A Minimization Number of Final Exponentiations and Inversions for Reducing the Decryption Process Time in ELiPS-Based CP-ABE

Le Hoang Anh [1,2,*], Yuta Kawada [1], Samsul Huda [3], Md. Arshad Ali [4], Yuta Kodera [1], and Yasuyuki Nogami [1]

[1] Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Japan
[2] An Giang University, Vietnam National University Ho Chi Minh City, Vietnam
[3] Green Innovation Center, Okayama University, Japan
[4] Faculty of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University, Bangladesh
Email: lhanh@s.okayama-u.ac.jp (L.H.A.); yuta_kawada@s.okayama-u.ac.jp (Y.K.); shuda@okayama-u.ac.jp (S.H.); arshad@hstu.ac.bd (Md.A.A.); yuta_kodera@okayama-u.ac.jp (Y.K.); yasuyuki.nogami@okayama-u.ac.jp (Y.N.)
*Corresponding author

***Abstract*—Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is an advanced encryption method used across various fields, including cloud storage, Personal Health Records, Internet of Things, Internet of Vehicles, and blockchain. However, challenges such as insufficient security levels and performance issues in modern applications pose significant drawbacks inherent to the CP-ABE scheme. Previously, we studied Efficient Library for Pairing Systems (ELiPS)-Based CP-ABE to enhance performance and strengthen security level. This approach involved adopting ELiPS as an efficient library for pairing systems. This study increased the security level up to 128 bits and reduced the computation cost, excluding the decryption process. The decryption part primarily utilizes inversion in the Lagrange coefficient part and pairing, which includes the Miller loop and final exponentiation. Both the final exponentiation and inversion are equivalent to the number of attributes. In this paper, we further explore reducing the decryption process time by minimizing the number of final exponentiations and inversions. The effectiveness of the proposal is confirmed through security and experimental analyses where the decryption time in the proposed scheme decreased by an average of 45.45% compared to previous work.**

***Keywords*—Pairing-Based Cryptography, Attribute-Based Encryption, Ciphertext-Policy Attribute-Based Encryption (CP-ABE), Efficient Library for Pairing Systems (ELiPS), ELiPS-Based CP-ABE, pairing**

I. Introduction

Ciphertext-Policy Attribute-Based Encryption is a type of advanced encryption scheme that allows for access control based on specific attributes assigned to users and data [1, 2]. CP-ABE finds applications in diverse fields, including but not limited to cloud storage [3–5], Personal Health Record [6–8], Internet of Things [9–12], blockchain [13–15], and other prominent fields [16–20].

Cloud computing has become increasingly popular in our lives, with users storing a wide range of data in the cloud. CP-ABE is commonly employed as a mechanism to safeguard data in cloud computing [3–5].

The Personal Health Record (PHR) contains extensive private information, including the user's health conditions, medical history, medications, and other personal details. Recognizing the sensitive nature of PHR, CP-ABE has been considered a suitable choice for access control and safeguarding private data within PHR systems [6–8].

The growing proliferation of the Internet of Things generates vast amounts of data, leading to an increased emphasis on data access control in security [9–12]. CP-ABE meets this demand by allowing data sources to encrypt data while enforcing a security access policy cryptographically.

Blockchain stands out as one of the most talked-about technologies in recent years, ushering in a genuine revolution in the financial sector. Its capability to offer cryptographically validated transactions and data, free from the influence of any third-party organization, highlights its significance. Overall, blockchain technology boasts key advantages, including decentralization, persistence, anonymity, and auditability. To enhance the security and privacy of data without relying on a third party for control, CP-ABE has been integrated with blockchain technology [13–15].

CP-ABE has various important applications; however, the original CP-ABE based on the Pairing-Based Cryptography (PBC) library has not been updated for a significant time. The PBC library provides only an 80-bit security level. Thus, PBC-Based CP-ABE has some drawbacks such as performance issues and a lack of sufficient security.

Manuscript received January 16, 2024; revised February 18, 2024; accepted March 1, 2024; published June 20, 2024.

doi: 10.12720/jait.15.6.748-755

However, Anh *et al.* [2] addressed these shortcomings by proposing an ELiPS-Based CP-ABE scheme, which enhanced security strength and increased performance. Their work in [2] integrated ELiPS into the CP-ABE framework. ELiPS serves as a foundation for cryptosystems and offers a 128-bit security level. They proposed several methods to improve the performance of the CP-ABE [2]. The results indicated ELiPS-Based CP-ABE enhanced the security strength and increased the setup, key generation, and encryption speeds of the CP-ABE system. Nevertheless, decryption processing in ELiPS-Based CP-ABE remains a challenge due to its heaviness.

Accordingly, in this paper, the authors aim to reduce the decryption processing time of Anh *et al.*'s work [2] by proposing methods to minimize the number of final exponentiations and inversions. Through formula analysis, the proposed scheme reduces $2n-1$ times final exponentiations and $n-1$ times inversions. We also conducted several experiments to assess the performance of our proposed formula while increasing the number of attributes from 5 to 100. Experimental analysis shows that the decryption time in the proposed scheme decreased by an average of 45.45% compared to previous work [2].

Some of the primary contributions of this study are as follows:

(1) Transform the decryption equation to a Miller loop and final exponentiation bases.
(2) The number of final exponentiations is proportional to the number of attributes.
(3) Transform the decryption equation by performing final exponentiation only once at the last step.
(4) The count of inversion operations in the Lagrange coefficient function is minimized by employing a single inversion operation.

