ELiPS-based Ciphertext-Policy Attribute-Based Encryption

September, 2024

Le Hoang Anh

Graduate School of
Natural Science and Technology
(Doctor's Course)

OKAYAMA UNIVERSITY

DOCTORAL THESIS

ELiPS-based Ciphertext-Policy Attribute-Based Encryption

Author:	Le Hoang ANH
Supervisor:	Yasuyuki NOGAMI
Co-supervisors:	Yoshitaka TOYOTA
	Yukinobu FUKUSHIMA

A dissertation submitted to
OKAYAMA UNIVERSITY
in fulfillment of the requirements for the degree of
Doctor of Philosophy in Engineering
in the
Graduate School of Natural Science and Technology

September, 2024

Declaration Authorship

This dissertation and the work presented here for doctoral studies were conducted under the supervision of Professor Yasuyuki Nogami. I, Le Hoang Anh, declare that this thesis titled, "**ELiPS-based Ciphertext-Policy Attribute-Based Encryption**" and the work presented in it are my own. I confirm that:

- The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, while enrolled in the Graduate School of Natural Science and Technology at Okayama University as a candidate for the degree of Doctor of Philosophy in Engineering.
- This work has not been submitted for a degree or any other qualification at this University or any other institution.
- The published work of others cited in this thesis is clearly attributed. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help to pursue this work.
- The experiments and results presented in this thesis and in the articles where I am the first author were conducted by myself.

Signed: _____

Date: _____

Abstract

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is an advanced cryptographic technique that enhances the flexibility and security of access control in data encryption. Unlike traditional encryption methods where access is determined by the possession of a single key, CP-ABE enables access based on a user's attributes, providing a more fine-grained and expressive approach to data security.

The CP-ABE scheme operates through four main functions such as setup, key generation, encryption, and decryption. In the setup function, the algorithm generates a master key and a public key. The public key is distributed to users, while the master key is kept secret. The master key and public key are then used to create secret keys for users based on their attributes. These secret keys enable authorized users to decrypt ciphertexts that adhere to specified access policies, ensuring fine-grained access control over encrypted data.

In CP-ABE, data is encrypted under an access policy specified by the data owner. Access to the encrypted data is granted only if the user's attributes satisfy the access policy embedded in the ciphertext. This approach integrates the encryption and access control processes, ensuring that only authorized users can decrypt the data.

Therefore, CP-ABE is not only to encrypt data but also to provide fine-grained access control over encrypted data. CP-ABE is a powerful cryptographic tool for keeping data safe in places like cloud storage, the Internet of Things (IoT), personal health records, and blockchain, using pairing-based cryptography.

Cloud computing enables the storage and remote access of data via the internet. However, issues with access control and privacy arise when data is stored by a third party. On the other hand, IoT is a rapidly developing technology in the modern digital era. The large amounts of data generated by the expanding IoT have led to a greater focus on privacy and data access control in security. To meet these requirements, CP-ABE is utilized to provide privacy and fine-grained access control in both cloud storage and IoT applications.

Despite CP-ABE has various important applications, the original CP-ABE scheme relies on the pairing-based cryptography (PBC) library. The PBC library is an open-source library carrying out the essential mathematical

operations in pairing-based cryptosystems. Speed and portability are crucial considerations as the PBC library is intended to serve as the foundation for pairing-based cryptosystem implementations. It offers functions like elliptic curve arithmetic, hash-to-curve, and pairing. The PBC utilizes symmetric pairing, which offers a security level limited to 80 bits. This level of security is now considered outdated and vulnerable to various attacks, failing to meet the current demands for high-level security.

The Efficient Library for Pairing Systems (ELiPS), on the other hand, offers efficient operations related to pairing-based cryptography, delivering high performance while upholding a substantial security standard. Such cryptography involves mathematical pairings between points on an elliptic curve. The ELiPS library offers a range of functionalities, including point arithmetic operations, exponentiation, hash-to-curve, and pairing. ELiPS is specifically designed to support bilinear pairing using the BLS-12 curve, providing a 128-bit security level.

In our first study, to deal with the shortcomings of the original CP-ABE, we adopt and implement the ELiPS as an efficient library for pairing systems into the CP-ABE framework, namely ELiPS-based CP-ABE. However, the integration process is not straightforward due to differences between PBC and ELiPS libraries, including function parameters, data types, and the type of pairing. Notably, ELiPS supports asymmetric pairing, while the original CP-ABE relies on symmetric pairing. To bridge this gap and ensure compatibility, we designed three procedures to adapt ELiPS for CP-ABE. Our approach begins with the generation of a generator g. Then, we utilize Shirase's method to transform asymmetric pairing to symmetric pairing, establishing compatibility between ELiPS and CP-ABE. Subsequently, we make several modifications to the CP-ABE framework and choose the appropriate ELiPS functions for integration.

Afterward, we validate our proposal through several experiments involving data access authorization scenarios. Firstly, we evaluate the efficacy of setup, key generation, encryption, and decryption in PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE with a two-attribute scenario. The results show that the setup time in ELiPS-based CP-ABE reduces by 26.8% and in MCL-based CP-ABE decreases by 28.6% compared to PBC-

based CP-ABE. In addition, the key generation time in MCL-based CP-ABE is lower than that in PBC-based CP-ABE by 74.8%, while in ELiPS-based CP-ABE, it is lower than other schemes by 2.6% compared to MCL-based CP-ABE and by 75.5% compared to PBC-based CP-ABE. Moreover, the results confirm that the encryption time in ELiPS-based CP-ABE is the lowest among the three versions, namely PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE. Whereas encryption time in MCL-based CP-ABE decreases by 74.0%, encryption time in ELiPS-based CP-ABE reduces by 75.3% compared to that in PBC-based CP-ABE. On the other hand, the decryption time for MCL-based CP-ABE and ELiPS-based CP-ABE increases by 31.5% and 50.7%, respectively, compared to the decryption time for PBC-based CP-ABE. Hence, further evaluation with increasing the number of attributes is necessary. Secondly, since the setup part is not affected by the number of attributes, we do not need to evaluate it further. Instead, we focus on experiments and evaluations of key generation, encryption, and decryption with the numbers of attributes ranging from 2 to 20. The experimental results depict the key generation time in MCL-based CP-ABE is lower than that in PBC-based CP-ABE by 74.7%, while in ELiPS-based CP-ABE, it is lower than other schemes by 3.7% compared to MCL-based CP-ABE and by 75.6% compared to PBC-based CP-ABE. Encryption time in ELiPS-based CP-ABE is the lowest among the three versions. Encryption time in ELiPS-based CP-ABE decreases by 75.0% compared to that in PBC-based CP-ABE and reduces by 4.9% compared to that in MCL-based CP-ABE. The decryption time of both MCL-based CP-ABE and ELiPS-based CP-ABE is higher than that of the PBC-based CP-ABE across scenarios.

Overall, the experimental results confirm that our ELiPS-based CP-ABE performs comparably to the competitive MCL library, showcasing its efficiency and effectiveness in modern cryptographic applications. Additionally, compared to PBC-based CP-ABE, our ELiPS-based solution demonstrates reduced computational costs across most functions, except for decryption. Therefore, in the next study, we aim to reduce the decryption process time in ELiPS-based CP-ABE. In ELiPS-based CP-ABE, 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. Performing these operations can be very expensive, especially when the number of attributes is large.

In our second study, we further explore reducing the decryption process time in the initial version of ELiPS-based CP-ABE by proposing two optimization methods, such as minimizing the number of final exponentiations and inversions. The decryption cost comparison shows that our methods reduce the number of final exponentiations from $2n+1$ to 2 and the number of inversions from $n+1$ to 2. The experimental results show that the equation with minimizing the number of final exponentiations reduces the execution time by an average of 43.6% compared to the original equation, and our proposed equation with minimizing the number of inversions decreases the execution time by an average of 74.4% compared to the equation without minimizing the number of inversions. In addition, we already successfully integrated these minimization methods into the ELiPS-based CP-ABE and implemented several scenarios, which increase the number of attributes from 5 to 100, to measure the decryption time. The effectiveness of the proposal is confirmed through experimental analyses where the decryption time in the ELiPS-based with these optimizations decreased by an average of 45.5% compared to the initial version of ELiPS-based CP-ABE.

In our third study, we further evaluate and analyze the impact of these optimizations on decryption efficiency. Moreover, we compare the ELiPS-based CP-ABE with these improvements to the initial version of ELiPS-based CP-ABE and the original PBC-based CP-ABE. As a result, the combination of both optimization techniques resulted in an average 43.1% overall reduction in decryption time compared to the initial version of the ELiPS-based CP-ABE scheme, while in total execution, it led to a 25.3% improvement. Furthermore, our optimized construction also outperformed the original PBC-based CP-ABE by an average of 53.8%, while providing a higher 128-bit security level.

Our research demonstrates that integrating the ELiPS library into the CP-ABE framework significantly enhances the efficiency and security of the CP-ABE scheme. By implementing optimization techniques, we further reduced computational costs, particularly during the decryption process. This makes ELiPS-based CP-ABE a highly viable option for modern cryptographic applications, providing robust security and efficient performance.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Yasuyuki Nogami, for his support throughout my doctoral courses at Okayama University. Without his extraordinary understanding and cooperation, I would not have been able to complete my doctoral research. I also appreciate my co-supervisors, Professor Yoshitaka Toyota, and Associate Professor Yukinobu Fukushima, who gave me a lot of effort to improve this thesis. They also gave me knowledge of electronics and networks through the classes in my courses.

I would like to express my gratitude to Specially Appointed Assistant Professor Samsul Huda at the Green Innovation Center, Okayama University, for the in-depth discussion of scientific topics. His strong work ethic and passion for research helped us publish some remarkable collaborative works. He was always there to help while any difficulty arose from attending conferences to publishing papers.

I sincerely thank Assistant Professor Yuta Kodera who is my respected senior and gave me a lot of influence from his great attitude for research. I also appreciate other teachers who have imparted a lot of knowledge to me through the classes in my courses.

Special thanks also to student members of the Information Security Laboratory (Nogami Lab.) for creating a great work atmosphere and their generous support. My special thanks to Mr. Yuta Kawada for his kind support in discussions, cooperation, and publishing papers.

Thanks to MEXT, Japan for the scholarship that fulfilled my dream to pursue doctoral study in Japan. I sincerely acknowledge all the funds that afforded me to join several conferences and conduct research activities.

I am also grateful to all administrative officers of the Faculty of Engineering who directly or indirectly made an impact on my doctoral course studies. My special thanks to Ms. Yuri Kunisada and Ms. Yumi Sato for their kind support in administrative work.

Last but not least, I cannot thank my wife enough for her sacrifices and support. I would also like to express my heartfelt thanks to my parents, my wife, and my little daughter for allowing me to learn in the doctoral course. They gave me much encouragement, which motivated me to carry out research.

Publications

Peer-Reviewed Journal Papers:

1. **L. H. Anh**, Y. Kawada, S. Huda, M. A. Ali, Y. Kodera, and Y. Nogami, "ELiPS-based Ciphertext-Policy Attribute-Based Encryption," *International Journal of Networking and Computing (IJNC),* vol. 14, no. 2, pp. 186-205, 2024, doi: 10.15803/ijnc.14.2_186.

2. **L. H. Anh**, Y. Kawada, S. Huda, M. A. Ali, Y. Kodera, and Y. Nogami, "A Minimization Number of Final Exponentiations and Inversions for Reducing the Decryption Process Time in ELiPS-Based CP-ABE," *Journal of Advances in Information Technology (JAIT),* vol. 15, no. 6, pp. 748–755, 2024, doi: 10.12720/jait.15.6.748-755.

Peer-Reviewed International Conference Papers (First author):

3. **L. H. Anh**, Y. Kawada, S. Huda, M. A. Ali, Y. Kodera, and Y. Nogami, "An implementation of ELiPS-based Ciphertext-Policy Attribute-Based Encryption," *Eleventh International Symposium on Computing and Networking Workshops (CANDARW 2023),* Matsue, Japan, pp. 220–226, 2023, doi: 10.1109/CANDARW60564.2023.00044.

4. **L. H. Anh**, Y. Kawada, S. Huda, Y. Kodera, and Y. Nogami, "Performance Analysis of ELiPS-based CP-ABE with Optimized Decryption Functions," *8th International Conference on Information and Communication Technology for Intelligent Systems (ICTIS 2024),* Las Vegas, USA, 2024.

Peer-Reviewed International Conference Papers (Co-author):

5. S. Huda, Y. Nogami, T. Akada, M. Rahayu, Md. B. Hossain, M. B. Musthafa, **L. H. Anh**, and Y. Jie, "A Proposal of IoT Application for Plant Monitoring System with AWS Cloud Service," *2023 International Conference on Smart Applications, Communications and Networking (SmartNets),* Istanbul, Turkiye, 2023, pp. 1–5, doi: 10.1109/SmartNets58706.2023.10215620.

6. S. Huda, Y. Nogami, Md. B. Hossain, Y. Jie, **L. H. Anh**, M. B. Musthafa, M. Rahayu, and T. Akada, "A Secure Authentication for Plant Monitoring System Sensor Data Access," *2024 IEEE International Conference on Consumer Electronics (ICCE),* Las Vegas, NV, USA, 2024, pp. 1–2, doi: 10.1109/ICCE59016.2024.10444465.

Table of Contents

List of Figures

List of Tables

Notations and Abbreviations

E	Elliptic curve
e	Pairing
$E(\mathbb{F}_q)$	$\mathbb{F}_q$-rational point groups of E
$E/\mathbb{F}_q$	Elliptic curve defined over $\mathbb{F}_q$
$E[r]$	r-torsion subgroup of E
E'	Twist of E
e_{asy}	Asymmetric pairing
e_{sym}	Symmetric pairing
F	Field
$\mathbb{F}_p$	Prime field of order p
$\mathbb{F}_q$	Finite field of order q
G	Group
$\mathcal{O}$	Point at infinity
p	Prime
q	Prime or power of prime
r	Prime order of a subgroup of $E(\mathbb{F}_q)$
R	Ring
$\mathbb{Z}$	Set of all integers
π_p	p-th power Frobenius map in E or $\mathbb{F}_p$
3DES	Triple DES
AES	Advanced Encryption Standard
BLS	Barreto-Lynn-Scott
CP-ABE	Ciphertext-Policy Attribute-Based Encryption

CT	Ciphertext
DDH	Decision Diffie-Hellman
DES	Data Encryption Standard
DLP	Discrete Logarithm Problem
ECA	Elliptic Curve Addition
ECC	Elliptic Curve Cryptography
ECD	Elliptic Curve Doubling
ECDH	Elliptic Curve Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ELiPS	Efficiency Library for Pairing Systems
IBE	Identity-Based Encryption
IoT	Internet of Things
KP-ABE	Key-Policy Attribute-Based Encryption
MK	Master key
NIST	National Institute of Standards and Technology
PBC	Pairing-Based Cryptography
PK	Public key
RELIC	Efficient LIbrary for Cryptography
RSA	Rivest-Shamir-Adleman
SCM	Elliptic curve Scalar Multiplication
SK	Secret key
SSL	Secure Sockets Layer
TLS	Transport Layer Security
zk-SNARKs	zero-knowledge Succinct Non-interactive ARgument of Knowledge

Chapter 1

Introduction

This chapter presents the literature review, outlines the problems, and highlights the contributions of this work.

1.1 Cryptography

Cryptography is a crucial field in modern computing and communications, providing essential security measures for protecting sensitive information. Cryptography is vital and sets the foundation for understanding its significance in various applications [1, 2].

- **Data confidentiality:** Cryptography ensures that information is accessible only to those authorized to view it. Through encryption, readable data (plaintext) is transformed into an unreadable format (ciphertext), which can only be deciphered by those possessing the appropriate key. This process safeguards sensitive data from unauthorized access [2].
- **Data integrity:** Ensuring that data remains unaltered during storage or transmission is critical. Cryptographic techniques, such as hashing, generate a unique fixed-size string from data. This enables verification that the original data has not been tampered with, thereby maintaining its integrity [2].
- **Authentication:** Authentication is a fundamental aspect of cryptography, verifying the identities of users and devices. Digital signatures and certificates are used to confirm that a message or document originates from a trusted source and has not been forged. This process is essential for establishing trust in digital interactions [2].
- **Non-repudiation:** Non-repudiation prevents entities from denying their actions. Through digital signatures, cryptography provides proof of the origin and integrity of a message, ensuring that the sender cannot later

claim they did not send it. This is particularly important in legal and financial transactions [3].

- **Secure communication:** Cryptography enables secure communication over insecure channels. Protocols such as SSL/TLS use encryption to protect data transmitted over the internet, ensuring privacy and security for activities like online banking and shopping. This secure communication is vital for maintaining trust in digital ecosystems [2].

- **Secure storage:** Protecting data at rest is another critical function of cryptography. By encrypting files and databases, cryptography safeguards data from unauthorized access and breaches, ensuring that sensitive information remains confidential even when stored [2].

- **Key management:** Effective cryptography involves the secure generation, distribution, and storage of keys. Proper key management is vital for maintaining the security of cryptographic systems, as the strength of encryption directly depends on the secrecy and integrity of the cryptographic keys [4].

- **Financial transactions:** Cryptography secures online transactions and digital currencies. Protocols like SSL/TLS and technologies like blockchain rely on cryptographic principles to ensure the security and integrity of financial exchanges. This protection is fundamental for the functioning of modern financial systems [4].

- **Protecting national security:** National security relies heavily on cryptography to safeguard military communications, government data, and critical infrastructure from cyber threats and espionage. The robustness of cryptographic systems is therefore a cornerstone of national defense strategies [2].

Cryptography is essential for ensuring the confidentiality, integrity, and authenticity of information. Its role in underpinning the security of digital communications and data is indispensable across various applications, from personal privacy to national security. There are many cryptographic algorithms, such as AES, ECC, CP-ABE, etc. They are divided into two cate-

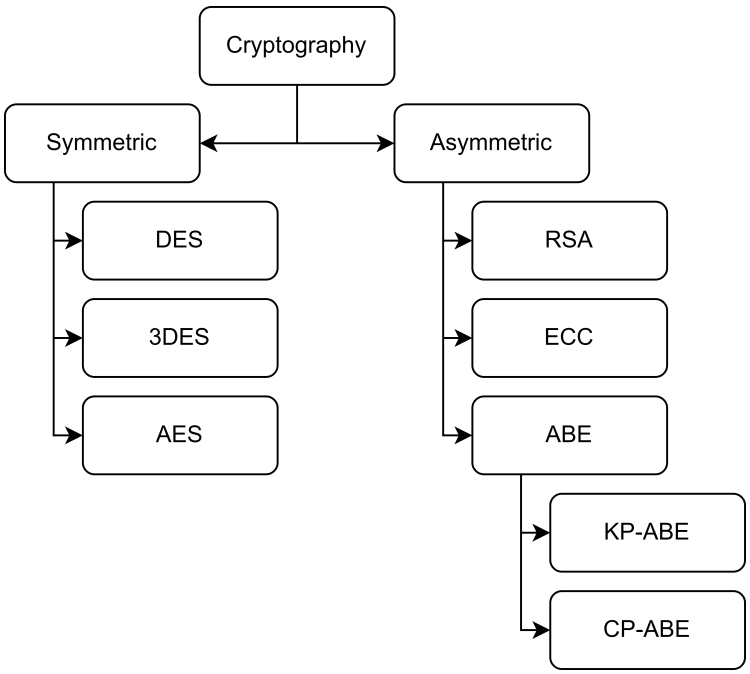


Figure 1.1: The classification of some primary encryption algorithms.

gories: symmetric-key cryptography and asymmetric-key cryptography [1], as shown in Figure 1.1.

