UNIVERSITÀ DEGLI STUDI DI MILANO

SCUOLA DI DOTTORATO
INFORMATICA

DIPARTIMENTO
GIOVANNI DEGLI ANTONI

CORSO DI DOTTORATO
INFORMATICA, XXX CICLO

Efficient and Secure Data Sharing Using Attribute-Based Cryptography

Tesi di Dottorato di Ricerca di:
Masoomeh Sepehri

Relatore:
Prof. Paolo Boldi

Correlatore:
Prof. Alberto Trombetta
Prof. Ernesto Damiani

Coordinatore del Dottorato:
Prof. Paolo Boldi

Anno Accademico 2016/17

Abstract

With the ever-growing production of data coming from multiple, scattered, and highly dynamical sources, many providers are motivated to upload their data to the cloud servers and share them with other persons for different purposes. However, storing data on untrusted cloud servers imposes serious concerns in terms of security, privacy, data confidentiality, and access control. In order to prevent privacy and security breaches, it is vital that data is encrypted first before it is outsourced to the cloud. However, designing access control models that enable different users to have various access rights to the shared data is the main challenge. To tackle this issue, a possible solution is to employ a cryptographic-based data access control mechanism such as attribute-based encryption (ABE) scheme, which enables a data owner to take full control over data access. However, access control mechanisms based on ABE raise two challenges: (i) *weak privacy:* they do not conceal the attributes associated with the ciphertexts, and therefore they do not satisfy attribute-hiding security, and (ii) *inefficiency:* they do not support efficient access policy change when data is required to be shared among multiple users with different access policies. To address these issues, this thesis studies and enhances inner-product encryption (IPE), a type of public-key cryptosystem, which supports the attribute-hiding property as well as the flexible fine-grained access control based payload-hiding property, and combines it with an advanced cryptographic technique known as proxy re-encryption (PRE).

The first part of this thesis discusses the necessity of applying the inner-product proxy re-encryption ($IPPRE$) scheme to guarantee secure data sharing on untrusted cloud servers. More specifically, we propose two extended schemes of IPE: in the first extended scheme, we propose an inner-product proxy re-encryption ($IPPRE$) protocol derived from a well-known inner-product encryption scheme [1]. We deploy this technique in the healthcare scenario where data, collected by medical devices according to some access policy, has to be changed afterwards for sharing with other medical staffs. The proposed scheme delegates the re-encryption capability to a semi-trusted proxy who can transform a delegator's ciphertext associated with an attribute vector to a new ciphertext associated with delegatee's attribute vector set, without knowing the underlying data and private key. Our proposed policy updating scheme enables the delegatee to decrypt the shared data with its own key without requesting a new decryption key. We analyze the proposed protocol in terms of its performance on three different types of elliptic curves such as the SS curve, the MNT curve, and the

BN curve, respectively. Hereby, we achieve some encouraging experimental results. We show that our scheme is adaptive attribute-secure against chosen-plaintext under standard Decisional Linear (*D-Linear*) assumption. To improve the performance of this scheme in terms of storage, communication, and computation costs, we propose an efficient inner-product proxy re-encryption (*E-IPPRE*) scheme using the transformation of Kim's inner-product encryption method [2]. The proposed *E-IPPRE* scheme requires constant pairing operations for its algorithms and ensures a short size of the public key, private key, and ciphertext, making it the most efficient and practical compared to state of the art schemes in terms of computation and communication overhead. We experimentally assess the efficiency of our protocol and show that it is selective attribute-secure against chosen-plaintext attacks in the standard model under Asymmetric Decisional Bilinear Diffie-Hellman assumption. Specifically, our proposed schemes do not reveal any information about the data owner's access policy to not only the untrusted servers (e.g, cloud and proxy) but also to the other users.

The second part of this thesis presents a new lightweight secure data sharing scheme based on attribute-based cryptography for a specific *IoT*-based healthcare application. To achieve secure data sharing on *IoT* devices while preserving data confidentiality, the *IoT* devices encrypt data before it is outsourced to the cloud and authorized users, who have corresponding decryption keys, can access the data. The main challenge, in this case, is on the one hand that *IoT* devices are resource-constrained in terms of energy, *CPU*, and memory. On the other hand, the existing public-key encryption mechanisms (e.g., *ABE*) require expensive computation. We address this issue by combining the flexibility and expressiveness of the proposed *E-IPPRE* scheme with the efficiency of symmetric key encryption technique (*AES*) and propose a light inner-product proxy re-encryption (*L-IPPRE*) scheme to guarantee secure data sharing between different entities in the *IoT* environment. The experimental results confirm that the proposed *L-IPPRE* scheme is suitable for resource-constrained *IoT* scenarios.