II. Reliminariers

This section, we first introduce background information on arithmetic operations over elliptic curves, hash-to-curve, and pairings, which play a vital role in the CP-ABE scheme. Next, we present the Discrete Logarithm Problem and Elliptic Curve Discrete Logarithm Problem. Then, an overview of the ELiPS-Based CP-ABE algorithm is also introduced.

A. Arithmetic Operations over the Elliptic Curve

An elliptic curve E of short Weierstrass form defined over $\mathbb{F}_{p^m}$ is presented as follows [21]:

$$E: y^2 = x^3 + ax + b, \tag{1}$$

where a and b are coefficients satisfying following condition:

$$4a^3 + 27b^2 \neq 0. \tag{2}$$

Let r be a prime integer of number of points on E given as $\#E(\mathbb{F}_p) = p + 1 - t$, where $|t| \leq 2\sqrt{p}$. Let k be the smallest integer such that $r|(p^k - 1)$, which is called the embedding degree. When the embedding degree k is greater than one, then $E[r]$ is defined over $\mathbb{F}_{p^k}$.

Let $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$, and $R = (x_R, y_R)$ be affine rational points on E, as can be seen in Eq. (1). The arithmetic operations over the elliptic curve are defined as follows.

1) Elliptic Curve Addition (ECA)

If $P \neq Q$, point addition formula for computing $R = P + Q$ is given as follows:

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}, \tag{3}$$

$$\begin{cases} x_R = \lambda^2 - x_P - x_Q, \\ y_R = \lambda(x_P - x_R) - y_P. \end{cases} \tag{4}$$

2) Elliptic Curve Doubling (ECD)

If $P = Q$, point doubling formula for computing $R = P + Q = 2P$ is given as follows:

$$\lambda = \frac{3x_P^2 + a}{2y_P}, \tag{5}$$

$$\begin{cases} x_R = \lambda^2 - 2x_P, \\ y_R = \lambda(x_P - x_R) - y_P. \end{cases} \tag{6}$$

3) Elliptic curve Scalar Multiplication (SCM)

Repeating to use $+$ for P leads to the definition of a point sP, which is P multiplied by s. Point scalar multiplication formula for calculating $R = sP$ as:

$$R = sP = \underbrace{P + P + \cdots + P}_{s-\text{terms}}, \tag{7}$$

where $s > 0$.

4) Hash function $\mathcal{H}$ onto elliptic curve

Hash function $\mathcal{H}$ maps any attribute described as a binary string to a random group element.

$$\mathcal{H}: \{0,1\}^* \rightarrow \mathbb{G}. \tag{8}$$

Hash function $\mathcal{H}$ holds following properties:

- Pre-image resistance: For a given output h, it is computationally infeasible to find a value m such that $\mathcal{H}(m) = h$.
- 2nd pre-image resistance: For a given input m, it is computationally infeasible to find a value m', where $m \neq m'$ such that $\mathcal{H}(m) = \mathcal{H}(m')$.
- Collision resistance: It is computationally infeasible to find two values m and m', where $m \neq m'$ such that $\mathcal{H}(m) = \mathcal{H}(m')$.

5) Pairing map

The subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$ over $E(\mathbb{F}_{p^{12}})$ are defined as follows [2]:

$$\begin{cases} \mathbb{G}_1 = E[r] \cap \mathrm{Ker}(\pi_p - [1]), \\ \mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\pi_p - [p]), \end{cases} \tag{9}$$

where π_p is a Frobenius map.

A pairing e is a map from two elements in groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to an element in group $\mathbb{G}_T$, defined as:

$$e: \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{G}_T, \tag{10}$$

which has the following properties [2]:

- Bilinear map: For all rational points $P \in \mathbb{G}_1$, and $Q, Q' \in \mathbb{G}_2$, and integers $a, b \in \mathbb{Z}_p$, we have:

$$e(Q + Q', P) = e(Q, P) \cdot e(Q', P), \tag{11}$$

$$e(aQ, bP) = e(Q, P)^{ab}. \tag{12}$$

- Non-degenerate: If $P \neq \mathcal{O}$ and $Q \neq \mathcal{O}$, then:

$$e(Q, P) \neq 1. \tag{13}$$

6) Types of pairings

The groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are elliptic curve subgroups, and the group $\mathbb{G}_T$ is the multiplicative group of a finite field. We have three types of pairings:

- Type I: When $\mathbb{G}_1 = \mathbb{G}_2$.
- Type II: When $\mathbb{G}_1 \neq \mathbb{G}_2$ but an efficiently computable isomorphism $\phi: \mathbb{G}_2 \to \mathbb{G}_1$ is known, while none is known in the other direction.
- Type III: When $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable isomorphism is known between $\mathbb{G}_1$ and $\mathbb{G}_2$, in either direction.

Pairing Type I is also referred to as symmetric pairing, while pairing Types II and III are known as asymmetric pairings.

7) Sextic twist

The element of $\mathbb{G}_2$ is a rational point in $E(\mathbb{F}_{p^{12}})$. However, it is known to only possess an equal amount of information with a rational point existing on $E'(\mathbb{F}_{p^2})$. Let z be a quadratic non-residue and cubic non-residue over $\mathbb{F}_{p^2}$ and defines two elliptic curves as follows:

$$\begin{cases} E: y^2 = x^3 + b \text{ over } \mathbb{F}_{p^{12}}, \\ E': y^2 = x^3 + bz \text{ over } \mathbb{F}_{p^2}. \end{cases} \tag{14}$$

The sextic twist $\phi: E' \to E$ is defined as follows [2]:

$$\phi: E' \to E \qquad (x, y) \mapsto (z^{-\frac{1}{3}}x, z^{-\frac{1}{2}}y)\,. \tag{15}$$

B. Discrete Logarithm Problem and Elliptic Curve Discrete Logarithm Problem

The security of pairing-based cryptography relies on the hardness of the Discrete Logarithm Problem (DLP) and the Elliptic Curve Discrete Logarithm Problem (ECDLP).