1.1.1 Symmetric-key cryptography

Symmetric-key cryptography, also known as secret-key or private-key cryptography, is a fundamental category of cryptographic systems that uses a single key for both encryption and decryption of data [1]. This type of cryptography is known for its efficiency and speed, making it particularly suitable for applications requiring high throughput and low latency.

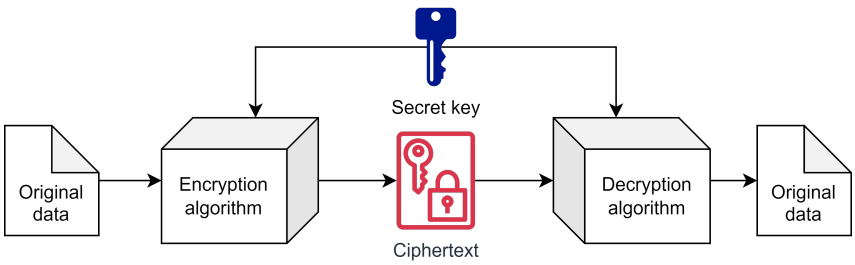


Figure 1.2: The process of symmetric-key cryptography.

In symmetric-key cryptography, the same key is employed to encrypt and decrypt the data, as shown in Figure 1.2. The security of the system relies on the secrecy of the key, which must be shared between the communicating parties through a secure channel. The primary advantage of symmetric-key cryptography is its computational efficiency, as the algorithms are typically faster and require less computational power compared to other types of cryptography.

Several well-known symmetric-key algorithms are widely used today, each with distinct characteristics and applications:

- **Data Encryption Standard (DES):** DES was one of the earliest symmetric-key algorithms standardized by the U.S. National Institute of Standards and Technology (NIST). It uses a 56-bit key. Due to its relatively short key length, DES is now considered insecure against brute-force attacks and has been largely replaced by AES [2]. Despite its obsolescence, DES played a crucial role in the development and popularization of cryptographic standards.

- **Triple DES (3DES):** Triple DES enhances the security of DES by applying the DES algorithm three times with three different keys, effectively increasing the key length to 168 bits. While more secure than DES, 3DES is slower and has been largely superseded by AES in new implementations [2].

- **Advanced Encryption Standard (AES):** AES is a symmetric encryption algorithm established as a standard by NIST in 2001. AES supports key sizes of 128, 192, and 256 bits, providing a robust level of security [2]. AES is extensively used in a variety of applications, including secure communications, financial transactions, and data storage.

Symmetric-key algorithms are generally faster and require less computational power, making them suitable for high-speed encryption and decryption. The conceptual simplicity of using a single key for both encryption and decryption simplifies the implementation and reduces the computational overhead [1, 3].

However, the primary challenge of symmetric-key cryptography is the secure distribution of the key. Both parties must share the key through a secure channel, which can be complex and impractical, especially in large-scale systems. As the number of participants increases, the number of keys required for secure communication grows exponentially, complicating key management [1, 3].

Symmetric-key cryptography remains a fundamental component of modern cryptographic systems due to its efficiency and speed. While it presents challenges in key distribution and scalability, its advantages make it indispensable for numerous applications requiring secure, high-speed data encryption [1, 3]. The next section will explore asymmetric-key cryptography, which addresses some of the limitations inherent in symmetric-key systems.

1.1.2 Asymmetric-key cryptography

Asymmetric-key cryptography, also known as public-key cryptography, is a pivotal branch of cryptographic systems that utilizes a pair of keys: a public key for encryption and a private key for decryption, as shown in Figure 1.3. This innovative approach addresses some of the inherent limitations of symmetric-key cryptography, particularly in the realms of key distribution and scalability [1, 3].

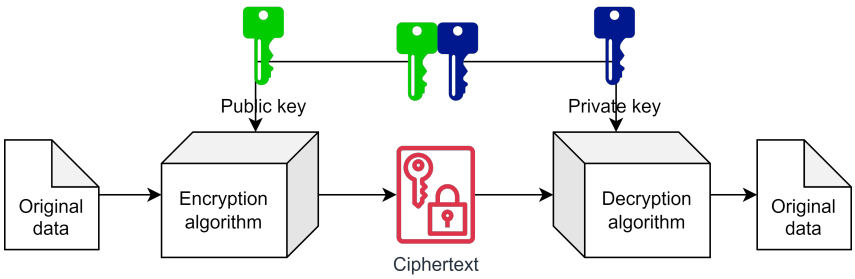


Figure 1.3: The process of asymmetric-key cryptography.

In asymmetric-key cryptography, each participant possesses a pair of keys:

- **Public key:** This key is shared openly and can be used by anyone to encrypt data intended for the key owner.

- **Private key:** This key is kept secret and is used by the key owner to decrypt encrypted data with their public key.

The security of asymmetric-key cryptography relies on the computational difficulty of deriving the private key from the public key. This property facilitates secure communication without the need for a pre-shared secret, overcoming the key distribution problem of symmetric-key cryptography.

Several asymmetric-key algorithms are prevalent in securing digital communications:

- **Rivest-Shamir-Adleman (RSA):** RSA, one of the earliest public-key cryptosystems, is based on the mathematical difficulty of factoring large composite numbers. RSA supports various key lengths, typically ranging from 1024 to 4096 bits, providing flexible security levels [1, 3]. Widely used for secure data transmission, digital signatures, and key exchange.

- **Elliptic Curve Cryptography (ECC):** ECC relies on the algebraic structure of elliptic curves over finite fields, offering comparable security to RSA but with much smaller key sizes. ECC keys are more efficient, providing faster computations and reduced power consumption, making them ideal for mobile devices and embedded systems [1, 3]. ECC is used in various standards and protocols, including TLS, digital signatures, and encryption.

Asymmetric-key cryptography eliminates the need for secure key exchange, as public keys can be freely distributed while keeping private keys secure. The system scales efficiently, as each participant only needs a single pair of keys to communicate securely with any number of others. However, asymmetric algorithms may generally be slower and require more computational resources, making them less suitable for encrypting large volumes of data. The mathematical operations involved in asymmetric cryptography are more complex, requiring careful implementation to ensure security and efficiency [1, 3].

Asymmetric-key cryptography underpins many essential security protocols and applications, including: Digital signatures, secure communications, and key exchange [1, 3].

Asymmetric-key cryptography is a cornerstone of modern security systems, addressing critical challenges in key distribution and scalability. While it complements symmetric-key cryptography, its unique properties enable a wide range of secure communication and authentication protocols essential in today's interconnected world. The following section will introduce the Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) algorithms, two types of ABE, which are advanced asymmetric-key cryptographic techniques.

1.1.3 Key-Policy Attribute-Based Encryption

Key-Policy Attribute-Based Encryption is an advanced cryptographic system that enhances access control mechanisms by associating attributes with data and access policies with cryptographic keys. It is a type of Attribute-Based Encryption where the access control policy is embedded in the private keys of users, and the ciphertext is associated with a set of attributes.

In KP-ABE, the encryption process involves the following components:

- **Attributes:** Descriptive identifiers that characterize the data. For example, attributes could include roles (e.g., "Doctor", "Nurse"), data types (e.g., "Medical Records"), or other contextual information.
- **Access Policy:** A logical expression embedded in the user's private key that defines which combinations of attributes allow decryption. Access policies are typically expressed as monotonic access structures like AND, OR, and threshold gates.
- **Encryption:** Data is encrypted with a set of attributes, and only users whose private keys satisfy the corresponding access policy can decrypt the data.
- **Decryption:** Users obtain private keys that contain embedded access policies. If the attributes associated with the ciphertext satisfy the access policy in the user's private key, the user can decrypt the data.

The KB-ABE has four key components, as follows:

1. **Setup:** The setup algorithm generates a public key and a secret key. This phase is typically executed by a trusted authority.

2. **Key Generation:** The key generation algorithm uses the secret key to generate a private key for a user based on an access policy. The access policy specifies which attribute combinations allow decryption.

3. **Encryption:** In the encryption phase, data is encrypted using the public key and a set of attributes. The resulting ciphertext can only be decrypted by users whose private keys have an access policy that matches the attributes.

4. **Decryption:** The decryption algorithm allows a user to decrypt the ciphertext if the attributes in the ciphertext satisfy the access policy in the user's private key.

Consider a medical data system where attributes represent different roles, such as "Doctor", "Nurse", and "Researcher". An access policy might specify that only users with the role "Doctor" OR ("Nurse" AND "Researcher") can decrypt the medical records.

- **Setup:** The trusted authority generates the public key and secret key.
- **Key Generation:**
  - A doctor receives a private key with an access policy "Doctor".
  - A nurse-researcher receives a private key with an access policy "Nurse AND Researcher".
- **Encryption:** Medical records are encrypted with the attribute set "Doctor", "Medical Records".
- **Decryption:**
  - The doctor can decrypt the medical records because the attribute "Doctor" satisfies their access policy.
  - The nurse-researcher cannot decrypt the records because their access policy requires both "Nurse" and "Researcher" attributes, which are not all present in the ciphertext.

Key-Policy Attribute-Based Encryption is a powerful cryptographic technique that enhances data security and access control by embedding access policies within user keys. It is especially useful in environments requiring fine-grained and flexible access control, such as healthcare, cloud storage, and secure communications. While it offers significant advantages in terms of access control and flexibility, implementing KP-ABE systems requires careful consideration of complexity and performance challenges.

1.1.4 Ciphertext-Policy Attribute-Based Encryption

Ciphertext-Policy Attribute-Based Encryption is an advanced encryption algorithm that enhances the flexibility and security of access control in data encryption. Unlike traditional encryption methods where access is determined by the possession of a single key, CP-ABE enables access based on a user's attributes, providing a more granular and expressive approach to data security [5].

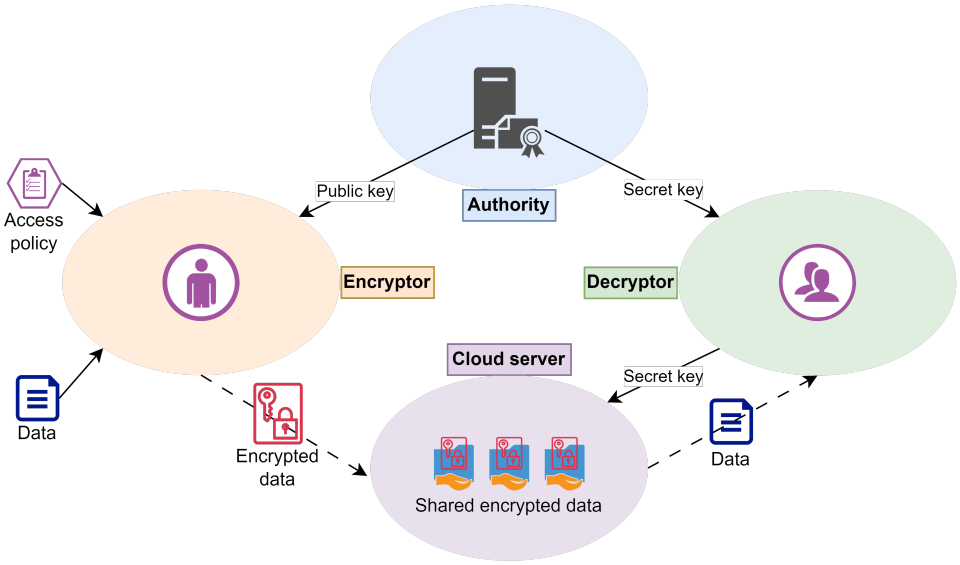


Figure 1.4: Ciphertext-Policy Attribute-Based Encryption system.

Figure 1.4 illustrates the process of the CP-ABE system. In CP-ABE, data is encrypted under an access policy specified by the data owner. Users are issued keys associated with a set of attributes. Access to the encrypted data is granted only if the user's attributes satisfy the access policy embedded in

the ciphertext [5]. This approach integrates the encryption and access control processes, ensuring that only authorized users can decrypt the data.

- **Attributes:** Characteristics or properties assigned to users (e.g., role, department, clearance level).
- **Access policy:** A logical expression specifying which combinations of attributes are required to decrypt the data (example: *"Faculty: Engineering* **AND** *Position: Professor"*).
- **Ciphertext:** The encrypted data, which includes the access policy.
- **User's key:** A private key or secret key associated with the user's attributes.

The CP-ABE has four primary phases as follows [6]:

1. **Setup:** The authority generates a master key and a public key. While the public key is distributed to users, the master key is kept secret.
2. **Key generation:** The authority responsible for key management uses the master key and public key to generate secret keys for users based on their attributes. These secret keys enable authorized users to decrypt ciphertexts that adhere to specified access policies, ensuring fine-grained access control over encrypted data.
3. **Encryption:** The data owner/encryptor defines an access policy and uses it to encrypt the data. The ciphertext includes the access policy and the encrypted data.
4. **Decryption:** A user/decryptor attempts to decrypt the ciphertext. The decryption is successful only if the user's attributes match the access policy embedded in the ciphertext.

CP-ABE allows data owners/encryptors to define detailed access policies, enabling complex and precise control over who can access the data. Policies can be defined using any combination of attributes, making the system adaptable to various scenarios and organizational structures. Unlike traditional methods

that require a separate key for each possible access combination, CP-ABE simplifies key management by associating keys with attributes [5].

The setup and management of CP-ABE systems can be complex, particularly in environments with a large number of attributes and users. While CP-ABE reduces some key management burdens, the system must still handle the distribution and management of attribute-based keys, which can become challenging in large-scale deployments.

CP-ABE is well-suited for scenarios where data security and flexible access control are critical. Common applications include: Cloud storage, Internet of Things (IoT), Healthcare, Enterprise data management, Digital Rights Management (DRM) [7].

Ciphertext-Policy Attribute-Based Encryption represents a significant advancement in cryptographic access control, providing flexible, fine-grained, and secure data protection. By allowing access policies to be embedded directly into the ciphertext, CP-ABE ensures that only users with the appropriate attributes can decrypt and access the data. This approach is particularly valuable in dynamic and large-scale environments where traditional access control mechanisms fall short.

1.2 Problem outline and motivation

CP-ABE is utilized across diverse domains like cloud storage, IoT, and healthcare due to its attribute-based access control capabilities [7]. However, its reliance on the Pairing-Based Cryptography library, now outdated and vulnerable to attacks, poses significant security risks. The lack of sufficient security measures in this library, which supports only an 80-bit security level, renders CP-ABE unsuitable for modern applications, threatening data confidentiality and integrity [8,9]. As a result, CP-ABE's effectiveness is compromised, hindering its adoption in contemporary systems requiring robust security solutions. Addressing these limitations is essential to ensure the viability and effectiveness of CP-ABE in safeguarding sensitive information across various domains.

On the other hand, the ELiPS library offers efficient computational performance while ensuring high security, making it a specialized cryptographic

library focused on pairing-based cryptography [10]. It leverages the BLS-12 curve, providing a robust 128-bit security level, which addresses the limitations of older cryptographic libraries [11]. ELiPS includes a variety of functions to support the implementation of algorithms and protocols that rely on pairing, enhancing the development process and operational efficiency [10, 11].

Additionally, the ELiPS library has been utilized in advanced cryptographic applications, such as Pairing-based Homomorphic Encryption by Kanenari et al. [12], demonstrating its capability to manage complex schemes securely. Moreover, ELiPS has been used in the implementation of zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge), essential for privacy-preserving protocols in blockchain and other secure systems. This versatility and effectiveness make ELiPS a valuable resource for developers and researchers in the field of modern cryptography.

In our first study, to deal with these challenges, we propose an ELiPS-based CP-ABE scheme, integrating ELiPS into the CP-ABE framework. Despite differences between PBC and ELiPS libraries, including function parameters and data types, as well as the type of pairing, we establish compatibility by generating a generator g and employing Shirase's method [13] to convert asymmetric to symmetric pairing. After that, we make essential modifications to ensure the integration of ELiPS into the CP-ABE framework. These modifications span the setup, key generation, encryption, and decryption algorithms.

To validate the proposal, we conducted several experiments, utilizing a data access authorization process at the university level with various attribute policy scenarios. We compare our ELiPS-base solution with MCL to see if it works as well and stays up to date with today's security needs. This comparison confirms that our approach meets today's security standards while operating efficiently. Moreover, the analysis and experimental results demonstrate the effectiveness of the proposal, revealing not only an increase in the security level but also a reduction in computational requirements for the setup, key generation, and encryption functions. However, the decryption in CP-ABE remains heavy.

In our second study, therefore, we aim to improve the decryption processing of ELiPS-based CP-ABE. The decryption part in ELiPS-based CP-ABE mainly employs inversions and pairings, including Miller loops and final expo-

nentiations. Thus, we propose two methods, minimizing the number of final exponentiations and inversions, to reduce the decryption time in ELiPS-based CP-ABE. Through formula analysis, the proposed scheme reduces by $2n - 1$ times final exponentiations and by $n - 1$ 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 45.5% compared to the initial version of ELiPS-based CP-ABE.

In our third study, furthermore, we evaluate and analyze the impact of these optimizations on decryption efficiency. Moreover, we compare the ELiPS-based CP-ABE with these improvements to the initial version and the original PBC-based CP-ABE. As a result, the combination of both optimization techniques resulted in an average 43.1% overall reduction in decryption time compared to the initial version of the ELiPS-based CP-ABE scheme, while in total execution, it led to a 25.3% improvement. Furthermore, there was an average 53.8% overall reduction in total execution time compared to the original PBC-based CP-ABE method.

1.3 Major contributions

The following are some of the main contributions:

- **Implementation of ELiPS-based CP-ABE (first study).**

  The key works were published in the CANDAR 2023 conference [14] and the IJNC journal [15]. These works are presented in Chapter 4. The primary contributions of the first study can be summarized as follows:

  - Introduce generator g over $E(\mathbb{F}_{p^{12}})$ specifically designed to convert asymmetric pairing to symmetric pairing, which is a significant contribution in addressing the compatibility issue between ELiPS and the original CP-ABE scheme.
  - Employ Shirase's technique [13] to accomplish the conversion of asymmetric pairing to symmetric pairing using the generated generator g.

- Make several modifications to the CP-ABE framework and carefully select appropriate ELiPS functions to ensure compatibility of ELiPS with CP-ABE.
- Evaluate the proposal in terms of computation time.
- Compare our proposal's performance with other competitive pairing libraries.

- **Improvement decryption process in ELiPS-based CP-ABE (second study):**

  Chapter 5 provides detailed information on these works, and the key findings of this study were published in the JAIT journal [16]. The second study summarizes its main contributions as follows:

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

- **Further evaluates the impact of these decryption optimizations (third study):**

  These works of this study are introduced in Chapter 6 and in our publication at the ICTIS 2024 conference [17], which presented the primary findings of the third study. The key contributions of this study are:

  - Evaluates the impact of these decryption optimizations on the efficiency of the ELiPS-based CP-ABE approach.
  - Analyze the performance gains achieved through the optimizations and compare our optimized construction to both the initial ELiPS-based version and the original PBC-based CP-ABE scheme.

1.4 Thesis outline

Some of the results from this thesis have been published, particularly as follows: A portion of Chapter 4 was published in the CANDAR 2023 conference [14] and the IJNC journal [15]. Our paper in the JAIT journal [16] presents part of the results from Chapter 5, while Chapter 6 was introduced at the ICTIS 2024 conference [17]. The remaining structure of this thesis is as follows:

Chapter 2: We briefly discuss the mathematical concepts necessary for understanding this thesis. We introduce modular arithmetic, group, ring, field, Frobenius map, elliptic curve, sextic twist, hash function, pairing, types of pairing, the discrete logarithm problem, and the elliptic curve discrete logarithm problem.

