Journal of Scientific & Industrial Research
Vol. 81, December 2022, pp. 1326-1335
DOI: 10.56042/jsir.v81i12.69313

A New Pairing-Free Certificateless Signcryption Scheme

Srinivasa Rao Gopisetti[1], Ramesh Babu Amarapu[2], Gowri Thumbur[3] & Vasudeva P Reddy[1]*

[1]Department of Engineering Mathematics, Andhra University, Visakhapatnam, 530 003 India

[2]Department of Engineering Mathematics, Anil Neerukonda Institute of Technology and Science, Visakhapatnam 531 163, India

[3]Department of Electronics and Communication Engineering, GITAM University, Visakhapatnam 530 045, India

Received 03 June 2022; *revised 12 August 2022*; *accepted 07 October 2022*

Signcryption is a cryptographic primitive which provides unforgeability and confidentiality for digital communications. Many signcryption schemes have been constructed in the literature for secure communication between smart objects. But, many of these existing schemes are not secure and inefficient for resource constrained applications like WSNs, Mobile computing, VANETs and IoT applications. To enrich the security and efficiency issues, in this paper, we propose a new signcryption scheme in certificateless based framework and prove its security under the CDHP and ECDLP assumptions. The efficiency analysis indicates that our scheme is more efficient than other existing signcryption schemes and is well suitable for resource-constrained applications.

Keywords: Authentication and confidentiality, Certificateless based cryptography, EUF-CLSC-CMA, IND-CLSC-CCA, IoT applications

Introduction

With the rapid development of wireless and communication technologies, the Internet of things (IoT) is one of the most debatable topics among the research community. The IoT applications influences our daily lives, i.e., it is deployed in smart cities, smart homes and e-health, VANETS etc.[1–4] It needs millions of devices to be connected and communicate each other. So that the reliable connectivity and their security are of great challenges in the design of IoT applications.[2,3] Many of these applications will be realized as embedded systems which rely heavily on security and efficiency mechanisms. When data is transmitted through open network, the authenticity and confidentiality of data must be considered as basic security factors in the design of many IoT applications and these security properties can be achieved through digital signature and encryption mechanisms respectively. In 1997, Zheng[5] combine these two processes in the single mechanism called signcryption. But, in the year 2001, Jung *et al*.[6] spotted out that the scheme of Zheng *et al*.[5] is unable to produce forward secrecy. Signcryption cost is less than the traditional encryption and then signature. In 2007, Baek *et al*.[7] produced a frame work of Signcryption scheme and its security.

*Author for Correspondence
E-mail: vasucrypto@andhrauniversity.edu.in

In Public Key Infrastructure (PKI) based setting and certificate based systems, storing, updating certificates, revocation, filing certificates leads to the complex certificate management process and is the highly complicated situation. Identity based (ID-based) setting is the solution for this critical certificate management. In the year 2002, with the help of bilinear pairings, Lee *et al*.[8] constructed an ID–based Signcryption scheme. In 2003, Libert and Quisquater[9] improved their scheme, also presented an efficient Signcryption scheme under the q-strong Diffie-Hellman Problem. Later on the flow of research in ID-based Signcryption is happened and several schemes are proposed in the literature[10–12] ID-based cryptography requires Private Key Generator (PKG) to compute the private keys of users based on their identities. Hence the private key of the identities are known by PKG then it will generates malicious key escrow problem. Thus ID-based systems get rid of certificate management issues, but such systems leads to have an inherent key escrow problem. To eliminate it, the Certificateless Public Key Cryptography (CL-PKC) is invented by Al-Riyami and Paterson[13] in 2003. In this methodology, the Key Generation Centre (KGC) combines the partial private key and secret value of the user, to generate the full private key of the user. This combination will exclude the key-escrow problem. Therefore, CL-PKC has many suitable characteristics of real-time applications so that such certificateless frame work has been widely used in practice and hence the schemes based on CL-PKC have attracted the attention of many cryptographic researchers.[4, 12–17]

Many certificateless Signcryption schemes (CLSC) are proposed with bilinear pairings and without using bilinear pairings.[4,15,16] Implementation of Pairing based cryptographic schemes is more complex and is less efficient because of high computation cost and large bandwidth. In view of this, cryptographic schemes with pairing free environment over ECC are desirable to implement complex cryptographic schemes. In 2015, Won *et al.*[4] proposed a secure CLSC with Tag key encapsulation mechanism. To improve the computational efficiency in the year 2018, Cao *et al.*[15] analyzed several CLSC schemes and cryptanalyzed. Also, Cao *et al.*[15] constructed a pairing free signcryption scheme based on the GDH and ECDLP problems. In the year 2018, Cui *et al.*[16] constructed a new CLSC scheme without bilinear pairings. In the year 2019, Zhou *et al.*[17] proposed an improved lightweight CLSC scheme for mobile-health applications but the scheme is not secure. Some Signcryption schemes with the additional functionalities are also appeared in the literature.[18–24] The details of security issues for signcryption schemes are discussed in the literature.[12,15,24]

While numerous studies have been published on addressing the security and efficiency issues of CLSC schemes, most of the existing CLSC schemes are not much efficient in view of security and computational costs to deal with the low computation, less bandwidth and less memory devices for real world applications. Therefore, the designing of secure and efficient signcryption schemes for IoT and other applications is still major challenging. In view of this, in this work, a new paring free CLSC scheme (PF-CLSC) is constructed. The advantages and main contributions of the present work are as follows:

- The proposed efficient pairing free CLSC scheme is constructed based on the ECC. This scheme avoids the complex bilinear pairings operations and used the lightweight operations on elliptic curve. This improves the computational efficiency in signcryption and unsigncryption process.
- The security of the scheme relies on the intractability of the ECDLP & CDH problems.
- This scheme also improves the efficiency when compared with the relevant and existing schemes.

Preliminaries

This section presents basic facts about mathematical preliminaries on Elliptic curves, related computational hard problems.