In a finite group, computing $a = b^x$ (where $a, b \in \mathbb{F}_p$, $x \in \mathbb{Z}$) is easy but finding x from a and b is a hard problem, known as the DLP.

In an elliptic curve group, we can easily calculate $R = sP$, where $P, R \in E(\mathbb{F}_p)$ and $s \in \mathbb{Z}$. However, recovering s from P and R is a hard problem, known as the ECDLP.

C. Efficient Library for Pairing Systems

The ELiPS[1] library serves as a foundation for cryptosystems. It is a specifically designed cryptographic library that provides various functionalities such as hash-to-curve, pairing, elliptic curve addition, elliptic curve doubling, and elliptic curve scalar multiplication [22].

The ELiPS library, which utilizes the Barreto-Lynn-Scott (BLS)-12 curve, provides a 128-bit security level. It employs the BLS curve E with an embedding degree of $k = 12$. The work by Hattori *et al.* [22] demonstrated that ELiPS provides a slightly faster execution time compared to previous libraries while maintaining a parameter set that ensures a high-security level.

Anh *et al.* [2] conducted a comparative analysis of four prominent libraries (PBC, MCL, RELIC, and ELiPS libraries) in this research area. They compared these libraries in terms of hash-to-curve, pairing, exponentiation, scalar multiplication domains, and security level. Table I suggests that the ELiPS library holds an advantage in these domains. Based on this evidence, the ELiPS library is deemed suitable for the CP-ABE system. Therefore, we aim to address the drawbacks of ELiPS-Based CP-ABE.

TABLE I. COMPARISON AMONG PAIRING LIBRARIES [2]

Parameters		PBC	MCL	RELIC	ELiPS
Security level		80-bit	128-bit	128-bit	128-bit
Hash-to-curve		3.2 [ms]	0.3 [ms]	0.6 [ms]	0.1 [ms]
Pairing		0.9 [ms]	1.1 [ms]	2.6 [ms]	2.2 [ms]
Exponentiation		0.1 [ms]	0.8 [ms]	1.3 [ms]	0.6 [ms]
Scalar	$\mathbb{G}_1$	1.2 [ms]	0.3 [ms]	0.3 [ms]	0.2 [ms]
multiplication	$\mathbb{G}_2$	1.2 [ms]	0.4 [ms]	0.7 [ms]	0.5 [ms]

D. Overview of ELiPS-Based CP-ABE

CP-ABE is an encryption system that enables fine-grained access control for encrypted information [1, 2]. In CP-ABE, data encryption relies on a set of attributes and access to the encrypted data is authorized based on predetermined access policies linked to those attributes [1, 2]. This method permits adaptable and tailored access control, empowering data owners to stipulate the specific attributes necessary for decryption.

The ELiPS-Based CP-ABE algorithm, primarily based on hash-to-curve and pairing processes, includes four primary components [2]:

1) Setup

It primarily employs pairing and exponentiation operations. This stage initiates by creating the $\mathbb{G}$ group, which possesses an order denoted as r.

Fig. 1 shows the $\mathbb{G}$ generation process. Firstly, the algorithm generates g_1' over $E(\mathbb{F}_p)$ and g_2' over $E'(\mathbb{F}_{p^2})$. Secondly, g_1' and g_2' are mapped to g_1 in group $\mathbb{G}_1$ andg_2 in group $\mathbb{G}_2$, respectively. Thirdly, the generator g in group $\mathbb{G}$ is formed by the operation $g_1 + g_2$.

$$\mathbb{G} =< g_1 + g_2 >. \tag{16}$$

[1] *ELiPS*. Information Security laboratory Okayama University. [Online]. Available: https://github.com/ISecOkayamaUniv/ELiPS

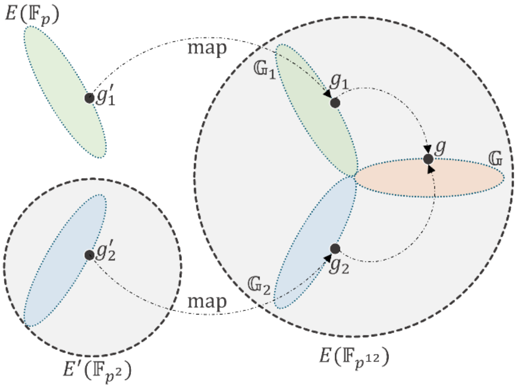

Fig. 1. The process for generating $\mathbb{G}, \mathbb{G}_1$, and $\mathbb{G}_2$.

Then, the algorithm generates random values $\alpha, \beta \in \mathbb{Z}_r$, and utilizes a bilinear map $e_{\text{sym}}: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ to calculate the master key MK and public key PK as follows [2]:

$$MK = (\beta, d), \tag{17}$$

$$PK = (g, h, u, v), \tag{18}$$

where:

- $d = \alpha g,$
- $h = \beta g,$
- $u = \beta^{-1} g,$
- $v = e(g, g)^{\alpha}.$

However, ELiPS library employs asymmetric pairing; therefore, it requires the transformation of asymmetric pairing into symmetric pairing. In previous work, Anh *et al.* [2] successfully utilized Shirase's method to achieve this transformation by extracting $P' \in \mathbb{G}_1$ and $Q' \in \mathbb{G}_2$ from P and $Q \in \mathbb{G}$, respectively. Then, the algorithm uses asymmetric pairing for pairing calculation. Fig. 2 illustrates the concept of the extraction procedure.