Chapter 3: We introduce an overview of prominent pairing-based cryptography libraries, such as PBC, RELIC, MCL, and ELiPS. We then show a comparison between these prominent libraries in terms that are mainly utilized in the CP-ABE algorithm. Additionally, we present an overview of an access tree, which plays a crucial role in determining who can access encrypted data. Finally, we provide an overview of the CP-ABE scheme.

Chapter 4: We present the implementation of ELiPS-based CP-ABE, including generating a generator g, transforming asymmetric pairing to symmetric pairing, and modifying CP-ABE to integrate ELiPS into the CP-ABE framework. We also discuss the experimental evaluation and results.

Chapter 5: We propose two methods to improve the decryption process time in ELiPS-based CP-ABE by minimizing the number of final exponentiations and inversions. We then evaluate our proposal and discuss the results.

Chapter 6: We perform a further performance analysis of ELiPS-based CP-ABE with optimized decryption functions. We describe the evaluation setup and assess the performance of ELiPS-based CP-ABE with these optimizations in terms of decryption time and total execution time.

Chapter 7: We conclude the thesis and outline future works.

Chapter 2

Fundamental mathematics

In this chapter, we briefly discuss the mathematical concepts necessary for understanding this thesis. We present modular arithmetic, group, ring, field, Frobenius map, elliptic curve, sextic twist, hash function, pairing, types of pairing, the discrete logarithm problem, and the elliptic curve discrete logarithm problem.

2.1 Modular arithmetic

Modular arithmetic is an essential tool in modern cryptography, providing the foundation for cryptographic algorithms. By leveraging the principles of modular arithmetic, cryptographic systems can achieve secure encryption, key exchange, and digital signatures, ensuring the confidentiality and integrity of sensitive information in digital communications.

Definition 1. If a is an integer and m is a positive integer, we define $a \bmod m$ to be the remainder when a is divided by m. The integer m is called the **modulus**. Thus, for any integer a, we can write as follows [2]:

$$a = nm + r, \qquad \text{where: } 0 \leq r < m, \quad n = \lfloor a/m \rfloor. \tag{2.1}$$

Example 1. $13 \mod 11 = 2$; and $-13 \mod 11 = 9$.

Definition 2. Two integers a and b are said to be congruent modulo m, if $(a \bmod m) = (b \bmod m)$. This is written as $a \equiv b \pmod{m}$.

Example 2. $18 \equiv 5 \pmod{13}$; $11 \equiv -5 \pmod{8}$.

Congruences have the following properties [2]:

- $a \equiv b \pmod{m}$ if $m \mid (a - b)$.
- $a \equiv b \pmod{m}$ implies $b \equiv a \pmod{m}$.
- $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$ imply $a \equiv c \pmod{m}$.

2.2 Group, Ring, Field

In finite field arithmetic, mathematical structures such as group, ring, and field play fundamental roles in defining the operations and properties of arithmetic within the finite field.

2.2.1 Group

In mathematics, a group is a fundamental algebraic structure consisting of a set of elements and a binary operation defined on those elements.

Definition 3. A group $G = (S, \circ)$ is a set S of elements together with a binary operation [3]

$$\circ : S \times S \to S. \tag{2.2}$$

The operation must satisfy four properties [2, 18, 19]:

1. **Closure:** For any two elements a and b in the group, their combination under the operation, denoted as:

$$a \circ b, \text{must also be an element of the group.}$$

2. **Associativity:** The order in which operations are performed does not affect the result. That is:

$$(a \circ b) \circ c = a \circ (b \circ c), \text{for all elements } a, b, \text{ and } c \text{ in the group.}$$

3. **Identity element:** There exists an element, usually denoted as e or 1, such that combining it with any other element leaves the other element unchanged. Formally, for any element a in the group:

$$a \circ e = e \circ a = a.$$

4. **Inverse element:** For every element a in the group, there exists an element, usually denoted as a^{-1} in the case of the multiplicative group, such that combining a with its inverse yields the identity element. In other words:
$$a \circ a^{-1} = a^{-1} \circ a = e.$$

Definition 4. A group is said to be **abelian** or **commutative** if it satisfies the following additional condition [2, 3]:

5. **Commutative:** $a \circ b = b \circ a$ for all $a, b \in G$.

Groups are used extensively in various branches of mathematics, including abstract algebra, number theory, and geometry. They serve as the foundation for understanding symmetry, transformations, and many other mathematical concepts. In particular, groups play a crucial role in cryptography, where they are used to define mathematical structures that underpin encryption algorithms and protocols.

2.2.2 Ring

In mathematics, a ring is an algebraic structure consisting of a set equipped with two binary operations: addition and multiplication. These operations must satisfy specific properties that generalize arithmetic operations on integers.

Definition 5. A ring $\{R, +, \cdot\}$ is a set R with two binary operations, that we shall call addition $(+)$ and multiplication $(\cdot)$, such that satisfy the following properties [2, 3, 18, 19]:

1. **Closure:** The set R is closed under both addition and multiplication. This means that for any elements a, b in R:
   - The sum $a + b$ is also in R.
   - The product $a \cdot b$ is also in R.
2. **Associativity:**

- Addition is associative: $(a + b) + c = a + (b + c)$ for all $a, b, c \in R$.
- Multiplication is associative: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in R$.

3. **Additive identity:** There exists an element $0 \in R$ such that: $a + 0 = 0 + a = a$ for all $a \in R$.

4. **Additive inverses:** For each $a \in R$, there exists an element $-a \in R$ such that: $a + (-a) = (-a) + a = 0$.

5. **Distributivity:** Multiplication distributes over addition: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ and $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$ for all $a, b, c \in R$.

A ring may also have a multiplicative identity, an element $a \cdot 1 = 1 \cdot a = a$ for all $a \in R$. Rings with a multiplicative identity are often referred to as "rings with unity".

Rings can be classified into different types based on additional properties [2]. For example:

- **Commutative ring:** A ring in which multiplication is commutative, i.e., $a \cdot b = b \cdot a$ for all $a, b \in R$.
- **Ring with unity:** A ring that has a multiplicative identity.

Rings are fundamental in various areas of mathematics, including number theory, algebraic geometry, and functional analysis. They also form the basis for many cryptographic algorithms and error-correcting codes, where the ring structure provides a framework for constructing and analyzing these systems.

2.2.3 Field

In mathematics, a field is an algebraic structure that provides a framework for understanding and performing arithmetic operations such as addition, subtraction, multiplication, and division (except by zero).

Definition 6. A field is a ring $\{F, +, \cdot\}$ for which multiplication is commutative and every nonzero element in F has an inverse under multiplication. That is, a field is a ring that is a group under addition and for which the elements other than the additive identity form an abelian group under multiplication [2, 3].

A field $(F, +, \cdot)$ satisfies the following properties [2, 19]:

1. **Closure:** The set F is closed under both addition and multiplication. This means that for any elements a, b in F:
   - The sum $a + b$ is also in F.
   - The product $a \cdot b$ is also in F.
2. **Associativity:**
   - Addition is associative: $(a + b) + c = a + (b + c)$ for all $a, b, c \in F$.
   - Multiplication is associative: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in F$.
3. **Commutativity:**
   - Addition is commutative: $a + b = b + a$ for all $a, b \in F$.
   - Multiplication is commutative: $a \cdot b = b \cdot a$ for all $a, b \in F$.
4. **Identity elements:**
   - Additive identity: There exists an element $0 \in F$ such that $a + 0 = 0 + a = a$ for all $a \in F$.
   - Multiplicative identity: There exists an element $1 \in F$ (where $1 \neq 0$) such that $a \cdot 1 = 1 \cdot a = a$ for all $a \in F$.
5. **Inverses:**
   - Additive inverse: For each $a \in F$, there exists an element $-a \in F$ such that $a + (-a) = (-a) + a = 0$.
   - Multiplicative inverse: For each $a \in F$ (where $a \neq 0$), there exists an element $a^{-1} \in F$ such that $a \cdot a^{-1} = a^{-1} \cdot a = 1$.
6. **Distributivity:** Multiplication distributes over addition: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ for all $a, b, c \in F$.

Fields are integral to various branches of mathematics, including algebra, number theory, and geometry. They provide the underlying structure for many mathematical concepts and systems. In particular, finite fields (also known as

Galois fields) are extensively used in cryptography and coding theory. Finite fields offer a robust mathematical foundation for constructing secure encryption algorithms, error-correcting codes, and other applications where precise and reliable arithmetic operations are crucial.

2.3 Frobenius map

The Frobenius map, named after the mathematician Ferdinand Frobenius, is an important concept in the theory of finite fields, especially in the context of algebra and number theory. It is a special type of endomorphism (a map from a field to itself) that plays a crucial role in the arithmetic and algebraic properties of finite fields [18, 19].

Definition 7. In a finite field $\mathbb{F}_q$, where q is a power of a prime p (i.e., $q = p^n$), the Frobenius map is defined as follows [18]:

$$\varphi(a) = a^p, \tag{2.3}$$

for any element $a \in \mathbb{F}_q$.

The Frobenius map has several key properties [18, 19]:

1. **Homomorphism:** It preserves the field structure, meaning that it is a ring homomorphism. Specifically:
   - $\varphi(a + b) = \varphi(a) + \varphi(b)$.
   - $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$.
2. **Injectivity and surjectivity:** In a finite field $\mathbb{F}_q$, the Frobenius map is both injective and surjective, hence it is a bijection. This means every element in the field is mapped to a unique element and covers the entire field.
3. **Iterative nature:** Repeated application of the Frobenius map results in raising elements to higher powers of p:
   - $\varphi^k(a) = a^{p^k}$.

- For $k = n$, where n is the dimension of the field extension, $\varphi^n(a) = a$ for all $a \in \mathbb{F}_q$.

4. **Fixed points:** In the prime subfield $\mathbb{F}_p$, the Frobenius map acts as the identity function since $a^p = a$ for any $a \in \mathbb{F}_p$.

The Frobenius map has significant applications in various areas of mathematics and cryptography:

- **Field automorphisms:** It generates the Galois group of a finite field extension $\mathbb{F}_q/\mathbb{F}_p$, which is cyclic and generated by the Frobenius map.
- **Polynomial equations:** It is used in solving polynomial equations over finite fields, especially in counting solutions (e.g., in the proof of the Weil conjectures).
- **Cryptographic algorithms:** It plays a role in certain cryptographic algorithms, such as those based on elliptic curves over finite fields, where the Frobenius endomorphism can be used to construct efficient algorithms for point multiplication and other operations.

The Frobenius map is a fundamental concept in the study of finite fields, providing deep insights into their structures and properties. Its homomorphic nature, combined with its role in generating field automorphisms, makes it a powerful concept in both theoretical and applied mathematics, particularly in areas such as cryptography and coding theory.

2.4 Elliptic curve

Elliptic curve mathematics is a branch of algebraic geometry and number theory that studies the properties and applications of elliptic curves. An elliptic curve is defined as a smooth, non-singular projective algebraic curve of genus one, with a specified point at infinity. Over a field $\mathbb{F}_q$ (where $q = p^m$), an elliptic curve can be described by a simplified Weierstrass equation of the form [18–20]:

$$y^2 = x^3 + ax + b, \tag{2.4}$$

where m is an extension degree, a and b are coefficients in $\mathbb{F}_q$ satisfying following condition:

$$4a^3 + 27b^2 \neq 0. \tag{2.5}$$

The pair (x, y) that satisfies Equation (2.4) is called a rational point of E. $\#E(\mathbb{F}_q)$ is number of rational points of $E(\mathbb{F}_q)$:

$$\#E(\mathbb{F}_q) = q + 1 - t, \tag{2.6}$$

where $\mid t \mid \leq 2\sqrt{q}$.

Let r be the largest prime factor that divides $\#E(\mathbb{F}_q)$. Then, k be the minimal integer such that satisfies $r \mid (q^k - 1)$, which is called the embedding degree of the order r group.

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 Equation (2.4). The arithmetic operations over the elliptic curve are defined as follows [18].

- **Elliptic Curve Addition (ECA)**

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

$$\begin{aligned} &\lambda = \frac{y_Q - y_P}{x_Q - x_P}, \\ &\begin{cases} x_R = \lambda^2 - x_P - x_Q, \\ y_R = \lambda(x_P - x_R) - y_P. \end{cases} \end{aligned} \tag{2.7}$$

- **Elliptic Curve Doubling (ECD)**

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

$$\begin{aligned} &\lambda = \frac{3x_P^2 + a}{2y_P}, \\ &\begin{cases} x_R = \lambda^2 - 2x_P, \\ y_R = \lambda(x_P - x_R) - y_P. \end{cases} \end{aligned} \tag{2.8}$$

- **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:

$$R = sP = \underbrace{P + P + \cdots + P}_{s\text{-1 times additions.}}. \tag{2.9}$$

2.5 Sextic twist

A sextic twist is a mathematical concept used in the field of elliptic curve cryptography, particularly when dealing with pairing-based cryptographic protocols. It involves a special kind of isomorphism between elliptic curves that simplifies certain calculations, making cryptographic operations more efficient [19, 20].

Sextic twists are particularly useful in pairing-based cryptography, which relies on bilinear pairings such as the Tate pairing or the Weil pairing. These pairings are crucial for various cryptographic protocols, including Identity-Based Encryption, Attribute-Based Encryption, and Short Digital Signatures. The use of sextic twists can significantly reduce the computational complexity of evaluating these pairings.

Definition 8. Let E and E' be two elliptic curves defined over $\mathbb{F}_q$, for q, a power of a prime number p. Then, the curve E' is a twist of degree d of E if we can define an isomorphism ϕ_d over $\mathbb{F}_{q^d}$ from E' into E and such that d is minimal [19]:

$$\phi_d : E'(\mathbb{F}_q) \to E(\mathbb{F}_{q^d}) \tag{2.10}$$

Let E be given by $y^2 = x^3 + b$ defined over $\mathbb{F}_{p^{12}}$. The sextic twist E' of E is given by $y^2 = x^3 + b/z$ defined over $\mathbb{F}_{p^2}$, where z is quadratic and cubic non-residue in $\mathbb{F}_{p^2}$:

$$\begin{cases} E : y^2 = x^3 + b & \text{defined over } \mathbb{F}_{p^{12}}, \\ E' : y^2 = x^3 + b/z & \text{defined over } \mathbb{F}_{p^2}. \end{cases} \tag{2.11}$$

The sextic twist $\phi_6 : E'(\mathbb{F}_{p^2}) \to E(\mathbb{F}_{p^{12}})$ is defined as follows [19]:

$$\phi_6 : E'(\mathbb{F}_{p^2}) \to E(\mathbb{F}_{p^{12}}), \qquad (x, y) \mapsto (z^{\frac{1}{3}}x, z^{\frac{1}{2}}y). \tag{2.12}$$

Sextic twist is used in various cryptographic schemes, particularly those based on elliptic curves and pairings. This includes:

- **Pairing-Based Cryptography:** Enhancing the efficiency of pairings in protocols like identity-based encryption and attribute-based encryption [19].
- **Homomorphic Encryption:** Certain implementations of homomorphic encryption schemes utilize sextic twists for improved efficiency [19].
- **Zero-Knowledge Proofs:** Cryptographic protocols that rely on zero-knowledge proofs can benefit from the performance enhancements provided by sextic twists [19].

Sextic twist is a valuable method in the realm of elliptic curve cryptography, offering a means to optimize the performance of pairing-based protocols. By leveraging the mathematical relationship between elliptic curves and their twists, cryptographic operations can be made more efficient, enhancing the overall security and performance of cryptographic systems.

2.6 Hash function $\mathcal{H}$ onto elliptic curve

A hash function onto an elliptic curve is a cryptographic technique that maps arbitrary input data (such as a message) to a point on an elliptic curve [2, 19]. This process is essential for various cryptographic protocols, including Digital Signatures, Public Key Cryptography, Zero-Knowledge proofs, and CP-ABE, where it is often necessary to securely convert data into a point on an elliptic curve.

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

$$\mathcal{H} : \{0, 1\}^* \to \mathbb{G}. \tag{2.13}$$

Hash function $\mathcal{H}$ has the following properties [19]:

- **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 $\mathcal{H}(m) = \mathcal{H}(m')$.

Hash function onto elliptic curve has various applications, as follows:

- **Digital Signatures:** Hash functions onto elliptic curves are used in schemes like Elliptic Curve Digital Signature Algorithm (ECDSA), where messages are hashed to points on the curve to create and verify signatures.
- **Public Key Cryptography:** Mapping data to elliptic curve points is crucial for protocols like Elliptic Curve Diffie-Hellman (ECDH) for secure key exchange.
- **Zero-Knowledge Proofs:** Cryptographic proofs often require mapping statements or secrets to elliptic curve points to ensure security and anonymity.

Hash function onto elliptic curve is a fundamental component of modern cryptographic systems, enabling secure and efficient mapping of data to elliptic curve points. By ensuring uniform distribution, collision resistance, and other cryptographic properties, these functions enhance the security and performance of various cryptographic protocols.

2.7 Pairing map

Bilinear pairing is a mathematical operation that plays a crucial role in modern cryptographic protocols, particularly in the context of pairing-based cryptography. A bilinear pairing is a map that takes two points from two

groups and maps them to a third group in such a way that the operation is bilinear [18]. This property enables a range of advanced cryptographic applications, including Short Signatures, Identity-Based Encryption, and Ciphertext-Policy Attribute-Based Encryption.

The subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$ of $E(\mathbb{F}_{p^{12}})$ are defined as follows [13]:

$$\begin{cases} \mathbb{G}_1 = E[r] \cap \mathrm{Ker}(\pi_p - [1]), \\ \mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\pi_p - [p]), \end{cases} \tag{2.14}$$

where $E[r]$ is a subgroup of order r on an elliptic curve over $\mathbb{F}_{p^{12}}$; $\pi_p : A \mapsto A^p$ is called a Frobenius endomorphism, a low-cost mapping for calculating p-th powering; $\mathrm{Ker}(\varphi)$ is the set of points mapped to the point at infinity $\mathcal{O}$ by the specified map φ: $\mathrm{Ker}(\varphi) = \{P \in E(\mathbb{F}_{p^{12}}) : \varphi(P) = \mathcal{O}\}$.

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:

$$e : \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{G}_T, \tag{2.15}$$

which has the following properties:

- **Bilinear map**

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

  $$\begin{aligned} e(Q + Q', P) &= e(Q, P) \cdot e(Q', P), \\ e(aQ, bP) &= e(bQ, aP) = e(Q, P)^{ab}. \end{aligned} \tag{2.16}$$

- **Non-degeneracy**

  For all $P \neq \mathcal{O}$ and $Q \neq \mathcal{O}$, then:

  $$e(Q, P) \neq 1. \tag{2.17}$$

Bilinear pairing has been employed in various applications, as follows:

- **Identity-Based Encryption (IBE):** Bilinear pairing allows the construction of encryption schemes where the public key can be derived

from an arbitrary string, such as an email address, enabling simplified key management.