Elliptic Curve Group

The equation of the elliptic curve $E: y^2 = x^3 + ax + b;\ a,b \in F_p$, where F_p represents a finite field and p is prime. The discriminant is $4a^3 + 27b^2 \neq 0 \mod p$. The set of all solutions on the elliptic curve and an infinite point $\mathcal{O}$ is represented by the $E(F_p)$, that is $E(F_p) = \{(x,y)/x,y \in F_p\} \cup \mathcal{O}$. The number of points on $E(F_p)$ is represented by q, which becomes the order of the elliptic curve

Computational Hard Problems

Definition 1: Discrete Logarithm Problem (DLP): Let $(G,+)$ be a cyclic group with prime order q and P is the generator of G. For a given $P,Q \in G$ such that $Q = aP$, the ECDLP is to find $a \in Z_q^*$. The computation of ECDLP is hard by any polynomial-time bounded algorithm.

Definition 2: Computational Deffie-Hellman Problem (CDHP): Given a $(P,aP,bP) \in G$ for two unknown elements $a,b \in Z_q^*$, the CDHP is to find abP from aP and bP. For anonymous values $a,b \in Z_q^*$, computing abP is difficult.

Framework for PF-CLSC Scheme

Our proposed PF-CLSC scheme composed with the below six probabilistic polynomial time algorithms:

(1) ***Setup:*** KGC performs this stage with the security parameter $k \in Z^+$ as input and outputs the common necessary parameters *params* and also master secret key *msk*. KGC keeps the *msk* as secret and publishes the system parameters *params* publicly.

(2) ***Set Secret Value:*** This algorithm is implemented by user with *params*, his identity $ID \in \{0,1\}^*$ as inputs and selects a random $x_{ID} \in Z_q^*$ as his secret value.

(3) ***Set Partial Private Key:*** This algorithm is performed by KGC to generate the user's partial private key D_i. KGC sends the PPK D_i to the user ID_i through a secure way

(4) ***Set Private and Public Keys:*** This algorithm executed by the user with inputs *params*, identity $ID \in \{0,1\}^*$ and the corresponding partial private key d_{ID} and generates the users public key PK_{ID} and private key (or secret key) SK_{ID}.

(5) ***Signcryption:*** For a given *params*, a message m, a sender and receiver's public keys (PK_S, PK_R) and a signer's private key SK_s, the sender ID_S perform this algorithm to generate signcryptext δ.

(6) ***Unsigncryption:*** The receiver runs this Unsigncryption algorithm after receiving a signcryption δ, *params*, public keys PK_s, PK_R of the corresponding identities as ID_s, ID_R and his own secret key SK_R to decrypts the signcryptext δ. If the signcryptext is valid, it is accepted, otherwise rejected

The workflow of our CLSC scheme is depicted in the following Fig. 1.

Security Model

Based on the potential behavior[16,24], we consider two types of adversaries to discuss the security of our CLSC scheme: Adversary $\mathcal{A}_I$ (Type-I adversary) and Adversary $\mathcal{A}_{II}$ (Type-II adversary).

Type-I adversary $\mathcal{A}_I$ unaware of the master secret key, but it can replace the any ones public keys. Therefore, adversary $\mathcal{A}_I$ is also treating as a malicious user.

Type-II adversary $\mathcal{A}_{II}$ knows the master secret key but cannot substitute the public keys of the users. It is a malicious KGC and it constructs the user's secret key.

Important notations and their meanings are given in Table 1.

Proposed Pairing-Free Certificateless Signcryption Scheme

This segment presents a new PF-CLSC scheme with the following six PPT algorithms. The workflow of these algorithms is presented in Fig. 2.

Setup: KGC run this phase. For a given security parameter k, KGC chooses the secure hash functions $H_1, H_2, H_3 : \{0,1\}^* \to Z_p^*$. Let the primes p, q. The KGC selects $s \in_R Z_p^*$ as the master secret key (*msk*), determine $P_{pub} = sP$ as the system public key and outputs the system public parameters as $params = \{P, p, q, E, G, H_1, H_2, H_3, P_{pub}\}$.

Table 1 — Notations and their meanings

Notation	Meanings
k	Security parameter,
msk	Master secret key
CLSC	Certificateless Signcryption
params	System parameters
KGC	Key Generation Centre
PPK	Partial Private Key
PPT algorithm	Probabilistic Polynomial Time algorithm
ROM	Random Oracle Model
IND-CLSC-CCA	Indistinguishable Certificateless Signcryption Chosen Cipher text Attack
IND-CLSC-CMA	Indistinguishable Certificateless Signcryption Chosen Message Attack
ID_i	Identity of the user *i*
H_i	Collision resistant hash functions
D_i	Partial private key of the user *i*
PK_i	Public key of the user *i*
SK_i	Secret key of the user *i*
δ	Signcryptext
m	Message