The transformation between asymmetric and symmetric pairing can be defined as [2]:

$$e_{\text{sym}}(Q, P) = e_{\text{asy}}(\text{ext2}(Q), \text{ext1}(P)) \tag{19}$$

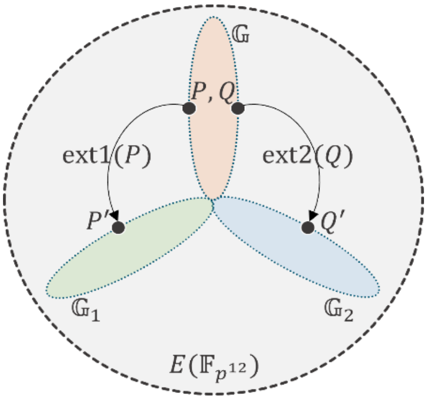

Fig. 2. Extraction of P' *and* Q' in transforming asymmetric pairing to symmetric pairing.

Next, a method for extracting $\mathbb{G}_1$ and $\mathbb{G}_2$ from $\mathbb{G}$ is conducted in the following manner:

Let $l = (p-1)^{-1} (\operatorname{mod} r)$, where r is an order of subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$. Then, the values of ext1 and ext2 can be calculated as follows [2]:

$$\begin{cases} \text{ext1} = \big([p] - \pi_p\big)[l], \\ \text{ext2} = \big(\pi_p - [1]\big)[l]. \end{cases} \tag{20}$$

The symmetric pairing procedure in ELiPS-Based CP-ABE is processed as follows:

(1) Calculate the rational points P and Q, where $P, Q \in \mathbb{G}$.

(2) Then, it calls the ext1 and ext2 functions to calculate P' and Q', as demonstrated in Fig. 2.

$$P' = \text{ext1}(P),$$
$$Q' = \text{ext2}(Q),$$

where $P' \in \mathbb{G}_1$ and $Q' \in \mathbb{G}_2$.

(3) Afterward, the algorithm calls $e_{\text{asy}}(Q', P')$ to calculate asymmetric pairing. Asymmetric pairing uses Miller loop and final exponentiation to calculate and return asymmetric pairing value.

2) *Key generation*

This step mainly utilizes scalar multiplication and hash-to-curve operations. The algorithm takes the master key MK and the set of attributes A as inputs. It proceeds to calculate the secret key SK, which is associated with the attribute set A. The algorithm selects a random value $\gamma \in \mathbb{Z}_r$, and for each attribute $i \in A$, it selects a random value $\gamma_i \in \mathbb{Z}_r$. A hash function $\mathcal{H} = \{0,1\}^* \rightarrow \mathbb{G}$ is utilized. Subsequently, the secret key SK is computed as [2]:

$$SK = (D, \{D_i, D_i'\}_{\forall i \in A}), \tag{21}$$

where:

- $D = (\alpha + \gamma)\beta^{-1} g,$
- $D_i = \gamma g + \gamma_i \mathcal{H}(i),$
- $D_i' = \gamma_i g.$

3) *Encryption*

It mainly employs scalar multiplication and hash-to-curve operations. The data is encrypted using the tree structure policy $\mathcal{T}$. The function $\text{ind}(t)$ returns the value for node t, while the function $\text{par}(t)$ returns the parent node of t in the tree. For each node t, a polynomial q_t is chosen. The process chooses a random value $s \in \mathbb{Z}_r$, starting with the R node, setting $q_R(0) = s$. Then, for every $t \in \mathcal{T}, q_t(0) = q_{\text{par}(t)}(\text{ind}(t))$. The leaf nodes in $\mathcal{T}$ are denoted as $\mathcal{L}$, and the function $\text{att}(t)$ provides the attribute value of each node in $\mathcal{T}$. The message is encrypted using the access policy $\mathcal{T}$, as follows [2]:

$$CT = \big(\mathcal{T}, \tilde{C}, C, \{C_l, C_l'\}_{\forall l \in \mathcal{L}}\big), \tag{22}$$

where:

- $\tilde{C} = M v^s,$
- $C = sh,$
- $C_l = q_l(0) g,$
- $C_l' = q_l(0) \mathcal{H}(\text{att}(l)).$

4) Decryption

In this part, pairing and multiplication operations are predominantly utilized. The decryption process involves taking the secret key SK and the ciphertext CT, then computing the plaintext M. The algorithm computes $\text{dec\_node}(CT, SK, t)$, which receives CT, SK, and node t as inputs. If t is a leaf node, the attribute of node t is obtained as $i = \text{att}(t)$. Then, $\text{dec\_node}(CT, SK, t)$ is computed as [2]:

$$\text{dec\_node}(CT, SK, t) = \begin{cases} \frac{e(D_i, C_t)}{e(D_i', C_t')} & \text{if } i \in A, \\ null & \text{if } i \notin A. \end{cases} \tag{23}$$

In this step, a pair of pairing functions is executed for each attribute. The algorithm also requires a transformation between asymmetric and symmetric pairing, as described in Fig. 2 and Eq. (19).