- **Attribute-Based Encryption (ABE):** Extends IBE by allowing encryption and decryption based on attributes (e.g., user roles, policies), providing fine-grained access control.
- **Short Signatures:** Pairing-based signatures, such as the Boneh-Lynn-Shacham (BLS) signatures, are much shorter than traditional signatures, making them useful in resource-constrained environments.
- **Zero-Knowledge Proofs:** Pairings enable efficient construction of zero-knowledge proofs, where a prover can convince a verifier that they know a value without revealing the value itself.
- **Key Exchange Protocols:** Pairing-based versions of the Diffie-Hellman key exchange protocol provide enhanced security features and are used in various secure communication standards.

Bilinear pairings are a powerful tool in modern cryptography, enabling a wide range of advanced cryptographic protocols and applications. Their unique properties of bilinearity, non-degeneracy, and computability make them suitable for constructing efficient and secure cryptographic schemes that are widely used in practical implementations.

2.8 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. There are three types of pairings [19]:

- **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.

Type III pairings have steadily replaced Type I pairings, despite Type I pairings being the norm in the early days of pairing-based encryption. In fact, because Type I includes very huge curves, it is not desirable enough from an efficiency standpoint nowadays.

Additionally, Type III pairings are consistent with a number of computational assumptions that do not hold in Type I pairings, such as the Decision Diffie-Hellman (DDH) in $\mathbb{G}_1$ or $\mathbb{G}_2$, often known as the XDH assumption.

2.9 Discrete Logarithm Problem and Elliptic Curve Discrete Logarithm Problem

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

2.9.1 Discrete Logarithm Problem

The DLP is a fundamental problem in the field of number theory and cryptography. It underpins the security of many cryptographic protocols, including the Diffie-Hellman key exchange and the Digital Signature Algorithm (DSA). The problem can be described as follows [18]:

Definition 9. Given a finite cyclic group $\mathbb{G}$ generated by an element g, and an element h in $\mathbb{G}$, the Discrete Logarithm Problem is to find the integer x (if it exists) such that:

$$g^x = h, \tag{2.18}$$

where the group operation is written multiplicatively. This integer x is called the discrete logarithm of h to the base g, and is denoted by $\log_g h$.

Example 3. In the multiplicative group of integers modulo a prime p, the

DLP involves finding x such that:

$$g^x \equiv h \pmod{p}. \tag{2.19}$$

The difficulty of solving the DLP is the basis for the security of many cryptographic systems. While there are algorithms for solving the DLP, such as the baby-step giant-step algorithm and Pollard's rho algorithm, these are generally inefficient for large groups, making the problem computationally hard for sufficiently large group sizes.

2.9.2 Elliptic Curve Discrete Logarithm Problem

The ECDLP is analogous of the DLP, but it is defined within the context of elliptic curves over finite fields. The problem can be described as follows [2]:

Definition 10. Given an elliptic curve E defined over a finite field $\mathbb{F}_q$, a point P on E which generates a cyclic subgroup, and a point Q in the subgroup generated by P, the Elliptic Curve Discrete Logarithm Problem is to find the integer k (if it exists) such that:

$$Q = kP, \tag{2.20}$$

where kP denotes the scalar multiplication of the point P by the integer k.

Example 4. Consider an elliptic curve E over a finite field $\mathbb{F}_q$ with a point P of order r. Given a point Q in the subgroup generated by P, the ECDLP involves finding k such that:

$$Q = kP. \tag{2.21}$$

The ECDLP is the foundation of elliptic curve cryptography. Solving the ECDLP is significantly harder than solving the DLP for the same group size. This means that elliptic curve cryptosystems can achieve the same level of security as traditional systems like RSA or DSA with much smaller key sizes. The best-known algorithms for solving the ECDLP, such as Pollard's rho algorithm for elliptic curves, are much less efficient than their counterparts for the DLP, making ECC highly efficient and secure.

2.10 Summary

This chapter presented the related mathematical fundamentals such as modular arithmetic, group theory, ring theory, field theory, ECC, pairings, and hash functions on elliptic curves. These concepts play a crucial role in cryptographic protocols such as CP-ABE.

Chapter 3

Efficient pairing libraries and Ciphertext-Policy Attribute-Based Encryption

In this chapter, we introduce an overview of prominent pairing-based cryptography libraries. We then show a comparison between these prominent libraries in terms that are mainly utilized in the CP-ABE algorithm. Additionally, we present an overview of an access tree, which plays a crucial role in determining who can access encrypted data. Afterward, we introduce the CP-ABE algorithm.

3.1 Efficient libraries for pairing systems

This section presents an overview of the PBC, RELIC, MCL, and ELiPS libraries as competitive pairing libraries. Then, we demonstrate a comparison among four notable libraries in terms that are mainly utilized in the CP-ABE scheme.

3.1.1 Pairing-Based Cryptography (PBC) library

The GNU Multiple Precision (GMP) arithmetic library served as the foundation for the PBC library, an open-source library carrying out the essential mathematical operations in pairing-based cryptosystems [21]. Speed and portability are crucial considerations as the PBC library is intended to serve as the foundation for pairing-based cryptosystem implementations. It offers functions like pairing computation and elliptic curve arithmetic.

In PBC, which utilizes symmetric pairing, let $\mathbb{G}$ be an additive group over an elliptic curve and $\mathbb{G}_T$ be a multiplicative cyclic group. Both groups $\mathbb{G}$ and $\mathbb{G}_T$ have order r [6]. The pairing operation is defined as:

$$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T.$$

There are eight different parameter types available in PBC. In each case, the curve group has a group order of 160-bit. Type A is known to be the fastest pairing and is suitable for cryptosystems where the group size is not a critical factor [22]. However, this type only provides an 80-bit security level and is vulnerable to multiple attacks [8, 9, 22]. Type A utilizes a supersingular curve, which is defined as follows:

$$E : y^2 = x^3 + x.$$

3.1.2 Efficient LIbrary for Cryptography (RELIC)

RELIC is an Efficient LIbrary for Cryptography, developed by Aranha et al. [23]. The first version was released in 2010. It is a contemporary cryptographic library, prioritizing efficiency and adaptability. RELIC focuses on portability, including architecture-dependent code, flexible configuration, and maximum efficiency [23].

RELIC utilizes both BN curves and BLS curves for configuration options. It supports a wide range of security levels such as 128-bit, 192-bit, and 256-bit [24]. The RELIC library supports nearly all functions necessary for the implementation of CP-ABE, such as elliptic curve addition, elliptic curve scalar multiplication, inversion, hash-to-curve, and pairing.

3.1.3 MCL

The MCL library, developed by Mitsunari et al. [24], is a high-performance library specializing in cryptographic operations and multi-core computation. It offers efficient implementations of mathematical operations crucial for modern cryptography, including elliptic curve cryptography and pairing-based cryptography.

With a focus on optimization and parallelism, MCL leverages multi-core processors to achieve fast execution time, making it ideal for applications requiring high computational efficiency. MCL is compatible with various operating systems and hardware architectures [25]. By providing developers with robust and optimized cryptographic primitives, the MCL library serves as a

valuable tool for building secure and efficient cryptographic systems and protocols. MCL supports the BLS curve with a 255-bit size of r and an embedding degree of 12, providing a 128-bit security level.

3.1.4 Efficient Library for Pairing Systems (ELiPS)

The ELiPS library is a specialized cryptographic library that focuses on efficient operations related to pairing-based cryptography. Such cryptography involves mathematical pairings between points on elliptic curves. The ELiPS library offers a range of functionalities, including point arithmetic operations, exponentiation, and pairing computations [10, 11]. ELiPS is specifically designed to support bilinear pairing using the BLS-12 curve, providing a 128-bit security level [10, 11].

The ELiPS library was evaluated and verified by Takahashi et al. [11]. It has gained attention for its applications in advanced cryptographic schemes such as identity-based encryption, attribute-based encryption, and functional encryption. Furthermore, it has been applied not only in the realization of Pairing-based Homomorphic Encryption by Kanenari et al. [12] but also being employed in the implementation of zk-SNARKs. This library is currently in development with regular updates, suggesting its potential as a promising resource.

3.1.5 A comparison among prominent pairing libraries in terms of primary domains used in CP-ABE

We conduct a comparative analysis of four prominent libraries in this research area: PBC [22], RELIC [23], MCL [25], and ELiPS [26]. We evaluate them across various metrics including hash-to-curve, pairing, exponentiation, scalar multiplication domains, security level, type of pairing, etc.

Our findings, as summarized in Table 3.1, show that some important tasks like hash-to-curve, pairing, exponentiation, and scalar multiplication are slower in PBC and RELIC compared to MCL and ELiPS. Since our goal is to improve the CP-ABE method, relying on PBC, we do not compare it to RELIC. Instead, we adopt ELiPS as it shows promise in our tests. We then utilize

Table 3.1: Comparison among pairing libraries [14, 15]

		PBC	**RELIC**	**MCL**	**ELiPS**
Security level		80-bit	128-bit	128-bit	128-bit
Hash-to-curve		3.2 [ms]	0.6 [ms]	0.3 [ms]	0.1 [ms]
Pairing		0.9 [ms]	2.6 [ms]	1.1 [ms]	2.2 [ms]
Exponentiation		0.1 [ms]	1.3 [ms]	0.8 [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]	0.7 [ms]	0.4 [ms]	0.5 [ms]
Type of pairing		I	III	III	III

ELiPS to enhance the CP-ABE method, calling it ELiPS-based CP-ABE. Our experiments demonstrate that ELiPS-based CP-ABE performs similarly to the MCL library, which is known for its effectiveness. This indicates that ELiPS could be a valuable option for enhancing this type of security method.

3.2 Access tree

In this section, we introduce an access tree, which plays a crucial role in access control for CP-ABE. Then, we present how to check if a user's attributes match an access tree and provide an example.

3.2.1 Define an access tree

An access tree is used to describe the access policy of an encrypted message [21]. For example, Figure 3.1 gives information about the access tree, which expresses the access policy as follows: (*Position: Professor* **OR** *Position: Researcher* **OR** *Position: Student*) **AND** (*Faculty: Engineering* **OR** *Faculty: Technology*).

Each non-leaf node of the access tree represents a threshold gate, described by its children and a threshold value. If num_x is the number of children of a non-leaf node x and k_x is its threshold value, then $0 < k_x \leq num_x$. For instance, two particular cases are **AND** and **OR** gates:

- **AND** gate: $k_x = num_x$.

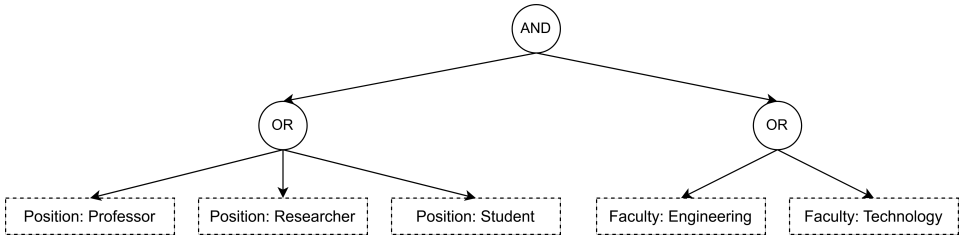


Figure 3.1: An example of a simple access tree $\mathcal{T}$ [15].

- **OR** gate: $k_x = 1$.

Every leaf node x of the access tree is described by an attribute and a threshold value $k_x = 1$.

Some functions are defined to facilitate working with access trees:

- par(x) : denotes the parent of the node x in the tree.
- att(x) : is defined only if x is a leaf node and denotes the attribute associated with the leaf node x in the tree.
- ind(x) : denotes the order of the node x between its brothers. The nodes are randomly numbered from 1 to num.

3.2.2 Satisfying an access tree

Let $\mathcal{T}$ be an access tree with root r. $\mathcal{T}_x$ denotes the subtree of $\mathcal{T}$, which has root at the node x. Hence the tree $\mathcal{T}$ is the same as the $\mathcal{T}_x$. If a set of attributes A satisfies the access tree $\mathcal{T}_x$, we denote it as $\mathcal{T}_x(A) = 1$, where A is a set of attributes, which is associated with the user's secret key. We compute $\mathcal{T}_x(A)$ recursively as follows:

- If x is a non-leaf node, evaluate $\mathcal{T}_{x'}(A)$ for all children x' of node x. $\mathcal{T}_x(A)$ returns 1 if and only if at least k_x children return 1.
- If x is a leaf node, then $\mathcal{T}_x(A)$ returns 1 if and only if att(x) $\in A$.

For instance, if the receiver/decryptor possesses a secret key with the attribute set *{Position: Researcher, Faculty: Engineering}*, it satisfies the access tree as described in Figure 3.1. However, if the receiver/decryptor possesses a secret key with the attribute set *{Position: Researcher, Faculty: Agriculture}*, it does not satisfy the access tree as described in Figure 3.1.

3.3 Ciphertext-Policy Attribute-Based Encryption algorithm

CP-ABE is an encryption scheme that provides fine-grained access control over encrypted data. In CP-ABE, data is encrypted based on a set of attributes, and access to the encrypted data is granted based on predefined access policies associated with those attributes [14]. This approach allows for flexible and customizable access control, where data owners can define specific attributes required for decryption [7].

CP-ABE offers several advantages in scenarios where access control needs to be managed carefully. It enables data sharing among multiple users or organizations while ensuring that the data can only be accessed by those with the necessary credentials. The usage of CP-ABE is particularly relevant in cloud services, Internet of Things environments, and scenarios involving sensitive data storage and communication [27]. By leveraging Attribute-Based Encryption, CP-ABE offers robust protection of data confidentiality and privacy [28]. It allows for secure data sharing, collaboration, and compliance with regulatory requirements.

The original CP-ABE implementation is based on the PBC library. In this work, we refer to it as PBC-based CP-ABE. The CP-ABE algorithm primarily relies on hash-to-curve and pairing procedures, comprising four main components.

3.3.1 Setup

The setup primitive is executed only once by the trusted party/server in the initial phase. This phase mainly uses scalar multiplication, pairing, and exponentiation operations for the computations. It outputs the master key MK and public key PK. Whereas the master key MK is kept secret, the public key PK is shared with all participants.

The algorithm begins by generating the $\mathbb{G}$ and $\mathbb{G}_T$ groups, where $\mathbb{G}$ has a generator g and both groups have an order r. Next, it randomly generates values α and $\beta \in \mathbb{Z}_r$. Then the master key MK and public key PK are

calculated as follows [6]:

$$\begin{aligned} MK &= (\beta, g^{\alpha}), \\ PK &= (\mathbb{G}, g, h, f, v), \end{aligned} \tag{3.1}$$

where: $h = g^{\beta}, f = g^{\beta^{-1}}, v = e(g,g)^{\alpha}, e$ is a bilinear map: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

3.3.2 Key generation

The key generation algorithm is also run once by the trusted party/server for each user. This phase primarily includes scalar multiplication and hash-to-curve operations. The algorithm takes the master key MK as well as the attribute set $A = \{att_1, att_2, ...\}$ as input. It proceeds to calculate the secret key SK, which is associated with the set of attributes A.

Firstly, the algorithm selects a random value $\gamma \in \mathbb{Z}_r$. Secondly, for each attribute $i \in A$, it selects a random value $\gamma_i \in \mathbb{Z}_r$. This part utilizes a hash function $\mathcal{H}$ to map each attribute into an element in $\mathbb{G} : \mathcal{H} = \{0,1\}^* \rightarrow \mathbb{G}$. Subsequently, the secret key SK is computed as [6]:

$$SK = (D, \{D_i, D_i'\}_{\forall i \in A}), \tag{3.2}$$

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

3.3.3 Encryption

This activity is executed by the sender/encryptor, who encrypts data on their devices. It primarily involves scalar multiplication and hash-to-curve operations. The encryption algorithm takes as input the public key PK, a message M, and an access policy $\mathcal{T}$ over the universe of attributes. It will encrypt message M and output a ciphertext CT such that only the receiver/decryptor who possesses a set of attributes associated with their secret key SK that satisfies the access tree $\mathcal{T}$ will be able to decrypt the message.

The encryption process is run as follows. A polynomial q_t is chosen for each node t in the access tree $\mathcal{T}$. The process chooses a random value $s \in \mathbb{Z}_r$,

starting with the root R node, setting $q_R(0) = s$. Then, for every node $t \in \mathcal{T}$, it sets $q_t(0) = q_{\text{par}(t)}(\text{ind}(t))$. The leaf nodes in $\mathcal{T}$ are denoted as $\mathcal{L}$, and the function $\text{att}(t)$ provides the attribute value of each leaf node in the access tree. The message M is encrypted using the access policy $\mathcal{T}$, as follows [6]:

$$CT = (\mathcal{T}, \tilde{C}, C, \{C_l, C'_l\}_{\forall l \in \mathcal{L}}), \tag{3.3}$$

where: $\tilde{C} = Me(g,g)^{\alpha s}, C = h^s, C_l = g^{q_l(0)}, C'_l = \mathcal{H}(\text{att}(l))^{q_l(0)}$.

3.3.4 Decryption

The algorithm is run by the receiver/decryptor to decrypt the encrypted message on the server. The server will check whether the user's attributes satisfy the access policy. If the user's attributes match the access policy, the decryption process is successful, and the user gains access to the message; otherwise, they are denied access. This stage primarily employs pairing and multiplication operations for computations. It takes as input ciphertext CT, which contains an access policy $\mathcal{T}$, and a secret key SK constructed from a list A of attributes. If the set A of attributes satisfies the access tree $\mathcal{T}$ then the algorithm will be able to decrypt the ciphertext and return a message M.

The algorithm computes dec_node(CT, SK, t), which receives CT, SK, and node t as input. If t is a leaf node, the attribute of node t is obtained as $i = \text{att}(t)$. Then, dec_node(CT, SK, t) is computed as [6]:

$$\text{dec\_node}(CT, SK, t) = \begin{cases} \frac{e(D_i, C_t)}{e(D'_i, C'_t)} & \text{if } i \in A, \\ null & \text{if } i \notin A. \end{cases} \tag{3.4}$$

The dec_node(CT, SK, t) function operates on leafless node t as follows: For each child node c of t, the algorithm calls dec_node(CT, SK, c) and stores the result in F_c. A_t is a list of children c, where $F_c \neq null$. If no such set exists, the function returns $null$. Otherwise, the following calculation is performed [6]:

$$\text{Let: } k = \text{ind}(c), \qquad A'_t = \{\text{ind}(c), \forall c \in A_t\},$$
$$\Delta_{k,A'_t(0)} = \prod_{j \in A'_t, j \neq k} \frac{-j}{k-j}, \tag{3.5}$$

$$\begin{aligned} F_t &= \prod_{c \in A_t} F_c^{\Delta_{k,A'_t(0)}} \\ &= \prod_{c \in A_t} \left(e(g,g)^{\gamma q_c(0)}\right)^{\Delta_{k,A'_t(0)}} \\ &= \prod_{c \in A_t} \left(e(g,g)^{\gamma q_{\text{par}(c)}(\text{ind}(c))}\right)^{\Delta_{k,A'_t(0)}} \\ &= \prod_{c \in A_t} e(g,g)^{\gamma q_t(k) \Delta_{k,A'_t(0)}} \\ &= e(g,g)^{\gamma q_t(0)}. \end{aligned} \tag{3.6}$$

If the set of attributes A match the tree access policy $\mathcal{T}$, the algorithm then calls the dec_node(CT, SK, R) function as follows [6]:

$$\tilde{A} = \text{dec\_node}(CT, SK, R) = e(g,g)^{\gamma q_R(0)} = e(g,g)^{\gamma s}. \tag{3.7}$$

Then, the ciphertext is decrypted using the following formula [6]:

$$\frac{\tilde{C}}{\frac{e(C,D)}{\tilde{A}}} = M. \tag{3.8}$$

3.4 Summary

This chapter introduced several prominent libraries for pairing systems, such as PBC, RELIC, MCL, and ELiPS. Then, we demonstrated a comparative analysis of these well-known libraries using terminology that are primarily used in the CP-ABE algorithm. We afterward presented the advantages of the CP-ABE algorithm and its primary functions. However, CP-ABE relies on the PBC library, which has not been updated for a significant amount of time and lacks sufficient security strength, making it vulnerable to various attacks and

unsuitable for modern cryptographic applications. In the next chapter, we introduce an ELiPS-based CP-ABE scheme to enhance the security level and increase the performance of CP-ABE by integrating ELiPS into the CP-ABE framework.

