Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 2** : 1–10.
ISSN-0037-8712
doi:10.5269/bspm.80554

An Efficient Anonymous Certificateless Proxy Signcryption Scheme without Bilinear Pairings over Elliptic Curves

G. Swapna, G. Naga Malleswari, Kusuma Tummala, Gowri Thumbur

ABSTRACT: In real-world applications, anonymity, confidentiality, and unforgeability play pivotal roles in secure communication, especially in the scenarios where privacy and security are dominant concerns. The novel paradigm of certificateless anonymous proxy signcryption introduced in this work allows the original signer to expose the proxy signer's identity if it is misused, while simultaneously granting anonymity to the proxy signer. We design a CL-APSC mechanism and it's security is proven by using the Diffie-Hellman and discrete logarithm problem in the elliptic curve. Additionally, we provide a security model and a formal definition of the certificateless anonymous proxy signcryption (CL-APSC) technique. Since the proposed approach operates in a certificateless environment, it eliminates the overhead of maintaining certificates and avoids the key escrow problem. Our aim in this article is to reduce the computational cost with the existing schemes and also provides anonymity to the proxy identity.

Keywords: Certificateless cryptography, anonymous proxy signcryption, delegation of rights, provable security, elliptic curve.

Contents

1. Introduction

Confidentiality and authentication are the two crucial aspects in public key cryptography (PKC). Confidentiality is provided through encryption, while a digital signature guarantees authenticity. Encoding a message and then developing a digital signature on the ciphertext, and vice versa, are the two fundamental features in PKC. To reduce communication and computation cost, Zheng [1] developed a new cryptographic technique, i.e., signcryption. It executes both encryption and digital signature instantaneously in a single step. This significantly reduces the computing cost, as modular exponentiation is an important component of computational expense. Given that modular exponentiation requires more processing, this drastically lowers the cost of calculation. Due to the improvements of signcryption, different variants of these signcryption schemes are presented in [3], like proxy, hybrid, ring, aggregate, and multi-signcryption etc. To assist people in delegating their authority to an individual or group of individuals in order to complete a task on time, such as digital contract signing, work transfers for deputies, or online proxy auctions, new application demands might occasionally necessitate the privilege delegation mechanism. The first proxy signature was developed by Mambo et al. [4], which enables to transfer of signing ability to a new user called a proxy signer acting on behalf of the original sender, to meet the necessities of the applications mentioned above. The only disadvantage of using a proxy signature is that while it possesses signature legitimacy, it cannot guarantee message confidentiality. To overcome this, the authors introduced proxy signcryption [5] by fusing the functions of proxy signature and encryption. In a proxy signcryption approach, the sender or delegator can transfer their signing rights to another user,

2020 *Mathematics Subject Classification:* 11G07, 11T71.
Submitted December 02, 2025. Published March 14, 2026

Typeset by $\mathcal{B}^{\mathcal{S}}\mathcal{P}_{\mathcal{M}}$ style.
© Soc. Paran. de Mat.

called the proxy sender. The proxy sender then performs the signcryption on behalf of the delegator, as the recipient can verify using the delegator's public key.

Motivation: An anonymous proxy signcryption scheme [2] holds promising applications across various domains due to its ability to combine the functionalities of proxy and signcryption while preserving the anonymity of the proxy signer. Here are some potential applications: Secure Communication in Sensitive Environments: Industries that handle sensitive information, such as healthcare or finance, can use anonymous proxy signcryption to ensure secure and anonymous communication between entities. For example, doctors communicating with patients or financial institutions sharing sensitive data with clients could benefit from this approach. Secure Online Voting Systems: Anonymous proxy signcryption can help develop secure and anonymous online voting systems, ensuring voter anonymity and voting process integrity to enhance trust and participation in democracy. Anonymized Financial Transactions: In financial transactions where parties wish to remain anonymous, such as peer-to-peer lending or blockchain-based transactions, anonymous proxy signcryption can enable secure communication and transaction verification without disclosing the identities of the transacting parties. Nowadays, there is a great need to keep proxy signers' identities hidden from other users on the network. To address this, we develop an algorithm that applies the anonymity principle to a proxy signing mechanism in elliptic curve cryptography.