The $\text{dec\_node}(CT, SK, t)$ function operates on leafless node t as follows: For each child node c of t, the algorithm calls $\text{dec\_node}(CT, SK, c)$ and stores the result in F_c. A_t is a list of nodes c, where $F_c \neq null.$ If no such set exists, the function returns $null$. Otherwise, the following calculation is performed [2]:

Let: $k = \text{ind}(c)$, $A_t' = \{\text{ind}(c), \forall c \in A_t\}$,

$$\Delta_{k, A_t'(0)} = \prod_{j \in A_t', j \neq k} \frac{-j}{k-j}. \tag{24}$$

$$\begin{aligned} F_t &= \prod_{c \in A_t} F_c^{\Delta_{k, A_t'(0)}} \\ &= \prod_{c \in A_t} [e(g, g)^{\gamma q_c(0)}]^{\Delta_{k, A_t'(0)}} \\ &= \prod_{c \in A_t} [e(g, g)^{\gamma q_{\text{par}(c)}(\text{ind}(c))}]^{\Delta_{k, A_t'(0)}} \\ &= \prod_{c \in A_t} e(g, g)^{\gamma q_{\text{t}}(\text{k}) \Delta_{k, A_t'(0)}} \\ &= e(g, g)^{\gamma q_{\text{t}}(0)}. \end{aligned} \tag{25}$$

To decrypt the data, the algorithm first calls the $\text{dec\_node}(CT, SK, R)$ function. If the attributes A match the tree access structure $\mathcal{T}$, we set [2]:

$$\begin{aligned} \tilde{A} &= \text{dec\_node}(CT, SK, R) \\ &= e(g, g)^{\gamma q_R(0)} \\ &= e(g, g)^{\gamma s}. \end{aligned} \tag{26}$$

The encrypted data is decrypted using the following Eq. (27) [2]:

$$\frac{\tilde{C} \cdot \tilde{A}}{e(C, D)} = M. \tag{27}$$

III. Proposed Methods

The objective of this paper is to reduce decryption processing time of ELiPS-Based CP-ABE. To accomplish this goal, the authors suggest:

- The decryption Eq. (23) is transformed to Eq. (28), which includes a Miller loop and final exponentiation bases.
- The number of final exponentiation operations is proportional to the number of attributes.
- The final exponentiation is minimized to be executed only once at the last step.
- The number of inversion operations in the Lagrange coefficient part is reduced by using one inversion operation.

A. Minimizing the Number of Final Exponentiations Method

For data decryption, in accordance with Eqs. (23) and (25), the algorithm conducts a pair of pairing calculations for each attribute. The pairing operation includes Miller loop and final exponentiation. Therefore, we propose to transform the decryption equation to Miller loop and final exponentiation as follows:

$$\begin{aligned} &\prod_{i=1}^{n} \left[\frac{e(D_i, C_i)}{e(D_i', C_i')}\right]^{\Delta_i} \\ &= \prod_{i=1}^{n} \left[\frac{\left(f_{D_i, C_i}\right)^{\frac{p^{k_{-1}}}{r}}}{\left(f_{D_i', C_i'}\right)^{\frac{p^{k_{-1}}}{r}}}\right]^{\Delta_i}, \end{aligned} \tag{28}$$

where:

- n is the number of attributes.
- $f_{P,Q}$ be a Miller loop result with P and Q on elliptic curve as inputs.
- Δ_i is the Lagrange coefficient, $\Delta_i \in \mathbb{Z}_r$,

$$\Delta_i = \sum_{j=1, j \neq i}^{n} \frac{-j}{i-j} = \sum_{j=1, j \neq i}^{n} -j(i-j)^{-1}, \tag{29}$$

- $(i-j)^{-1}$ is the inverse of $i - j$ over $\mathbb{Z}_r$.

The pairing e is formed through the Miller loop and final exponentiation. Thus, in Eq. (28), both the Miller loop and final exponentiation are utilized for each pairing per attribute.

Therefore, the authors propose a formula aiming to decrease the number of final exponentiations. From Eq. (28), the authors propose a formula transformation as follows:

$$\begin{aligned} &\prod_{i=1}^{n} \left[\frac{e(D_i, C_i)}{e(D_i', C_i')}\right]^{\Delta_i} \\ &= \prod_{i=1}^{n} \left[\frac{\left(f_{D_i, C_i}\right)^{\frac{p^{k_{-1}}}{r}}}{\left(f_{D_i', C_i'}\right)^{\frac{p^{k_{-1}}}{r}}}\right]^{\Delta_i} \\ &= \left[\prod_{i=1}^{n} \left(\frac{f_{D_i, C_i}}{f_{D_i', C_i'}}\right)^{\Delta_i}\right]^{\frac{p^{k_{-1}}}{r}}. \end{aligned} \tag{30}$$

In Eq. (30), it decreases the number of final exponentiations. It also employs the Miller loop for each pairing; however, it utilizes the final exponentiation only once at the end.

Consequently, Eqs. (28) and (30) demonstrate that our proposed method effectively reduces the number of final exponentiations by $2n-1$ times, improving the efficiency of ELiPS-Based CP-ABE.

B. Minimizing the Number of Inversions Method

The inversion operation is one of the operations that has an expensive calculation cost. However, the Lagrange coefficient in decryption part is calculated as follows:

$$\Delta_i = \sum_{j=1,j\neq i}^{n}[-j(i-j)^{-1}]. \tag{31}$$

According to Eq. (31), the inversion operation is used in the Lagrange coefficient part for every number of attributes. Consequently, in this paper, we propose a method to improve the efficiency of the decryption part by minimizing the number of inversions as follows:

(1) Calculate product:

$$\mathcal{A}_i = \prod_{j=1,j\neq i}^{n}(i-j). \tag{32}$$

(2) Then, calculate inverse of $\mathcal{A}_i$:

$$\mathcal{A}_i^{-1} = \frac{1}{\mathcal{A}_i}. \tag{34}$$

(3) Afterward, we can calculate the inversion as follows:

$$(i-j)^{-1} = \mathcal{A}_i^{-1}\prod_{k=1,k\neq i,k\neq j}^{n}(i-k). \tag{35}$$

The calculation in Eq. (35) decreases $n-1$ times inversion operations and increases $3(n-1)$ times multiplication operations. This algorithm is known as Montgomery's trick. However, the cost of multiplication is much lower than that of inversion. Therefore, this method is more effective than Eq. (31).