Chapter 4

An implementation of ELiPS-based Ciphertext-Policy Attribute-Based Encryption

In this chapter, we present the main procedures required to implement CP-ABE using ELiPS, which we refer to as ELiPS-based CP-ABE. However, PBC and ELiPS use several different operations and parameters. Additionally, while CP-ABE uses symmetric pairing, ELiPS utilizes asymmetric pairing. Therefore, before that, we introduce three procedures to make ELiPS appropriate for CP-ABE. These procedures include generating g, g_1, and g_2, as well as transforming asymmetric to symmetric pairing and modifying CP-ABE framework functions. Afterward, we experiment and evaluate the performance of the proposal. Firstly, we evaluate the efficacy of setup, key generation, encryption, and decryption in PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE with a two-attribute scenario. Secondly, we validate the key generation, encryption, and decryption parts with an increasing number of attributes. On the other hand, the primary findings of this chapter were published in the CANDAR 2023 conference [14] and the IJNC journal [15].

4.1 Introduction

The Ciphertext-Policy Attribute-Based Encryption [6] is an advanced cryptographic protocol that safeguards privacy data in environments such as cloud storage [29] and the Internet of Things [30]. Data is encrypted and protected based on an access policy. Only users who possess keys with attributes that satisfy the access policy can access and decrypt the encrypted data.

Cloud computing enables the storage and remote access of data via the internet. However, issues with access control and privacy arise when data is stored by a third party. On the other hand, IoT is a rapidly developing technology in the modern digital era. The large amounts of data generated by the expanding IoT have led to a greater focus on privacy and data access control in security. To meet these requirements, CP-ABE is utilized to provide

privacy and fine-grained access control in both cloud storage [31–35] and IoT applications [36–40].

However, the CP-ABE employed on the PBC library [22], which has not been updated for a significant period and lacks sufficient security strength, may pose a potential weakness in modern cryptographic applications. The PBC library supports only an 80-bit security level, rendering it vulnerable to various attacks and limiting its practicality. Bos et al. [8] recommend that transitioning to a security level greater than 80-bit is necessary. According to Barker [9], an 80-bit of security is no longer regarded as being sufficiently secure.

On the other hand, the ELiPS[1] library provides efficient calculation costs while ensuring high security. It is a specialized cryptographic library that concentrates on efficient operations related to pairing-based cryptography [10]. ELiPS utilizes the BLS-12 curve and offers a 128-bit security level [11]. It provides several functions that support the implementation of algorithms and protocols that utilize pairing. Additionally, the ELiPS library has not only been used for the implementation of Pairing-based Homomorphic Encryption by Kanenari et al. [12] but also has been utilized in the implementation of zk-SNARKs.

To deal with these challenges, we propose an ELiPS-based CP-ABE scheme, integrating ELiPS into the CP-ABE framework [14]. Despite differences between PBC and ELiPS libraries, including function parameters and data types, as well as the type of pairing, we establish compatibility by generating a generator g and employing Shirase's method [13] to convert asymmetric to symmetric pairing. After that, we make essential modifications to ensure the integration of ELiPS into the CP-ABE framework. These modifications span the setup, key generation, encryption, and decryption algorithms.

To validate the proposal, we conducted several experiments, utilizing a data access authorization process at the university level with various attribute policy scenarios. The analysis and experimental results demonstrate the effectiveness of the proposal, revealing not only an increase in the security level but also a reduction in computational requirements for the setup, key generation,

1. *ELiPS*. Information Security laboratory Okayama University. https://github.com/ISecOkayamaUniv/ELiPS

and encryption functions. Moreover, we compare our ELiPS-base solution with MCL to see if it works as well and stays up to date with today's security needs. This comparison confirms that our approach meets today's security standards while operating efficiently.

The following are some of the main contributions of this chapter:

- Introduce generator g over $E(\mathbb{F}_{p^{12}})$ specifically designed to convert asymmetric pairing to symmetric pairing, which is a significant contribution in addressing the compatibility issue between ELiPS and the original CP-ABE scheme.
- Employ Shirase's technique [13] to accomplish the conversion of asymmetric pairing to symmetric pairing using the generated generator g.
- Make several modifications to the CP-ABE framework and carefully select appropriate ELiPS functions to ensure compatibility of ELiPS with CP-ABE.
- Evaluate the proposal in terms of computation time.
- Compare our proposal's performance with other competitive pairing libraries.

4.2 Proposed schemes

In this section, we present the main procedures required to implement CP-ABE using ELiPS, which we refer to as ELiPS-based CP-ABE. PBC and ELiPS use several different function names, as shown in Table 4.1. Therefore, we have

Table 4.1: Comparison of the main function names used for implementing CP-ABE between the PBC and ELiPS libraries [15]

	PBC	**ELiPS**
Function names in $\mathbb{G}_T$	element_mul element_pow_zn	g3_mul g3_exp
Function names in $\mathbb{G}_1, \mathbb{G}_2$	element_mul element_pow_zn	g1_eca, g2_eca g1_scm, g2_scm
Type of pairings	Symmetric pairing	Asymmetric pairing

designed three procedures to make ELiPS appropriate for CP-ABE. These procedures include generating g, g_1, and g_2, as well as transforming asymmetric to symmetric pairing and modifying CP-ABE framework functions.

4.2.1 Generator g generation

Generating a generator g serves the purpose of transforming asymmetric pairing, which is the basis of ELiPS, into symmetric pairing. This transformation is crucial for compatibility with the original CP-ABE, which relies on symmetric pairing.

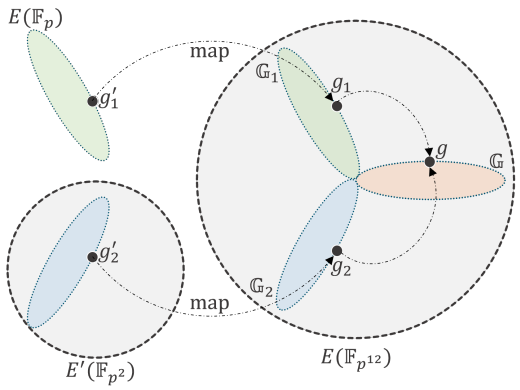


Figure 4.1: The process for generating g, g_1, and g_2.

Figure 4.1 illustrates the process of generating g, as follows:

1. Firstly, the algorithm generates two generators g_1' and g_2' over $E(\mathbb{F}_p)$ and $E'(\mathbb{F}_{p^2})$, respectively.

2. Secondly, g_1' and g_2' are mapped to g_1 in subgroup $\mathbb{G}_1$ and g_2 in subgroup $\mathbb{G}_2$, respectively. Since $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of $E[r]$, we can add the elements of $\mathbb{G}_1$ and $\mathbb{G}_2$, and the result is an element of $E[r]$.

3. Finally, the generator g of group $\mathbb{G}$ be generated as follows:

$$g = g_1 + g_2. \tag{4.1}$$

This method is significantly faster compared to directly creating a generator over $E(\mathbb{F}_{p^{12}})$. Since $\mathbb{G}$ is a subgroup of order r of $E[r](\subset E(\mathbb{F}_{p^{12}}))$,

addition and scalar multiplication can be defined over $\mathbb{G}$ in the same way as those on $E(\mathbb{F}_{p^{12}})$ [13].

Algorithm 1 demonstrates the process of generating a generator $g \in \mathbb{G}$ over $E(\mathbb{F}_{p^{12}})$.

Algorithm 1 Generate a generator g

Input:

Output: $g \in \mathbb{G}$ over $E(\mathbb{F}_{p^{12}})$

1: $g_1' \leftarrow$ efp_generate() // g_1' over $E(\mathbb{F}_p)$
2: $g_2' \leftarrow$ efp2_generate() // g_2' over $E'(\mathbb{F}_{p^2})$
3: $g_1 \leftarrow$ g1_map_from_efp(g_1') // $g_1 \in \mathbb{G}_1$ over $E(\mathbb{F}_{p^{12}})$
4: $g_2 \leftarrow$ g2_map_from_efp2(g_2') // $g_2 \in \mathbb{G}_2$ over $E(\mathbb{F}_{p^{12}})$
5: $g \leftarrow g_1 + g_2$ // $g \in \mathbb{G}$ over $E(\mathbb{F}_{p^{12}})$
6: return g

4.2.2 Asymmetric to symmetric transformation

We successfully implemented Shirase's method [13] for converting asymmetric pairing to symmetric pairing. Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be generator rational points. Then g is a generator point of $\mathbb{G}$, and this can be calculated as shown in Equation (4.1).

For two rational points $P, Q \in \mathbb{G}$ and we can use symmetric pairing $e_{\text{sym}}(Q, P)$ by defining a symmetric pairing as follows [13]:

$$e_{\text{sym}} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T.$$

Since ELiPS uses asymmetric pairing, we need to transform asymmetric pairing into symmetric pairing. This is done by extracting P' in group $\mathbb{G}_1$ and Q' in group $\mathbb{G}_2$ from P and Q, respectively. Figure 4.2 shows the concept of the extraction procedure. The transformation between asymmetric and symmetric pairing can be defined as [13]:

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

Next, we provide a method for extracting P' in group $\mathbb{G}_1$ and Q' in group

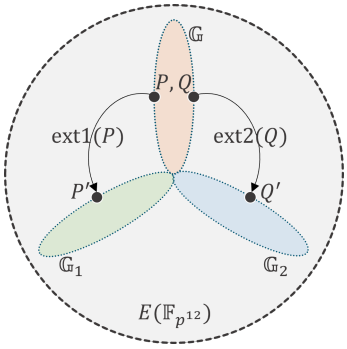


Figure 4.2: Extraction of P' and Q' in transforming asymmetric pairing to symmetric pairing.

$\mathbb{G}_2$ from P and Q in group $\mathbb{G}$, respectively.

Let $l = (p-1)^{-1} \pmod{r}$, 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 [13]:

$$\begin{cases} \text{ext1} = ([p] - \pi_p)[l], \\ \text{ext2} = (\pi_p - [1])[l]. \end{cases} \tag{4.2}$$

The symmetric pairing procedure 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 Figure 4.2.

$$\begin{aligned} P' &= \text{ext1}(P), \\ Q' &= \text{ext2}(Q), \end{aligned} \tag{4.3}$$

   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. The asymmetric pairing function uses the Miller loop and final exponentiation to calculate and return the pairing value.

Algorithm 2 illustrates the detailed computation process of transforming

an asymmetric pairing to a symmetric pairing.

Algorithm 2 Asymmetric pairing to symmetric pairing transformation

Input: $Q, P \in \mathbb{G}$

Output: $e \in \mathbb{F}_{p^{12}}$ //Asymmetric pairing result

1: $r' \leftarrow (p-1)^{-1} \pmod{r}$ // r be order of groups $\mathbb{G}_1$ and $\mathbb{G}_2$
2: $P' \leftarrow (p - \pi_p) \times r' \times P$ // This is ext1 function, π_p be Frobenius map
3: $Q' \leftarrow (\pi_p - 1) \times r' \times Q$ // This is ext2 function
4: $e \leftarrow e_{\text{asy}}(Q', P')$ // $Q' \in \mathbb{G}_2$ and $P' \in \mathbb{G}_1$
5: return e

4.3 CP-ABE algorithm modifications

We present some modifications to enable ELiPS to work within the CP-ABE framework. Then, we conduct a security analysis on ELiPS-based CP-ABE, showing its alignment with today's security needs.

4.3.1 Setup

The setup is executed once on the server at the beginning of the system. The algorithm generates g_1' and g_2' over $E(\mathbb{F}_p)$ and $E'(\mathbb{F}_{p^2})$, respectively. Then, it maps g_1' and g_2' to g_1 and g_2 over $E(\mathbb{F}_{p^{12}})$, as illustrated in Figure 4.1. The generator point g is calculated using the formula $g = g_1 + g_2$. Next, random $\alpha, \beta \in \mathbb{Z}_r$ are generated. The master key MK and public key PK are computed as follows [14]:

$$\begin{aligned} MK &= (\beta, \alpha g), \\ PK &= (g, h, f, v), \end{aligned} \tag{4.4}$$

where: $h = \beta g, f = \beta^{-1} g, v = e(g, g)^{\alpha}$.

The master key MK is kept secret, whereas the public key PK is shared with everyone.

4.3.2 Key generation

This function is run once on the server for each user as well. The key generation takes the master key MK and attribute set A as input. It calculates the secret key SK as [14]:

$$SK = (D, \{D_i, D_i'\}_{\forall i \in A}), \tag{4.5}$$

where: $D = (\alpha + \gamma)\beta^{-1}g, D_i = \gamma g + \gamma_i \mathcal{H}(i), D_i' = \gamma_i g, \gamma$ and γ_i are random numbers over $\mathbb{Z}_r$.

Then, the secret key SK is provided to the user.

4.3.3 Encryption

The encryption function is executed every time to encrypt data on the user's own devices. It takes the public key PK, message M, and access policy tree $\mathcal{T}$ as input. The output is the ciphertext CT, which is computed as follows [14]:

$$CT = (\mathcal{T}, \tilde{C}, C, \{C_l, C_l'\}_{\forall l \in \mathcal{L}}), \tag{4.6}$$

where: $\tilde{C} = Me(g,g)^{\alpha s}, C = sh, C_l = q_l(0)g, C_l' = q_l(0)\mathcal{H}(\text{att}(l)), s$ is a random number over $\mathbb{Z}_r, \mathcal{L}$ is the leaf node set in $\mathcal{T}$.

4.3.4 Decryption

The decryption phase is run on the server. Firstly, it verifies the user's attributes and access policy. If the user's attributes meet the access policy, the decryption process is successful, and then the user can access the plain message. Otherwise, access is denied. The inputs for the procedure are ciphertext CT and secret key SK. It calls the dec_node(CT, SK, R) function to calculate $\tilde{A}$ as [14]:

$$\tilde{A} = \text{dec\_node}(CT, SK, R) = e(g,g)^{\gamma s}. \tag{4.7}$$

In this function, the algorithm calls the recursive dec_node(CT, SK, t)

function to calculate the value $\tilde{A}$ and verify whether the secret key SK matches the access policy, where t is a leaf node, as follows [14]:

$$\text{dec\_node}(CT, SK, t) = \begin{cases} \frac{e_{\text{sym}}(D_i, C_t)}{e_{\text{sym}}(D'_i, C'_t)} & \text{if } i \in A, \\ null & \text{if } i \notin A. \end{cases} \tag{4.8}$$

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 children c, where $F_c \neq null$. If no such set exists, the function returns $null$. Otherwise, the following calculation is performed:

$$\begin{aligned} &\text{Let: } k = \text{ind}(c), \qquad A'_t = \{\text{ind}(c), \forall c \in A_t\}, \\ &\Delta_{k, A'_t(0)} = \prod_{j \in A'_t, j \neq k} -j(k-j)^{-1}, \end{aligned} \tag{4.9}$$

$$\begin{aligned} F_t &= \prod_{c \in A_t} F_c^{\Delta_{k, A'_t(0)}} \\ &= \prod_{c \in A_t} \left(e(g,g)^{\gamma q_c(0)}\right)^{\Delta_{k, A'_t(0)}} \\ &= \prod_{c \in A_t} \left(e(g,g)^{\gamma q_{\text{par}(c)}(\text{ind}(c))}\right)^{\Delta_{k, A'_t(0)}} \\ &= \prod_{c \in A_t} e(g,g)^{\gamma q_t(k) \Delta_{k, A'_t(0)}} \\ &= e(g,g)^{\gamma q_t(0)}. \end{aligned} \tag{4.10}$$

The original message is decrypted using the following formula [14]:

$$\frac{\tilde{C}\tilde{A}}{e_{\text{sym}}(C, D)} = \frac{\tilde{C}\tilde{A}}{e_{\text{asy}}(\text{ext2}(C), \text{ext1}(D))} = M. \tag{4.11}$$

4.3.5 Security analysis

Numerous cryptographic protocols rely on computational assumptions to demonstrate their security. Mrabet et al. [19] noted that pairings of Type III

are compatible with various computational assumptions, such as the Decision Diffie-Hellman in $\mathbb{G}_1$ or $\mathbb{G}_2$, also referred to as the External Diffie-Hellman assumption, which is not upheld in Type I pairings [20]. While the ELiPS-based CP-ABE relies on ELiPS, employing asymmetric pairing (Type III), the PBC-based CP-ABE utilizes symmetric pairing (Type I). This implies that ELiPS-based CP-ABE is better than PBC-based CP-ABE from a security perspective.

Additionally, according to Equation (4.6) and Equation (4.11), to decrypt encrypted data, one needs to calculate the value of $e(g,g)^{\alpha s}$ or $e(C,D)/e(g,g)^{\gamma s}$, as follows:

- Recovering the value $e(g,g)^{\alpha s}$ requires attackers to determine α and s. However, based on the Discrete Logarithm Problem and the Elliptic Curve Discrete Logarithm Problem, computing α from $d = \alpha g$ or $v = e(g,g)^{\alpha}$ and s from $C = sh$ is infeasible.

- Calculating the value $e(C,D)/e(g,g)^{\gamma s}$ allows adversaries to compute $e(C,D)$ 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.

  where $g, h \in E$, and $\alpha, \beta, \gamma, s \in \mathbb{Z}_r$.

Computing discrete logarithms is evidently difficult, which is related to the bit length of r. In PBC-based CP-ABE, r is 160 bits, while in ELiPS-based CP-ABE, it is 308 bits. This demonstrates that the proposed scheme increases the security level.

4.4 Experimental evaluation and discussion

In this section, we experiment and evaluate the performance of the proposal. Firstly, we evaluate the efficacy of setup, key generation, encryption, and decryption in PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE with a two-attribute scenario. Secondly, we validate the key

generation, encryption, and decryption parts with an increasing number of attributes.

4.4.1 Experimental evaluation setup

Table 4.2 shows the devices and software used during the evaluation. In our experiments, we employed a data access authorization for administration procedures at the university level and attribute policy scenarios, as depicted in Figure 4.3, which involve three entities:

- University administrator: Authority.
- President: Sender.
- Professors: Receiver.

Table 4.2: Experimental environments

OS	Ubuntu 22.04.1 LTS (WSL2)
CPU	Intel(R) Core(TM) i7-6600U @ 2.60GHz
Memory	4 GB
Language	C
GMP version	6.2.1
GCC version	11.3.0
GCC optimization level	-O2

Assuming the university president wishes to share private data exclusively with professors in the Faculty of Engineering, the president only encrypts the data once and shares the encrypted data with all intended recipients. The president also needs to define an access policy $\mathcal{T}$ structure to determine who can decrypt the encrypted data, as shown in Figure 4.3. On the recipients' side, if their attributes satisfy the access policy, they can successfully decrypt the data; otherwise, they are unable to decrypt it.

