A Novel Pairing-Free ECC-Based Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Secure Cloud Storage

Shivangi Shukla[a] and Sankita J. Patel

Department of Computer Science and Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat 395007, India

Keywords: Pairing-Free, Elliptic Curve Cryptography, Access Control, CP-ABPRE, Cloud Storage.

Abstract: Proxy re-encryption (PRE) is a cryptographic primitive enabling data owner to delegate ciphertext access rights without leaking underlying plaintext to honest-but-curious cloud servers. The delegation of ciphertext access rights enhances the efficiency of outsourced data on cloud servers. Ciphertext-policy attribute-based proxy re-encryption (CP-ABPRE) employs PRE in attribute-based encryption to enable ciphertext transformation from specified access policy to new access policy without leaking underlying plaintext. However, current state-of-the-art schemes incorporate expensive bilinear pairing operations to transform ciphertext access policy. The escalating adoption of cloud computing in real-time applications demands a pairing-free CP-ABPRE mechanism for resource-limited users in the network. The agenda of this paper, for the first time, is to design a novel pairing-free elliptic curve cryptography (ECC) based ciphertext-policy attribute-based proxy re-encryption, abbreviated as ECC-CP-ABPRE scheme. It incorporates linear secret sharing scheme (LSSS) for the expressiveness of access policies. To reduce overall communication and computational overheads, ECC-CP-ABPRE scheme replaces expensive bilinear pairing operations with scalar multiplication on elliptic curve. The security analysis illustrates that ECC-CP-ABPRE scheme is secure under collusion attack and ensures data confidentiality. Furthermore, the performance evaluation demonstrates that ECC-CP-ABPRE scheme incurs significant reduction in computational and communication overheads than existing CP-ABPRE schemes.

1 INTRODUCTION

In the modern era of Internet, cloud computing has become mainstream solution for outsourcing data that can be accessed anytime and anywhere (Wang et al., 2023). Although cloud computing offers reliable and cost-effective data storage, users are reluctant to cloud services as they lose physical control over their outsourced data. Furthermore, cloud service providers are untrusted third parties that can access or disclose outsourced data to unauthorized organizations for financial benefits (Dhakad and Kar, 2022). Thus, data confidentiality and access control mechanisms are pivotal security requirements for cloud storage.

CP-ABE associates user's secret key with attributes, and ciphertext incorporates access policy defined over attributes of the system. The ciphertext can be decrypted only if user's attribute secret key satisfies access policy in the ciphertext (Bethencourt et al., 2007). However, CP-ABE lacks the provision of encrypted data sharing in collaborated scenarios. It is not desirable in practical scenarios that require frequent ciphertext-policy updates. Additionally, heavy computational overheads incurred in decrypting ciphertext and encrypting plaintext under new access policy are inconvenient for resource-limited data users in the network. Thus, PRE technique is integrated with CP-ABE, known as CP-ABPRE, to enable efficient sharing of encrypted data stored on clouds. (Liang et al., 2009) first introduced CP-ABPRE to enable ciphertext transformation from specific access policy to new access policy. Assume a delegator (i.e. original data user) wants to delegate access rights of CT_1 under access policy AP_1 to delegatee (i.e. shared data user) that satisfies access policy AP_2. The original data user generates re-encryption key for ciphertext transformation of CT_1 under AP_1 to CT_2 under AP_2 and transmits it to proxy server. The *semi-trusted* proxy server transforms CT_1 under AP_1 to CT_2 under AP_2 without gaining underlying plaintext or secret keys of participating data users.

The existing CP-ABPRE schemes are based on expensive bilinear pairings that impede in escalating

[a] https://orcid.org/0000-0001-8945-2721

Shukla, S. and Patel, S. J.
A Novel Pairing-Free ECC-Based Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Secure Cloud Storage.
DOI: 10.5220/0013138600003899
In *Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP 2025) - Volume 1*, pages 225-233
ISBN: 978-989-758-735-1; ISSN: 2184-4356

adoption of cloud computing in real-time applications of smart cities incorporating resource-constrained end-users (Rezaeibagha et al., 2021). This motivates us to design a lightweight CP-ABPRE scheme that efficiently shares encrypted data on clouds. The salient contributions are:

- To enhance the efficiency of CP-ABPRE schemes, we propose a novel pairing-free ECC-CP-ABPRE that eliminates expensive bilinear pairing with scalar multiplications on elliptic curves.
- Security analysis illustrates that ECC-CP-ABPRE is resilient against collusion attack, offers specificity of re-encryption keys and data confidentiality. Performance evaluation illustrate its efficiency.

2 RELATED WORK

(Liang et al., 2009) designed CP-ABPRE with access policy based on *AND* gates of positive and negative attributes. (Luo et al., 2010) extended it for *AND* gates access policy based on negative, multi-value attributes and wildcards, and (Liang et al., 2015) designed adaptively chosen-ciphertext secure CP-ABPRE for arbitrary network settings. (Yang et al., 2016) designed CP-ABPRE that enables user revocation by facilitating cloud server with re-encryption keys from each data owner to data user however, it necessitates PRE for each data access request. (Deng et al., 2020) proposed CP-ABPRE that enables partial decryption of re-encrypted ciphertext by proxy server to minimize computational overhead. (Ge et al., 2021) designed CP-ABPRE to support verifiability wherein shared data user verifies correctness of re-encrypted data and proxy server proves its fairness if the re-encryption process is performed honestly. (Zhou et al., 2021) designed CP-ABPRE with accountability to trace malicious users. However, it lacks revocation mechanism for such malicious data users to prevent any further unauthorized access. (Chen et al., 2022) proposed CP-ABPRE for encrypted sharing of medical data that facilitates user revocation mechanism in healthcare centres. However, existing CP-ABPRE schemes incorporate complex bilinear pairing to update or modify access policy of ciphertext. It is inefficient for cloud-based applications in real-world resource-constrained environments such as smart homes etc. Thus, designing of lightweight CP-ABPRE scheme that supports access rights delegation of ciphertext is an interesting open problem. In this direction, we aim to design a lightweight ECC-based CP-ABPRE scheme while retaining security of the system.