IV. Evaluation and Discussion

In this section, the authors present a brief discussion about the security analysis for the proposed scheme. Subsequently, we outline our experiment aimed at assessing the correctness of proposed formulas and their performance. Additionally, we compare the performance of our proposed decryption method in ELiPS-Based CP-ABE with that of previous work [2].

A. Security Analysis

1) Security level

According to Eqs. (22) and (27), to decrypt encrypted data, one needs to calculate the value of $e(g,g)^{\alpha s}$ or $e(C,D)/e(g,g)^{\gamma s}$.

- To recover value $e(g,g)^{\alpha s}$, attackers have to find α and s. However, based on DLP and ECDLP, computing α from $d=\alpha g$ or $v=e(g,g)^{\alpha}$ and s from $C=sh$ is infeasible.
- To calculate value $e(C,D)/e(g,g)^{\gamma s}$, adversaries can calculate $e(C,D)$ using C from the ciphertext and D from the user's secret key. However, the value of $e(g,g)^{\gamma s}$ remains blinded. Recovering γ from $D=(\alpha+\gamma)\beta^{-1}g$ and s from $C=sh$ are challenging problems, according to DLP and ECDLP. Another approach for attackers to recover $e(g,g)^{\gamma s}$ is through collusion attacks. Next, we present resistant colluding users' analysis in our proposal.

where $g,h \in E(\mathbb{F}_p), \alpha,\beta,\gamma,s \in \mathbb{Z}_r$. In our system, r has a 308-bit length, therefore, the proposed scheme remains at a 128-bit security level.

2) Collusion attack

According to Ref. [23], the primary challenge in the CP-ABE scheme is collusion attack. Based on the CP-ABE algorithm and Eq. (27), the attacker needs to recover two values such as $e(C,D)$ and $e(g,g)^{\gamma s}$. However, the attacker is still blinded by the value $e(g,g)^{\gamma s}$. This value can only be recovered if and only if enough the user has the secret key component to satisfy the access policy embedded in the ciphertext. As a result, collusion attacks are ineffective because the blinding value γ is randomized to the randomness from a particular user's secret key.

It is clear that, considering the security level, collusion attack analysis, Eqs. (30) and (35), the proposed scheme not only reduces the number of final exponentiations and inversions by $2n-1$ times and $n-1$ times, respectively, but also maintains the required security level and is resistant to potential attacks.

B. Decryption Cost Comparison

Here, we present a comparison of decryption costs among CP-ABE schemes, as shown in Table II. These data reveal that the decryption cost of our scheme reduces the number of final exponentiations and inversions by $2n-1$ times and $n-1$ times compared to Refs. [2] and [23]. When compared to Refs. [24] and [25], our scheme not only remains the number of inversions at constant 2 but also reduces the number of final exponentiations, consistently remaining at only 2 final exponentiations. Table II demonstrates that the proposed scheme reduces the number of final exponentiations and inversions to a constant of 2. Therefore, our scheme may be effective and competitive with other schemes.

TABLE II. Comparison of Decryption Cost among CP-ABE Schemes

Schemes	Inversion	Miller loop	Final exp
[23]	$n+1$	$2n+1$	$2n+1$
[24]	2	$n+2$	$n+2$
[25]	2	$\tau+2$	$\tau+2$
[2]	$n+1$	$2n+1$	$2n+1$
Proposal	2	$2n+1$	2

Note: n is the number of attributes, τ is the maximum number of multi-use [25].

C. Evaluation of the Proposed Formula, Reducing the Number of Final Exponentiations

Firstly, through experimentation, we assess the correctness and performance of the previous formula and our proposed formula, which reduces the number of final exponentiations. The author implemented Eqs. (28) and (31) to measure the execution time, progressively increasing the number of pairs pairing from 5 to 20. During the experiment, we used the devices and software as depicted in Table III.

TABLE III. EXPERIMENTAL ENVIRONMENTS

Devices and Software	Descriptions
OS	Ubuntu 22.04.1 LTS
CPU	Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz
Memory	4 GB
Language	C
GMP version	6.2.1
GCC version	11.3.0
GCC optimization level	-O2

We executed the calculations 10,000 times for each scenario to measure the computation time and then calculated the average values. The experimental results reveal that the outcome of Eq. (28) is identical to the result of Eq. (30). These results validate the correctness of our proposed formula. Moreover, as depicted in Fig. 3, our proposed Eq. (30) reduces the execution time by an average of 43.61% compared to the previous Eq. (28).

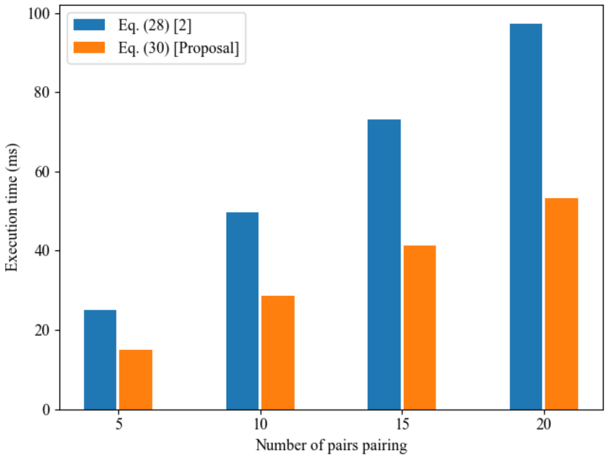

Fig. 3. Comparison of execution time between Eqs. (28) and (30).