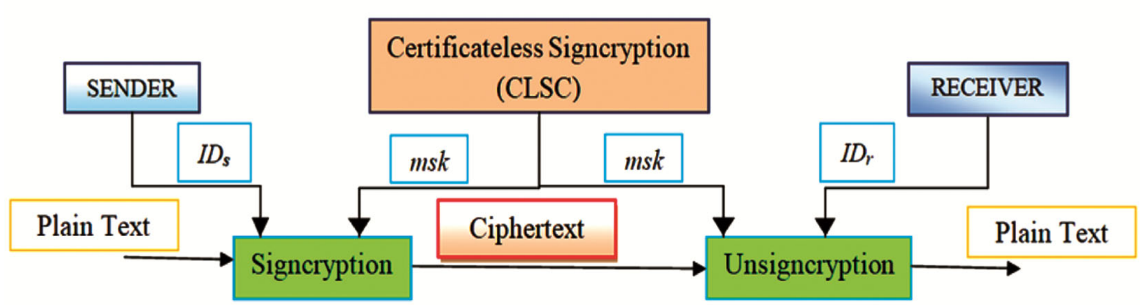

Fig. 1 — Schematic diagram of CLSC

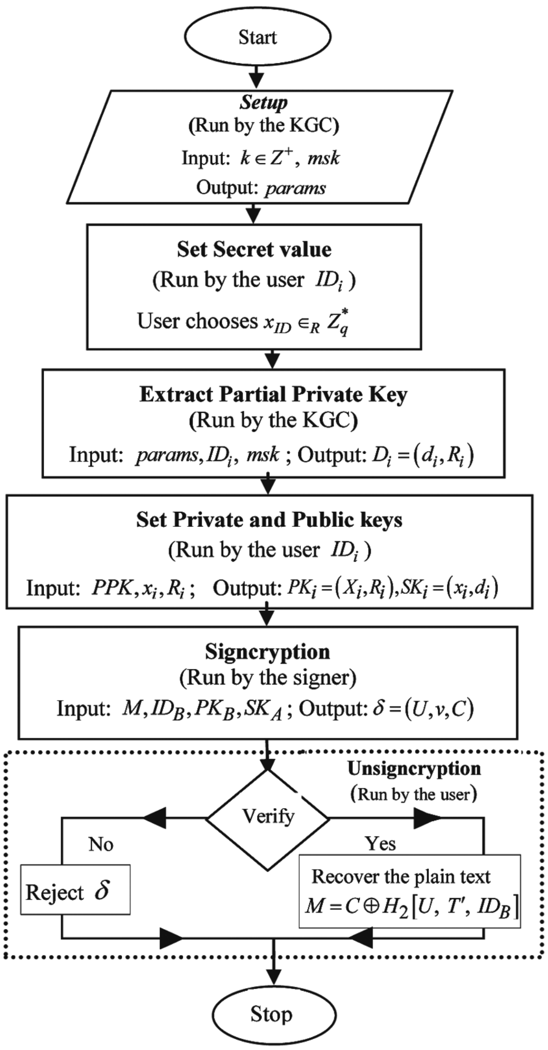

Fig. 2— Workflow of the proposed PF-CLSC scheme

Set Secret Value: The user ID_i selects $x_i \in_R Z_q^*$ and computes $X_i = x_i P$.

Extract Partial Private Key: With the input of $params$, $msk\, s$, user $ID_i \in \{0,1\}^*$; KGC implements the following to generate the Partial private key.

- KGC chooses a random integer $r_i \in Z_q^*$ and creates $R_i = r_i P,\ d_i = (r_i + s h_{1i}) \bmod \mathrm{q}$, where $h_{1i} = H_1(ID_i, R_i, X_i) \in \{0,1\}^*$.
- KGC sends the $D_i = (d_i, R_i)$ as the PPK to the user ID_i.

The user can validate D_i by verifying $d_i P = R_i + h_{1i} P_{pub}$.

Set Private and Public Keys: Each user ID_i generates his private key as $SK_i = (x_i, d_i)$ and sets his publickey as $PK_i = (X_i, R_i)$.

Signcryption: The signer run this algorithm with the inputs public parameters, message, public keys of the sender and receiver and a signer's private key, the signer or sender performs the following to generate signcryptext δ.

On inputting $params$, message M, for the receiver identity ID_B with public key $PK_B = (X_B, R_B)$ and the secret key $SK_A = (x_A, d_A)$, the signer ID_A generates a Signcryption on M through the following steps:

Signer chooses $u \in_R Z_q^*$, computes $U = uP$, $T = u\left[h_{1B} P_{pub} + R_B + X_B\right]$.

$C = H_2[U,\ T,\ ID_B] \oplus M,\ \ v = u + h_{3B}(d_A + x_A) \bmod p$, where $h_{3B} = H_3(ID_A, ID_B, C, U, X_B)$.

Signcryption on m_i is $\delta = (U, v, C)$.

Unsigncryption: The receiver runs this Unsigncryption algorithm after receiving a signcryption, public parameters, public keys and the corresponding identities of the sender and the receiver. The receiver uses his own secret key to decrypts the signcryptext δ. If the signcryptext is valid, the receiver accepts the signcryption text, otherwise he rejects its. Receiver B takes the public parameters $params$, public keys PK_A, PK_B of ID_A, ID_B and his own secret key SK_B to decrypts the signcryptext $\delta = (U, v, C)$ as follows:

Receiver B computes $h_{3B} = H_3(ID_A, ID_B, C, U, X_B)$.

Verifies $vP = U + h_{3B}(R_A + X_A + h_{1A} P_{Pub})$.

Computes $T' = U(x_B + d_B)$ and recover the message $M = C \oplus H_2[U,\ T',\ ID_B]$.

Analysis of our PF-CLSC Scheme

This section presents the correctness and the security aspects against the adversaries $\mathcal{A}_I$ and $\mathcal{A}_{II}$ for the proposed PF-CLSC scheme.

Proof of Correctness

$$vP = \left[u + h_{3B}(d_A + x_A)\right]P = uP + h_{3B}(d_A P + x_A P)$$
$$= U + h_{3B}\left[R_A + h_{1A} P_{pub} + X_A\right].$$

Security Analysis

Theorem 1: In the ROM model, our PF-CLSC scheme is IND-PF-CLSC-CCA secure against Type-I and Type-II adversaries with the claim of that the CDH problem is intractable.

Proof: The proof of this theorem can be derived from the following Lemma 1 and Lemma 2.

Lemma 1: Our PF-CLSC protocol is PF-CLSC-CCA2 secure against $\mathcal{A}_I$ with the intractability of the CDH problem.

Proof: Suppose there is an adversary $\mathcal{A}_I$ attempting to break our PF-CLSC security. ξ is given with an instance of an CDHP. The challenger ξ uses $\mathcal{A}_I$ to find the solution of the CDHP instance.

The challenger ξ sets $P_{pub} = sP$ and treated H_i $(1 \le i \le 3)$ are random oracles. The algorithm ξ gives *params* to $\mathcal{A}_I$. To keep uniformity, ξ maintains lists $\mathcal{L}_i$ $(1 \le i \le 3)$ and $\mathcal{L}_k$. ξ chooses ID_t as the target identity.

The algorithm ξ c hoose $h_{1t}, x_t \in Z_q^*$ and sets H_1 $(ID_t, R_t, X_t) = -h_{1t}$, then creates $R_t = h_{1t}P_{pub} + aP - x_tP$ and also the $X_t = x_tP$. The value of a is unknown to ξ and aP is the instance of the CDHP problem. ξ adds the tuples $\langle ID_t, R_t, X_t, -h_{1t} \rangle$ and $\langle ID_t, \perp, x_t, R_t, X_t \rangle$ into the lists $\mathcal{L}_1$ and $\mathcal{L}_k$. ξ responds as follows for the queries formed by the adversary $\mathcal{A}_I$.

1) H_1 ***queries:*** When $\mathcal{A}_I$ makes a H_1 query with the tuple (ID_i, R_i, X_i), then ξ searches the list $\mathcal{L}_1$ for $(ID_i, R_i, X_i, -h_{1i})$. ξ gives $-h_{1i}$ if it already available. Otherwise, ξ selects $-h_{1i} \in_R Z_q^*$ and adds $(ID_i, R_i, X_i, -h_{1i})$ to $\mathcal{L}_1$. Finally, the algorithm ξ returns $-h_{1i}$ as answer to $\mathcal{A}_I$.