3 PRELIMINARIES

The preliminaries are as follows:

1. **Access Structure:** Assume $\mathcal{P} = \{P_1, P_2 \dots, P_n\}$ denotes set of parties. $\mathbb{AS} \subseteq 2^{\mathcal{P}}$ is monotone if $\forall B, C$: if $B \in \mathbb{AS}$ and $B \subseteq C$ then $C \in \mathbb{AS}$. The access structure (respectively, monotonic access structure) is a collection (respectively, monotone collection) $\mathbb{AS}$ of non-empty subsets of $\mathcal{P}$ i.e., $\mathbb{AS} \subseteq 2^{\mathcal{P}} \backslash \emptyset$. The sets in $\mathbb{AS}$ are authorized sets, else unauthorized sets. $\mathbb{AS}$ incorporates all authorized set of attributes.
2. **LSSS:** The secret sharing scheme $\prod$ based on set of parties is called as LSSS if: (a) Shares of all participating entities form a vector over $\mathbb{Z}_q$. (b) There exists share-generating matrix $\mathbb{A}$ of $l \times n$ and function ρ where $\rho(i) \in \mathcal{P}$, $i \in \{1, 2, \dots, l\}$. Assume column vector $\vec{v} = (s, v_2, v_3, \dots, v_n)$ where $v_2, \dots, v_n \in \mathbb{Z}_q$ are randomly chosen and $s \in \mathbb{Z}_q$ is secret to be shared, then $\mathbb{A} \cdot v$ denotes vector of l shares of s related to $\prod$, and $\mathbb{A}_i \cdot \vec{v}$ belongs to $\rho(i)$. Note that no such constants exists for any unauthorized attribute set. *Linear construction* property of LSSS: Let $\prod$ be LSSS for $\mathbb{AS}$ and $S \models (\mathbb{A}, \rho)$ denotes that S is authorized attribute set hence, it satisfies access structure $\mathbb{AS}$ and assume $I = \{i : \rho(i) \in S\} \subset \{1, 2 \dots, l\}$. Suppose for $\prod$, λ_i are valid shares of s then, constants $\{c_i \in \mathbb{Z}_q\}_{i \in I}$ exists such that $\sum_{i \in I} c_i \cdot \lambda_i = s$. These $\{c_i\}$ can be found in polynomial time with knowledge of $\mathbb{A}$ and I (Beimel, 1996). Any unauthorized set with corresponding matrix includes no target vector $(1, 0, \dots, 0)$ in row span of I. Also, it will incorporate a vector $\vec{d}$ such that $d \cdot (1, 0, \dots, 0) = -1$ and $\vec{d} \cdot \mathbb{A}_i = 0$ for all $i \in I$.
3. **ECC:** Elliptic curve $E_p(a, b)$ is defined on finite field F_p as $y^2 = x^3 + ax + b \pmod{p}$ and $4a^3 + 27b^2 \neq 0$ where p is large prime number and a, b are elements of F_p (Miller, 1985). Given $Q = k.G$ where G is group generator of order q, it is hard to calculate k in polynomial time. ECDLP is more difficult to solve than integer factorization hence, requiring smaller key size than RSA. Alice and Bob performs following: (a) Key Generation: Alice selects private key $s_a \in \mathbb{Z}_q$ and computes public $P_a = s_a.G$ Bob selects private $s_b \in \mathbb{Z}_q$ and computes public $P_b = s_b.G$ (b) Encryption: Alice maps plaintext m to point M, selects random integer $k \in \mathbb{Z}_q$, computes $CT_1 = k.G$, $CT_2 = M + k.P_b$ and transmits $\{CT_1, CT_2\}$ to Bob. (c) Decryption: Bob utilizes his s_b to compute $CT_2 - s_b.CT_1 = M + k.P_b - s_b k.G = M$. Bob maps M to obtain m.
4. **DDH Assumption:** Assume P is cyclic group of prime order q with G as generator, and b, c are random integers in $\mathbb{Z}_q$. Given (G, bG, cG, bcG) to $\mathcal{A}$, it is difficult to distinguish between $bcG \in P$ and ran-

dom $R \in P$. The algorithm $\mathcal{B}$'s advantage in solving DDH assumption is ε if $Prob[\mathcal{B}(G, bG, cG, Z = bcG) = 0] - Prob[\mathcal{B}(G, bG, cG, Z = R) = 0]| \geq \varepsilon$

Definition 1. *The DDH assumption holds if the advantage of polynomial time algorithm in solving DDH problem is at most negligible.*

4 SYSTEM OVERVIEW

This section elaborates system architecture, threat model, and formal structure of ECC-CP-ABPRE.

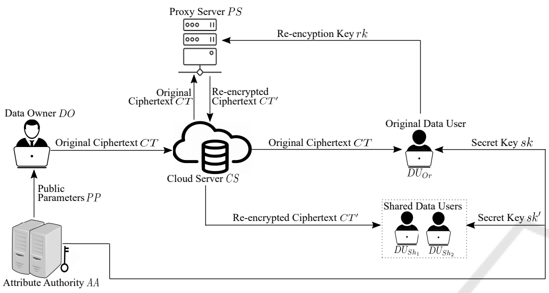


Figure 1: System Architecture of ECC-CP-ABPRE.

Table 1: Notations Summary.

Notation	Description
AA, DO	Attribute authority, Data owner
CS, PS	Cloud server, Proxy server
DU_{Or}, DU_{Sh}	Original data user, Shared data user
$\mathcal{A}$, C	Adversary, Challenger
$\mathbb{A}$	Share-generating matrix
ρ	Associate rows of $\mathbb{A}$ to attribute
U, S	Universal attribute set, DU's attribute set
PK_i	Public key of attribute i
msk, PP	Master secret key, public parameters
sk, rk	Secret key of DU, Re-encryption key
CT, CT'	Original, Re-encrypted ciphertext
$\vec{\lambda}$, $\vec{w}$	Random vectors

4.1 System Architecture

The entities incorporated are: *AA* is *trusted third party* responsible for system initialization, generates *PP* and distributes *sk* to data users. Fig. 1 and Table 1 depict and describe, respectively, entities and notations of ECC-CP-ABPRE. *DO* formulates access policy over system attributes and encrypts ans stores his data as *CT* on *CS*. *CS* and *PS* are *honest-but-curious entities* that execute all authorized requests but attempt to retrieve some information from results. The *CS* stores ciphertexts generated by participating entities, and *PS* generates CT' using *CT* and *rk* of *DU*. *DU* request access to ciphertext stored on *CS* and can successfully decrypt it if corresponding attributes satisfy the underlying access policy. *DU* are categorized as delegator i.e. DU_{Or} and delegatee i.e. DU_{Sh}. DU_{Or} that satisfies the access policies of *CT*, generates *rk* to update the ciphertext-policy. CT' can be decrypted by DU_{Sh} with their sk' that satisfies the updated ciphertext-policy.

4.2 Threat Model and Security Requirements

An unauthorized user without valid secret key is an adversary $\mathcal{A}$ that attempts to gain underlying plaintext of encrypted data. $\mathcal{A}$ attempts following attacks: firstly, $\mathcal{A}$ colludes with *CS* and *PS* to download and decrypt either *CT* or CT' to gain underlying plaintext. Secondly, *PS* leverages *rk* to update access policy of unspecified ciphertext (i.e. *rk* that does not satisfies ciphertext-policy). Following are security goals:

Data confidentiality: It ensures that the outsourced encrypted data should be decrypted by authorized entities with valid *sk* that satisfy ciphertext-policy. Similarly, CT' should be decrypted by authorized users that satisfy the updated access policy in CT'. Additionally, the underlying plaintext of both *CT* and CT' should be inaccessible to both *CS* and *PS*.