Aknowledgements

I feel greatly privileged that God gave me the patience, perseverance, and courage to finish this thesis.

I would like to thank my supervisor, Prof. Paolo Boldi, and my co-supervisors, Prof. Alberto Trombetta and Prof. Ernesto Damiani, for their guidance, support and trust.

I would like also to say thank to all my friends and colleagues at Milan university for all the fun we had. Especially I would like to take this opportunity to thank Antonio Capoduro, Marco Frasca, Corrado Mio, Muneeb Kiani, Niloofar Aprin and Fatima Hachem. You helped me to integrate in this new and astonishing environment of Italy and supported me and my work in many ways. My kind Corrado Mio, your insights in the world of programming are amazing and I am glad that you shared so much of your experience and unique jokes with me. You always made me smile. I want to especially thank Antonio Capoduro, you supported me when I needed someone at my side. I am very grateful for Marco Frasca, your guidance, especially during the first year, gave me hope, when my sister could not support me you have been with me. All of you made me feel comfortable during these three years of my stay in Italy. I am glad to call you my friends. Thank you.

I would like to express my deep appreciation to my best friend Armin. You supported and encouraged me in every possible way. You continuously pushed me to do my best and reinforced my strength in times when I doubted myself. Thank you very much.

Last but not least, I want to thank my beloved and amazing sister Maryam. You always gave guidance and support for me. Your words encouraged me to continue my thesis. Without you I would not have enjoyed my stay in Italy nor finished my thesis. Although you are far from me now, I always feel like you are next to me. I will never forget our conversations in the middle of the night, due to the time-shift, where we laughed and cried, discussed and argued, and dreamed of our future. From the deepest of my heart I want to thank you Maryam, you are the best sister in the world.

I will dedicate this thesis to my parents and family, I would not be here without your love and support. I LOVE YOU.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The emerging trend of sharing information among different users (esp. businesses and organizations) aiming to gain profit, has recently attracted a tremendous amount of attention from both research and industry communities. However, despite all benefits that data sharing inevitably provides [3], many organizations are reluctant to share their data with others due to the large initial investments of expensive infrastructure setup, large equipment and daily maintenance cost [4]. With the advent of cloud computing, data outsourcing paradigm makes shared data much more accessible as users can retrieve them from anywhere with significant cost benefits. There are major concerns, with data confidentiality in the cloud as organizations lose control of their data and disclose sensitive information to a service provider that is not fully trusted. In addition, most organizations do not wish to grant full access privilege to other users. To this purpose, many research efforts have been dedicated to solve these issues by proposing cryptographically enforced access control mechanisms to set access policies for encrypted data such that only users with appropriate authorization can have access. In this thesis, we advance research in the area of cloud data sharing (i) Providing fine-grained access control over encrypted data, (ii) Minimizing the amount of computation used for sharing data between users with different access rights (policy change) adopting a novel re-encryption (PRE) method, and (iii) Verifying the feasibility and practicability of the proposal schemes for constrained devices in the Internet-of-Things (IoT).

1.1 Data sharing on cloud computing

With the growing popularity of cloud computing, more and more enterprises are motivated to outsource their data to the cloud. Cloud data sharing has

found many practical applications in healthcare, where patients are willing to share their personal health records among a group of users (e.g., doctors, nurses, care-practitioners, and etc.,) for remote monitoring and diagnosis without having to leave their home. In one particular scenario, a patient is equipped with smart objects to collect his health data that is analyzed through his mobile phone and outsourced to a cloud service provider. Then authorized healthcare professionals monitor patient's health status remotely without visiting the patient hence saving costs and time. The benefits of cloud data sharing paradigm are many for users as providing unlimited and elastic storage resources, low costs and time due to provide services on demand anywhere and at anytime, lowering the chance of data loss and high scalability computing resources with high reliability and availability since data is usually replicated among the number of servers [3]. On the other hand, when cloud cannot be trusted enough to have data in plaintext, the data privacy becomes a primary concern [5]. In addition, it is that only authorized users have access to stored data in the cloud. Hence, the major requirements of secure data sharing in the cloud can be summarized as below [3]:

- Data owner should be able to specify a list of authorized users who allow having access to their data from anywhere at anytime without interaction,
- Data owner should have access control over her data,
- Data owner should be able to add new authorized users,
- Data owner should be able to revoke access rights of any member of the group to access her data,
- No member of the group should be allowed to revoke rights or join new users to the group.