2) H_2 ***queries:*** Suppose $\mathcal{A}_I$ makes a H_2 query on $\langle U, T, Y, ID_i \rangle$, ξ searches the list $\mathcal{L}_2$ for the tuple $\langle aP, U, T \rangle$. If it exists, ξ returns l_i and replaces the symbol $*$ with T. Else, ξ selects $l_i \in_R \{0,1\}^n$ and inserts in $\mathcal{L}_2$. finally, ξ returns l_i to $\mathcal{A}_I$ as an answer to H_2 query.

3) H_3 ***queries:*** When $\mathcal{A}_I$ makes a H_3 query on the tuple $\langle ID_i, C, U, X_i / R_i \rangle$. ξ returns h_{1i} to $\mathcal{A}_I$ if $\langle ID_i, C, U, X_i / R_i, h_{1i} \rangle$ already in the list $\mathcal{L}_3$, else ξ chooses $h_{1i} \in_R Z_q^*$, and inserts into the list $\mathcal{L}_3$ and response $H = h_{1i}$ to $\mathcal{A}_I$. For other queries made by $\mathcal{A}_I$, ξ responds as below.

Phase-I

i) Set user key queries: If $\mathcal{A}_I$ request secret value query on ID_i, ξ responds as follows. ξ aborts if PK_{ID_i} for ID_i is replaced, otherwise returns x_i from $\mathcal{L}_k$.

ii) Extract Partial Private Key queries: Suppose that $\mathcal{A}_I$ makes a PPK query on ID_i to ξ, then ξ aborts if $ID_i = ID_t$. Otherwise if $ID_i \ne ID_t$, ξ searches $\mathcal{L}_k$ for a tuple $\langle ID_i, d_i, x_i, R_i, X_i \rangle$ and returns d_i. If no such tuple exists then ξ uses the PPK algorithm to computes PPK of ID_i and adds $\langle ID_i, d_i, x_t, R_i, X_i \rangle$ to $\mathcal{L}_k$ as a response to PPK query.

iii) Set private key queries: $\mathcal{A}_I$ asks ξ for full private key of a user with ID_i. ξ stops the process if $ID_i = ID_t$. Otherwise ξ searches for $\langle ID_i, d_i, x_i, R_i, X_i \rangle$ in $\mathcal{L}_k$ and gives (x_i, d_i) if it appears. Otherwise, ξ picks $h_{1i}, b_i, x_i \in_R Z_q^*$, and sets H_1 $(ID_i, R_i, X_i) = -h_{1i}$, $R_i = h_{1i}P_{pub} + b_iP$ and computes $X_i = x_iP$, $d_i = b_i$. These values satisfies $d_iP = R_i + H_1$ $(ID_i, R_i, X_i)P_{pub}$. ξ includes the tuple $\langle ID_i, R_i, X_i, -h_{1i} \rangle$ in $\mathcal{L}_1$ and the $\langle ID_i, d_i, x_i, R_i, X_i \rangle$ in $\mathcal{L}_k$ lists and replies (x_i, d_i) as an answer to the private key query.

iv) Set public key queries: When $\mathcal{A}_I$ submits a public key query on ID_i, ξ inspects $\mathcal{L}_k$ for a tuple $\langle ID_i, d_i, x_i, R_i, X_i \rangle$ and returns (X_i, R_i), if it appears. Otherwise, ξ proceeds as above in set private key queries and returns (X_i, R_i).

v) Public-key-replacement queries: $\mathcal{A}_I$ replaces (X_i, R_i) by (X_i', R_i') for a user ID_i. ξ updates the list $\mathcal{L}_k$ as $\langle ID_i, \_\_, \_\_, X_i', R_i' \rangle$. ξ uses the new public key (X_i', R_i') for further computations or responses of queries asked by the adversary $\mathcal{A}_I$.

vi) CLSC-Signcryption queries: $\mathcal{A}_I$ submits a signcryption query on (ID_A, ID_B) with senders and receivers public keys (X_A, R_A) and (X_B, R_B) a message m to ξ, ξ do the following:

- If $ID_A \neq ID_t$, ξ executes the private key algorithm and computes the full private key SK_A of ID_A. Then, ξ executes the CLSC signcryption algorithm and outputs the signcryptext δ. ξ sends it to $\mathcal{A}_I$.
- If $ID_A = ID_t$, (and hence $ID_B \neq ID_t$), then the challenger ξ chooses $u_t, h_t^{-1} \in Z_q^*$ and computes $U = u_t P - h_t^{-1}(aP - x_t P)$, $T = u_t\left[H_1(ID_B, R_B, X_B)P_{pub} + R_B + X_B\right]$, $C = H_3(U, T, ID_B) \oplus m$. Algorithm ξ sets $H_3(ID_A, C, U, T, R_A) = h_t$, $H_3(ID_A, C, U, T, X_A) = h_t'$ and adds the tuple $\langle ID_A, C, U, T, X_A, h_t' \rangle$ and $\langle ID_A, C, U, T, R_A, h_t \rangle$ to the list $\mathcal{L}_3$. ξ computes $v = u_t + x_t h_t'$ and $(ID_A, ID_B, \delta = (U, v, C))$ as the signcryption.

vii) CLSC-Unsigncryption queries: $\mathcal{A}_I$ makes this query on $\delta = (U, v, C)$ and ID_A, ID_B to the challenger ξ. ξ runs CLSC-verify algorithm and results $\perp$ if the validation fails. If $ID_B \neq ID_t$, ξ retrieves the private key and go through the CLSC-unsigncryption and gives m_i to $\mathcal{A}_I$. If $ID_B = ID_t$, ξ inspects in the list $\mathcal{L}_2$ for the tuple $\langle U, T, Y, ID_B, h_i \rangle$ and returns h_i if it exists. Otherwise, ξ adds the tuple $\langle U, \_\_, \_\_, ID_B, h_i \rangle$ for a random h_i to the list $\mathcal{L}_2$.