D. *Evaluation of the Proposed Formula, Decreasing the Number of Inversions*

Secondly, the authors assess the correctness and performance of the method, which decreases the number of inversions. We implemented Eq. (31) and our method to measure the execution time, progressively increasing the number of variables from 5 to 20. We ran the experiment 10,000 times, then took the average execution time. The experimental results show that the inversion result of Eq. (31) is identical to the result of Eq. (35). These results demonstrate the correctness of our proposed method.

Furthermore, Fig. 4 illustrates our proposed Eq. (35) decreases the execution time by an average of 74.39% compared to the Eq. (31).

Subsequently, we successfully implemented our proposed method into ELiPS-Based CP-ABE. Additionally, we conducted several evaluations to compare our work with previous research [2].

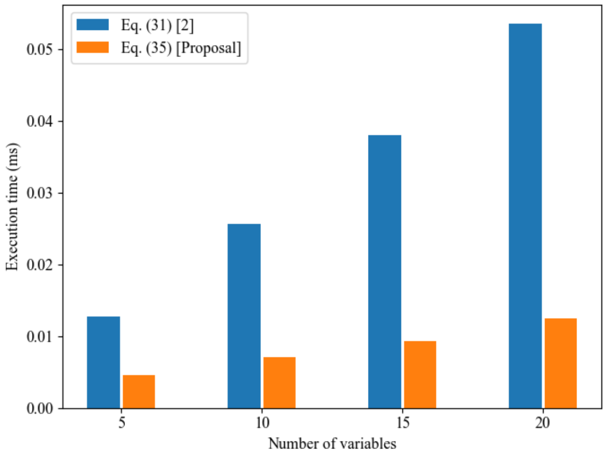

Fig. 4. Comparison of execution time between Eqs. (31) and (35).

E. *Evaluation of Decryption Performance with Our Proposed Methods*

We successfully employed our proposed methods into ELiPS-Based CP-ABE and implemented several scenarios, which increase the number of attributes from 5 to 100, to measure the decryption time. Then, the authors compare the decryption time of their proposed method with the previous work [2]. In this evaluation, we run setup, keygen, and encrypt functions once. However, we executed the decrypt function 10,000 times to measure the decryption time for each scenario, then calculated the average values.

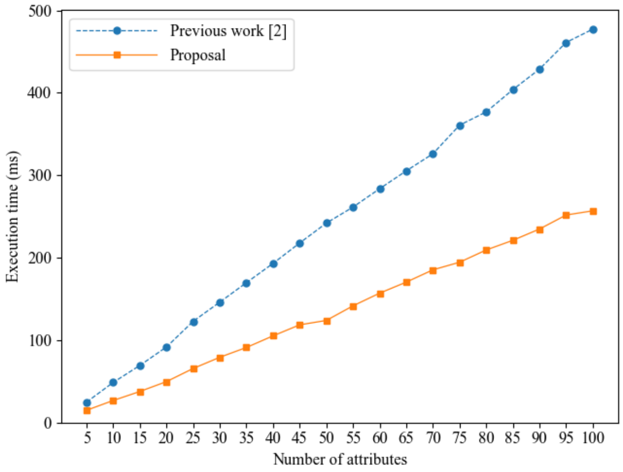

Fig. 5. Compare decryption time between previous work [2] and our work.

Fig. 5 gives information that our decryption performance is faster than the previous work [2] across various scenarios. Decryption time in our proposed scheme decreased by an average of 45.45%. In addition, Fig. 5 also shows that the decryption time in the proposed scheme is more efficient when the number of attributes increases and can effectively handle a large number of attributes. This is because our proposed formula decreases the number of final exponentiations by $2n - 1$ times and the number of inversions by $n - 1$ times, where n is the number of attributes.

V. Conclusion

The authors proposed methods that minimize the number of final exponentiations and inversions to reduce the decryption processing time of ELiPS-Based CP-ABE. The proposed formula effectively decreased the number of final exponentiations and inversions by $2n-1$ times and $n-1$ times, respectively. Experimental results demonstrate that our scheme decreases the decryption time by an average of 45.45% compared to previous work. Our system has been successfully implemented within the CP-ABE framework, making it applicable in practical applications. Future developments will further improve the decryption cost of ELiPS-Based CP-ABE. Additionally, we will try to integrate our proposal into the blockchain system.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Le Hoang Anh conducted the research. Yasuyuki Nogami and Samsul Huda supervised the entire research development. Yuta Kawada provided critical feedback on the mathematical background. Le Hoang Anh wrote the manuscript. Yuta Kawada, Samsul Huda, Md. Arshad Ali, Yuta Kodera, and Yasuyuki Nogami reviewed the manuscript and offered critical feedback. All authors have approved the final version.

References

[1] V. C. Hu, "Overview and considerations of access control based on attribute encryption," *NIST*, pp. 1–41, 2023. doi: 10.6028/NIST.IR.8450-upd1

[2] L. H. Anh, Y. Kawada, S. Huda *et al.*, "An implementation of ELiPS-based ciphertext-policy attribute-based encryption," in *Proc. 2023 Eleventh International Symposium on Computing and Networking Workshops (CANDARW)*, Matsue, Japan, 2023, pp. 220–226. doi: 10.1109/CANDARW60564.2023.00044

[3] T. P. Ezhilarasi, N. S. Kumar, T. P. Latchoumi *et al.*, "A secure data sharing using IDSS CP-ABE in cloud storage," in *Advances in Industrial Automation and Smart Manufacturing*, Singapore: Springer, 2021, pp. 1073–1085. doi: 10.1007/978-981-15-4739-3_92