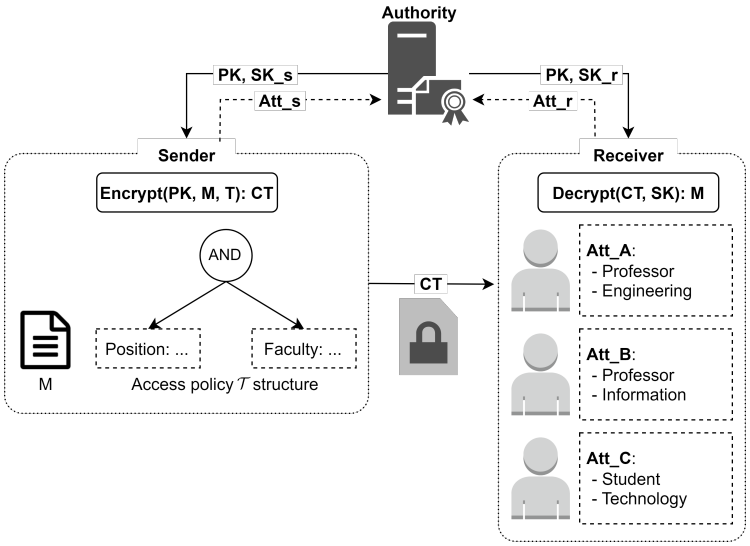


Figure 4.3: An example of data access authorization for administrative procedures at the university level.

4.4.2 Performance evaluation with two-attribute scenario

We employed two attributes to implement a data access authorization for the administration scenario. We performed 10,000 executions to measure the computation time of setup, key generation, encryption, and decryption functions for PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE, then we took the average values.

Table 4.3: A comparison among PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE in a two-attribute scenario

Functions	**PBC-based CP-ABE**	**MCL-based CP-ABE**	**ELiPS-based CP-ABE**
Setup	5.6 [ms]	4.0 [ms]	4.1 [ms]
Keygen	15.5 [ms]	3.9 [ms]	3.8 [ms]
Encryption	15.0 [ms]	3.9 [ms]	3.7 [ms]
Decryption	7.3 [ms]	9.6 [ms]	11.0 [ms]

Table 4.3 summarizes the comparison results. Overall, it shows that most of the functions in ELiPS-based CP-ABE perform faster than their counterparts in PBC-based CP-ABE, except for the decryption function. The per-

formance of both ELiPS-based CP-ABE and MCL-based CP-ABE are closely competitive, with no significant difference between them. It shows that our ELiPS-based solution is working well and operating efficiently.

In detail, Table 4.3 shows that while the setup speed in ELiPS-based CP-ABE is faster than that in PBC-based CP-ABE by 26.8%, and MCL-based CP-ABE is faster than that in PBC-based CP-ABE by 28.6%.

In addition, the data illustrates that the key generation performance in MCL-based CP-ABE is better than that in PBC-based CP-ABE by 74.8%, while the key generation in ELiPS-based CP-ABE is better than other schemes by 2.6% compared to MCL-based CP-ABE and by 75.5% compared to PBC-based CP-ABE.

Regarding the encryption part, Table 4.3 shows that encryption time in ELiPS-based CP-ABE is the best among three versions, namely PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE. Whereas encryption time in MCL-based CP-ABE decreases by 74.0%, encryption time in ELiPS-based CP-ABE reduces by 75.3% compared to that in PBC-based CP-ABE.

On the other hand, the decryption time for MCL-based CP-ABE and ELiPS-based CP-ABE increases by 31.5% and 50.7%, respectively, compared to the decryption time for PBC-based CP-ABE. Therefore, further evaluation with an increase in the number of attributes is necessary.

4.4.3 Evaluating the key generation, encryption, and decryption with an increasing number of attributes

Since the setup part is not affected by the number of attributes, we do not need to evaluate it further. Instead, we focus on experiments and evaluations of key generation, encryption, and decryption with varying numbers of attributes.

We conducted experiments 10,000 times to measure the key generation time, encryption time, and decryption time in PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE. We then calculated the average results.

Figure 4.4 shows the key generation performance in MCL-based CP-ABE is better than that in PBC-based CP-ABE by 74.7%, while the key generation

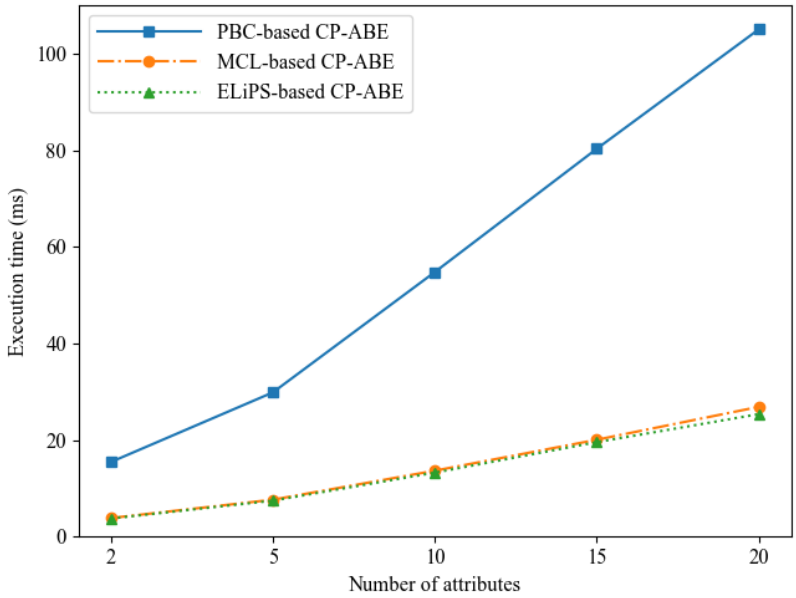


Figure 4.4: Key generation time for several scenarios of PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE.

in ELiPS-based CP-ABE is better than other schemes by 3.7% compared to MCL-based CP-ABE and by 75.6% compared to PBC-based CP-ABE. The increase in performance of key generation in MCL-based CP-ABE and ELiPS-based CP-ABE can be attributed to the fact that the key generation algorithm primarily utilizes hash-to-curve and SCM operations for each attribute, as illustrated in Table 4.4. These operations exhibit superior performance in MCL and ELiPS compared to PBC, as indicated in Table 3.1.

Table 4.4: Computations cost in CP-ABE algorithm

Functions	**Computational cost**
Key generation	$1\mathcal{J} + n\mathcal{H} + (n+1)\mathcal{A} + 2(n+1)\mathcal{S}$
Encryption	$1\mathcal{M} + 1\mathcal{E} + n\mathcal{H} + (2n+1)\mathcal{S}$
Decryption	$2\mathcal{M} + 1\mathcal{J} + (2n+1)\mathcal{P}$

where: $\mathcal{M}$ is the multiplication cost over $\mathbb{F}_{p^{12}}$, $\mathcal{E}$ is the exponentiation cost over $\mathbb{F}_{p^{12}}$, $\mathcal{J}$ is the inversion cost over $\mathbb{F}_{p^{12}}$, $\mathcal{H}$ is the hash-to-curve cost, $\mathcal{P}$ is the pairing cost, $\mathcal{A}$ is the elliptic curve addition cost, $\mathcal{S}$ is the elliptic curve scalar multiplication cost, n is the number of attributes.

Figure 4.5 shows a similar trend to the key generation part. Encryption time in ELiPS-based CP-ABE is the best among the three versions. Encryp-

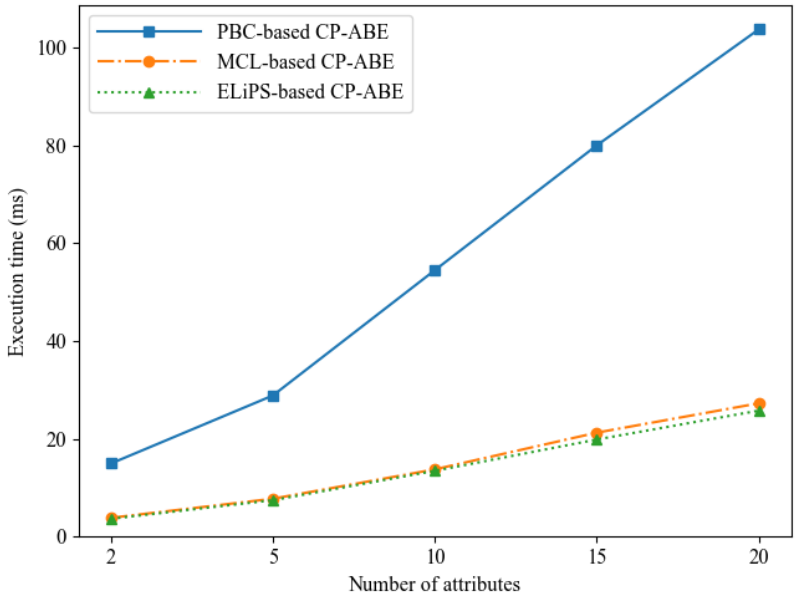


Figure 4.5: Encryption time for several scenarios of PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE.

tion time in ELiPS-based CP-ABE decreases by 75.0% compared to that in PBC-based CP-ABE and reduces by 4.9% compared to that in MCL-based CP-ABE. Similar to the key generation algorithm, Table 4.4 shows that the encryption algorithm primarily utilizes hash-to-cure and SCM operations, which are employed for each attribute in the access policy. The results in Table 3.1 indicate that the computational cost of these operations in MCL and ELiPS significantly reduces compared to that in PBC. Hence, the encryption time in the proposal decreases by around 3.9-fold compared to that in PBC-based CP-ABE.

Figure 4.6 demonstrates that the decryption time of both MCL-based CP-ABE and ELiPS-based CP-ABE is higher than that of the PBC-based CP-ABE across scenarios. The decryption time increases linearly as the number of attributes increases. As indicated in Table 4.4, the number of pairing operations depends on the number of attributes. Additionally, the pairing cost in both MCL and ELiPS is heavier than that in PBC, as demonstrated in Table 3.1.

In IoT scenarios, efficient encryption is crucial for resource-constrained IoT devices acting as senders. While the setup and key generation are one-time operations performed on the server, encryption and decryption must be

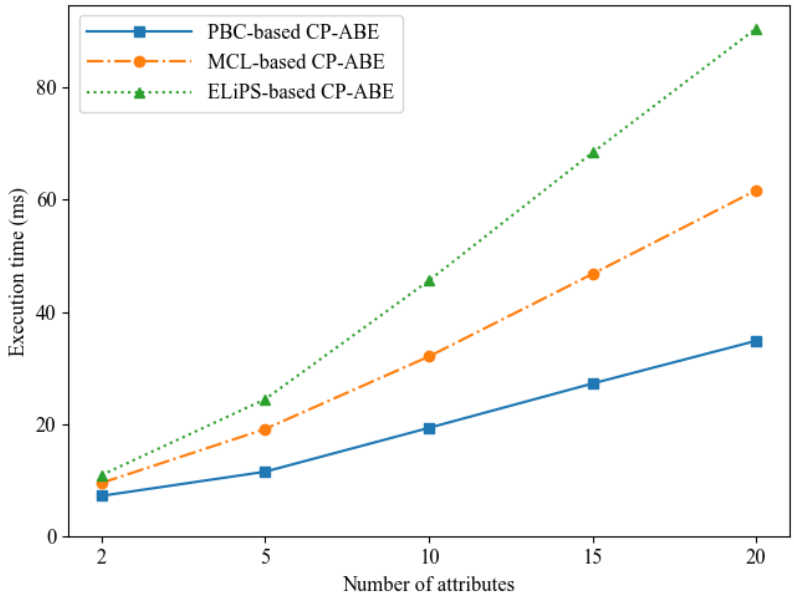


Figure 4.6: Decryption time for several scenarios of PBC-based CP-ABE, MCL-based CP-ABE, and ELiPS-based CP-ABE.

executed repeatedly for securing and accessing data. According to the results in Table 4.3 and Figure 4.5, the ELiPS-based CP-ABE scheme outperforms other compared schemes like PBC-based CP-ABE and MCL-based CP-ABE in terms of encryption performance. Since IoT devices have limited resources, the superior encryption efficiency of ELiPS-based CP-ABE makes it suitable for practical implementation in IoT scenarios where the encryption is handled by the devices, and the server handles the decryption.

On the other hand, the PBC library utilizes symmetric pairing, while the ELiPS library uses asymmetric pairing (Type III). The symmetric pairing is not robust enough from a security point of view [19]. Pairings categorized as Type III align with various computational assumptions, including the Decision Diffie-Hellman assumption in $\mathbb{G}_1$ or $\mathbb{G}_2$, also referred to as the External Diffie-Hellman assumption, which does not hold in Type I pairings [19]. Therefore, the ELiPS-based solution is more compatible with various computational assumptions than the PBC-based CP-ABE.

Moreover, the security levels for ELiPS and PBC are different. Comparing them will be more appropriate when PBC-based CP-ABE and ELiPS-based CP-ABE use the same security level.

4.5 Summary

We introduced an ELiPS-based CP-ABE scheme by integrating ELiPS as an efficient library for pairing systems into the CP-ABE encryption method. Here, generating a generator g served the purpose of transforming asymmetric pairing, which was the basis of ELiPS, into symmetric pairing. This transformation was crucial for compatibility with the original CP-ABE, which relied on symmetric pairing. Then, we transformed asymmetric pairing to symmetric pairing using Shirase's technique and made several modifications to the CP-ABE framework for the integration. Comparing ELiPS-based CP-ABE with MCL showed that our ELiPS-based solution functioned efficiently while matching today's security needs. Additionally, the experimental results confirmed that the proposal not only improved the performance but also boosted the security level to 128 bits for CP-ABE. The superior encryption efficiency of ELiPS-based CP-ABE made it suitable for practical implementation in IoT scenarios where the encryption was handled by the devices, and the server handled the decryption. On the other hand, the decryption process remains heavy. Therefore, in the next study, we aim to improve the decryption process in ELiPS-based CP-ABE.

Chapter 5

Improvement decryption process in ELiPS-based CP-ABE

In this chapter, we propose two methods to reduce the number of final exponentiations and inversions, which improve decryption process time in ELiPS-based CP-ABE. Then, we validate the correctness of the proposed formulas and their performance. Additionally, we compare the performance of our proposed decryption method in ELiPS-based CP-ABE with the initial version of ELiPS-based CP-ABE. The primary contributions in this chapter have been published in the JAIT journal [16].

5.1 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 [14]. CP-ABE finds applications in diverse fields, including but not limited to cloud storage [33–35], Internet of Things [37–40], Personal Health Record [41–43], blockchain [44–46], and other prominent fields [47–50].

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 [33–35].

The growing proliferation of the Internet of Things generates vast amounts of data, leading to an increased emphasis on data access control in security [37–40]. CP-ABE meets this demand by allowing data sources to encrypt data while enforcing a security access policy cryptographically.

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 [41–43].

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 [44–46].

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.

Therefore, in our previous study [14, 15], we addressed these shortcomings by proposing an ELiPS-based CP-ABE scheme, which enhanced security strength and increased performance. Previous work in [14,15] integrated ELiPS into the CP-ABE framework. ELiPS serves as a foundation for cryptosystems and offers a 128-bit security level. We proposed several methods to improve the performance of the CP-ABE [14, 15]. The results indicated that 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, we aim to reduce the decryption processing time of ELiPS-based CP-ABE 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 45.5% compared to previous work [14].

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

- Transform the decryption equation to a Miller loop and final exponentiation bases.

- The number of final exponentiations is proportional to the number of attributes.
- Transform the decryption equation by performing final exponentiation only once at the last step.
- The count of inversion operations in the Lagrange coefficient function is minimized by employing a single inversion operation.

5.2 Proposed methods

The objective of this study is to reduce the decryption processing time of ELiPS-based CP-ABE. To accomplish this goal, we propose two methods as:

5.2.1 Minimizing number of final exponentiations

For data decryption, in accordance with Equation (4.8), the algorithm conducts a pair of pairing calculations for each attribute. The pairing operation includes the Miller loop and final exponentiation. Therefore, we propose to transform the decryption equation to Miller loop and final exponentiation as follows [16]:

$$\prod_{i=1}^{n}\left[\frac{e(D_i,C_i)}{e(D_i',C_i')}\right]^{\Delta_i}=\prod_{i=1}^{n}\left[\frac{(f_{D_i,C_i})^{\frac{p^k-1}{r}}}{(f_{D_i',C_i'})^{\frac{p^k-1}{r}}}\right]^{\Delta_i}. \tag{5.1}$$

where:

- n is the number of attributes.
- $f_{P,Q}$ be a Miller loop result with P and Q on an elliptic curve as inputs.
- $f_{P,Q}^{\frac{p^k-1}{r}}$ is the final exponentiation.

- Δ_i is the Lagrange coefficient, $\Delta_i \in \mathbb{Z}_r$,

$$\Delta_i = \sum_{j=1, j\neq i}^{n} \frac{-j}{i-j} = \sum_{j=1, j\neq i}^{n} \Big[-j(i-j)^{-1}\Big]. \tag{5.2}$$

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

The pairing e is formed through the Miller loop and final exponentiation. Thus, in Equation (5.1), both the Miller loop and final exponentiation are utilized for each pairing per attribute. Therefore, we propose a formula aiming to decrease the number of final exponentiations. From Equation (5.1), we propose a formula transformation as follows [16]:

$$\begin{aligned}
\prod_{i=1}^{n} \left[\frac{e(D_i, C_i)}{e(D_i', C_i')}\right]^{\Delta_i} &= \prod_{i=1}^{n} \left[\frac{(f_{D_i,C_i})^{\frac{p^k-1}{r}}}{(f_{D_i',C_i'})^{\frac{p^k-1}{r}}}\right]^{\Delta_i} \\
&= \prod_{i=1}^{n} \left[\left(\frac{f_{D_i,C_i}}{f_{D_i',C_i'}}\right)^{\frac{p^k-1}{r}}\right]^{\Delta_i} \\
&= \prod_{i=1}^{n} \left[\left(\frac{f_{D_i,C_i}}{f_{D_i',C_i'}}\right)^{\Delta_i}\right]^{\frac{p^k-1}{r}} \\
&= \left[\prod_{i=1}^{n} \left(\frac{f_{D_i,C_i}}{f_{D_i',C_i'}}\right)^{\Delta_i}\right]^{\frac{p^k-1}{r}}.
\end{aligned} \tag{5.3}$$

In Equation (5.3), 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 last step.

Consequently, Equation (5.1) and Equation (5.3) demonstrate that our proposed method effectively reduces the number of final exponentiations by $2n-1$ times, improving the efficiency of ELiPS-based CP-ABE.

The method for minimizing the number of final exponentiations is also shown in Algorithm 3.