Challenge: After the Phase *I*, $\mathcal{A}_I$ came up with two dissimilar messages M_0^* and M_1^*, ID_A^*, ID_B^* to ξ. ξ aborts the game if $ID_B^* = ID_t$. Otherwise, ξ do the following:

- Retrieve PK_A^*, PK_B^* from $\mathcal{L}_k$.
- Sets $U^* = bP$, where bP is given instance of the CDHP and $b \in_R Z_q^*$, choose $T^* \in_R G_q$.
- Chooses $\gamma \in \{0,1\}$, h' and sets $C^* = m_\gamma \oplus h$, choose h_{3A}', $h_{2A}^{-1} \in Z_q^*$, insert $\left(ID_i, C^*, U^*, T^*, R^*, h_{2A}^{-1}\right)$ to the list $\mathcal{L}_3$, computes $v^* = u_i^* + h_{3A}'\left(d_A^* + x_A^*\right)$, where d_A^*, x_A^* can be retrieved from the set-private-key queries.
- Returns $\delta^* = \left(C^*, U^*, v^*\right)$.

Phase II:

On receiving the challenge ciphertext δ^*, the $\mathcal{A}_I$ allows to ask queries as in the Phase *I*, and $\mathcal{A}_I$ should not make any unsigncryption query δ^*.

Guess: Since the adversary $\mathcal{A}_I$ can breaks the security IND-CLSC-CCA2-*I* of the proposed CLSC, $\mathcal{A}_I$ makes a H_1 query with the tuple $\left(U^*, T^*, Y^*, ID_B^*\right)$ as an inputs, here $T^* = b\left[\left(h_{1i}P_{pub} + R_B + X_B\right)\right] = abP$ i.e., one of $T's$ in the list $\mathcal{L}_2$ is the query corresponds to ID_A^* and receiver ID_B^*; such T^* is the solution of the instance of the CDHP.

Lemma 2: If an adversary $\mathcal{A}_{II}$ succeeded in the Game *II* with the non-negligible probability in polynomial time against IND-CLSC-CCA2-*II* security, then there exists an algorithm that resolves the CDHP.

Proof: Assume that there exists an algorithm $\mathcal{A}_{II}$ which can breach the IND-CLSC-CCA2-*II* security of the CLSC. ξ take the help of $\mathcal{A}_{II}$ to find abP. ξ sets $P_{pub} = sP$ and H_i $(1 \le i \le 3)$ as random oracles. ξ sends *params* to $\mathcal{A}_{II}$. ξ preserve the lists $\mathcal{L}_i$ $(1 \le i \le 3)$ and $\mathcal{L}_k$. Assume that ξ fix ID_t as the target identity. ξ picks $a_t, l_t \in Z_q^*$ at random and takes H_1 $(ID_t, R_t, X_t) = l_t$, and calculates the values $R_t = a_t P$, $d_t = a_t + l_t s$ and $X_t = aP$. Here, a is unknown to ξ. $\mathcal{A}_{II}$ asks queries to random oracles H_i $(1 \le i \le 3)$. For these queries, ξ responds as follows.

1) ***H_1 queries:*** When adversary $\mathcal{A}_{II}$ asking a H_1 query on (ID_i, R_i, X_i), ξ look over the list $\mathcal{L}_1$ for a tuple $\langle ID_i, R_i, X_i, l_i \rangle$. ξ returns l_i. Otherwise, ξ randomly selects $l_i \in_R Z_q^*$ and results l_i as the output to a H_1 query. Then, ξ adds (ID_i, R_i, X_i, l_i) to the list $\mathcal{L}_1$.

2) H_2 queries: If $\mathcal{A}_{II}$ came up with this on $\langle U,T,Y,ID_i\rangle$, ξ searches and outputs Y. Otherwise, ξ searches the list with $\mathcal{L}_2$ with entries $\langle U,T,*,ID_i,l_i\rangle$ for different l_i, such that to output 1 as answer to the query tuple $\langle aP,U,Y\rangle$.

3) H_3 queries: Suppose that $\mathcal{A}_{II}$ raises a H_3 query on $\langle ID_i, C,U,T,X_i/R_i\rangle$, then algorithm ξ searches $\langle ID_i, C,U,T,X_i/R_i,h_i\rangle$ in $\mathcal{L}_3$ and gives $H=h_i$ to the adversary $\mathcal{A}_{II}$. Else, ξ selects $h_{1i}\in_R Z_q^*$, and inserts in $\mathcal{L}_3$ and outputs $H=h_{1i}$ to $\mathcal{A}_{II}$. for the remaining queries of $\mathcal{A}_{II}$, ξ acts as follows.

Phase I:

i) Set user key queries: $\mathcal{A}_{II}$ may submits ID_i to ξ and makes a query on with ID_i. If $ID_i = ID_t$, then ξ abbots. If $ID_i \neq ID_t$, then ξ searches $\langle ID_i,\ d_i,x_i,R_i,X_i\rangle$ in $\mathcal{L}_k$ list and outputs x_i. Otherwise ξ selects $a_i,x_i\in_R Z_q^*$, and sets $H_1\ (ID_i,R_i,X_i)=l_i,\ R_i=a_iP,\ \ d_i=a_i+l_is,\ X_i=a_iP.$ Finally, ξ adds $\langle ID_i,\ R_i,X_i,l_i\rangle$ and $\langle ID_i,\ d_i,x_i,R_i,X_i\rangle$ to the lists $\mathcal{L}_1$ and $\mathcal{L}_k$ respectively and gives x_i to the adversary.

ii) Set private key queries: When $\mathcal{A}_{II}$ came up with this query on ID_i to ξ. If $ID_i = ID_t$, then ξ abbots. If $ID_i \neq ID_t$, then ξ searches for $\langle ID_i,\ d_i,x_i,R_i,X_i\rangle$ in $\mathcal{L}_k$ and returns (x_i,d_i) if it exists. Otherwise ξ randomly takes $a_i,x_i,l_i\in_R Z_q^*$ to compute $H_1\ (ID_i,R_i,X_i)=l_i$, and also calculates $R_i=a_iP,\ d_i=a_i+l_is,\ X_i=x_iP.$ ξ add the tuple $\langle ID_i,\ R_i,X_i,l_i\rangle$ to $\mathcal{L}_1$ and $\langle ID_i,\ d_i,x_i,R_i,X_i\rangle$ to $\mathcal{L}_k$ and then finally outputs (x_i,d_i).