[4] Y. W. Hwang and I. Y. Lee, "A study on lightweight anonymous CP-ABE access control for secure data protection in cloud environment," in *Proc. the 2019 International Conference on Information Technology and Computer Communications (ITCC'19)*, USA, 2019, pp. 107–111. doi: 10.1145/3355402.3355405

[5] Y. Zhang, R. H. Deng, S. Xu *et al.*, "Attribute-based encryption for cloud computing access control: A survey," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–41, 2020. doi: 10.1145/3398036

[6] Y. W. Hwang and I. Y. Lee, "A study on CP-ABE-based medical data sharing system with key abuse prevention and verifiable outsourcing in the IoMT environment," *Sensors*, vol. 20, no. 17, pp. 1–23, 2020. doi: 10.3390/s20174934

[7] B. Ying, N. R. Mohsen, and A. Nayak, "Efficient authentication protocol for continuous monitoring in medical sensor networks," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 130–138, 2021. doi: 10.1109/OJCS.2021.3055365

[8] H. Y. Lin and Y. R. Jiang, "A multi-user ciphertext policy attribute-based encryption scheme with keyword search for medical cloud system," *Applied Sciences*, vol. 11, no. 1, pp. 1–14, 2020. doi: 10.3390/app11010063

[9] R. Cheng, K. Wu, Y. Su *et al.*, "An efficient ECC-based CP-ABE scheme for power IoT," *Processes 2021*, vol. 9, no. 1176, pp. 1–16, 2021. doi: 10.3390/pr9071176

[10] B. Girgenti, P. Perazzo, C. Vallati *et al.*, "On the feasibility of attribute-based encryption on constrained IoT devices for smart systems," in *Proc. 2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, Washington, DC, USA, 2019, pp. 225–232. doi: 10.1109/SMARTCOMP.2019.00057

[11] P. Perazzo, F. Righetti, M. L. Manna, and C. Vallati, "Performance evaluation of attribute-based encryption on constrained IoT devices," *Computer Communications*, vol. 170, pp. 151–163, 2021. doi: 10.1016/j.comcom.2021.02.012

[12] D. Ziegler, J. Sabongui, and G. Palfinger, "Fine-grained access control in industrial internet of things," *IFIP Advances in Information and Communication Technology*, Springer, vol. 562, pp. 91–104, 2019. doi: 10.1007/978-3-030-22312-0_7

[13] T. Hu, S. Yang, Y. Wang *et al.*, "N-accesses: A blockchain-based access control framework for secure IoT data management," *Sensors*, vol. 23, no. 20, pp. 1–17, 2023. doi: 10.3390/s23208535

[14] G. Zhang, X. Chen, L. Zhang, *et al.*, "STAIBT: Blockchain and CP-ABE empowered secure and trusted agricultural IoT blockchain terminal," *International Journal of Interactive Multimedia and Artificial Intelligence*, pp. 66–75, 2022. doi: 10.9781/ijimai.2022.07.004

[15] R. Hu, Z. Ma, L. Li *et al.*, "An access control scheme based on blockchain and ciphertext policy-attribute based encryption," *Sensors*, vol. 23, no. 19, 2023. doi: 10.3390/s23198038

[16] Y. Zhao, H. Li, Z. Liu *et al.*, "A lightweight CP-ABE scheme in the IEEEP1363 standard with key tracing and verification and its application on the Internet of Vehicles," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 7, 2023. doi: 10.1002/ett.4774

[17] L. Meng, H. Xu, R. Tang *et al.*, "Dual hybrid CP-ABE: How to provide forward security without a trusted authority in vehicular opportunistic computing," *IEEE Internet of Things Journal*, 2023. doi: 10.1109/JIOT.2023.3321563

[18] K. Sowjanya and M. Dasgupta, "A ciphertext-policy attribute based encryption scheme for wireless body area networks based on ECC," *Journal of Information Security and Applications*, vol. 54, 2020. doi: 10.1016/j.jisa.2020.102559

[19] F. Meng, L. Cheng, and M. Wang, "Ciphertext-policy attribute-based encryption with hidden sensitive policy from keyword search techniques in smart city," *J. Wireless Com. Network*, pp. 1–22, 2021. doi: 10.1186/s13638-020-01875-2

[20] B. Chandrasekaran, R. Balakrishnan, and Y. Nogami, "TF-CPABE: An efficient and secure data communication with policy updating in wireless body area networks," *ETRI Journal*, vol. 41, no. 4, pp. 465-472, 2019. doi: 10.4218/etrij.2018-0320

[21] Y. Nanjo, M. Shirase, Y. Kodera *et al.*, "Efficient final exponentiation for cyclotomic families of pairing-friendly elliptic curves with any prime embedding degrees," *International Journal of Networking and Computing*, vol. 12, no. 2, pp. 317–338, 2022.

[22] D. Hattori, Y. Takahashi, T. Tatara *et al.*, "An optimal curve parameters for BLS12 elliptic curve pairing and its efficiency evaluation," in *Proc. 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Penghu, Taiwan, 2021, pp. 1–2. doi: 10.1109/ICCE-TW52618.2021.9602941

[23] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. 2007 IEEE Symposium on Security and Privacy (SP'07)*, Berkeley, CA, USA, 2007, pp. 321–334. doi: 10.1109/SP.2007.11

[24] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," *Public Key Cryptography*, vol. 6571, pp. 53–70, 2017. doi: 10.1007/978-3-642-19379-8_4

[25] D. Riepel and H. Wee, "FABEO: Fast attribute-based encryption with optimal security," in *Proc. the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*, New York, NY, USA, 2022, pp. 2491–2504. doi: 10.1145/3548606.3560699