Related work: Numerous additional techniques and advancements have been presented since the theory of signcryption was initiated [1]. A security model of signcryption was first presented by Baek et al. [6], which admits formal proofs for signcryption's confidentiality and unforgeability. The identity-based signcryption (IBC) technique was created by Malone-Lee [7]. After this, many IBC such as proxy, Multi, Aggregate, Hybrid, and Ring [3] signcryption schemes, are presented in the literature. But these identity-based signcryptions are suffer from key escrow problems. To prevent key management in PKC and key escrow problems in IBC, Barbosa [8] presented a certificateless signcryption scheme (CLSC) in 2008. In this process, KGC creates only a partial private key of the user, with the legitimate user's cooperation, and remains unaware of the full private key of the user within the network. With the advantage of the certificateless technique, Yang Feng initially presented the concept of certificateless proxy signcryption [9] without using costly pairings. But Bhatia et al. [10] first demonstrated that the mechanism [9] can't resist a forgery attack by a type II adversary and proposed a proxy signcryption in a certificateless setting. In 2018, Li et al. [11] evidenced that Bhatia's scheme [10] is insecure and can't prevent key replacement attacks and forgery attacks. Yu and Wang [12] combined the concept of proxy signature with certificateless signcryption by using cyclic multiplicative groups in 2019. A proxy signcryption employing ECC was proposed by Muhammad Asghar Khan et al. [13] and is used in the drone industry. None of these approaches offers the proxy's anonymity from the recipient. Thus, developing a new certificate-less (CL) anonymous proxy signcryption is focused and helpful in the sensitive applications like health care, smart homes, and financial transactions. At present, no proxy signcryption scheme guarantees proxy anonymity from the receiver in CLSC. To overcome this, we propose an anonymous proxy signcryption (CL-APSC) scheme that provides proxy sender anonymity using a pseudonym strategy. The security is based on the hardness of the DLP and ECDHP on elliptic curves. The original sender can reveal the proxy's identity if misuse occurs, while maintaining confidentiality otherwise.

2. Our Proposed CL-APSC Scheme

Here, we describe a novel certificateless anonymous proxy signcryption (CL-APSC), to guarantee the anonymity of the proxy sender. By using this approach, the original signer may also disclose the proxy signer's identity in an instance of misuse. Setup, Partial-key- Generation, Set-Private and Public-Keys, Proxy-Key-Generation, Anonymous-Proxy-Signcryption, Un-sigcryption, and Proxy-Revelation are the algorithms of our CL-APSC scheme.

Set Up: The PKG describes a message $m \in \{0,1\}^n$ and security parameter k, and then choose a generator P of cyclic group G_1 with a prime order $q = 2^n$. Also defines the following hash functions, $H_1 : \{0,1\}^* G \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$, $H_3 : G \rightarrow \{0,1\}^n$ $H_4 : \{0,1\}^n \rightarrow G$ The PKG randomly chooses his master key $s \in Z_q^*$ and computes $P_{pub} = sP$. PKG publishes the public parameters $params$=$\{k, n, q, Z_q^*, H_1, H_2, H_3, H_4, P, P_{pub}\}$.

Partial Key Generation: After receiving user's identity ID_i, KGC randomly chooses $x_i \in Z_q^*$ and then computes $X_i = x_iP$ and $d_i = x_i + sH_{1i}(X_i, ID_i, P_{pub})$. Then KGC sends the pair (X_i, d_i) to the user i through the secure channel.

Proxy-Key-Generation: Original signcrypter computes the following steps to attain a warrant m_w, it involves the delegation period, the type of message, pseudonym of the proxy signer etc. Finally, the algorithm creates the proxy sender's private and public keys.