iii) Set public key queries: Suppose that $\mathcal{A}_{II}$ asking this query on ID_i, ξ looks $\mathcal{L}_k$ for $\langle ID_i,\ d_i,x_i,R_i,X_i\rangle$. If it is available, ξ gives (X_i,R_i). Otherwise, ξ chooses randomly $a_i,x_i,l_i\in_R Z_q^*$, to form $H_1\ (ID_i,R_i,X_i)=l_i$, $R_i=a_iP,\ d_i=a_i+l_is$ and $d_iP=R_i+h_{1i}P_{pub}$. Then the public key as $X_i=x_iP.$ ξ adds $\langle ID_i,\ R_i,X_i,l_i\rangle$ in $\mathcal{L}_1$ and $\langle ID_i,\ d_i,x_i,R_i,X_i\rangle$ into $\mathcal{L}_k$ and returns (X_i,R_i).

iv) CLSC-signcrypt queries: When $\mathcal{A}_{II}$ makes a signcryption query on inputs $ID_A,\ ID_B$, public keys responds (X_A,R_A), (X_B,R_B) and a message m to ξ. ξ proceeds as follows:

- If $ID_A\neq ID_t$, ξ runs set private key algorithm and obtain the full secret key SK_A. Then, ξ obtains the signcryptext δ by implementing the actual CLSC Signcryption algorithm. ξ forwards δ to $\mathcal{A}_{II}$.
- If $ID_A = ID_t$, (and hence $ID_B\neq ID_t$), then ξ can obtains the full private key SK_B represents to ID_B. ξ chooses $u_t,h_t,h_t'\in Z_q^*$, computes $U=u_tP-h_tX_t$, $T_B=U(r_B+d_B)$. The algorithm ξ sets hash values $H_3(ID_A,ID_B,C,U,X_B)=h_t$ and $H_3(ID_A,ID_B,C,U,X_B)=h_t$ and adds the tuple $\langle ID_A,m,U,T,X_A,h_t\rangle$ to the list $\mathcal{L}_3$. ξ computes ciphertext as $C=H_2(U,T,ID_B)\oplus m$ and $v=u_t+h_td_t$, outputs $(ID_A,ID_B,\delta=(U,v,C))$ as the signcryptiontext.

The signcryptext is valid because of the following:

$$
\begin{aligned}
vP &= U_t+h_t\left[X_t+R_t+l_tP_{pub}\right]=u_tP-h_tX_t+h_t\left[X_t+R_t+lsP\right]\\
&=u_tP-h_tX_t+h_tX_t+h_tR_t+h_tlsP\\
&=u_tP+h_t\left[r_tP+lsP\right]=\left[u_t+h_t\left(r_t+ls\right)\right]P=\left[u_t+h_td_t\right]P.
\end{aligned}
$$

v) CLSC-Unsigncryption queries: $\mathcal{A}_{II}$ submits the signcryption text $\delta=(C,U,v)$ and ID_A,ID_B to the challenger ξ. If $ID_B\neq ID_t$, ξ executes the CLSC-unsigncrypt algorithm, and outputs the of CLSC-unsigncrypt to $\mathcal{A}_{II}$. Or else, ξ sieves the list $\mathcal{L}_3$ for the tuples are of the forms $\langle ID_A,M,C,U,T,X_A,R_A,h_i\rangle$ and $\langle ID_B,M,U,T,X_B,R_B,h_i\rangle$ and retrieve T_i. The algorithm ξ searches the list $\mathcal{L}_2$ for a tuple $\langle U,T,Y,ID_B,l_i\rangle$. If such tuple exists, then ξ the retrieve the message as $C\oplus l_i$.

Challenge: Finally, $\mathcal{A}_{II}$ selects two distinct and same length of messages M_0^* and M_1^*, identities ID_A^* and ID_B^*. Here, the PPK of ID_B^* was not queried in Phase *I*. ξ fails to challenge if $ID_B^*\neq ID_t$. Otherwise, ξ proceeds as follows to produce the challenge ciphertext.

- Sets $U^*=bP$, where bP is given instance of the CDHP problem, $b\in_R Z_q^*$ and choose $T^*\in_R G_q$.
- Selects randomly a bit $\gamma\in\{0,1\}$, selects hash value h_{2i}^{-1} at random and computes $C^*=m_\gamma\oplus h_{2i}^{-1}$,

takes $h'_{3i}, h_{2i}^{-1} \in Z_q^*$, adds $\left(ID_i, C^*, U^*, T^*, R_i^*, h'_{3i}\right)$, $\left(ID_i, C_i^*, U_i^*, T_i^*, R_i^*, h_{2i}^{-1}\right)$ to $\mathcal{L}_3$, computes $v^* = u_A^* + h'_{3i}\left(d_A^* + x_A^*\right)$, where d_s^*, x_s^* can be obtained as the answers of the set-private-key queries.

- Returns $\delta^* = \left(C^*, U^*, v^*\right)$.

Phase II:

$\mathcal{A}_{II}$ makes a adaptive queries as in Phase *I*. However, in this phase II, the adversary $\mathcal{A}_{II}$ cannot run CLSC-Unsigncryption query on δ^*.

Guess: Since $\mathcal{A}_{II}$ is capable to breach the IND-CLSC-CCA2-*II* security of the proposed CLSC scheme and $\mathcal{A}_{II}$ must be submit a H_2 query on $\left(U^*, T^*, Y^*, ID_B^*\right)$ with have $Y^* = x_B^* \cdot U^* = abP$. Thus, one of the T value is stored in $\mathcal{L}_2$ as the answer of H_2 query corresponding to ID_A^* and ID_B^* and is the solution for the CDHP problem.

Theorem 2: The PF-CLSC Scheme is existentially unforgeable in the ROM model under the intractability of the ECDL problem.

Proof: The proof of this theorem follows from Lemma 3 and Lemma 4.

Lemma 3: Our PF-CLSC scheme is secure against the adversary $\mathcal{F}_I$ in the ROM under the intractability of the ECDL problem.

Lemma 4: Our PF-CLSC scheme is secure against the adversary $\mathcal{F}_{II}$ in the ROM with the assumption that ECDLP is hard.

Performance Analysis

The efficiency analysis of the proposed PF-CLSC scheme including the computation and communication costs by computing Signcryption cost, Decryption cost and Ciphertext length are presented. Since the nature of IoT devices requires limited computing operations, limited band-width for communication and less memory. Our scheme consists of such lightweight operations only. Also, the symbols $|G|$, $|Z_q^*|$, $|m|$ represents the bit lengths of an element in G, Z_q^* and a message *m* respectively. To evaluate the operations or costs, a list of basic cryptographic operations and their average run time are considered from the works [14–15, 24–25] and are presented in Table 2.