Achieving the above-mentioned objectives, data owner should employ a kind of cryptography technique to enforce security policies on her data satisfying: (a) *Confidentiality*, that only authorized users are able to get access to the data, (b) *Authentication*, that an entity is who or what it claims to be, (c) *Authorization*, that a user has sufficient rights to perform the requested operation, (d) *Integrity*, that data is protected from unauthorized or unintentional alteration, modification, or deletion, (e) *User revocation*, that revoked users are not able to get access to the data, and (f) *Collusion resistance* that when users collude, they are not able to access the data without

the permission of data owner. In order to preserve data confidentiality, a possible solution is to employ some kind of public key infrastructure (PKI). In a traditional PKI, a data owner encrypts its data based on a user's public key before outsourcing it to the cloud (i.e., *payload-hiding*) and the user then can decrypt the encrypted data with her private key. However, this manner has some disadvantages: the data owner requires the public keys of users to encrypt its data and then sends the encrypted data separately to the cloud, which increases the computational overhead and lots of storage overhead would spend for storing the same plaintexts with different public keys belonging to different users. On the other hand, the data owner needs to obtain the list of authorized users before encryption and the cloud requires to specify and enforce the access control policies for stored data to control "who can have access to what?".

To meet these challenges, many cryptographic-based approaches have been proposed and among them, attribute-based encryption (ABE) schemes look very promising, since it binds fine-grained access control policies to the data and it does not require an access control manager to check the access policies in real time. Initially, access to data in the cloud is provided through access control list (ACL) [6] for fine-grained access control. However, this mechanism would introduce an extremely high complexity when it is enforced by cryptography schemes, the complexity of each data in terms of its ciphertext size and corresponding data encryption operations is linear to the number of system users; therefore, it makes the system less scalable and is just able to provide coarse-grained access control to data [7]. ABE scheme [8] provides a more flexible and scalable fine-grained access control to data compare to ACL.

In ABE, data is encrypted based on the set of attributes (key-policy ABE) or according to an access control policy over attributes (ciphertext-policy ABE), such that the decryption of ciphertext is possible only if a set of attributes in the user's private key matches with the attributes of the ciphertext, so that the data can be encrypted without exact knowledge of the users set that will be able to decrypt. Moreover, in ABE scheme senders and recipients are decoupled because they do not need to pre-share secrets, which simplifies key management for large-scale and dynamic systems and which makes data distribution more flexible. In particular, in group-oriented publish-subscribe systems like IoT, ABE does not require a shared group key to be updated for every new group member who joins, which causes improving scalability. Furthermore, ABE is more strongly resistant to collusion attacks than traditional public key encryption schemes [9].

Although access control systems based on ABE schemes present advan-

tages regarding reduced communication, storage management and provide a fine-grained access control, along with a decentralized access control mechanism, they are not suitable for scenarios in which data must be shared among different parties with different access policies. For example, think of a healthcare scenario in which, once data is collected by medical devices, it is encrypted according to some access policy that has to be changed afterwards since the data has been shared with other medical staff. Therefore, one of the challenges of secure data sharing is: "How can we change access policies, under which the data is encrypted, efficiently?".

A simple solution for applying a new access policy to the data is to decrypt the data and then re-encrypt it with a new access policy. However, this approach is very time-consuming and causes much computational overhead that could be a challenging issue when using on resource-constraint IoT devices such as smartphone or tablet.

Attribute-based proxy re-encryption ($ABPRE$) scheme offers a good solution for this drawback by delegating the re-encryption capability to a semi-trusted proxy who can transform the encrypted data to those encrypted under a different access policy by using the re-encryption key, which reduces the computational overhead of the data owner and the sensitive information and the decryption key cannot be revealed to the proxy. Therefore, this approach is an efficient way of preserving the privacy for shared data among users; however, the security issue still remains, i.e., the attributes that are associated with each ciphertext can reveal to users who can not decrypt. For example, in a healthcare scenario medical data requires a high degree of privacy since they are accessed by many parties such as the patient or staff (e.g. doctors, nurses, care practitioners, etc.,) from the different department or belonging to different hospitals. Therefore, even partial exposure of those attributes could hurt the patient's privacy. Thus, access control system based on ABE are not enough to provide appropriate protection for sensitive data in some scenario like healthcare. Instead, predicate encryption (PE) scheme [10] can solve the above problems by offering the "*attribute-hiding*"property (which means that is not possible to determine the set of attributes with which the ciphertext is encrypted) as well as the "*payload-hiding*" property.