Specificity of re-encryption keys: The *rk* issued by DU_{Or} should re-encrypt the specified ciphertext (i.e. *rk* generated for policy update of specified ciphertext). This *rk* should neither re-encrypt unspecified ciphertexts nor deduce any other valid re-encryption keys to re-encrypt other ciphertexts in *CS*.

4.3 Formal Structure

ECC-CP-ABPRE incorporates following algorithms:

- $Setup(\lambda, U) \rightarrow (PP, msk)$: *AA* inputs *U* and security parameter λ to generate *msk* and *PP*.
- $KeyGenr(S, msk) \rightarrow sk$: *AA* inputs *msk* and user's attribute set *S* and outputs secret key *sk* for *S*.
- $Enc(m, (\mathbb{A}, \rho)) \rightarrow CT$: *DO* generates *CT* by encrypting *m* under $(\mathbb{A}, \rho)$.
- $Dec_O(CT, sk) \rightarrow m/\perp$: DU_{Or} with *sk* executes this algorithm to decrypt *CT*. If attribute set $S \models (\mathbb{A}, \rho)$ then *m* is generated as output, otherwise symbol $\perp$ that indicates either *CT* is invalid or $S \not\models (\mathbb{A}, \rho)$.
- $ReEncKeyGenr(sk, (\mathbb{A}', \rho'), CT) \rightarrow rk$: DU_{Or} with *sk* of *S* where $S \models (\mathbb{A}, \rho)$, outputs *rk*. It inputs updated $(\mathbb{A}', \rho')$, *CT* under $(\mathbb{A}, \rho)$, *sk*, and outputs *rk*.
- $ReEncr(CT, rk) \rightarrow CT'$: *PS* inputs *CT*, *rk* to generate CT' which is outsourced on *CS*.
- $Dec_R(CT', sk') \rightarrow m/\perp$: DU_{Sh} with sk' of S' decrypts CT'. It inputs sk', CT' under $(\mathbb{A}', \rho')$, and outputs *m* if $S' \models (\mathbb{A}', \rho')$, else $\perp$ for $S' \not\models (\mathbb{A}', \rho')$ or invalid CT'.

4.4 Security Model

ECC-CP-ABPRE adopts selective model where adversary commits challenge policy before security game similar to (Waters, 2011; Ge et al., 2021).

Semantic Security: Sem-O and Sem-R security game illustrates semantic security of original and re-encrypted ciphertext, respectively.

Game Sem-O: ECC-CP-ABPRE is original ciphertext semantic secure if $\mathcal{A}$'s advantage is negligible.

Initialization: $\mathcal{A}$ selects challenge $(\mathbb{A}'', \rho'')$.

Setup: $\mathcal{C}$ outputs $\{PP, msk\}$ and provides PP to $\mathcal{A}$.

Query Phase 1: $\mathcal{A}$ queries: (1) $O_{sk}(S)$: $\mathcal{A}$ queries sk with S, $\mathcal{C}$ executes $KeyGenr(S, msk)$ and outputs sk to $\mathcal{A}$. (2) $O_{rk}(S, (\mathbb{A}', \rho'), CT)$: $\mathcal{A}$ queries rk on $(S, (\mathbb{A}', \rho'))$ wherein $S \not\models (\mathbb{A}', \rho')$. $\mathcal{C}$ computes $sk = KeyGenr(S, msk)$, $rk = ReEncKeyGenr(sk, (\mathbb{A}', \rho'), CT)$, and sends rk to $\mathcal{A}$. (3) $O_{re}(S, (\mathbb{A}', \rho'), CT)$: $\mathcal{A}$ queries re-encrypted ciphertext on $(S, (\mathbb{A}', \rho'), CT)$ where $\mathcal{C}$ computes $sk = KeyGenr(S, msk)$, $rk = ReEncKeyGenr(sk, (\mathbb{A}', \rho'), CT)$, $CT' = ReEncr(CT, rk)$ and transmits CT' to $\mathcal{A}$. $\mathcal{A}$ cannot execute: (1)$O_{sk}(S)$ if $S \models (\mathbb{A}'', \rho'')$ (2)$O_{rk}(S, (\mathbb{A}', \rho'), CT)$ if $S \models (\mathbb{A}'', \rho'')$ and $\mathcal{A}$ queried $O_{sk}(S')$ where $S' \models (\mathbb{A}', \rho')$

Challenge: $\mathcal{A}$ transmits (m_0, m_1) of same length to $\mathcal{C}$. $\mathcal{C}$ calculates challenge $CT'' = Enc(m_\beta, (\mathbb{A}'', \rho''))$ where $\beta \in \{0, 1\}$ and sends CT'' to $\mathcal{A}$.

Query Phase 2: $\mathcal{A}$ queries similar to Query Phase 1 except: (1) $O_{sk}(S)$ if $S \models (\mathbb{A}'', \rho'')$ (2) $O_{rk}(S, (\mathbb{A}', \rho'), CT)$ and $O_{sk}(S')$ if $S \models (\mathbb{A}'', \rho'')$ and $S' \models (\mathbb{A}', \rho')$ (3) $O_{re}(S, (\mathbb{A}', \rho'), CT'')$ and $O_{sk}(S')$ if $S \models (\mathbb{A}'', \rho'')$ and $S' \models (\mathbb{A}', \rho')$

Guess: $\mathcal{A}$ outputs guess bit β'. $\mathcal{A}$ wins if $\beta' = \beta$. $\mathcal{A}$'s advantage to win this game is defined as $Adv^{Sem-O}(\lambda) = |Prob[\beta' = \beta] - 1/2$

Game Sem-R: ECC-CP-ABPRE is re-encrypted ciphertext semantic secure if the advantage of $\mathcal{A}$ is negligible in the below game.

Initialization: $\mathcal{A}$ selects $(\mathbb{A}'', \rho'')$.

Setup: $\mathcal{C}$ generates $\{PP, msk\}$ and provides PP to $\mathcal{A}$.

Query Phase 1: $\mathcal{A}$ queries: (1)$O_{sk}(S)$: $\mathcal{A}$ queries sk with S, $\mathcal{C}$ executes $sk = KeyGenr(S, msk)$ and transmits sk to $\mathcal{A}$. (2) $O_{rk}(S, (\mathbb{A}', \rho'), CT)$: $\mathcal{A}$ queries rk on $(S, (\mathbb{A}', \rho'))$ wherein $S \not\models (\mathbb{A}', \rho')$. $\mathcal{C}$ computes $sk = KeyGenr(S, msk)$ and $rk = ReEncKeyGenr(sk, (\mathbb{A}', \rho'), CT)$, and returns rk to $\mathcal{A}$. $\mathcal{A}$ cannot execute $O_{sk}(S)$ if $S \models (\mathbb{A}'', \rho'')$.

Challenge: $\mathcal{A}$ transmits (m_0, m_1) of same length to $\mathcal{C}$. $\mathcal{C}$ computes challenge CT''' as $CT = Enc(m_\beta, (\mathbb{A}, \rho))$, $rk = ReEncKeyGenr(S, (\mathbb{A}'', \rho''), CT)$, $CT''' = ReEncr(CT, rk)$ where $\beta \in \{0, 1\}$ and $S \models (\mathbb{A}, \rho)$. $\mathcal{C}$ returns CT''' to $\mathcal{A}$.

