1Cost_I + 1Cost_M \simeq 5Cost_{Pa}$	$8Cost_{Pa}$	$5 \times 1024 = 5120$
Caixue 2018 [16]	TD-q-ABDHE, MDBDH, q-SDH, Squ-CDH	$4Cost_{Pa} + 7Cost_{Pow} + 5Cost_M \simeq 4Cost_{Pa}$	$4Cost_{Pa} + 5Cost_{Pow} + 1Cost_{Sq} + 6Cost_M \simeq 4Cost_{Pa}$	$8Cost_{Pa}$	$6 \times 1024 = 6144$
Luo 2018 [12]	DBDH, CDH	$1Cost_{Pa} + 1Cost_{Pow} + 6Cost_M \simeq 1Cost_{Pa}$	$6Cost_{Pa} + 2Cost_{Pow} + 1Cost_I + Cost_{Sq} + 14Cost_M \simeq 6Cost_{Pa}$	$7Cost_{Pa}$	$3 \times 1024 = 3072$
Shan 2019 [17]	Modified-PS	$1Cost_{Pa} + 9Cost_{Pow} + 1Cost_I + 8Cost_M \simeq Cost_{Pa}$	$5Cost_{Pa} + 3Cost_{Pow} + 7Cost_M \simeq 5Cost_{Pa}$	$6Cost_{Pa}$	$8 \times 160 = 1280$
Gao 2019 [19]	CDH	$1Cost_I + 3Cost_M \simeq 1Cost_I$	$4Cost_M$	$1Cost_I$	$3 \times 1024 = 3072$
Liu 2020 [21]	DLP, RSA	$6Cost_{Pow} + 4Cost_M + 2Cost_I \simeq 6Cost_{Pow}$	$6Cost_{Pow} + 4Cost_M + 1Cost_I \simeq 6Cost_{Pow}$	$12Cost_{Pow}$	$4 \times 1024 = 4096$
Kasyoka 2020 [22]	DLP, ECDLP	$4Cost_M$	$4Cost_M$	$8Cost_M$	$3 \times 1024 = 3072$
Our Scheme	PFP (or HCDLP)	$1Cost_I + 4Cost_M \simeq 1Cost_I$	$4Cost_M$	$1Cost_I$	$2 \times 80 + 256 = 416$

Table 6. Comparison of User Side Cost

Phase ⇒ Scheme ⇓	**User key generation**	**Key generation**
Zhou 2016 [9]	$2Cost_{Pow} + 1Cost_I$	$2Cost_M$
Rastegari 2017 [10]	$2Cost_{Pow} + 1Cost_I$	$4Cost_{Pow} + 2Cost_{Sq} + 4Cost_M$
Caixue 2018 [16]	$3Cost_{Pow} + 1Cost_M$	$2Cost_M$
Luo 2018 [12]	$1Cost_{Pow}$	$1Cost_{Pow} + 1Cost_M$
Shan 2019 [17]	$1Cost_{Pa}$	$1Cost_M$
Gao 2019 [19]	$2Cost_M$	$1Cost_M$
Liu 2020 [21]	$1Cost_{Pow}$	$1Cost_{Pow}$
Kasyoka 2020 [22]	$4Cost_M$	$1Cost_{Pow}$
Our scheme	$1Cost_M$	$1Cost_M$

Table 7. Comparison of User Side Execution Time

Phase ⇒ Scheme ⇓	**Estimated time**	**Execution time** (ms)
Zhou 2016 [9]	$722T_M$	527.782
Rastegari 2017 [10]	$1924T_M$	1406.444
Caixue 2018 [16]	$723T_M$	528.513
Luo 2018 [12]	$481T_M$	351.611
Shan 2019 [17]	$496T_M$	362.576
Gao 2019 [19]	$3T_M$	2.193
Liu 2020 [21]	$480T_M$	350.88
Kasyoka 2020 [22]	$244T_M$	178.364
Our scheme	$2T_M$	1.462

References

[1] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (Feb. 1978), 120-126. DOI:https://doi.org/10.1145/359340.359342

[2] R. C. Merkle, "Protocols for Public Key Cryptosystems", 1980 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 1980, pp. 122-122, doi: 10.1109/SP.1980.10006.

[3] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", in IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469-472, July 1985, doi: 10.1109/TIT.1985.1057074.