Algorithm 3 Minimizing the number of final exponentiations in the decryption function

Input: $n \in \mathbb{Z}_r, C, D, C', D'$

Output: $F = \left[\prod_{i=1}^{n} \left(\frac{f_{D_i,C_i}}{f_{D'_i,C'_i}}\right)^{\Delta_i}\right]^{\frac{p^k-1}{r}}$

1: $F \leftarrow 1$
2: $i \leftarrow 1$
3: **while** $i \leq n$ **do**
4: $\quad F = F \times \left(\frac{f_{D_i,C_i}}{f_{D'_i,C'_i}}\right)^{\Delta_i}$ // $f_{P,Q}$ be a Miller loop result,
5: // Δ_i be the Lagrange coefficient.
6: $\quad i \leftarrow i + 1$
7: **end while**
8:
9: $F \leftarrow F^{\frac{p^k-1}{r}}$
10: return F

5.2.2 Minimizing number of inversions

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

$$\Delta_i = \sum_{j=1, j\neq i}^{n} \Big[-j(i-j)^{-1} (\text{mod } r)\Big]. \tag{5.4}$$

According to Equation (5.4), the inversion operation is used in the Lagrange coefficient part for every number of attributes. Consequently, in this study, we propose a method to improve the efficiency of the decryption part by minimizing the number of inversions as follows [16]:

1. Calculate product:

$$\mathcal{A}_i = \prod_{j=1, j\neq i}^{n} (i-j) (\text{mod } r). \tag{5.5}$$

2. Calculate inverse of $\mathcal{A}_i$:

$$\mathcal{B}_i = \mathcal{A}_i^{-1} = \frac{1}{\mathcal{A}_i} (\text{mod } r). \tag{5.6}$$

3. We can calculate the inversion as follows:

$$\mathcal{C}_i = \sum_{j=1, j \neq i}^{n} -j\mathcal{B}_i \prod_{k=1, k \neq i, k \neq j}^{n} (i-k) (\text{mod } r). \tag{5.7}$$

Algorithm 4 describes the calculation of our proposal, which involves minimizing the number of inversions in the Lagrange coefficient in the decryption function.

The calculation in Equation (5.7) 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 Equation (5.4).

Algorithm 4 Minimizing the number of inversions in Lagrange coefficient in decryption function

Input: $i \in \mathbb{Z}_r, n \in \mathbb{Z}_r$
Output: Δ_i

$\Delta_i \leftarrow 0$
$\mathcal{A}_i \leftarrow 1$
$j \leftarrow 1$
while $j < i$ **do**
    $\mathcal{A}_i \leftarrow \mathcal{A}_i \times (i-j)$
    $j \leftarrow j+1$
end while
$j \leftarrow i+1$
while $j < n$ **do**
    $\mathcal{A}_i \leftarrow \mathcal{A}_i \times (i-j)$
    $j \leftarrow j+1$
end while
$\mathcal{A}_i^{-1} \leftarrow 1/\mathcal{A}_i \pmod{r}$
$j \leftarrow 1$

15: **while** $j < i$ **do**
16: $(i-j)^{-1} \leftarrow 1$
17: $k \leftarrow 1$
18: **while** $k < j$ **do**
19: $(i-j)^{-1} \leftarrow (i-j)^{-1} \times (i-k)$
20: $k \leftarrow k+1$
21: **end while**
22: $k \leftarrow j+1$
23: **while** $k < i$ **do**
24: $(i-j)^{-1} \leftarrow (i-j)^{-1} \times (i-k)$
25: $k \leftarrow k+1$
26: **end while**
27: $k \leftarrow i+1$
28: **while** $k < n$ **do**
29: $(i-j)^{-1} \leftarrow (i-j)^{-1} \times (i-k)$
30: $k \leftarrow k+1$
31: **end while**
32: $\Delta_i = \Delta_i + (i-j)^{-1}$
33: $j \leftarrow j+1$
34: **end while**
35: $j \leftarrow i+1$
36: **while** $j < n$ **do**
37: $(i-j)^{-1} \leftarrow 1$
38: $k \leftarrow 1$
39: **while** $k < i$ **do**
40: $(i-j)^{-1} \leftarrow (i-j)^{-1} \times (i-k)$
41: $k \leftarrow k+1$
42: **end while**
43: $k \leftarrow i+1$
44: **while** $k < j$ **do**
45: $(i-j)^{-1} \leftarrow (i-j)^{-1} \times (i-k)$
46: $k \leftarrow k+1$
47: **end while**
48: $k \leftarrow j+1$
49: **while** $k < n$ **do**
50: $(i-j)^{-1} \leftarrow (i-j)^{-1} \times (i-k)$
51: $k \leftarrow k+1$
52: **end while**
53: $\Delta_i \leftarrow \Delta_i + (i-j)^{-1}$
54: $j \leftarrow j+1$
55: **end while**
56: return Δ_i

5.3 Evaluation and discussion

In this section, we present a brief comparison of the decryption costs among CP-ABE schemes. 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 [14].

5.3.1 Decryption cost

Here, we present a comparison of decryption costs among CP-ABE schemes, as shown in Table 5.1. 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. [6] and [14]. When compared to Refs. [51], 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 5.1 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 5.1: Decryption cost comparison among CP-ABE schemes

Schemes	**No. of Final Exponentiations**	**No. of Inversions**
[6]	$2n+1$	$n+1$
[51]	$n+2$	2
[14]	$2n+1$	$n+1$
Proposal	2	2

Note: n is the number of attributes.

5.3.2 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. We implemented Equation (5.1) and Equa-

tion (5.3) 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 4.2.

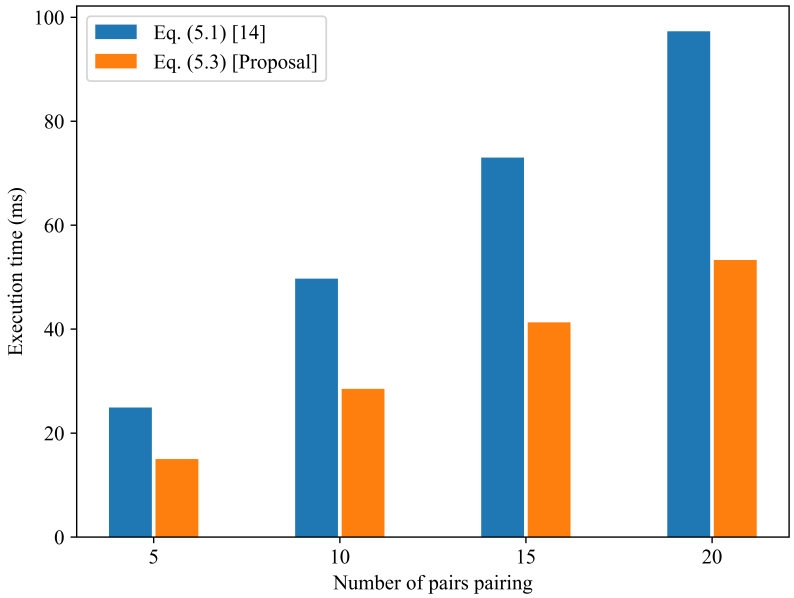


Figure 5.1: Comparison of execution time between Equation (5.1) and Equation (5.3).

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 Equation (5.1) is identical to the result of Equation (5.3). These results validate the correctness of our proposed formula. Moreover, as depicted in Figure 5.1, our proposed Equation (5.3) reduces the execution time by an average of 43.6% compared to the previous Equation (5.1).

5.3.3 Evaluation of the proposed formula, reducing the number of inversions

Secondly, we assess the correctness and performance of the method, which decreases the number of inversions. We implemented Equation (5.4) 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 and then took

the average execution time. The experimental results show that the inversion result of Equation (5.4) is identical to the result of Equation (5.7). These results demonstrate the correctness of our proposed method.

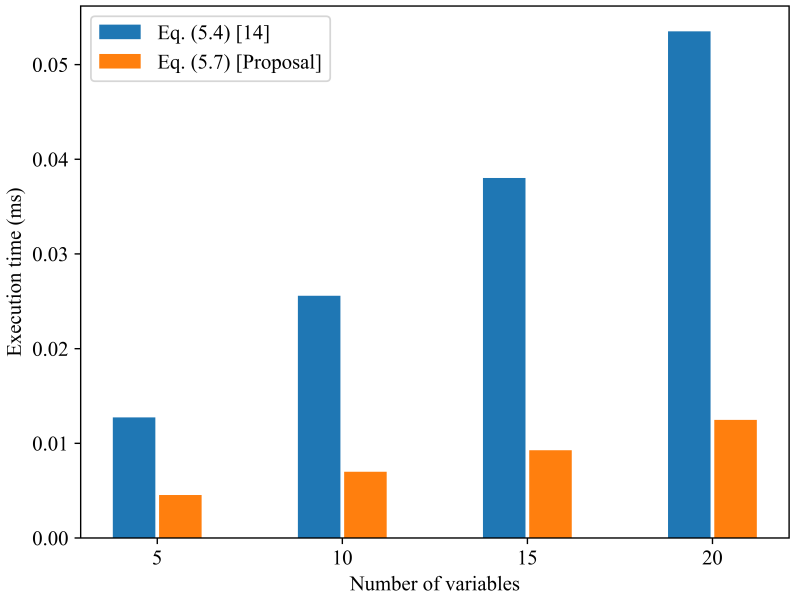


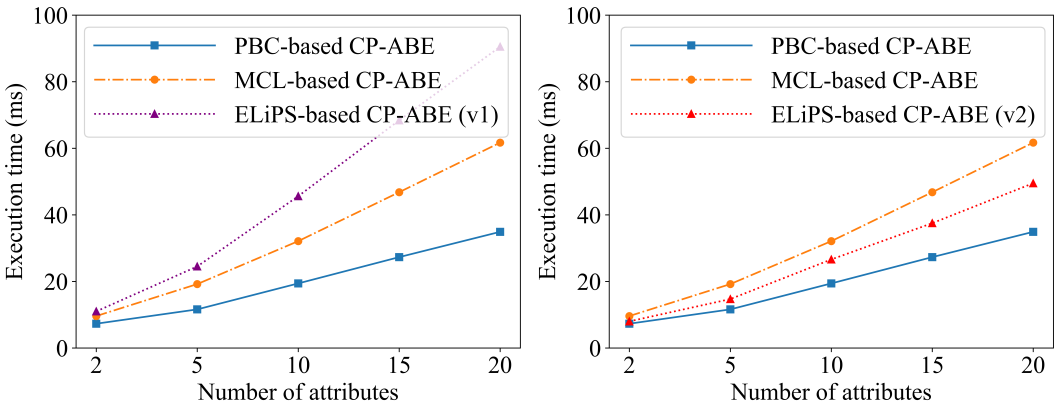
Figure 5.2: Comparison of execution time between Equation (5.4) and Equation (5.7)

Furthermore, Figure 5.2 illustrates our proposed Equation (5.7) decreases the execution time by an average of 74.4% compared to the Equation (5.4).

Subsequently, we successfully implemented our proposed methods into ELiPS-based CP-ABE. Additionally, we conducted several evaluations to compare our work with previous research [14].

5.3.4 Evaluation of decryption performance with our proposed methods

After validating our proposed methods, we successfully integrated these optimizations into the ELiPS-based CP-ABE system. We then compared the decryption performance of our current work with that of the previous work [14]. Using a similar scenario to that described in Section 4.4.3, Figure 5.3 demonstrates that our proposal significantly reduces decryption time. The decryption



(a) Decryption time in previous work

(b) Decryption time in current work

Figure 5.3: A comparison of decryption time in previous work and current work.

performance of the optimized ELiPS-based CP-ABE is not only better than the initial version but also outperforms the MCL-based CP-ABE.

To further evaluate performance, we implemented several scenarios in which the number of attributes varied from 5 to 100 to measure decryption time. We compared the decryption time of our proposed method with that of the previous work [14]. In this evaluation, we ran the setup, key generation,

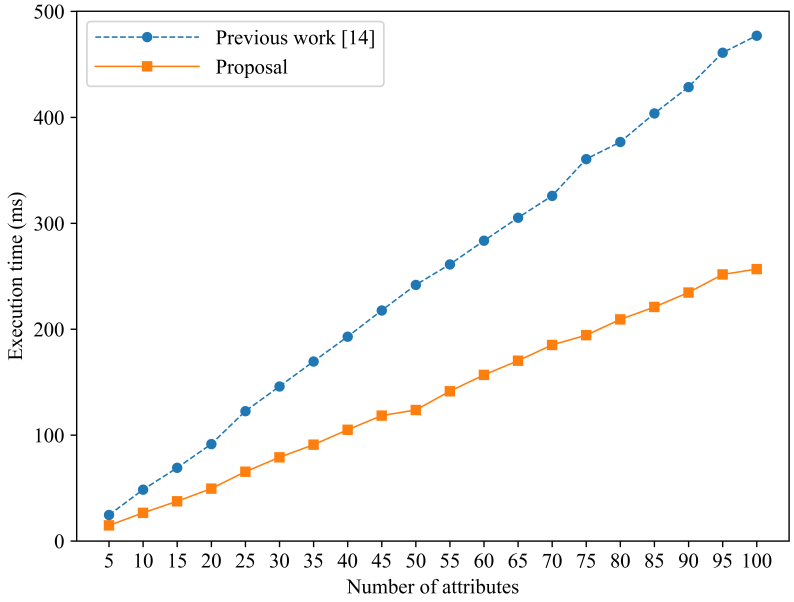


Figure 5.4: Compare decryption time between previous work [14] and our work.

and encryption functions once, while the decryption function was executed 10,000 times for each scenario to obtain average decryption times.

Figure 5.4 gives information that our decryption performance is faster than the previous work [14] across various scenarios. Decryption time in our proposed scheme decreased by an average of 45.5%. In addition, Figure 5.4 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.

5.4 Summary

We proposed methods that minimize the number of final exponentiations and inversions to reduce the decryption processing time of ELiPS-based CP-ABE. The proposed formulas 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 45.5% compared to previous work. Our system has been successfully implemented within the CP-ABE framework, making it applicable in practical applications.

Chapter 6

Performance analysis of ELiPS-based CP-ABE with optimized decryption functions

In this chapter, we further evaluate and analyze the impact of these optimizations, which reduce the number of final exponentiations and inversions, on decryption efficiency. We compare the ELiPS-based CP-ABE with these improvements to the initial version and the original PBC-based CP-ABE. As a result, the combination of both optimization techniques resulted in an average 43.1% overall reduction in decryption time compared to the initial version of the ELiPS-based CP-ABE scheme, while in total execution, it led to a 25.3% improvement. Furthermore, there was an average 53.8% overall reduction in total execution time compared to the original PBC-based CP-ABE method. Additionally, a portion of these results has been presented at the ICTIS 2024 conference [17].

6.1 Introduction

The Internet of Things and cloud computing have revolutionized how data is generated, transmitted, and stored. However, they also introduce significant security and privacy challenges. Ensuring the confidentiality of data and controlling access are essential, especially when sensitive information is involved. Ciphertext-Policy Attribute-Based Encryption has emerged as an advanced cryptographic solution, enabling fine-grained access control within encrypted form [6].

In the CP-ABE scheme, user attributes are used to determine who can decrypt ciphertexts associated with access policy. This grants for flexible, expressive access control without the need for complex key management. CP-ABE has found applications in areas like secure data sharing, access control in IoT and cloud environments, and secure electronic health records [30].

While CP-ABE provides robust security, early constructions were based on the PBC library, which offers only an 80-bit security level. This level of

security is now considered insufficient and vulnerable to attacks from powerful adversaries. To address this limitation, we previously introduced an ELiPS-based CP-ABE scheme that achieves 128-bit security using the ELiPS library for bilinear pairings [14].

Although the ELiPS-based construction enhanced security, the decryption process remained computationally expensive due to the costly pairing, final exponentiation, and inversion operations involved. In our previous work, we optimized the decryption function by minimizing the number of final exponentiations and inversions, which are the most expensive steps [16].

This study evaluates the impact of these decryption optimizations on the efficiency of the ELiPS-based CP-ABE approach. We analyze the performance gains achieved through the optimizations and compare our optimized construction to both the initial ELiPS-based version and the original PBC-based CP-ABE scheme. Our study confirms the effectiveness of ELiPS-based CP-ABE with these optimizations.

6.2 Decryption optimizations

6.2.1 Minimizing final exponentiations

The decryption algorithm in ELiPS-based CP-ABE involves a pair of pairing calculations for each attribute. The pairing operation includes the Miller loop and final exponentiation. Performing these operations can be very expensive, especially when the number of attributes is large.

In accordance with Equation (4.8) and Equation (4.10), the algorithm performs two pairings for each attribute. The pairing operation includes the Miller loop and final exponentiation. We proposed a formula that declined the number of final exponentiations as follows [17]:

$$\prod_{i=1}^{n}\left[\frac{e(D_i,C_i)}{e(D'_i,C'_i)}\right]^{\Delta_i}=\left[\prod_{i=1}^{n}\left(\frac{f_{D_i,C_i}}{f_{D'_i,C'_i}}\right)^{\Delta_i}\right]^{\frac{p^k-1}{r}}, \tag{6.1}$$

where $f_{P,Q}$ be a Miller loop result with P and Q on an elliptic curve as inputs.

It reduces the number of final exponentiations by $2n-1$ times and employs the Miller loop for each pairing. Finally, the decryption operation utilizes the final exponentiation only once at the end, improving performance.

6.2.2 Minimizing inversions

The decryption process also requires the inversion operation. We further optimized the decryption part by minimizing the number of inversions in Equation. (4.9) as follows [17]:

1. Calculate $\mathcal{P}_i$:

$$\mathcal{P}_i = \prod_{j=1, j\neq i}^{n} (i-j) (\text{mod } r).$$

2. Then, calculate inverse of $\mathcal{P}_i$:

$$\mathcal{P}_i^{-1} = \frac{1}{\mathcal{P}_i} (\text{mod } r).$$

3. Afterward, calculate the inversion $(i-j)^{-1}$:

$$(i-j)^{-1} = \mathcal{P}_i^{-1} \prod_{k=1, k\neq j}^{n} (i-k) (\text{mod } r).$$

It reduces the number of inversion operations by $n-1$ times and increases the number of multiplication operations by $3(n-1)$ times. 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 efficient than Equation (4.9).

6.3 Implementation and performance evaluation

6.3.1 Evaluation setup

We implemented the ELiPS-based CP-ABE approach and optimizations in software, devices, and environments as described in Table 4.2.

For evaluation, we utilized an authorization for data access regarding administrative procedures at the university level using the defined access policy $\mathcal{T}$. In our scenario, the confidential information can be obtained by the user whose attribute possession of Professor in the Engineering faculty which deals with $\mathcal{T}$. The access policy $\mathcal{T}$ as illustrated in Figure 6.1 is depicted as:

$$\mathcal{T} = (\text{'Professor'} \ \textbf{AND} \ \text{'Engineering'}).$$

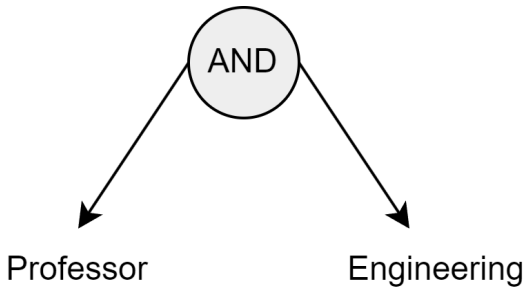


Figure 6.1: Structure of access policy $\mathcal{T}$ for administrative exchange in university.

Firstly, we assessed the decryption process in a scenario that included ten user attributes across various file sizes ranging from 8 KB to 96 KB. We then compared the performance of the proposed techniques with the initial version of ELiPS-based CP-ABE.

Secondly, we conducted a comparison of the total execution time, which includes the setup, key generation, encryption, and decryption phases. This comparison was performed while expanding the number of attributes from 10 to 50. We then compared it with the initial version of ELiPS-based CP-ABE and the original PBC-based CP-ABE.

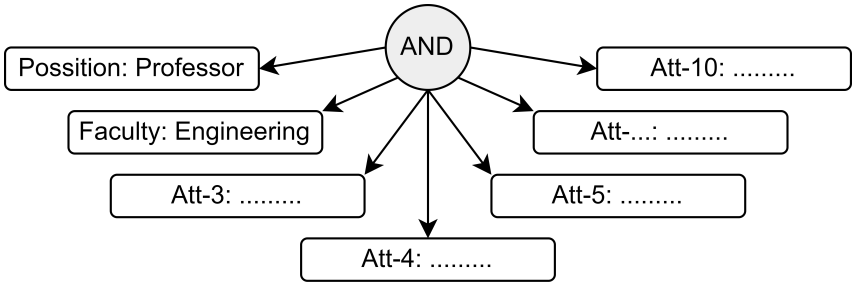


Figure 6.2: An access tree is used for the decryption time evaluation.