Query Phase 2: $\mathcal{A}$ queries similar to Query Phase 1 except $O_{sk}(S)$ if $S \models (\mathbb{A}'', \rho'')$.

Guess: $\mathcal{A}$ outputs guess bit β'.

There is no constraint on re-encryption query i.e. $\mathcal{A}$ generates appropriate rk followed by re-encryption query hence, omitted in this game. $\mathcal{A}$'s advantage to win this game is $Adv^{Sem-R}(\lambda) = |Prob[\beta' = \beta] - 1/2|$

Definition 2. *The ECC-CP-ABPRE is semantic secure if both Sem-O and Sem-R are secure.*

5 ECC-CP-ABPRE CONSTRUCTION

ECC-CP-ABPRE incorporates following algorithms:

***Setup*(λ, U):** AA generates $(p, E_p(a, b), G)$ with security parameter λ as input. Assume hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ for one-way mapping of ID_U to elements in $\mathbb{Z}_q^*$. AA inputs $U = \{A_1, A_2, \ldots, A_n\}$ and for each $A_i \in U$, AA selects random $a_i \in \mathbb{Z}_q$ as secret key and computes $PK_i = a_i.G$ as public key for A_i. AA secretly stores $msk = \{a_1, a_2 \ldots, a_n\}$ and publicize $PP = \{U, PK_1, PK_2, \ldots PK_n, G, H\}$ as PP.

***KeyGenr*(S, msk):** AA inputs S and msk, and assigns unique ID_U to DU and generates random $r_i \in \mathbb{Z}_q, \forall i \in S$ to compute $sk_{i,ID_U} = a_i + H(ID_U).r_i$. Finally, AA securely transmits $sk = \{sk_{i,ID_U}, \forall i \in S\}$ to DU.

***Enc*$(m, (\mathbb{A}, \rho))$:** DO performs following:

1.DO maps m to point M on $E_p(a, b)$, selects random $s \in \mathbb{Z}_q$ to calculate $CT_0 = M + s.G$

2.DO selects $\vec{\lambda} = (s, v_2, v_3, \ldots, v_n) \in \mathbb{Z}_q^n$ and $\vec{w} = (0, \zeta_2, \zeta_3, \ldots, \zeta_n) \in \mathbb{Z}_q^n$. Assume $\mathbb{A}_x$ denotes x^{th} row of $\mathbb{A}$ where $x = [1, l]$, DO calculates $\lambda_x = \vec{\lambda} \cdot \mathbb{A}_x^\intercal$ and $w_x = \vec{w} \cdot \mathbb{A}_x^\intercal$ where $\mathbb{A}_x^\intercal$ denotes transpose of $\mathbb{A}_x$. Further, for all $x = [1, l]$, DO computes: $CT_{1,x} = \lambda_x.G + w_x.PK_{\rho(x)}$ and $CT_{2,x} = w_x.G$ DO transmits $CT = ((\mathbb{A}, \rho), CT_0, \{CT_{1,x}, CT_{2,x}\}_{x=[1,l]})$ to CS.

***Dec$_O$*(CT, sk):** DU_{Or} with S and $sk = \{sk_{x,ID_U}, \forall x \in S\}$ decrypts CT as follows:

1.DU_{Or} generates $X = \{x | \rho(x) \in S\}$. If $S \models (\mathbb{A}, \rho)$ then, there exists $\{c_x \in \mathbb{Z}_q\}_{x \in X}$ such that $\sum c_x.\mathbb{A}_x = (1, 0, \ldots, 0)$. DU_{Or} computes:

$$\begin{aligned} D_x &= \sum CT_{1,x} - \sum CT_{2,x}.sk_{\rho(x),ID_U} \\ &= \sum \lambda_x.G + w_x.PK_{\rho(x)} - \left(\sum w_x a_x.G + w_x r_{\rho(x)} H(ID_U).G\right) \\ &= \sum \lambda_x.G - w_x r_{\rho(x)} H(ID_U).G \end{aligned} \tag{1}$$

2.DU_{Or} utilizes constant set $\{c_x \in \mathbb{Z}_q\}_{x \in X}$ as $\sum_{x \in X} c_x \lambda_x = s$ and $\sum_{x \in X} c_x w_x = 0$. DU_{Or} computes:

$$c_x.D_x = \sum c_x \lambda_x.G - c_x w_x r_{\rho(x)} H(ID_U).G = s.G \tag{2}$$

3.DU_{Or} calculates point M as $M = CT_0 - s.G$. Finally, DU_{Or} generates m by mapping point M on $E_p(a, b)$.

$\boldsymbol{ReEncKeyGenr(sk,(\mathbb{A}',\rho'),CT)}$: DU_{Or} with sk of S considers CT under $(\mathbb{A},\rho)$ and new $(\mathbb{A}',\rho')$ as input and computes rk. DU_{Or} can generate rk only if $S \models (\mathbb{A},\rho)$, else symbol $\perp$ is generated. DU_{Or} computes:

1. For $(\mathbb{A},\rho)$, DU_{Or} generates $\{c_x \in \mathbb{Z}_q\}_{x\in X}$ such that $\sum_{x\in X} c_x \cdot \mathbb{A} = (1,0,0,\ldots,0)$ where $X = \{x|\rho(x) \in S\}$. DU_{Or} computes $s.G$ using (1) and (2), generates random $s' \in \mathbb{Z}_q$ and computes $rk_0 = s'.G - s.G$.
2. Assume $\mathbb{A}'$ is of size $l' \times n'$ and $\mathbb{A}'_y$ denotes y^{th} row of $\mathbb{A}'$ where $y = [1,l']$. DU_{Or} selects two random vector $\vec{\lambda}' = (s', v'_2, v'_3, \ldots, v'_{n'}) \in \mathbb{Z}_q^n$ and $\vec{w}' = (0, \zeta'_2, \zeta'_3, \ldots, \zeta'_{n'}) \in \mathbb{Z}_q^n$. DU_{Or} calculates $\lambda'_y = \vec{\lambda}' \cdot \mathbb{A}'^{\mathsf{T}}_y$ and $w'_y = \vec{w}' \cdot \mathbb{A}'^{\mathsf{T}}_y$ where $\mathbb{A}'^{\mathsf{T}}_y$ denotes transpose of $\mathbb{A}'_y$. Further, for all $y = [1,l']$, DU_{Or} computes $rk_{1,y} = \lambda'_y.G + w'_y.PK_{\rho(y)}$ and $rk_{2,y} = w'_y.G$
3. DU_{Or} transmits re-encryption key $rk = ((\mathbb{A}',\rho'), rk_0, \{rk_{1,y}, rk_{2,y}\}_{y=[1,l']})$.