[4] Huang X., Susilo W., Mu Y., Zhang F. (2005) On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In: Desmedt Y.G., Wang H., Mu Y., Li Y. (eds) Cryptology and Network Security. CANS 2005. Lecture Notes in Computer Science, vol 3810. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11599371 - 2

[5] M. Barbosa and P. Farshim. 2008. Certificateless signcryption. In Proceedings

of the 2008 ACM symposium on Information, computer and communications security (ASIACCS '08). Association for Computing Machinery, New York, NY, USA, 369-372. DOI:https://doi.org/10.1145/1368310.1368364

[6] Dorothy E. Denning and Dennis K. Branstad. 1996. A taxonomy for key escrow encryption systems. Commun. ACM 39, 3 (March 1996), 34-40. DOI:https://doi.org/10.1145/227234.227239

[7] Zheng Y. (1997) Digital signcryption or how to achieve cost(signature & encryption) ≪ cost(signature) + cost(encryption). In: Kaliski B.S. (eds) Advances in Cryptology — CRYPTO '97. CRYPTO 1997. Lecture Notes in Computer Science, vol 1294. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0052234

[8] Yumin Yuan, Chenhui Wang, Certificateless signature scheme with security enhanced in the standard model, Information Processing Letters, Volume 114, Issue 9, 2014, Pages 492-499, ISSN 0020-0190, https://doi.org/10.1016/j.ipl.2014.04.004.

[9] Zhou, C., Gao, G. & Cui, Z. Certificateless Signcryption in the Standard Model. Wireless Pers Commun 92, 495-513 (2017). https://doi.org/10.1007/s11277-016-3554-8

[10] Rastegari, Parvin, and Mehdi Berenjkoub. "An Efficient Certificateless Signcryption Scheme in the Standard Model". ISeCure 9.1 (2017).

[11] Huifang Yu, Bo Yang, Pairing-Free and Secure Certificateless Signcryption Scheme, The Computer Journal, Volume 60, Issue 8, August 2017, Pages 1187-1196, https://doi.org/10.1093/comjnl/bxx005

[12] Luo, M., Wan, Y. An Enhanced Certificateless Signcryption in the Standard Model. Wireless Pers Commun 98, 2693-2709 (2018). https://doi.org/10.1007/s11277-017-4995-4

[13] Zhenhua Liu, Yupu Hu, Xiangsong Zhang, Hua Ma, Certificateless signcryption scheme in the standard model, Information Sciences, Volume 180, Issue 3, 2010, Pages 452-464, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2009.10.011.

[14] Songqin Miao, Futai Zhang, Sujuan Li, Yi Mu, On security of a certificateless signcryption scheme, Information Sciences, Volume 232, 2013, Pages 475-481, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2011.11.045.

[15] Jian Weng, Guoxiang Yao, Robert H. Deng, Min-Rong Chen, Xiangxue Li, Cryptanalysis of a certificateless signcryption scheme in the standard model, Information Sciences, Volume 181, Issue 3, 2011, Pages 661-667, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2010.09.037.

[16] Caixue, Z. H. O. U. "Certificateless signcryption scheme without random oracles". Chinese Journal of Electronics 27, no. 5 (2018): 1002-1008.

[17] Shan Shan, An Effcient certificateless Signcryption Scheme without Random Oracles I.J. of Electronics and Information Engineering, Vol.11, No.1, PP.9-15, Sept. 2019 (DOI: 10.6636/IJEIE.201909 11(1).02) 9 - 15

[18] Canard S., Trinh V.C. (2016) An Efficient Certificateless Signature Scheme in the Standard Model. In: Ray I., Gaur M., Conti M., Sanghi D., Kamakoti V. (eds) Information Systems Security. ICISS 2016. Lecture Notes in Computer Science, vol 10063. Springer, Cham. https://doi.org/10.1007/978-3-319-49806-5 - 9

[19] GaiMei Gao1, XinGuang Peng, and LiZhong Jin, Effcient Access Control Scheme with Certi cateless Signcryption for Wireless Body Area Networks, International Journal of Network Security, Vol.21, No.3, PP.428-437, May 2019 (DOI: 10.6633/IJNS.201905 21(3).09) 428 - 437

[20] Xi-Jun Lin, Lin Sun, Zhen Yan, Xiaoshuai Zhang, Haipeng Qu, On the Security Of A Certificateless Signcryption With Known Session-Specific Temporary Information Security In The Standard Model, The Computer Journal, Volume 63, Issue 8, August 2020, Pages 1259-1262, https://doi.org/10.1093/comjnl/bxz157

[21] Xiaoguang Liu, Ziqing Wang, Yalan Ye, Fagen Li, An efficient and practical certificateless signcryption scheme for wireless body area networks, Computer Communications, Volume 162, 2020, Pages 169-178, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2020.08.014.