1.2 Thesis Statement

The major goal of this research effort is to perform a realistic evaluation of the security and efficiency issues of data sharing schemes on cloud computing. Based on the findings of these evaluations, we also propose a technique to make data sharing paradigm more practical so that it can be adapted to

IoT environment.

Four main research objectives of this thesis are described below:

Objective 1: Efficient and Secure Access Control for Outsourced Data
Protecting data stored on cloud server from unauthorized access is another important issue to be addressed. Users with different roles should be granted different level of access privilege. Many papers [8, 11, 9] have been developed to propose techniques for cryptographically enforced access control to outsourced data. These techniques are mostly designed for implementing fine-grained access control via attribute-based encryption [8], which do not allow updating attribute set without re-encryption, making policy extremely inefficient. To this purpose, we evaluate the existing data sharing schemes in terms of several metrics (e.g, computation costs, execution time and etc.,) to identify and design scalable data sharing schemes with fine-grained access control among both data owners and their authorized users and different users with different access rights while preserving the security of encrypted message and its associated attribute set.

Objective 2: Lightweight Fine-grained Access Control in *IoT*
Due to the limited computational capability of mobile devices, there is a strong need to offload the intensive data access operations on the cloud server for execution and adopt lightweight access control mechanisms to access shared data [12]. Hence, the plan is to improve the proposed schemes (Objective 1) using lightweight symmetric encryption version in order to get significant improvement in results while preserving data privacy.

Objective 3: Correctness and Security Analysis
We aim to prove the correctness of the proposed schemes and provide a formal mathematical security proof.

Objective 4: Performance Analysis
The effort is to conduct experimental performance assessments of the proposed schemes in emulated platforms.

1.3 Main Contribution of this Research

This research contributes original ideas to the data sharing schemes on cloud computing. We identify an extensive evaluation for data sharing and used

that to determine which schemes work best for IoT scenario and constrained devices in terms of privacy and efficiency. We conduct several experiments to assess designed schemes in terms of computation cost and execution time and to show that the proposed schemes perform well compared to the previous relevant approaches. We propose an effective deployment IoT scenario namely $EVOTION$ to deploy the lightweight version of the proposed data sharing techniques.

The main contributions of this research are:

1. Fine-grained Access Control. The protocols proposed in this thesis are differs from attribute-based encryption (ABE) schemes used for fine-grained access policies since they do not sufficiently protect the attributes associated with the message ciphertext. The proposed schemes are based on using inner-product encryption (IPE) that is a well-known functional encryption primitive that allows decryption when the inner-product of the attribute vectors upon which the encrypted data and the decryption key depend is equal to zero. With using the proposed IPE protocols it is possible to define fine-grained access policies over encrypted data whose enforcement can be outsourced to the cloud where the data is stored.

2. Policy Change. Current IPE schemes do not support efficient access policy changes. The proposed scheme in this thesis adopts a novel inner-product proxy re-encryption scheme that provides the proxy server with a transformation key with which a ciphertext associated with an attribute vector can be transformed to a new ciphertext associated with a different attribute vector, providing a policy update mechanism whose performance is suitable for many practical applications.

3. Attribute Hiding. A common limitation of ABE approaches is that the ciphertexts do not conceal their corresponding attributes set, and therefore they do not guarantee attribute-hiding property. Furthermore, to the best of our knowledge, this concern has not yet been addressed satisfactorily in previous inner-product proxy re-encryption schemes. The proposed inner-product proxy re-encryption protocols in this thesis do a better job in hiding the vector used for data encryption than previous inner-product proxy re-encryption proposal.

4. Lightweight Fine-grained Access Control. Due to the limited computational power of mobile devices, it is important to design lightweight

ABE schemes. This thesis proposes a lightweight inner-product proxy re-encryption scheme using a symmetric key for data encryption/decryption.

1.4 Overview of Thesis

The remaining of the thesis is organized as follows.

Chapter 2 presents some basic notations, cryptographic primitives and security assumptions that are used throughout this thesis.

Chapter 3 outlines the primary data sharing techniques available on cloud computing and their privacy and performance issues. It reviews the relevant solutions to mitigate the limitations of the primary data sharing techniques including the state of the art discussed in cloud computing. The existing techniques are compared in terms of computation costs and execution time to investigate which schemes appear most promising for IoT environment.

Chapter 4 states the system model in which the proposed protocols are built upon and the details of problem statements. It also addresses Objective 1 and illustrates our access control system for securely sharing data stored at honest-but-curious servers and proposes an efficient approach managing access policy changes among users with different access rights.