$\boldsymbol{ReEncr(CT,rk)}$: PS inputs $CT = ((\mathbb{A},\rho), CT_0, \{CT_{1,x}, CT_{2,x}\}_{x=[1,l]})$, $rk = ((\mathbb{A}',\rho'), rk_0, \{rk_{1,y}, rk_{2,y}\}_{y=[1,l']})$ and computes transformed ciphertext as $CT'_0 = CT_0 + rk_0 = M + s.G + s'.G - s.G = M + s'.G$. For all $y = [1,l']$, PS computes $CT'_{1,y} = rk_{1,y}$, $CT'_{2,y} = rk_{2,y}$ and sends $CT' = ((\mathbb{A}',\rho'), CT'_0, \{CT'_{1,y}, CT'_{2,y}\}_{y=[1,l']})$ to CS.

$\boldsymbol{Dec_R(CT',sk')}$: DU_{Sh} with S' and $sk' = \{sk'_{y,ID_U}, \forall y \in S'\}$ decrypts CT' as:

1. DU_{Sh} generates set $Y = \{y|\rho'(y) \in S'\}$. If $S' \models (\mathbb{A}',\rho')$ then, there exists $\{c'_y \in \mathbb{Z}_q\}_{y\in Y}$ such that $\sum c'_y.\mathbb{A}'_y = (1,0,0,\ldots,0)$. DU_{Sh} performs:

$$\begin{aligned} D'_y &= \sum CT'_{1,y} - \sum CT'_{2,y}.sk'_{\rho(y),ID_U} \\ &= \sum \lambda'_y.G + w'_y.PK_{\rho(y)} - \left(\sum w'_y a_y.G + w'_y r_{\rho(y)} H(ID_U).G\right) \\ &= \sum \lambda'_y.G - w'_y r_{\rho(y)} H(ID_U).G \end{aligned} \tag{3}$$

2. DU_{Sh} utilizes $\{c'_y \in \mathbb{Z}_q\}_{y\in Y}$ as $\sum_{y\in Y} c'_y \lambda'_y = s'$ and $\sum_{y\in Y} c'_y w'_y = 0$. Hence, DU_{Sh} computes:

$$\sum c'_y.D'_y = \sum c'_y \lambda'_y.G - c'_y w'_y r_{\rho(y)} H(ID_U).G = s'.G \tag{4}$$

3. DU_{Sh} computes $M = CT'_0 - s'G$. DU_{Sh} calculates m by mapping point M on $E_p(a,b)$.

Correctness of Original Ciphertext

$$\begin{aligned} M &= CT_0 - c_x.\left(\sum CT_{1,x} - \sum CT_{2,x}.sk_{\rho(x),ID_U}\right) \\ &= CT_0 - c_x.\left(\sum \lambda_x.G + \sum w_x.PK_{\rho(x)} - \sum w_x a_x.G \right. \\ &\quad \left. - \sum w_x r_{\rho(x)} H(ID_U).G\right) \\ &= CT_0 - c_x.\left(\sum \lambda_x.G + \sum w_x a_x.G - \sum w_x a_x.G \right. \\ &\quad \left. - \sum w_x r_{\rho(x)} H(ID_U).G\right) \\ &= CT_0 - c_x.\left(\sum \lambda_x.G - \sum w_x r_{\rho(x)} H(ID_U).G\right) \\ &= CT_0 - \left(\sum c_x \lambda_x.G - \sum c_x w_x r_{\rho(x)} H(ID_U).G\right) \\ &= CT_0 - s.G = M + s.G - s.G = M \end{aligned}$$

It should be noted that $\sum c_x \lambda_x = s$ and $\sum c_x w_x = 0$.

Correctness of Re-Encrypted Ciphertext

$$\begin{aligned} M &= CT'_0 - c'_y.\left(\sum CT'_{1,y} - \sum CT'_{2,y}.sk'_{\rho(y),ID_U}\right) \\ &= CT'_0 - c'_y.\left(\sum \lambda'_y.G + \sum w'_y.PK_{\rho(y)} - \sum w'_y a_y.G \right. \\ &\quad \left. - \sum w'_y r_{\rho(y)} H(ID_U).G\right) \\ &= CT'_0 - c'_y.\left(\sum \lambda'_y.G + \sum w'_y a_y.G - \sum w'_y a_y.G \right. \\ &\quad \left. - \sum w'_y r_{\rho(y)} H(ID_U).G\right) \\ &= CT'_0 - c'_y.\left(\sum \lambda'_y.G - \sum w'_y r_{\rho(y)} H(ID_U).G\right) \\ &= CT'_0 - \left(\sum c'_y \lambda'_y.G - \sum c'_y w'_y r_{\rho(y)} H(ID_U).G\right) \\ &= CT'_0 - s'.G = M + s'.G - s'.G = M \end{aligned}$$

It should be noted that $\sum c'_y \lambda'_y = s'$ and $\sum c'_y w'_y = 0$.

6 SECURITY PROOF

Semantic security of ECC-CP-ABPRE is as follows:

Theorem 1. *The ECC-CP-ABPRE scheme is semantic secure under the DDH assumption.*

Proof. As per semantic security in Definition 2, following two lemma proves Sem-O and Sem-R security.

Lemma 1. *ECC-CP-ABPRE is Sem-O secure under the DDH assumption.*

Proof. Assume probability polynomial time (PPT), $\mathcal{A}$ breaks Sem-O security with non-negligible probability $\varepsilon > 0$, then there exists PPT algorithm $\mathcal{B}$ that distinguishes between random tuples and DDH with advantage of $\varepsilon/2$. Challenger $\mathcal{C}$ selects random $b,c \in \mathbb{Z}_q$, $\beta \in \{0,1\}$ and $R \in P$. Assume G is generator of group P with order q. Then $Z = bcG$ if $\beta = 0$, else $Z = R$. $\mathcal{C}$ transmits (G, bG, cG, Z) to $\mathcal{B}$. $\mathcal{B}$ maintains secret and re-encryption keys lists that are empty initially: (i) Li_{sk}: records (S, sk_S) tuple. (ii) Li_{rk}: stores $(S, (\mathbb{A}',\rho'), rk, ind)$ tuple where $ind = 1$ indicates valid rk and $ind = 0$ indicates random rk. $\mathcal{B}$ executes the following:

Initialization: $\mathcal{A}$ provides challenge $(\mathbb{A}'',\rho'')$ to $\mathcal{B}$.

Setup: $\mathcal{B}$ selects random $a_i \in \mathbb{Z}_q$ and computes $PK_i = a_i b.G$ as public key for A_i where $i = [1,n]$. $\mathcal{B}$ transmits $PP = \{PK_1, PK_2, \ldots, PK_n, G, H\}$ to $\mathcal{A}$.