- **Pseudonym-Generation:** Proxy signer P randomly picks η. Select random $\rho_\eta \in Z_q^*$, then perform $U_\eta = \rho_\eta P$ and $h_\eta = H_2(\eta, U_\eta) \in Z_q^*$ Computes $t_\eta = \rho_\eta + h_\eta(d_P + r_P h_P)$ $in Z_q^*$ where $h_p = H_2(R_P, ID_P)$. Use secure channel to send η and $\sigma_\eta = (U_\eta, t_\eta)$ to the original signcrypter O. The original signer O, agrees σ_η if $t_\eta P = U_\eta + h_\eta(X_P + P_{pub}H_{1P} + R_P h_P) \in Z_q^*$ and then creates the Pseudonym for proxy as $H_Q = U_\eta + H_P$, where $H_P = H_4(D_P)$.
- **Generate Delegation Rights:** Original sender O, generates delegation rights to proxy signcrypter with the inputs $SK_O = (d_O, r_O)$, public key $PK_O = (X_O, R_O)$ and message warrant M_w.
  - Original sender randomly selects $\rho_w \in Z_q^*$ and then computes $Z_w = \rho_w P$.
  - Computes $H_w = H_2(M_w, Z_w)$
  - And then calculate $T_w = \rho_w + H_w(d_O + r_O h_O)$ where $h_O = H_2(ID_O, R_O)$. Sends the warrant M_w and it's delegation $\sigma_w = (Z_w, T_w)$ to proxy signcrypter.
- **Delegation- Verify:** Proxy signcrypter accepts the delegation σ_w if $T_w P = U_w + H_w(X_O + P_{pub}H_{1O} + R_O h_O)$.
- **Proxy-Key-Generation:** P creates $D_Q = T_w + h_2\rho_\eta$, where $h_2 = H_2(M_W, U_w, P_{pub})$ if the delegation is valid.

Anonymous Proxy-Signcryption: On behalf of original sender O, the proxy sender signcrypt a message $m \in \{0,1\}^n$ anonymously for the receiver R, by taking the inputs of $\{Z_w, D_Q, (X_R, R_R)\}$. The proxy sender randomly selects $u \in Z_q^*$ and then computes

- $U = uP$
- $V = uX_R$ and $h = H_3(V)$
- $C = h \oplus m$
- $h_3 = H_4(C \parallel ID_O \parallel ID_R \parallel U_\eta)$
- $S = u + h_3 D_Q$.

Send the anonymous proxy signcryption $\sigma = \{M_W, U_\eta, U, V, C, Z_w\}$ to the receiver R.
Unsigncryption: After receiving the anonymous proxy signcryption text $\sigma = (M_W, U_\eta, U, V, C, Z_w)$. The receiver verifies the Authenticity using the following computations.

- Computes $V' = x_R U$.
- Accept the message if it holds the following $SP = h_3(U_w + H_w(H_{10}P_{pub} + h_0 R_0 + X_0) + h_2 U_\eta) + U$ reject otherwise.
- If Anonymous proxy signcryption is verified, then decrypt the message $m = C \oplus h$, where $h = H_3(V')$.

Proxy revelation: To disclose the proxy signer, original sender O can first expose the nonce η along with its signature (U_η, V_η) and prove that $H_Q = H_P + U_\eta$, where $H_P = H_4(ID_P)$.

3. Alalysis of CL-APSC Scheme

This section discusses about correctness of the scheme, Security analysis and efficiency analysis.

3.1. Correctness Algorithm:

The verification of proposed CL-APSC Scheme is described as follows.
Correctness of Decryption:

$$V^{'} = x_R U = x_R uP = uX_R = V$$

Correctness of Verification:

$$\begin{aligned} SP &= (h_3 D_Q + U)P \\ &= h_3[T_w + h_2\rho_\eta]P + UP \\ &= UP + h_3[T_w P + h_2\rho_\eta P] \\ &= UP + h_3[(\rho_w + H_w(d_0 + r_0 h_0))P + h_2 U_\eta] \\ &= UP + h_3[\rho_w P + H_w(x_0 P + sPH_{10}) + r_0 h_0 P) + h_2 U_\eta] \\ &= h_3(U_w + (X_0 + H_{10}P_{pub} + h_0 R_0)H_w + h_2 U_\eta) + U \end{aligned}$$

3.2. Security Analysis:

Based on the challenge of solving the Elliptic Curve Computational Diffie-Hellman problem (EC-CDHP), we will prove that the proposed CL-APSC mechanism is confidential and unforgeable like [12].

Confidentiality

Theorem 3.1 *Our CL-APSC scheme is secure under IND-CCA2 in the random oracle model. Provided that the ECCDH problem is difficult to resolve.*

Proof: This result is derived from Lemma 3.1 and Lemma 3.2.