6.3.2 Evaluation decryption time

In this experiment, we use the access tree shown in Figure 6.2. The number of attributes is fixed at ten, and we utilized various file sizes ranging from 8 KB to 96 KB. For each file size, we performed 10,000 executions to measure the computation time of the decryption function and then took the average values.

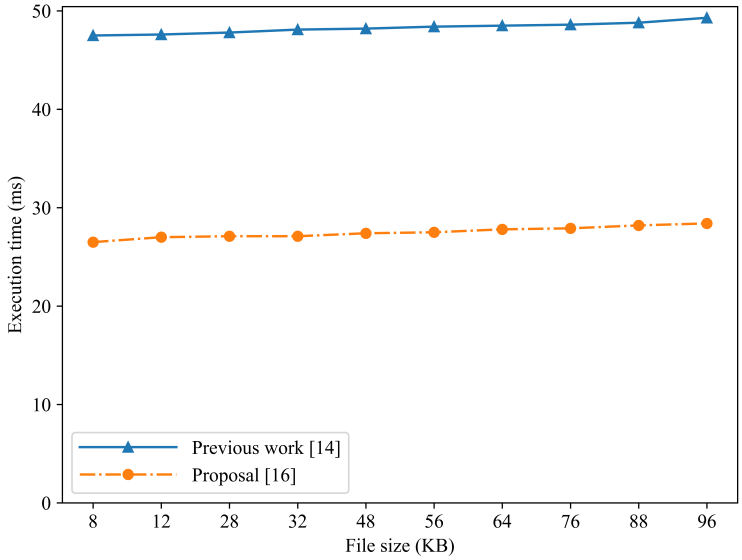


Figure 6.3: Comparison of decryption time between previous work [14] and proposal [16].

Figure 6.3 shows the comparison results of decryption time between initial ELiPS-based CP-ABE and the optimized versions. The results clearly demonstrate the significant performance improvements achieved through our decryption optimizations. By minimizing expensive final exponentiations and

inversions, we obtained up to 43.1% in average faster decryption times compared to the initial ELiPS-based implementation.

6.3.3 Evaluation total execution time

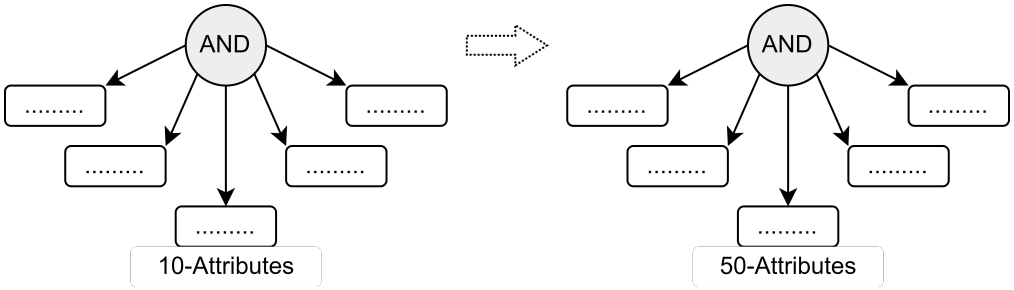


Figure 6.4: An access tree is used for the total execution time evaluation.

In this experiment, we employ the access tree shown in Figure 6.4. The number of attributes is increased from 10 to 50. We conducted the experiments 10,000 times to measure the total execution time, including setup time, key generation time, encryption time, and decryption time. We then calculated the average results.

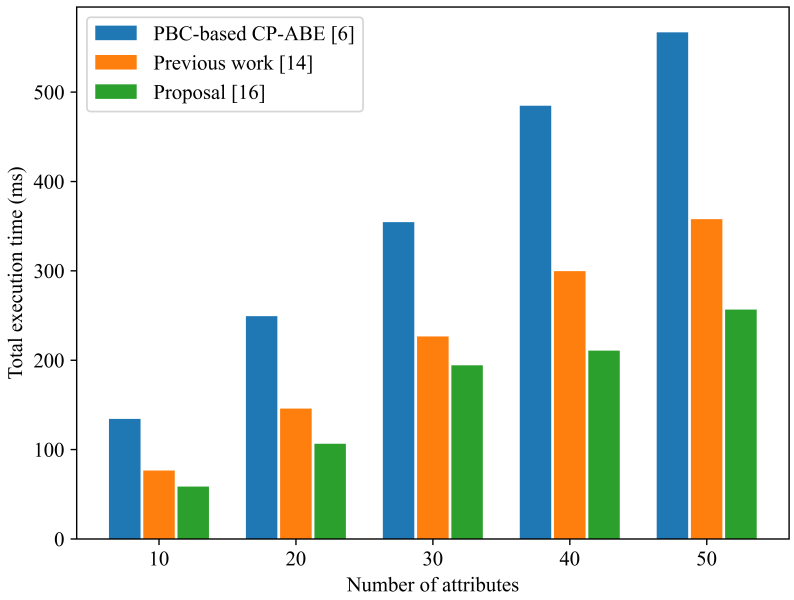


Figure 6.5: Comparison of total execution time among PBC-based CP-ABE [6], previous work [14], and proposal [16].

Figure 6.5 shows the comparison results of total execution time among original PBC-based CP-ABE, initial ELiPS-based CP-ABE, and optimized ELiPS-based CP-ABE versions.

When considering the total execution time of all CP-ABE algorithms, the optimized ELiPS scheme showed a 25.3% improvement over the initial version [14]. Moreover, it outperformed the original PBC-based construction by 53.8% on average, despite the higher security level.

These optimizations are particularly beneficial in scenarios with large attribute universes or complex access policies, where decryption can be the performance bottleneck.

6.4 Summary

We assessed the influence of decryption optimizations on the efficiency of our previously proposed ELiPS-based CP-ABE scheme. By minimizing final exponentiations and inversions, we achieved significant performance gains in the decryption process. The optimized ELiPS-based CP-ABE demonstrated up to 43.1% faster decryption times compared to the initial version. For total execution time, including setup, key generation, encryption, and decryption, we observed a 25.3% improvement. Notably, our optimized construction also outperformed the original PBC-based CP-ABE by an average of 53.8%, while providing a higher 128-bit security level. These results highlight the importance of carefully optimizing expensive cryptographic operations, especially in resource-constrained environments like IoT devices.

Chapter 7

Conclusion and future works

The key objective of this thesis is to enhance the security level and increase the performance of CP-ABE. The original CP-ABE advanced encryption algorithm has shortcomings, such as a lack of sufficient security and performance, making it vulnerable to various attacks and unsuitable for modern applications. To address these problems, we proposed several improvements to accelerate performance and strengthen security, as follows:

- Proposed an ELiPS-based CP-ABE scheme, integrating ELiPS into the CP-ABE framework. ELiPS-based CP-ABE enhanced the performance of most functions.
- Proposed two additional methods to minimize the number of final exponentiations and inversions, which reduce the decryption process time in ELiPS-based CP-ABE.
- Evaluated and analyzed the impact of these optimizations on decryption efficiency. Moreover, we compared the improved ELiPS-based CP-ABE to the initial version and the original PBC-based CP-ABE.

To deal with these shortcomings and purposes, we define the necessary fundamentals in Chapter 2. Chapter 3 provides an overview of prominent pairing-based cryptography libraries and the CP-ABE scheme. Chapter 4 presents an implementation of ELiPS-based CP-ABE, along with several proposed methods to ensure ELiPS compatibility with the CP-ABE scheme, such as generating a generator g, using Shirase's technique to transform asymmetric pairings into symmetric pairings, and carefully choosing functions and modifying the CP-ABE framework to integrate ELiPS into CP-ABE. Although the ELiPS-based CP-ABE scheme enhances the security level and performance of most parts, the decryption process remains computationally expensive due to the costly pairings, final exponentiations, and inversions involved. Therefore, Chapter 5 introduces two methods to reduce the number of final exponentiations and inversions, optimizing the decryption process in ELiPS-based CP-

ABE. Chapter 6 provides further evaluation and analysis of the impact of these optimizations on decryption efficiency.

From the experimental results presented in each chapter, it is evident that our proposed methods can substantially improve the security level and accelerate the processing time of CP-ABE. Our system was not only successfully implemented within the CP-ABE framework, but also analyzed and evaluated in several scenarios, indicating its potential applicability in practical applications. The main experimental results and analysis in this thesis are as follows:

- ELiPS-based CP-ABE enhances the security level of CP-ABE from 80 bits to 128 bits.
- ELiPS-based CP-ABE improves the efficiency of most phases. ELiPS-based CP-ABE reduces the setup time, key generation time, and encryption time by 26.8%, 75.5%, and 75.3%, respectively, compared to PBC-based CP-ABE.
- ELiPS-based CP-ABE with these optimizations reduces the number of final exponentiations by $2n - 1$ and inversions by $n - 1$ times. Additionally, the decryption time of ELiPS-based with these optimizations is decreased by an average of 45.5% compared to that of PBC-based CP-ABE.
- Further evaluation and analysis of the impact of these optimizations shows that the combination of both techniques resulted in an average 43.1% reduction in decryption time compared to the initial version of the ELiPS-based CP-ABE scheme, while in total execution, it led to a 25.3% improvement. Moreover, there was an average 53.8% reduction in total execution time compared to the original PBC-based CP-ABE method.

For future work, blockchain technology has gained traction across industries and organizations, offering a secure and transparent framework for exchanging data. Therefore, we will try to integrate our proposal into a blockchain system to establish fine-grained access control policies and enhance privacy-preserving distributed access control.

References

[1] C. Paar and J. Pelzl, *Understanding Cryptography A Textbook for Students and Practitioners,* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, doi: 10.1007/978-3-642-04101-3.

[2] W. Stallings, *Cryptography and Network Security: Principles and Practice,* seventh ed. Pearson, 2017.

[3] D. Buell, *Fundamentals of Cryptography,* Springer, 2021, doi: 10.1007/978-3-030-73492-3.

[4] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems,* 3rd ed. John Wiley & Sons, 2020.

[5] D. Huang, Q. Dong, and Y. Zhu, *Attribute-Based Encryption and Access Control,* CRC Press, 2020.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07),* Berkeley, CA, USA, pp. 321-334, 2007, doi: 10.1109/SP.2007.11.

[7] K. P. Praveen, K. P. Syam, and P. J. A. Alphonse, "Attribute based encryption in cloud computing: A survey, gap analysis, and future directions," *Journal of Network and Computer Applications,* vol. 108, pp. 37-52, 2018, doi: 10.1016/j.jnca.2018.02.009.

[8] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery, "On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography," *Cryptology ePrint Archive,* pp. 1-19, 2009. [Online]. Available: https://eprint.iacr.org/2009/389

[9] E. Barker, "Recommendation for Key Management," *National Institute of Standards and Technology,* 2020, doi: 10.6028/NIST.SP.800-57pt1r5.

[10] D. Hattori, Y. Takahashi, T. Tatara, Y. Nanjo, T. Kusaka, and Y. Nogami, "An Optimal Curve Parameters for BLS12 Elliptic Curve Pairing and Its Efficiency Evaluation," in *2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW),* Penghu, Taiwan, pp. 1-2, 2021, doi: 10.1109/ICCE-TW52618.2021.9602941.

[11] Y. Takahashi, Y. Nanjo, T. Kusaka, Y. Nogami, T. Kanenari, and T. Tatara, "An Implementation and Evaluation of Pairing Library ELiPS for BLS Curve with Several Techniques," *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC),* JeJu, Korea (South), 2019, doi: 10.1109/ITC-CSCC.2019.8793376.

[12] T. Kanenari, Y. Takahashi, Y. Hashimoto, Y. Kodera, T. Kusaka, Y. Nogami, and T. Nakanishi, "A Comparison of Relic-toolkit and ELiPS Libraries for a Pairing-based Homomorphic Encryption," *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC),* JeJu, Korea (South), 2019, doi: 10.1109/ITC-CSCC.2019.8793446.

[13] M. Shirase, "Symmetric Pairing on Ordinary Elliptic Curves," in *Information Processing Society of Japan Symposium Proceedings,* Japan, pp. 357-362, 2010.

[14] L. H. Anh, Y. Kawada, S. Huda, Md. A. Ali, Y. Kodera, and Y. Nogami, "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.

[15] L. H. Anh, Y. Kawada, S. Huda, Md. A. Ali, Y. Kodera, and Y. Nogami, "ELiPS-based Ciphertext-Policy Attribute-Based Encryption," *International Journal of Networking and Computing (IJNC),* vol. 14, no. 2, pp. 186-205, 2024, doi: 10.15803/ijnc.14.2_186.

[16] L. H. Anh, Y. Kawada, S. Huda, M. A. Ali, Y. Kodera, and Y. Nogami, "A minimization number of final exponentiations and inversions for reducing the decryption process time in ELiPS-based CP-ABE," *Journal of Advances in Information Technology (JAIT),* vol. 15, no. 6, pp. 748-755, 2024, doi: 10.12720/jait.15.6.748-755.

[17] L. H. Anh, Y. Kawada, S. Huda, Y. Kodera, and Y. Nogami, "Performance Analysis of ELiPS-based CP-ABE with Optimized Decryption Functions,"

in Proc. 8th International Conference on Information and Communication Technology for Intelligent Systems (ICTIS2024), Las Vegas, USA, 2024.

[18] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography,* Springer, 2014. doi: 10.1007/978-1-4939-1711-2.

[19] N. E. Mrabet, and M. Joye, "Guide to Pairing-Based Cryptography," *Chapman and Hall/CRC,* pp. 1-420, 2016, doi: 10.1201/9781315370170.

[20] Y. Nanjo, M. Shirase, Y. Kodera, T. Kusaka, and Y. Nogami, "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, doi: 10.15803/ijnc.12.2_317.

[21] J. Bethencourt, A. Sahai, and B. Waters. The University of Texas. *Advanced Crypto Software Collection.* (2006). Accessed: Jul. 15, 2023. [Online]. Available: https://acsc.cs.utexas.edu/cpabe

[22] B. Lynn. Stanford University. *PBC Library - Pairing-Based Cryptography.* (2006). Accessed: Jul. 15, 2023. [Online]. Available: https://crypto.stanford.edu/pbc

[23] D. F. Aranha, C. P. L. Gouvêa, T. Markmann, R. S. Wahby, and K. Liao. Github. *RELIC is an Efficient LIbrary for Cryptography.* (2013). Accessed: Jan. 19, 2024. [Online]. Available: https://github.com/relic-toolkit/relic

[24] Y. Sakemi, T. Kobayashi, T. Saito, and R. Wahby, "Pairing-Friendly Curves," The Internet Engineering Task Force (IETF), 2021. [Online]. Available: https://www.ietf.org/archive/id/draft-irtf-cfrg-pairing-friendly-curves-10.html

[25] S. Mitsunari. Github. *MCL - A portable and fast pairing-based cryptography library.* (2011). Accessed: Jan. 19, 2024. [Online]. Available: https://github.com/herumi/mcl

[26] Information Security laboratory Okayama University. Github. *ELiPS - Efficient Library for Pairing Systems.* (2019). Accessed: Jan. 19, 2024. [Online]. Available: https://github.com/ISecOkayamaUniv/ELiPS

[27] 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.

[28] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute Based Encryption with Privacy Protection and Accountability for CloudIoT," *IEEE Transactions on Cloud Computing,* vol. 10, no. 2, pp. 762-773, 2022, doi: 10.1109/TCC.2020.2975184.

[29] Y. Lu, Y. Wang, X. Dai, J. Li, J. Li, and M. Chen, "Survey of Attribute-Based Encryption in Cloud Environment," *Communications in Computer and Information Science.* vol. 1127, pp.375-384, 2020, doi: 10.1007/978-981-15-6113-9_43.

[30] M. Rasori, M. L. Manna, P. Perazzo, and G. Dini, "A Survey on Attribute-Based Encryption Schemes Suitable for the Internet of Things," *IEEE Internet of Things Journal,* vol. 9, no. 11, pp. 8269-8290, 2022, doi: 10.1109/JIOT.2022.3154039.

[31] S. Huda, A. Sudarsono, and T. Harsono, "Secure data exchange using authenticated ciphertext-policy attributed-based encryption," *2015 International Electronics Symposium (IES),* pp. 134-139, 2015, doi: 10.1109/ELECSYM.2015.7380829.

[32] S. Huda, A. Sudarsono, and T. Harsono, "Secure Communication and Information Exchange using Authenticated Ciphertext Policy Attribute-Based Encryption in Mobile Ad-hoc Network," *EMITTER International Journal of Engineering Technology,* vol. 4, no. 1, pp. 115-140, 2016, doi: 10.24003/emitter.v4i1.116.

[33] T. P. Ezhilarasi, N. S. Kumar, T. P. Latchoumi, and N. Balayesu, "A Secure Data Sharing Using IDSS CP-ABE in Cloud Storage," *Advances*

in Industrial Automation and Smart Manufacturing, Singapore: Springer, Singapore, pp. 1073-1085, 2021, doi: 10.1007/978-981-15-4739-3_92.

[34] Y. W. Hwang, and I. Y. Lee, "A Study on Lightweight Anonymous CP-ABE Access Control for Secure Data Protection in Cloud Environment," *2019 International Conference on Information Technology and Computer Communications (ITCC'19),* New York, NY, USA, pp. 107-111, 2019, doi: 10.1145/3355402.3355405.

[35] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng, "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.

[36] S. Huda, N. Fahmi, A. Sudarsono, and M.U.H. Al Rasyid, "Secure data sensor sharing on ubiquitous environmental health monitoring application," *Jurnal Teknologi (Sciences & Engineering),* vol. 78, no. 6-3, pp. 53-58, 2016. doi: 10.11113/jt.v78.8928.

[37] R. Cheng, K. Wu, Y. Su, W. Li, W. Cui, and J. Tong, "An Efficient ECC-Based CP-ABE Scheme for Power IoT," *Processes 2021,* vol. 9, no. 1176, pp. 1-16, 2021, doi: 10.3390/pr9071176.

[38] B. Girgenti, P. Perazzo, C. Vallati, F. Righetti, G. Dini, and G. Anastasi, "On the Feasibility of Attribute-Based Encryption on Constrained IoT Devices for Smart Systems," *2019 IEEE International Conference on Smart Computing (SMARTCOMP),* Washington, DC, USA, pp. 225-232, 2019, doi: 10.1109/SMARTCOMP.2019.00057.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] T. Hu, S. Yang, Y. Wang, G. Li, Y. Wang, G. Wang, and M. Yin, "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.

[45] G. Zhang, X. Chen, L. Zhang, B. Feng, X. Guo, J. Liang, and Y. Zhang, "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.

[46] R. Hu, Z. Ma, L. Li, P. Zuo, Z. Li, J. Wei, and S. Liu, "An access control scheme based on blockchain and ciphertext policy-attribute based encryption," *Sensors,* vol. 23, no. 19, 2023. doi: 10.3390/s23198038.

[47] Y. Zhao, H. Li, Z. Liu, and G. Zhu, "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.

[48] L. Meng, H. Xu, R. Tang, X. Zhou, and Z. Han, "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.

[49] 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.

[50] 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.

[51] 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.