Query Phase 1: In this phase: (1)$O_{sk}(S)$: $\mathcal{A}$ queries on S and $\mathcal{B}$ confirms whether $S \not\models (\mathbb{A}'',\rho'')$. If unsatisfied, $\mathcal{B}$ outputs $\perp$ otherwise, $\mathcal{B}$ selects $r_i \in \mathbb{Z}_q$ and computes $sk_i = a_i b + H(ID_{\mathcal{A}}).r_i$ where $ID_{\mathcal{A}}$ denotes identity of $\mathcal{A}$. (2)$O_{rk}(S,(\mathbb{A}',\rho'),CT)$: $\mathcal{B}$ verifies if $S \models (\mathbb{A}'',\rho'')$ and there exists $(S', sk_{S'})$ in Li_{sk} where $S' \models (A',\rho')$ then, $\perp$ is generated as output.

Else if $S \models (\mathbb{A}'', \rho'')$ and no entry $(S', sk_{S'})$ exists in Li_{sk} where $S' \models (\mathbb{A}', \rho')$ then, $\mathcal{B}$ generates random rk and $(S, (\mathbb{A}', \rho'), rk, 0)$ is added to Li_{rk}. Otherwise, $\mathcal{B}$ queries $O_{sk}(S)$ to receive sk_S, computes rk using *ReEncKeyGenr* and $(S, (\mathbb{A}', \rho'), rk, 1)$ is added to Li_{rk}.
(3)$O_{re}(S, (\mathbb{A}', \rho'), CT)$: If $S \models (\mathbb{A}'', \rho'')$ and there exists $(S', sk_{S'})$ in Li_{sk} wherein $S' \models (A', \rho')$ then, $\perp$ is generated as output. Else, if $(S, (\mathbb{A}', \rho'), rk, 0)$ or $(S, (\mathbb{A}', \rho'), rk, 1)$ exists in Li_{rk}, $\mathcal{B}$ encrypts CT with rk. Otherwise, $\mathcal{B}$ queries $O_{rk}(S, (\mathbb{A}', \rho'), CT)$ to compute rk followed by re-encryption of CT with rk.
Challenge: $\mathcal{A}$ selects and transmits (m_0, m_1) of same length to $\mathcal{B}$. $\mathcal{B}$ flips a coin β and selects random $s \in \mathbb{Z}_q$, $\vec{\lambda} = (s, v_2, v_3, \ldots, v_n) \in \mathbb{Z}_q^n$ and $\vec{w} = (0, \zeta_2, \zeta_3, \ldots, \zeta_n) \in \mathbb{Z}_q^n$. $\mathcal{B}$ generates challenge ciphertext $CT_0'' = M_\beta + s.G$, $CT_{1,x}'' = \lambda_x G + a_{\rho(x)} w_x.Z$ and $CT_{2,x}'' = w_x c.G$ where $x = [1, l'']$, $\lambda_x = \vec{\lambda} \cdot \mathbb{A}_x''^{\intercal}$, $w_x = \vec{w} \cdot \mathbb{A}_x''^{\intercal}$ and M_β denotes plaintext m_β on elliptic curve. $\mathcal{B}$ returns this challenge ciphertext $CT'' = \{(\mathbb{A}'', \rho''), CT_0'', \{CT_{1,x}'', CT_{2,x}''\}_{x \in [1, l'']}\}$ to $\mathcal{A}$.
Query Phase 2: Similar to Query Phase 1 with constraints mentioned in Sem-O model.
Guess: $\mathcal{A}$ outputs a guess β'. If $\mathcal{B}$ outputs 0, then $Z = bcG$ and $\beta' = \beta$; else, $\mathcal{B}$ returns 1 indicating $Z = R$.
Analysis: If $Z = bcG$, then CT'' is perfect ciphertext. Thus, $Prob[\mathcal{B}(G, bG, cG, Z = bcG) = 0] = 1/2 + \varepsilon$. If $Z = R$ then $Prob[\mathcal{B}(G, bG, cG, Z = R) = 0] = 1/2$. Hence, the advantage of $\mathcal{B}$ in breaking security is:

$$\begin{aligned} Adv^{Sem-O} &= \frac{1}{2}(Prob[\mathcal{B}(G, bG, cG, Z = bcG) = 0] \\ &\quad + Prob[\mathcal{B}(G, bG, cG, Z = R) = 0]) - \frac{1}{2} \\ &= \frac{1}{2}(\frac{1}{2} + \varepsilon + \frac{1}{2}) - \frac{1}{2} = \frac{\varepsilon}{2} \end{aligned}$$

Thus, $\mathcal{B}$ can solve DDH assumption with ε. □

Lemma 2. *ECC-CP-ABPRE scheme is Sem-R secure under the DDH assumption.*

Proof. Assume PPT $\mathcal{A}$ breaks Sem-O security with non-negligible probability $\varepsilon > 0$, then there exists PPT algorithm $\mathcal{B}$ that distinguishes between random tuple and DDH with advantage $\varepsilon/2$.
Initialization, *Setup* and *Query Phase 1* is same as Lemma 1.
Challenge: $\mathcal{A}$ selects (m_0, m_1) of same size and transmits to $\mathcal{B}$. $\mathcal{B}$ selects S where $S \not\models (\mathbb{A}'', \rho'')$ and outputs sk_S and $rk = ReEncKeyGenr(sk_S, (\mathbb{A}'', \rho''), CT)$. $\mathcal{B}$ flips a coin β and selects $(\mathbb{A}, \rho)$ where $S \models (\mathbb{A}, \rho)$ and computes $CT = Enc(m_\beta, (\mathbb{A}, \rho))$. $\mathcal{B}$ calculates $CT''' = ReEncr(CT, rk)$ and returns CT''' to $\mathcal{A}$.
Query Phase 2: Similar to Query in Sem-R model.
Guess: $\mathcal{A}$ outputs guess β'. If $\mathcal{B}$ outputs 0, it indicates $Z = bcG$ & $\beta' = \beta$; else, $\mathcal{B}$ outputs 1 indicating $Z = R$.
Analysis: If $Z = bcG$, then CT''' is perfect ciphertext. Thus, $Prob[\mathcal{B}(G, bG, cG, Z = bcG) = 0] = 1/2 + \varepsilon$. If $Z = R$ then $Prob[\mathcal{B}(G, bG, cG, Z = R) = 0] = 1/2$. Hence, the advantage of $\mathcal{B}$ in breaking security is:

$$\begin{aligned} Adv^{Sem-R} &= \frac{1}{2}(Prob[\mathcal{B}(G, bG, cG, Z = bcG) = 0] \\ &\quad + Prob[\mathcal{B}(G, bG, cG, Z = R) = 0]) - \frac{1}{2} \\ &= \frac{1}{2}(\frac{1}{2} + \varepsilon + \frac{1}{2}) - \frac{1}{2} = \frac{\varepsilon}{2} \end{aligned}$$