□

Lemma 3.1 *Suppose that an intruder I_1 has the capability to break CL-APSC mechanism with an advantage ζ' by $q_(H_i)$ number of queries for H_i oracle, q_{CU} be the total queries for create user oracle, q_{PPK} for proxy key oracle, q_{SK} for secret key oracle, q_x for replace public key oracle, q_{GD} number of queries for generate delegation oracle, q_{aP} number of queries for proxy key oracle, q_{SC} number of queries for proxy signcryption oracle and q_{USC} number of queries for unsigncryption oracle. There exists a probabilistic polynomial time algorithm, that can solve ECCDHP with an advantage*

$$\zeta' \geq \frac{1}{q_{H_3}}(2\zeta - q_{U_{SC}}(\tfrac{1}{2})^n)$$

$t' < t + O(t)q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}, q_{CU}, q_{PPK}, q_{SK}, q_x, q_{GD}, q_{aP}, q_{SC}, q_{USC}) + T_{SM}(2q_{CU} + q_{PPK} + 2q_{SC} + 6q_{USC})$ *Where T_{SM} is the running time for the computation of scalar multiplication in elliptic curve.*

Proof: Assume C receives an ECCDHP instance $< P, aP, bP >$ and it aims to calculate abP. Initially, C keeps a list $L_1, L_2, L_3, L_4, L_x, L_{PP}, L_P, L_{aP}, L_{uc}, L_{sn}, L_{usc}$ which are initially blank, and these lists are used for finding H_1, H_2, H_3, H_4, public key, PPK, proxy key, user-create, signcryption, and unsigncryption oracles, respectively. C randomly picks any identity $ID_i, i\ in 1, 2, 3, \ldots n$ as a challenge identity. When I_1 makes a request, C will respond as follows. *Phase 1:* In this adaptive way, I_1 issues a series of queries as follows.
$H_1 - Oracle$: I_1 presents H_1- oracle, C yields H_{1i} to I_1 if it exist in the list L_1. Else, C considers two cases to answer H_1 query.
Case 1: If $ID_i = ID^*$, then C outputs h_{ID_i} and stores $< ID_i, P_{pub}, h_{ID_i} >$ in L_1.
Case 2: If $ID_i \neq ID^*$, then C randomly selects $\tau \in Z_q^*$ and adds to L_1 and then returns to I_1.
$H_2 - Oracle$: When I_1 submits H_2- query, C produces h_w to I_1, if there is a tuple $< m_w, Z_w >$ in L_2. Else, C replaces with random $h_w \in \{0, 1\}$ and adds $< m_w, Z_w, h_w >$ to the list L_2.
$H_3 - Oracle$: When I_1 submits H_3- query, C produces h to I_1, if the tuple exists in L_3. Else, C returns random h to I_1 and adds new h to L_3.

$H_4 - Oracle$: When I_1 submits H_4- query, C produces h_3 to I_1, if corresponding tuple in L_4. Else, C returns random z of its choice to I_1 and adds new z to the list I_4.
Create-user oracle: C maintains the list $L_{uc} =< ID_i, X_i, R_i, d_i, r_i >$. If the ID_i exist in L_{uc}, then it proceeds the same. Otherwise, the oracle computes as follows.

- C arbitrarily chooses $x_0, d_0, r_0 \in z_q^*$.
- Compute $X_0 = x_0P, R_0 = r_0P, d_0 = x_0 + s\tau$.

Add the new tuple $< ID_0, X_0, R_0, d_0 >$ to L_uc and $< ID_0, X_0, \tau >$ to L_1 respectively. PPK Oracle: I_1 proceeds the PPK query for the random identity ID_i.

- If $ID_i = ID^*$, then C fails and stops the game.
- If $ID_i \neq ID^*$, C first runs H_1- oracle and create user query and stores in L_{uc} with $< ID_i, R_i, X_i, d_i, r_i >$ and returns the partial private key d_i to I_1.

Private-key-oracle: I_1 proceeds the private key query for the random identity ID_i of its choice.

- If $ID_i = ID^*$, then C terminates the game.
- If $ID_i \neq ID^*$, C first runs create user oracle along with PPK oracle and sends r_i to I_1.

Replace Public key oracle: I_1 wants to change the public key $< X_i, R_i >$ with a random choice of identity ID_i.

- If $ID_i = ID^*$, then C terminates.
- If $ID_i \neq ID^*$, C replaces the public key with his random choice $< X_i', R_i' >$ and update the list L_x with $< ID_i, X_i', R_i' >$.