Chapter 5 addresses Objective 2 and determines the constraints of the proposed techniques (Objective 1) in IoT domain. It states how the proposed techniques can be improved using lightweight cryptographic technique providing fine-grained access policies for a considered IoT case study.

Chapter 6 summarizes the contributions of this thesis and outlines future work.

Chapter 2

Cryptographic Background

In this chapter, we will provide a brief review of cryptographic background, which is necessary for understanding the rest of this thesis. We first introduce two encryption algorithms: symmetric and asymmetric encryption algorithms. Then, we present the background on different cryptographic preliminaries, including cyclic group, elliptic curve cryptography, pairing-based cryptography and bilinear map. Finally, we present the complexity assumptions which are used in our proposed protocols.

2.1 Encryption

Encryption is one of the most popular and effective data security methods used by organizations to protect the confidentiality of data stored on computer systems or transmitted via the Internet. In cryptography, encryption is the process of encoding a message/file in such a way that only authorized users can access it. In an encryption, the intended message often referred to as plaintext, is encrypted using an encryption algorithm. This process generates a ciphertext that can be decrypted with the proper key. Encryption algorithms are divided into two categories: symmetric key encryption and asymmetric key encryption (public key encryption).

2.1.1 Symmetric Key Encryption

Symmetric key encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption or single key encryption. Symmetric key encryption is simple and much faster than asymmetric key encryption and you require quite small key-sizes to get good cryptographic strength. But its main drawback is that the sender has to exchange the key used to encrypt the data with

the recipient before he can decrypt it; therefore, there has to be a second secure channel to transmit this key from one end to the other. In IoT environment, symmetric cryptography is a good choice, since most of the IoT device platforms already have hardware blocks for symmetric cryptography. But getting the symmetric key to the decrypting entity is a hard problem, because the key has to be transmitted via the unsecured wireless connection as a second channel. To secure the symmetric key in transit, asymmetric approaches are used. One of the most common symmetric key encryption is advanced encryption standard (AES) that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits.

2.1.2 Asymmetric Key Encryption

Asymmetric key encryption (public key encryption) uses pairs of keys, a public key, and a private key. A user's public key is public and anyone in the system can access it and a user's private key should be kept private. A message that is encrypted with a particular public key can only be decrypted by the corresponding private key. This type of encryption does not need the pre-determined shared secret to begin secure communication because only the intended recipient can decrypt the message. Security of the public key is not required because it is publicly available and can be passed over the Internet. Asymmetric key encryption has a far better power in ensuring the security of information transmitted during communication. The asymmetric key encryption requires significantly more computational effort than the symmetric key encryption. Popular asymmetric key encryption includes EIGamal, Rivest Shamir Adleman (RSA), Digital Signature Algorithm (DSA), and Elliptic Curve Cryptography (ECC).

2.2 Hybrid Encryption

Hybrid encryption is a cryptographic paradigm which combines the efficiency and large message space of symmetric key encryption techniques with the advantages of public key encryption techniques. In hybrid a encryption scheme first, a key encapsulation mechanism (KEM) is used to fix a random session key that is then fed into a highly efficient data encapsulation mechanism (DEM) to encrypt the actual message. A ciphertext output from a hybrid cryptosystem has two components: A hybrid encryption scheme consists of two independent cryptosystems:

- **A key encapsulation mechanism (KEM).** This cryptosystem generates simultaneously a random symmetric key together with its en-

cryption (encapsulate the symmetric key). Basically, this cryptosystem is a public key encryption scheme, except that the encryption algorithm generates the encryption of a random symmetric key.

- **A data encapsulation mechanism (*DEM*).** This cryptosystem encrypts the message under the obtained symmetric key from the *KEM*) using symmetric key encryption techniques.

Both the *KEM* and *DEM* ciphertexts are then sent to a user. First, the user decrypts the *KEM* ciphertext to get the symmetric key using his own private key and then uses that key to decrypt the message.

2.3 Security Level

In cryptography, a security level is a measure of the strength that a cryptographic primitive, such as a cipher or hash function, achieves. The security level is often measured in bits, e.g., 80-bit security. This measure origins from the key length of a symmetric key. If there is no structural weakness of the symmetrical encryption algorithm, then an attacker makes an exhaustive search by trying all possible keys in the entire key space. Exhaustive key search for the λ-bit key may involve up to 2^{λ} different keys. In general, a cryptographic system offers security level λ if a successful general attack can be expected to require effort approximately 2^{λ} [13]. For example, an 80-bit key length means that the key space consists of 2^{80} different keys. By increasing the key length, the number of possible keys grows exponentially and the security level is increased as well. In asymmetric encryption, the key length does not directly correspond to the security level. Instead, the security of asymmetric encryption is dependent on the intractability of the underlying mathematical problems such as integer factorization. However, the National Institute of Standards and Technology (*NIST*) and other institutes have guidelines for translating the key size of asymmetric encryption algorithms to a security level in terms of symmetric key length. For example, *NIST* guidelines state that for 80-bit security *RSA* a prime modulus of 1024-bit is needed.