Thus, $\mathcal{B}$ can solve DDH assumption with non-negligible advantage ε. □

Hence, Theorem 1 is proved. □

7 SECURITY ANALYSIS

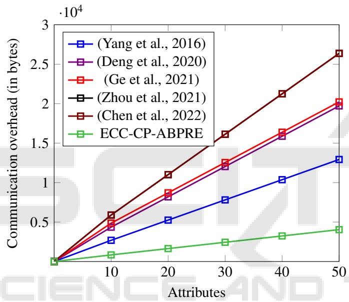
The security analysis are as follows:
Data Confidentiality: In ECC-CP-ABPRE, only valid users with corresponding attributes satisfying access policy can decrypt ciphertext. The security is based on ECDLP which ensures inefficacy of invalid users to compute master secret key $\{a_i\}$ from $PK_i = a_i.G$ in polynomial time. In *Enc*, assume M mapped as $m.G$ on elliptic curve where $m \in \mathbb{Z}_q$ and DO selects random $s \in \mathbb{Z}_q$ thus, $CT_0 = (m + s).G$. Note that CT_0 denotes a random point on elliptic curve from $\mathcal{A}$'s viewpoint hence, leaks no valuable information about M. Also, secret s is split by λ_x using LSSS that can be recovered by DU's attributes satisfying $(\mathbb{A}, \rho)$. Thus, any invalid user with attributes not satisfying $(\mathbb{A}, \rho)$, there exits no corresponding rows $\mathbb{A}_x$ such that $\sum c_x \mathbb{A}_x = (1, 0, 0 \ldots, 0)$ where $x = [1, l]$. Hence, secret s, first entry of vector $\vec{\lambda}$ cannot be computed thereby, ensuring data confidentiality of CT. In *ReEncKeyGenr*, DU_{Or} computes $rk_0 = (s' - s).G$ which denotes random point on elliptic curve. Thus, PS acquires no valuable information from rk_0 due to ECDLP. In *ReEncr* with $(\mathbb{A}', \rho')$, PS computes $CT_0' = (m + s').G$ where random $m, s' \in \mathbb{Z}_q$. Similar to CT, ECDLP and LSSS ensures security of CT'. Thus, invalid user with attributes not satisfying $(\mathbb{A}', \rho')$, there exits no corresponding rows $\mathbb{A}_y'$ such that $\sum c_y' \mathbb{A}_y' = (1, 0, 0 \ldots, 0)$ where $y = [1, l']$. Hence, secret s', first entry of vector $\vec{\lambda}'$ cannot be computed thereby, ensuring data confidentiality of CT'.
Resistant to Collusion Attack: ECC-CP-ABPRE should resist collusion attack i.e. if multiple users collude their secret keys, they should be incapable in

ciphertext decryption unless at-least one user can decrypt ciphertext independently. *KeyGenr* generates sk that binds unique ID_U of DU and random r_i with attributes of corresponding user. Hence, sk of different DU cannot be combined successfully to decrypt ciphertext. Assume Alice with attribute A, and Bob with attributes C and D, collude to gain underlying plaintext in ciphertext-policy $(A \vee B) \wedge C \wedge D$. Neither of them can decrypt the ciphertext individually, Alice computes $D_x^{Alice} = \sum \lambda_x.G - w_x r_{\rho(x)}^{Alice} H(ID_{Alice})G$ and Bob computes $D_x^{Bob} = \sum \lambda_x.G - w_x r_{\rho(x)}^{Bob} H(ID_{Bob})G$ for some x. Note that Alice and Bob have $H(ID_{Alice}) \neq H(ID_{Bob})$ and $r_{\rho(x)}^{Alice} \neq r_{\rho(x)}^{Bob}$ in their sk. Hence, they cannot compute constant set $\{c_x \in \mathbb{Z}_q\}$, such that $\sum c_x \mathbb{A}_x = (1,0,0,\ldots,0)$ and are unable to compute sG thus, resistant to collusion attack.

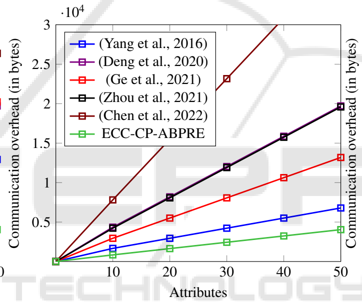
Specificity of Re-Encryption Key: In ECC-CP-ABPRE, rk should re-encrypt specified ciphertext i.e. rk generated for policy update of specified ciphertext. To update $(\mathbb{A}, \rho)$ to $(\mathbb{A}', \rho')$, DU_{Or} in *ReEncKeyGenr* generates random $s' \in \mathbb{Z}_q$ and computes $rk_0 = s'G - sG$ for original ciphertext $CT_0 = M + sG$ under $(\mathbb{A}, \rho)$. Thus, any unspecified ciphertext $CT_0^* = M^* + s^*G$ under $(\mathbb{A}^*, \rho^*)$ cannot be re-encrypted with rk_0 as it requires re-encryption key incorporating s^*G. Also, it is difficult to compute other re-encryption keys from $rk = \{rk_0, \{rk_{1,y}, rk_{2,y}\}_{y=[1,l']}\}$ due to random $s'.G$ hence, ensuring specificity of re-encryption keys.

8 PERFORMANCE ANALYSIS

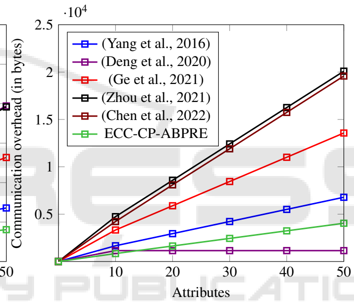
ECC-CP-ABPRE is compared with existing schemes by incorporating Java pairing-based cryptography package (De Caro and Iovino, 2011) for pairing operations on Type 1 symmetric pairing $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with 512-bit supersingular curve of embedding degree 2, size of the elliptic curve is 512 bits, the order of the elliptic curve group is 160 bits that are implemented



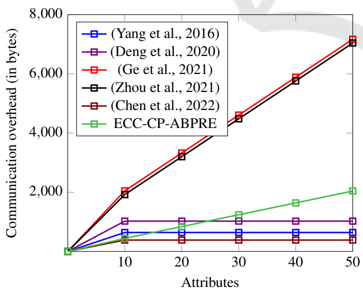
(a) Re-encryption Key.



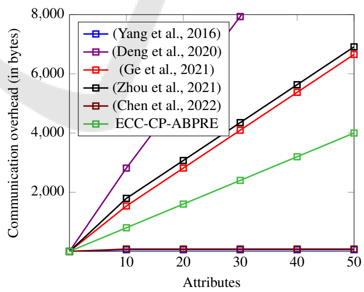
(b) Original Ciphertext.



(c) Re-encrypted Ciphertext.

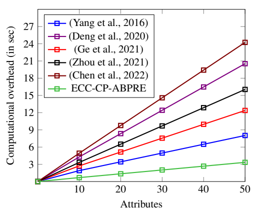


(d) Public Key.

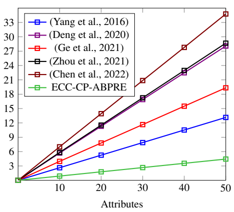


(e) Secret Key.