Proxy Key oracle: I_1 submits the query on $< ID_o, ID_P, m_w >$ then C runs the gen-delegation as given in the scheme and sends a tuple $< ID_o, T_w, m_w >$ to I_1. After verification of T_w, C computes $D_Q = T_w + h_2\rho_\eta$ and sends the tuple $< Z_w, T_w, m_w, D_Q >$ to I_1 and update the list L_{ap}. Proxy Signcryption Oracle: Intruder I_1 submits the query on the signcryption for the tuple $< ID_o, ID_P, ID_R, M_w, m >$. C initially proceeds H_1, H_4, create-user, and PPK oracles, then response to the proxy queries as follows.

- If $ID_R \neq ID^*$, C outputs signcryption text σ to I_1 by executing original proxy signcryption algorithm.
- If $ID_R = ID^*$, then C calculates signcryption text as follows.
  - Choose $a \in_R Z_q^*$ and creates $U = aP, V = abP$.
  - Calculate h by running H_3 oracle for V.
  - Computing $C = h \oplus m, S = a + zD_Q$.

  Add the tuple $< C, ID_0, ID_R, U_\eta, U, V, S >$ to the list L_4.

Unsigncryption query: Taking signcrypted message $\sigma =< M_w, U_\eta, U, V, C, Z_w >$ as inputs. C executes the following steps.

- If $ID_R \neq ID^*$, C output a result to I_1 through the execution of unsigncryption algorithm.
- If $ID_R = ID^*$, C performs as follows.
  - Taking receiver's secret key as an input and performing calculations $V = x_R U$.
  - Running H_3 oracle for V and returns H to calculate $C = h \oplus m$.

– Checking the list L_1, L_2, and L_4 for $< X_0, ID_0, P_{pub} >$, $< M_w, U_w, P_{pub} >$ and $< C \parallel ID_0 \parallel ID_R \parallel U_\eta >$ respectively to get H_10, h_w, h_2, z respectively. If $SP = U + z[U_w + h_w(X_0 + P_{pub}H_{10} + R_0h_0) + h_2$ is valid, then returns m. Else, the game is terminated.

Challenge: Finally, I_1 outputs $< m_0, m_1 > \in \{0,1\}^n$ along with ID_0^*, ID_R^*, M_w^* in phase I. I_1 is restricted from extracting $ID_R's$ private key. Furthermore, ID_R^* must not be an identity for which both the public key has been replaced and the PPK has been retrieved. C performs different queries on H_1, H_4 oracles together with the create user oracle and PPK oracle. Then the response is divided into two cases.

- If $ID_R^* \neq ID_i$, C terminates this game.
- If $ID_R^* = ID_i$, C arbitrarily choices t from $\{0,1\}^n$ and C runs H_3 query to get h^*, to calculate $C^* = h^* \oplus m_t \in \{0,1\}^n, S^* = U + Z^*D_Q^*$ and adds $< m_t \parallel ID_0^* \parallel ID_R^* \parallel Z^* >$ to the list and finally outputs the challenge signcryption test $\sigma^* =< M_w^*, U_n^*, U, V^*, C^*, Z_w^* >$.

Output: Finally, I_1 yields, τ^* as the guess value of τ, $\tau^* = \tau$, C outputs $V = aX_R = ax_RP$ as the solution of ECCDHP. Otherwise, the challenge fails. Phase-II: After obtaining σ^* the intruder I_1, continue to make a series of queries like phase-I, except for the unsigncrypt request on σ^* to find the message m_0 and m_1. Finally, I_1 outputs a bit $\tau^* \in \{0,1\}$ and succeed the game if $\tau^* = \tau$.

□

Lemma 3.2 *Assume that an intruder I_1 is capable to break the proposed CL-APSC mechanism with an advantage ζ' by using q_{H_i} number of queries for H_i oracle, q_{CU} for create user, q_{PPK} for PPK, q_{SK} for secret key, q_{GD} for generate delegation, q_{aP} for proxy key, q_{SC} for proxy signcryption and q_{USC} for unsigncryption oracles. There exists a probabilistic polynomial time algorithm, that can be able to find the solution for ECCDHP with an advantage*

$$\zeta' \geq \frac{1}{q_{H_4}}(2\zeta - q_{U_{SC}})(1/2)^n$$

$\tau' < \tau + O(\tau)(q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + q_{CU} + q_{PPK} + q_{SK} + q_{GD} + q_{aP} + q_{SC} + q_{USC}) + T_{SM}(2q_{CU} + 2q_{SC} + 6q_{USC})$

Proof: The proof of this game is equivalent to Game 1, so we will not go through it again. □

Unforgeability:

Theorem 3.2 *Our certificateless anonymous proxy signcryption mechanism is EUF-CMA-secure against adaptive chosen signcryption text attacks if no intruder with a non-negligible advantage can succeed in the succeeding games within polynomial time.*

Proof:

Like the proofs of unforgeability in [12], the unforgeability of our proposed CL-APSC scheme is also proved through the following lemmas 3.3 and 3.4. □

Lemma 3.3 *Assume that an intruder I_1 has the skill to break the proposed CL-APSC mechanism with an advantage $\zeta^p rime$ by using q_{H_i} number of queries for H_i oracle, q_{CU} for create user, q_{PPK} for PPK, q_{SK} for secret key, q_x for replace public key, q_{GD} for generate delegation, q_{aP} for proxy key, q_{SC} for proxy signcryption and q_{USC} for unsigncryption oracles. There exists a probabilistic polynomial time algorithm, that can solve ECDLP with an advantage*

$$\zeta' \geq \frac{1}{q_{H_3}q_{H_4}}(\zeta - (1 + q_{H_4})(1/2)^n)$$

$\tau' < \tau + O(\tau)(q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + q_{PPK} + q_{SK} + q_{CU} + q_x + q_{GD} + q_{aP} + q_{SC} + q_{USC}) + T_{SM}(2q_{CU} + q_{PPK} + 2q_{SC})$.

Proof:

Assume that an algorithm accepts an ECDLP with a provided instance the problem is to compute x becomes hard. To address this problem C acts as a challenger and runs the algorithm uses I_1 as a subroutine. In this algorithm, C Answering the queries honestly to I_1 and create the required parameters by C. We assume $P_{pub} = ap$. When I_1 makes a query, the challenger C generates and maintains a list $L_i, i = \{1, 2, 3, 4\}, L_x, L_{PPK}, L_P, L_{aP}, L_{uc}, L_{sn}, L_{usc}$ which are initially empty. C gives the responses to intruder queries.

Forgery: Finally, a signcrypted text $\sigma^* =< C^*, ID_O^*, ID_R^*, U_m^*, U^*, V^*, S^*, M_W^* >$ on the message m and warrant M_W^* given as a output. Without using a proxy signcryption oracle, unlike in conventional scenarios, ID_p^* is acts as a forged identity of the proxy signer on behalf of ID_A^*. I_1 takes the signcrypted text σ^* as an input of unsigncryption and request several queries to C other than PPK oracle and replace public key and private key oracle in PPT. If unsigncryption is verified, then I_1 achieves in this game otherwise I_1 flops.

□

Lemma 3.4 *Assume that an intruder I_2 has the capability to break the proposed CL-APSC mechanism with an advantage ζ' by using q_{H_i} number of queries for H_i oracle, q_{CU} for create user, q_{PPK} for PPK, q_{SK} for secret key, q_{GD} for generate delegation, q_{aP} for proxy key, q_{SC} for proxy signcryption and q_{USC} for unsigncryption oracles. There exists a PPT algorithm, that can solve ECDLP with an advantage*

$$\zeta' \geq \frac{1}{q_{H_3} q_{H_4}}(\zeta - (1 + q_{USC})(1/2)^n)$$

$$\tau' < \tau + O(\tau)(q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + q_{CU} + q_{PPK} + q_{SK} + q_{GD} + q_{aP} + q_{SC} + q_{USC}) + T_{SM}(2q_{CU} + 2q_{SC}).$$

Proof: The proof of this game is equivalent to Game 1, so we will not go through it again. □

Theorem 3.3 *Our Proposed scheme provides anonimity to the proxy signer from the reciever.*

Proof:

Since η is randomly selected from Z_q^* and evaluating η involves solving the ECDL problem, finding η from $U_\eta = \eta P$ is not achievable. As far as we are aware, there is no probabilistic algorithm for solving ECDLP. U_η is hidden from any intruder because it was transmitted through a secure anonymous channel. As a result, no attacker could determine the proxy sender's identity by using the formula $H_Q = H_P + U_\eta$.