[22] Philemon Kasyoka, Michael Kimwele, Shem Mbandu Angolo, Cryptanalysis of a Pairing-free Certificateless Signcryption scheme, ICT Express, 2020, ISSN 2405-9595, https://doi.org/10.1016/j.icte.2020.07.006.

[23] Luo, Wei, and Wenping Ma. "Secure and efficient data sharing scheme based on certificateless hybrid signcryption for cloud storage". Electronics 8, no. 5 (2019): 590.

[24] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K. R. Choo and Y. Park, "Certificateless-Signcryption-Based Three-Factor User Access Control Scheme for IoT Environment", in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3184-3197, April 2020, doi: 10.1109/JIOT.2020.2966242.

[25] H. Chen, Y. P. Hu, Z. Z. Lian, and H. W. Jia, "Efficient certificateless encryption schemes from lattices", J. Softw., vol. 27, no. 11, pp. 2884-2897, 2016.

[26] H. Yu, L. Bai, M. Hao and N. Wang, "Certificateless Signcryption Scheme From

Lattice", in IEEE Systems Journal, doi: 10.1109/JSYST.2020.3007519.

[27] Yuan, Y. Security Analysis of an Enhanced Certificateless Signcryption in the Standard Model. Wireless Pers Commun 112, 387-394 (2020). https://doi.org/10.1007/s11277-020-07031-9

[28] Menezes, Alfred, Robert Zuccherato, and Yi-Hong Wu. An elementary introduction to hyperelliptic curves. Faculty of Mathematics, University of Waterloo, 1996.

[29] Lange, T. Formulae for Arithmetic on Genus 2 Hyperelliptic Curves. AAECC 15, 295-328 (2005). https://doi.org/10.1007/s00200-004-0154-8

[30] Ganesan, Ramachandran, Mohan Gobi, and Kanniappan Vivekanandan. "A Novel Digital Envelope Approach for A Secure E-Commerce Channel". IJ Network Security 11, no. 3 (2010): 121-127.

[31] Sharma, G., Bala, S. & Verma, A.K. Pairing-Free Certificateless Ring Signcryption (PF-CLRSC) Scheme for Wireless Sensor Networks. Wireless Pers Commun 84, 1469-1485 (2015). https://doi.org/10.1007/s11277-015-2698-2.

[32] Yanwei Zhou, Bo Yang, Wenzheng Zhang, Provably secure and efficient leakage-resilient certificateless signcryption scheme without bilinear pairing, Discrete Applied Mathematics, Volume 204, 2016, Pages 185-202, ISSN 0166-218X, https://doi.org/10.1016/j.dam.2015.10.018.

[33] Mehibel N, Hamadouche M. Authenticated secret session key using elliptic curve digital signature algorithm. Security and Privacy. 2021;e148. https://doi.org/10.1002/spy2.148

[34] Tahat, N. and Abdallah, E.E., 2016. A proxy partially blind signature approach using elliptic curve cryptosystem. International Journal of Mathematics in Operational Research, 8(1), pp.87-95.

[35] Kumar V, Ahmad M, Kumari A, Kumari S, Khan MK. SEBAP: A secure and efficient biometric-assisted authentication protocol using ECC for vehicular cloud computing. Int J Commun Syst. 2019;e4103. https://doi.org/10.1002/dac.4103.

Saeed Banaeian Far received the B.Sc and M.Sc degrees in electrical engineering from Yadegar - e- Imam Khomeini (rah), shahr-e-rey Branch Islamic Azad University Tehran, Iran, in 2012, and 2016, respectively. He is currently a PhD student in the Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. His research interests include cryptography and network security, privacy, and privacy in blockchain-based networks.

Maryam Rajabzadeh Assar received the B.S. degree in electrical engineering from Shahid Bahonar University of Kerman, Kerman, Iran, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2008 and 2014, respectively. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. Her research interests include provable security, digital signatures, design and analysis of cryptographic protocols and network security, and security in industrial control systems.