The computation cost of the scheme consists of the several aspects: signcryption cost, Unsigncryption cost and total cost. These costs of are very high when the construction is with the use of bilinear pairing operations. But our scheme is constructed without using bilinear pairings. However, the contrasts of the constructed PF-CLSC with various existing signcryption schemes are presented in Table 3. The computational cost of signcryption, unsigncryption and total cost in milliseconds are presented in Table 4. The Zhou *et al.*[17] scheme requires $5T_{SM} = 2.21\ ms$ as the Signcryption cost, $7T_{SM} = 3.094\ ms$ as Unsigncryption cost. Hence the total computation cost for Zhou *et al.*[17] scheme is $5.304\ ms$. The Won *et al.*[4] scheme requires $4T_{SM} + 2T_{PA} = 1.7716\ ms$ as Signcryption cost, $7T_{SM} + 3T_{PA} = 3.0094\ ms$ as Unsigncryption cost. Hence the total cost for Won *et al.*[4] scheme is $4.871\ ms$. The Cao *et al.*[15] scheme

Table 2 — Cryptographic operations

Notations	Description
T_{SM}	Scalar point multiplication over elliptic curves $T_{SM} \approx 0.442\ ms$
T_{PA}	Point addition on Elliptic curve $T_{PA} \approx 0.0018\ ms$
T_{INV}	Modular inversion operation $T_{INV} \approx 0.18879\ ms$

Table 3 — Comparison of the computation cost of our PF-CLSC scheme

S.No	Name of the Scheme	Signcryption Cost	Unsigncryption Cost	Total Cost
1	Zhou *et al.*[17]	$5T_{SM}$	$7T_{SM}$	$12\ T_{SM}$
2	Won *et al.*[4]	$4T_{SM} + 2T_{PA}$	$7T_{SM} + 3T_{PA}$	$11T_{SM} + 5T_{PA}$
3	Cao *et al.*[15]	$6T_{SM} + 3T_{PA}$	$5T_{SM} + 3T_{PA}$	$11T_{SM} + 6T_{PA}$
4	Cui *et al.*[16]	$5T_{SM} + 3T_{PA} + 1T_{INV}$	$6T_{SM} + 2T_{PA}$	$11T_{SM} + 5T_{PA} + 1T_{INV}$
5	Our Scheme	$3T_{SM} + 2T_{PA}$	$4T_{SM} + 3T_{PA}$	$7T_{SM} + 5T_{PA}$

Table 4 — Comparison of the computation cost of our PF-CLSC scheme

S. No	Name of the Scheme	Signcryption Cost in *ms*	Unsigncryption Cost in *ms*	Total Cost in *ms*	Improvement
1	Zhou *et al.*[17]	2.21	3.094	5.304	41.50%
2	Won *et al.*[4]	1.7716	3.0094	4.871	36.30%
3	Cao *et al.*[15]	2.6574	2.2154	4.873	36.32%
4	Cui *et al.*[16]	2.4042	2.6556	5.06	38.68%
5	Our Scheme	1.3296	1.7734	3.103	—

Table 5— Comparison of the communication cost

S.No.	Name of the PF-CLSC Scheme	Total Communication Cost in *bits*
1	Zhou *et al.*[17]	$2\lvert G\rvert + \lvert Z_q^*\rvert = 800$
2	Won *et al.*[4]	$\lvert G\rvert + 2\lvert Z_q^*\rvert = 640$
3	Cao *et al.*[15]	$\lvert G\rvert + 2\lvert Z_q^*\rvert = 640$
4	Cui *et al.*[16]	$2\lvert G\rvert + \lvert Z_q^*\rvert = 800$
5	Our Scheme	$\lvert G\rvert + 2\lvert Z_q^*\rvert = 640$

requires $6T_{SM} + 3T_{PA} = 2.6574\ ms$ as Signcryption cost, $5T_{SM} + 3T_{PA} = 2.2154\ ms$ as Unsigncryption cost. For Cao *et al.*[15] scheme is $4.8728\ ms$. The Cui *et al.*[16] scheme requires $5T_{SM} + 3T_{PA} + 1T_{IN} = 2.40419\ ms$ as Signcryption cost, $6T_{SM} + 2T_{PA} = 2.6556\ ms$ as Unsigncryption cost. Therefore, the total computation cost for Cui *et al.*[16] scheme is $5.05979\ ms$. The proposed scheme requires $3T_{SM} + 2T_{PA} = 1.3296\ ms$ for Signcryption, $4T_{SM} + 3T_{PA} = 1.7734\ ms$ for Unsigncryption. Thus, our scheme needs $3.103\ ms$.

From Table 4, we can perceive that our PF-CLSC scheme is $\left(\frac{5.304 - 3.103}{5.304}\right)100 = 41.50\ \%$ faster than the scheme Zhou *et al.*[17] scheme, Our PF-CLSC scheme is 36.30% faster than Won *et al.*[4] scheme, also ours is 36.32% faster than Cao *et al.*[15] scheme and 38.68% faster than Cui *et al.*[16] scheme.

The computational improvements of our PF-CLSC scheme with existing schemes are given in Table 4. Another aspect to estimate the efficiency is communication cost. For computing such cost, the length of the signcryption text was considered. In our PF-CLSC scheme, the signcryption text is $\delta = (U, v, C)$. For ECC based pairing free scheme, the length of elements in a group G is considered as $\lvert G\rvert = 320\ bits$ and $\lvert m\rvert = \lvert ID\rvert = \lvert q\rvert = 160\ bits$.[15–17,24]

The communication costs for the proposed and other existing schemes are calculated and are given in Table 5. Zhou *et al.*[17] scheme has the communication cost $2\lvert G\rvert + \lvert Z_q^*\rvert = 2(320) + 160 = 800\ bits.$ The Won *et al.*[4] has the communication cost $\lvert G\rvert + 2\lvert Z_q^*\rvert = 640\ bits.$ The Cao *et al.*[15] has the communication cost $\lvert G\rvert + 2\lvert Z_q^*\rvert = 640\ bits.$ The Cui *et al.*[16] has the communication cost $2\lvert G\rvert + \lvert Z_q^*\rvert = 2(320) + 160 = 800\ bits.$

The proposed scheme requires the cost $\lvert G\rvert + 2\lvert Z_q^*\rvert = 320 + 2(160) = 640\ bits$ for the communication which is equivalent to Won *et al.*[4] and Cao *et al.*[15] schemes and fewer than the schemes Zhou *et al.*[17], Cui *et al.*[4] schemes.