□

3.3. Efficiency Analysis

In this instance, we evaluate our CL-APSC scheme's effectiveness in comparison to previous certificateless proxy signcryption schemes [9,10,11,12,13]. Analysis shows that the proposed scheme executes improved efficiency over existing schemes with respect to computation and operation time. The computational and communication costs of the proposed scheme are evaluated with reference to the experimental results reported in [14,15,16]. The corresponding results are summarized in Table 1. The approaches employed to perform these operations [14,15,16], as well as the conversions presented in Table 1, are based on elliptic curve computations over the curve $\frac{E}{F_P} : y^2 = x^3 + ax + b(modp)$, defined on the field Z_q^*. In this setting, the elements of the elliptic curve group G have an approximate length of 320 bits, where $a, b \in Z_q^*$ and the size of q is approximately 160 bits. A comparative analysis of the computational performance of the proposed scheme and existing schemes is presented in Table 2, highlighting efficiency improvements in terms of operational cost. Security analysis of proposed scheme is compared with related articles in table 3, in terms of confidentiality, unforgeability, proxy Anoninity, public verifiability.

Table 1: Notation and the Execution Time of Various Cryptographic Techniques

Notations	Descriptions and execution time
T_{ML}	The computational running time for execution of modular multiplication
T_P	The computational running time for one pairing $87T_{ML}$
T_H	The computational running time for one point hash function $\approx 29T_{ML}$
T_{ME}	The computational running time for one modular exponentiation $\approx 240T_{ML}$
T_{PE}	The computational running time for Pairing based exponentiation $\approx 43.5T_{ML}$
T_{SM}	The computational running time for one scalar multiplication $\approx 29T_{ML}$
T_{PA}	The computational running time for one point addition $\approx 0.12T_{ML}$
T_{MI}	The computational running time for one modular Inversion $\approx 11.6T_{ML}$

Table 2: Computational Efficiency

Scheme	Proxy Signcryption	Proxy Unsigncryption	Total	Total cost in T_{ML}
[9]	$5T_{SM} + 3T_H + T_{PA}$	$9T_{SM} + 8T_H + 7T_{PA}$	$11T_{SM} + 11T_H + 8T_{PA}$	638.96
[10]	$3T_{SM} + T_H + T_{PA}$	$6T_{SM} + 6T_H + 7T_{PA}$	$9T_{SM} + 7T_H + 8T_{PA}$	464.96
[11]	$3T_{SM} + 2T_H + T_{PA}$	$6T_{SM} + 6T_H + 7T_{PA}$	$9T_{SM} + 8T_H + 7T_{PA}$	498.84
[12]	$2T_{ME} + T_{PE} + 3T_H + 3T_{ML} + T_P$	$T_{ME} + 5T_H + 2T_{ML} + 5T_P$	$3T_{ME} + T_{PE} + 8T_H + 5T_{ML} + 6T_P$	1522.5
[13]	$2T_{SM} + 3T_H + 2T_{PA} + T_{MI}$	$3T_{SM} + 5T_H + 5T_{PA}$	$5T_{SM} + 8T_H + 7T_{PA} + T_{MI}$	389.44
OURS	$2T_{SM} + 2T_H$	$7T_{SM} + 6T_H + 5T_{PA}$	$9T_{SM} + 8T_H + 7T_{PA}$	493.84

Table 3: Security Concepts

Scheme	Confidentiality	Unforgeability	Proxy Anonimity	Public Verifiable	Certificate-Free Protocol	Key Escrow-Free Protocol
[9]	√	×	×	×	√	√
[10]	√	×	×	×	√	√
[11]	√	√	×	×	√	√
[12]	√	√	×	×	√	√
[13]	√	√	×	×	√	√
[OURS]	√	√	√	√	√	√

We can observe that our CL-APSC has less computational cost than schemes [9,11,12] in both signcryption and unsigncryption operations. Our scheme has a computational cost that is slightly higher than that of schemes [10] and [13]. However, the scheme in [10] is not secure, and [13] does not offer proxy signer anonymity. Examining it from a systemic perspective, the proposed approach is more efficient compared to other schemes.

4. Conclusion

This paper presents a novel certificate-less anonymous proxy signcryption (CL-APSC) scheme intended to address the challenges of achieving anonymity, unforgeability, and confidentiality in secure communication protocols. The proposed scheme ensures the anonymity of the proxy signer from other users within the network while maintaining strong security guarantees. Formal security analysis reveals that the proposed scheme achieves confidentiality and unforgeability under the certificateless cryptographic setting. The proposed construction is based on certificateless cryptography, which eliminates the traditional certificates, thereby reducing the risks associated with key escrow and certificate management.