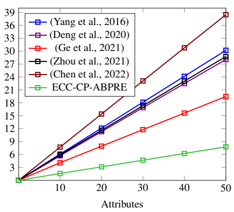
Figure 2: Comparison of Computational Overhead(in bytes).



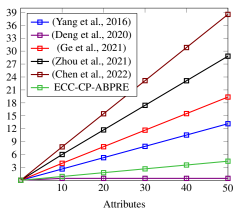
(a) Encrypt Plaintext.



(b) Decrypt CT.



(c) Re-encryption.



(d) Decrypt CT'.

Figure 3: Comparison of Communication Overhead(in sec).

using laptop configuration of Intel Core i5-1135G7 @ 2.40 GHz, 8 GB RAM. For communication overhead, $|\mathbb{G}|$, $|\mathbb{G}_T|$, $|P|$, and $|\mathbb{Z}_q|$ denotes size of elements in $\mathbb{G}$, $\mathbb{G}_T$, ECC point and random element in $\mathbb{Z}_q$. $|\mathbb{G}|$, $|\mathbb{G}_T|$, $|P|$ and $|\mathbb{Z}_q|$ are 1024 bits, 2048 bits, 320 bits and 160 bits, respectively. ECC-CP-ABPRE incurs less communication overhead to send rk and CT compared with (Yang et al., 2016; Deng et al., 2020; Ge et al., 2021; Zhou et al., 2021; Chen et al., 2022) as in Fig. 2(a) and 2(b), respectively. To transmit CT', ECC-CP-ABPRE performs better than (Yang et al., 2016; Ge et al., 2021; Zhou et al., 2021; Chen et al., 2022) as in Fig. 2(c) however, it is increased compared with (Deng et al., 2020) as latter scheme leverages PS to perform partial decryption of ciphertext in re-encryption. To transmit public key and secret key as in Fig. 2(d) and 2(e), ECC-CP-ABPRE increases by $\approx 320n$ bits and $\approx 160n'$ bits, respectively, compared to (Yang et al., 2016; Chen et al., 2022). This can be circumvented as AA transmits these keys once during system initialization and user registration.

For computational overhead comparison, T_{EXP}, T_{GM}, T_{BP}, T_{SM} and T_{PA} represent as time for exponential operation in $\mathbb{G}$, multiplication operation in $\mathbb{G}$, bilinear pairing, scalar multiplication, and point addition in ECC, respectively. T_{EXP}, T_{GM}, T_{BP}, T_{SM} and T_{PA} are 0.0765 sec, 0.0118 sec, 0.1099 sec, 0.0220 sec and 0.0002 sec, respectively. ECC-CP-ABPRE requires significantly less computation overhead than (Yang et al., 2016; Deng et al., 2020; Ge et al., 2021; Zhou et al., 2021; Chen et al., 2022) to encrypt plaintext, decrypt original ciphertext and re-encrypt ciphertext as shown in Fig. 3(a), 3(b), and 3(c) respectively. To decrypt CT', ECC-CP-ABPRE outperforms (Yang et al., 2016; Ge et al., 2021; Zhou et al., 2021; Chen et al., 2022) as in Fig. 3(d). However, computational overhead in (Deng et al., 2020) is less than ECC-CP-ABPRE as it leverages PS to partially decrypt CT in re-encryption phase. Nonetheless, computational overhead in re-encryption phase is significantly increased as PS performs both re-encryption and partial decryption of CT as in Fig. 3(c). Hence, the overall computational overhead of ECC-CP-ABPRE is less than (Deng et al., 2020). Thus, the overall efficiency of ECC-CP-ABPRE surpasses (Yang et al., 2016; Deng et al., 2020; Ge et al., 2021; Zhou et al., 2021; Chen et al., 2022) in terms of communication and computational overheads.

9 CONCLUSION

This paper designs a novel pairing-free ECC-based CP-ABPRE to enable efficient sharing of encrypted data in clouds. ECC-CP-ABPRE replaces expensive bilinear pairing operations with scalar multiplications to update ciphertext-policy. The security analysis illustrates semantic security of both original ciphertext and re-encrypted ciphertext under DDH assumption. It ensures data confidentiality and specificity of re-encryption keys while resisting collusion attack. The performance results demonstrate its efficiency. In future, ECC-CP-ABPRE will be extended to trace and revoke malicious data users leaking their secret keys to unauthorized users in the system.

REFERENCES

Beimel, A. (1996). Secure schemes for secret sharing and key distribution. *Technion-Israel Institute of technology, Faculty of Computer Science*.

Bethencourt, J., Sahai, A., and Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, pages 321–334. IEEE.

Chen, J., Wang, J., Yin, X., and Ning, J. (2022). A fine-grained medical data sharing scheme with ciphertext reencryption. *Wireless Communications and Mobile Computing*, 2022.

De Caro, A. and Iovino, V. (2011). jpbc: Java pairing based cryptography. In *2011 IEEE symposium on computers and communications (ISCC)*, pages 850–855. IEEE.

Deng, H., Qin, Z., Wu, Q., Guan, Z., and Zhou, Y. (2020). Flexible attribute-based proxy re-encryption for efficient data sharing. *Information Sciences*, 511:94–113.

Dhakad, N. and Kar, J. (2022). Eppdp: An efficient privacy-preserving data possession with provable security in cloud storage. *IEEE Systems Journal*.

Ge, C., Susilo, W., Baek, J., Liu, Z., Xia, J., and Fang, L. (2021). A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds. *IEEE Transactions on Dependable and Secure Computing*.

Liang, K., Au, M. H., Liu, J. K., Susilo, W., Wong, D. S., Yang, G., Yu, Y., and Yang, A. (2015). A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. *Future Generation Computer Systems*, 52:95–108.

Liang, X., Cao, Z., Lin, H., and Shao, J. (2009). Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 276–286.

Luo, S., Hu, J., and Chen, Z. (2010). Ciphertext policy attribute-based proxy re-encryption. In *International Conference on Information and Communications Security*, pages 401–415. Springer.

Miller, V. S. (1985). Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer.

Rezaeibagha, F., Mu, Y., Huang, K., Chen, L., and Zhang, L. (2021). Toward secure data computation and outsource for multi-user cloud-based iot. *IEEE Transactions on Cloud Computing*.

Wang, M., Xu, L., Hao, R., and Yang, M. (2023). Secure auditing and deduplication with efficient ownership management for cloud storage. *Journal of Systems Architecture*, 142:102953.

Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *International workshop on public key cryptography*, pages 53–70. Springer.

Yang, Y., Zhu, H., Lu, H., Weng, J., Zhang, Y., and Choo, K.-K. R. (2016). Cloud based data sharing with fine-grained proxy re-encryption. *Pervasive and Mobile computing*, 28:122–134.

Zhou, X., Xu, K., Wang, N., Jiao, J., Dong, N., Han, M., and Xu, H. (2021). A secure and privacy-preserving machine learning model sharing scheme for edge-enabled iot. *IEEE Access*, 9:17256–17265.