2.4 Cryptographic Preliminaries

In this section, some cryptography preliminaries needed for the understanding of our proposed schemes are introduced.

2.4.1 Cyclic Group

In mathematics, a group $\mathbb{G}$ is an algebraic structure consisting of a set of elements equipped with an operation $\cdot$ (called the group law of $\mathbb{G}$) that combines any two elements a and b to form a third element in the group, denoted $a \cdot b$ or ab. To qualify as a group, the set and operation, $(\mathbb{G}, \cdot)$, must satisfy four conditions called the group axioms:

1. **Closure:** if $a, b \in \mathbb{G}$, then $a \cdot b \in \mathbb{G}$.
2. **Associativity:** for all $a, b, c \in \mathbb{G}$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
3. **Identity element:** there exists an element e in $\mathbb{G}$ such that, for every element a in $\mathbb{G}$, the equation $e \cdot a = a \cdot e = a$ holds. Such an element is unique, and this one of the identity element.
4. **Inverse element:** for each a in $\mathbb{G}$, there exists an element b in $\mathbb{G}$, commonly denoted a^{-1} (or $-a$, if the operation is denoted "+"), such that $a \cdot b = b \cdot a = e$, where e is the identity element.

The result of an operation may depend on the order of the operands. In other words, the result of combining element a with element b need not yield the same result as combining element b with element a; the equation $a \cdot b = b \cdot a$ may not always be true.

In algebra, a cyclic group or monogenous group is a group that is generated by a single element g in the group $\mathbb{G}$. That is, it consists of a set of elements with a single invertible associative operation, and it contains an element g such that every other element of the group can be obtained by repeatedly applying the group operation or its inverse to g. Each element can be written as a power of g in multiplicative notation, or as a multiple of g in additive notation. This element g is called a generator of the group [14].

Definition 2.1. A group $\mathbb{G}$ is called cyclic if it is generated by an element $g \in \mathbb{G}$ such that every element in $\mathbb{G}$ has the form g^n for some integer n: $\mathbb{G} = \langle g \rangle = \{g^n \mid n \textit{ is an integer}\}$.

2.4.2 Finite Fields

A finite field is a mathematical group with a finite number of elements. An example of a finite field is all the integers modulo a number p, this finite field is denoted $\mathbb{Z}_p$.

2.4.3 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is a type of public key cryptosystem based on elliptic curve groups over finite fields. ECC encryption systems are based on the idea of using points on a curve to define the public/private key pair. An elliptic curve is the set of points defined by the following equation:

$$y^2 = x^3 + ax + b \, mod \, p, \tag{2.1}$$

where a and b are parameters that determine the shape of the curve and $4a^3 + 27b^2 \neq 0$ (this is required to exclude singular curves). Each point on the elliptic curve can be represented as a vector in affine coordinates such as $P = (p_x, p_y)$, where p_x and p_y must satisfy the equation 2.1. The only difference is that all coordinates must be integers in modular p.

An elliptic curve group $\mathbb{G}$ consists of the elliptic curve and a group operation called addition, denoted by "+". Furthermore, a point at infinity is needed, denoted **0** which serves as the identity element. In addition to the four group axioms the addition operation of elliptic curve groups has the property of being commutative, i.e. if $P, Q \in \mathbb{G}$ then $P + Q = Q + P$. For the purposes of this thesis, a high-level understanding of the elliptic curve group is sufficient, for more detailed information see [15].

Compared to other public key cryptosystems like RSA, elliptic curve cryptography requires less computing power. This type of public key has the advantage to reach a high-security level with rather small key sizes. A key of 160-bit ECC correlates to a 1024-bit RSA key, or 512-bit ECC correlates to 15,360-bit RSA. Computing 160-bit mathematical operations save a lot compared to 1024-bit operations [16].