In addition, performance evaluations reveal that the proposed scheme attains computational efficiency with minimal overhead. In summary, the proposed CL-APSC scheme offers a secure and efficient solution for ensuring anonymity, confidentiality, and unforgeability in communication protocols, effectively meeting the evolving security demands of modern information systems.

References

1. Zheng. Y.,*Digital signcryption or how to achieve cost(signature encryption*, in Proceedings of the cost(signature) + cost(encryption), in Advances in Cryptology - CRYPTO 97, 165-179, (1997).
2. Saraswat, Vishal, Sahu, Rajeev Anand and Awasthi, Amit. K.,*A secure anonymous proxy signcryption scheme*, Journal of Mathematical Cryptology, 11(2), 63-84, (2017).
3. Padmalaya Nayak, Swapna. G., *Security issues in IoT applications using certificateless aggregate signcryption schemes: An overview*, Internet of Things, 21, (2023).
4. Mambo. M, Usuda. K and Okamoto. E., *Proxy signatures: Delegation of the power to sign messages*, IEICE Trans. Fundam. Electron. Comm. Comput. Sci., 79(9), 1338–1354, (1996).
5. Gamage. C, Leiwo. J, and Zheng. Y., *An efficient scheme for secure mesasage transmission using proxy-signcryption*, Proc. 22nd Australasim Comput. Sci. Conf., Auckland, New Zealand: Springer-Verlag, 420–431, (1999).
6. Baek. J, Newmarch. J, Safavi-Naini. R and Susilo. W., *A Survey of Identity-Based Cryptography*, Proc. of the 10th Annual Conference for Australian Unix User's Group, 95-102, (2004).
7. Malone. J Lee., *Identity based Signcryption*, Cryptology e-Print Archive, Report 2002/098, (2002).
8. Barbosa. M and Farshim. P., *Certificateless signcryption*, Proc. ACM Symp. Inf., Comput. Commun., New York, NY, USA, 369–372, (2008).
9. Yanfeng. Q, Chunming. T, Yu L, Maozhi X, Baoan. G., *Certificateless proxy identity-based signcryption scheme without bilinear pairings*, China Commun 10(11), 37–41, (2013).
10. Bhatia. T and Verma. A. K.,*Cryptanalysis and improvement of certifcateless proxy signcryption scheme for e-prescription system in mobile cloud computing*, Annals of Telecommunications-Annales des Telecommunications, 72, (9-10), 563–576, (2017).
11. Li, Li, Zhou. S, Choo K. K. R, Li, X, He, D., *An Efficient and Provably Secure Certificateless Proxy Signcryption Scheme for Electronic Prescription System*, Secur. Commun. Netw. 95, 541-549, (2018).
12. H. Yu and Wang. Z.,*Construction of Certificateless Proxy Signcryption Scheme From CMGs*, IEEE Access, 7, 141910-141919, (2019).
13. Khan, M.A, Alhakami. H, Ullah. I, Alhakami, W, Mohsan. S.A.H, Tariq. U, Innab. N. A ., *Resource Friendly Certificateless Proxy Signcryption Scheme for Drones in Networks beyond 5G*, Drones 7, 321, (2023).
14. Ren. K, Lou. W, Zeng. K, Moran. P., *On Broadcast Authentication in Wireless Sensor Networks*, IEEE Trans. Wireless Commun. 6, 4136–4144, (2007).
15. Cao, Kou. W, Du. X., *A Pairing-free Identity-based Authenticated Key Agreement Protocol with Minimal Message Exchanges*, Inform. Sci. 180, 2895–2903, (2010).
16. Tan. S. Y, S.-H. Heng, B.-M. Goi., *Java Implementation for Pairing-Based Cryptosystems*, Computational Science and Its Applications – ICCSA, 188–198, (2010).

G. Swapna,
Department of Mathematics,
VNR Vignana Jyothi Institute of Engineering and Technology,
india.
E-mail address: swapnacrypto@gmail.com

and

G. Naga Malleswari,
Department of Mathematics,
Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology,
india.
E-mail address: malleswari.gn@gmail.com

2020 *Mathematics Subject Classification:* 35B40, 35L70.

and

T. Kusuma,
Department of Mathematics,
Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology,
india.
E-mail address: kusumatummala9@gmail.com

and

Gowri Thumbur,
Department of ECE,
Gitam University,
India.
E-mail address: gthumbur@gitam.edu