From the above discussion, our PF-CLSC scheme has better efficiency and high security and hence it can be well suitable for the construction of resource constrained IoT applications.

Conclusions

In this paper, we constructed an efficient and secure pairing-free certificateless signcryption scheme. This scheme ensures the security services' like confidentiality, authentication and is proven secure in the random oracle model with assumption that the CDH and ECDL problems are intractable. Furthermore, the designed approach of our PF-CLSC scheme improves the computational efficiency from 36.30% to 41.50% and improves the communicational efficiency by 20%, than the existing schemes. Based on the computational and communication efficiency and enhanced security, our PF-CLSC scheme is more attractive and is suitable for deployment in IoT applications.

References

1 Noor M B M & Hassan W H, Current research on internet of things (IoT) security: a survey. *J Comp Net* **148** (2019) 283–294.

2 Samalia M G, Neto M, Fernandes D B, Freire M & Inacio P M, Challenges of securing internet of things devices: a survey, *J Secu Priv* **01(02)** (2018) 1–32.

3 Ugnaya G, Radhika & Vijayaraj N, A survey on internet of things: applications, recent issues, attacks and security mechanisms, *J Circuits system and comp* **30(05)** (2021) 1–45.

4 Won, J, Seo S H & Berti E, A secure communication protocol for drones and smart objects, *Proc. ASIA Comp Comm Sec* (2015) 249–260.

5 Zheng Y, Digital signcryption or how to achieve cost (signature and encryption = cost (Signature) + cost (encryption*), Proc. Crypto, California* (1997) 165–179.

6 Jung H Y, Lee D H, Lim J I & Chang K S, Signcryption schemes with forward secrecy, *Proc. Info Secu Appl* (2001) 403–475.

7 Baek J, Steinfeld R & Zheng Y, Formal proofs for the security of signcryption, *J Cryptology*, **20(2)** (2007) 203–235.

8 Malone-Lee J, Identity based signcryption, *Cryptology e-Print Archive Report, IACR*, (2002).

9 Libert B & Quisquater J J, A new identity based signcryption scheme from pairings, *Proc. IEEE Info Theory Works* (2003) 155–158.

10 Swapna G & Vasudeva Reddy P, Efficient identity based aggregate signcryption scheme using bilinear pairings over elliptic curves, *Proc. IOP conf series, J Phys* **1344** (2019).

11 Yu Y, Yang B, Sun Y & Zhu S, Identity based signcryption scheme without random oracles, *J Comp stand interf* **31** (2009) 56–62.

12 Hussain S, Ullah S S, Uddin M, Iqbal J & Chen C L, A comprehensive survey on signcryption security mechanisms in wireless body area networks, *Sensors*, **22(3)** (2022) 1072. doi: 10.3390/s22031072.

13 Al-Riyami S S& Paterson K G, Certificateless public key cryptography, *Proc Int Conf Theory Appl Cryptol Info Sec*, (2003) 452–473.

14 Gowri Thumbur, Srinivasa Rao G, Vasudeva Reddy P, Gayathri N B & Rama Koti Reddy D V, Efficient pairing-free certificateless signature scheme for secure communication in resource-constrained devices, *J Commu Lett IEEE*, **24(8)** (2020), 1641–1645.

15 Cao L & Ge W, Analysis of certificateless signcryption schemes and construction of a secure and efficient pairing free one based on ECC, *KSII Trans Internet Info Sys*, **12(9)** (2018) 4527–4547.

16 Cui L, Yun B, Lin S & Wenhua B, A new certificateless signcryption scheme without bilinear pairing, *Proc. 13th Int Conf Comp Sci Edu (ICCSE),* (2018) 1–5. doi: 10.1109/ICCSE.2018.8468859.

17 Zhou C, An improved lightweight certificateless generalized signcryption scheme for mobile-health system, *Int J Distrib Sensor Netw*, **15(1)** (2009) 1–16.

18 Ullah I, Alkhalifah A, Rehman S U, Kumar N & Khan M A, An anonymous certificateless signcryption scheme for internet of health things, *IEEE Access*, **9** (2021) 101207–101216. doi: 10.1109/ACCESS.2021.3097403.

19 Kim T H, Kumar G, Saha R, Buchanan W J, Devgun T & Thomas R, LiSP-XK: Extended light-weight signcryption for iot in resource-constrained environments, *IEEE Access,* **9** (2021) 100972–100980.

20 Kasyoka P N, Kimwele M &Mbandu S A, Efficient certificateless signcryption scheme for wireless sensor networks in ubiquitous healthcare systems, *J Wirel Pers Commu* **18** (2021) 3349–3366. https://doi.org/10.1007/s11277-021-08183-y.

21 Ali U, Yamani Idna Idris Md,. Frnda Jaroslav, Bin Ayub, Mohamad Nizam, Alroobaea, Roobaea, Almansour F, Shagari N M, Ullah I & Ihsan Ali, Hyper elliptic curve based certificateless signcryption scheme for secure IIoT communications, *J Compu Materi Continua*, **71(2)** (2022), 2515–2532. http://doi:10.32604/ cmc.2022.019800.

22 Abdullah A M, Ullah I, Khan M A, Alsharif M H, Mostafa S M & Wu J M, An efficient multi-document blind signcryption scheme for smart grid-enabled industrial internet of things, *J Wirel Commu Mob Compu* (2022). http://dx.doi.org/10.1155/2022/7779152.

23 Yang Y, Zhang L, Zhao Y, Choo K K R & Zhang Y, Privacy preserving aggregation authentication scheme for safety warning system in fog-cloud based VANET, *IEEE Trans Info Forensics Secu* **17** (2022)317-331. http://doi: 10.1109/TIFS.2022.3140657.

24 Du H, Wen Q, Zhang S & Mingchu Gao, A pairing free certificateless signcryption scheme for vehicular ad hoc networks, *Chinese J Electro* **30(5)** (2021) 947–955. https://doi.org/10.1049/cje.2021.07.006.

25 MIRACL Library: http//certivox.com/miracl/2018 (12th May 2018).