2.4.4 Multiplicative and Additive Notations

There are two main notational conventions for abelian groups [1]: multiplicative and additive notation. The multiplicative notation is used for the operation in an arbitrary group, while the additive notation is used for modules and rings. The additive notation may also be used to emphasize that a particular group is abelian, whenever both abelian and non-abelian groups are considered, some notable exceptions being near-rings and partially ordered groups, where an operation is written additively even when non-abelian.

[1] An abelian group (commutative group), is a group in which the result of applying the group operation to two group elements does not depend on the order in which they are written.

Table 2.1 shows that how the operations of the different notations correspond to each other.

Convention	**Operation**	**Powers**	**Inverse**	**Identity**
Addition	$x + y$	nx	$-x$	0
Multiplication	$x \cdot y$ or xy	x^n	x^{-1}	e or 1

Table 2.1: Comparison between multiplicative and additive notations

2.4.5 Pairing-based Cryptography

In a multiplicative notation a pairing between elements of two cryptographic groups to a third group can be written as:

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

where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three multiplicative cyclic groups of prime order p and e is a bilinear map. Let g_1 be a generator of $\mathbb{G}_1$ and g_2 be a generator of $\mathbb{G}_2$. The bilinear map e satisfies the following properties:

- *Bilinearity:* for all $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p^*$, we have: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- *Non-degeneracy:* $e(g_1, g_2) \neq 1$.

In some instances the pairing is symmetric if $\mathbb{G}_1 = \mathbb{G}_2$; otherwise, is an asymmetric pairing i.e. when $\mathbb{G}_1 \neq \mathbb{G}_2$.

If the map e is symmetric, then we have: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} = e(g_1^b, g_2^a)$.

2.4.6 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p, and g be a generator of $\mathbb{G}$. A pairing (or bilinear map) $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a function that has the following properties [17]:

1. *Bilinear:* a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is bilinear if $e(u^a, v^b) = e(u, v)^{ab}$ for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$.
2. *Non-degenerate:* $e(g, g) \neq 1$. The map does not send all pairs in $\mathbb{G} \times \mathbb{G}$ to the identity in $\mathbb{G}_T$. Since $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order, this implies that if g is a generator of $\mathbb{G}$ then $e(g, g)$ is a generator of $\mathbb{G}_T$.

3. *Computable:* there is an efficient algorithm to compute the map $e(u, v)$ for any $u, v \in \mathbb{G}$.

A map e is an admissible bilinear map in $\mathbb{G}$ if satisfies the three properties above. Note that $e(\ ,\ )$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.4.7 Asymmetric Bilinear Pairing Groups

Asymmetric bilinear pairing groups $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ [18] are tuples composed of a prime order p, cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of a prime order p, $g_1 \neq 1 \in \mathbb{G}_1$, $g_2 \neq 1 \in \mathbb{G}_2$, and a polynomial-time computable non-degenerate bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ (e.g., $e(g_1^s, g_2^t) = e(g_1, g_2)^{st}$ and $e(g_1, g_2) \neq 1$).

The term asymmetric refers to the fact that the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are not the same. It is well-known that when $\mathbb{G}_1 = \mathbb{G}_2$ then Decision Diffie-Hellman (DDH) problem in $\mathbb{G}$ is easy; however when $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct and there is no efficiently computable map from $\mathbb{G}_1$ to $\mathbb{G}_2$ and the DDH problem in $\mathbb{G}$ can still be hard [19].

2.5 Complexity Assumptions

This thesis uses the following complexity assumptions.

- Decisional Bilinear Diffie-Hellman (DBDH) Assumption [17]
Let $a, b, c \in \mathbb{Z}_p^*$ be chosen at random and g be a generator for $\mathbb{G}$. The Decisional BDH assumption is defined as follows: given $(g, g^a, g^b, g^c, Z) \in \mathbb{G}^4 \times \mathbb{G}_T$ as input, determine whether $Z = e(g, g)^{abc}$ or Z is a random in $\mathbb{G}_T$.

- The Decision Linear (D-Linear) Assumption [20]
Let $z_1, z_2, z_3, z_4 \in \mathbb{Z}_p^*$ be chosen at random and g be a generator for $\mathbb{G}$. The Decision Linear assumption is defined as follows: given $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, Z) \in \mathbb{G}^6$ as input, determine whether $Z = g^{z_3 + z_4}$ or Z is random in $\mathbb{G}$. We consider an equivalently modified version such as: given $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z) \in \mathbb{G}^6$ as input, determine whether $Z = g^{z_2(z_3 + z_4)}$ or Z is random in $\mathbb{G}$.

Definition 2.2. We say that the {Decision BDH, Decision Linear} assumption holds in $\mathbb{G}$ if the advantage of any polynomial-time algorithm is solving the {Decision BDH, Decision Linear} problem is negligible.

- Asymmetric Decisional Bilinear Diffie-Hellman Problem (DBDH) [21], [22]
Consider the following two distributions for $g \in \mathbb{G}_1, h \in \mathbb{G}_2, a, b, c \in \mathbb{Z}_p^*$, and $T \in \mathbb{G}_T$ chosen uniformly at random:

- $\mathcal{P}_A := (g, g^a, g^c, h, h^a, h^b, e(g,h)^{abc}) \in \mathbb{G}_1^3 \times \mathbb{G}_2^3 \times \mathbb{G}_T$
- $\mathcal{R}_A := (g, g^a, g^c, h, h^a, h^b, T) \in \mathbb{G}_1^3 \times \mathbb{G}_2^3 \times \mathbb{G}_T$.

For an algorithm $\mathcal{A}$, we let $Adv_{\mathcal{A}}^{DBDH}$ be the advantage of $\mathcal{A}$ in distinguishing these two distributions:

$$Adv_{\mathcal{A}}^{DBDH} = |Pr[\mathcal{A}(D) = 1] - Pr[\mathcal{A}(R) = 1]|,$$

where is sampled from $\mathcal{P}_A$ and R is sampled from $\mathcal{R}_A$. We say that an algorithm $\mathcal{B}$ that outputs a bit in $\{0,1\}$ has advantage $Adv_{\mathcal{A}}^{DBDH} = \epsilon$ in solving the *DBDH* problem with asymmetric pairing if:

$$|Pr[\mathcal{B}(g, g^a, g^c, h, h^a, h^b, e(g,h)^{abc}) = 0] - Pr[\mathcal{B}(g, g^a, g^c, h, h^a, h^b, T) = 0]| \geq \epsilon,$$

where the probability is over the random choice of generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$, exponents $a, b, c \in \mathbb{Z}_p^*, T \in \mathbb{G}_T$, and the random bits used by $\mathcal{B}$.

As usual, to state the assumption asymptotically, we rely on a bilinear group generator $\mathcal{G}$ that takes a security parameter λ as input and outputs the description of a bilinear group.

Definition 2.3. Let $\mathcal{G}$ be a bilinear group generator. We say that the *DBDH* holds for $\mathcal{G}$ if, for all probabilistic polynomial-time *PPT* algorithms $\mathcal{A}$, the function $Adv_{\mathcal{A}}^{DBDH}(\lambda)$ is a negligible function of λ.

- $\mathcal{P}$-Asymmetric Decisional Bilinear Diffie-Hellman ($\mathcal{P}$-DBDH)[19], [23]
For $g \in \mathbb{G}_1, h \in \mathbb{G}_2, a, b, c \in \mathbb{Z}_p^*$, and $T \in \mathbb{G}_{\mathbb{T}}$ chosen uniformly at random, we consider the following two distributions:

- $\mathcal{D}_N := (g, g^a, g^{ab}, g^c, h, h^a, h^b, g^{abc}) \in \mathbb{G}_1^4 \times \mathbb{G}_2^3 \times \mathbb{G}_1$
- $\mathcal{D}_R := (g, g^a, g^{ab}, g^c, h, h^a, h^b, T) \in \mathbb{G}_1^4 \times \mathbb{G}_2^3 \times \mathbb{G}_1$.

For an algorithm $\mathcal{A}$, let $Adv_{\mathcal{A}}^{\mathcal{P}-DBDH}$ be the advantage of $\mathcal{A}$ in distinguishing these two distributions:

$$Adv_{\mathcal{A}}^{\mathcal{P}-DBDH} = |Pr[\mathcal{A}(N) = 1] - Pr[\mathcal{A}(P) = 1]|,$$

where N is sampled from $\mathcal{D}_N$ and P is sampled from $\mathcal{D}_R$. We say that an algorithm $\mathcal{B}$ that outputs a bit in $\{0,1\}$ has the advantage $Adv_{\mathcal{A}}^{\mathcal{P}-DBDH} = \epsilon\mathcal{P}$ in solving the $\mathcal{P} - DBDH$ problem in asymmetric pairing if:

$$|Pr[\mathcal{B}(g, g^a, g^{ab}, g^c, h, h^a, h^b, g^{abc}) = 0] - Pr[\mathcal{B}(g, g^a, g^{ab}, g^c, h, h^a, h^b, T) = 0]| \geq \epsilon\mathcal{P},$$

where the probability is over the random choice of generator $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$, exponents $a, b, c \in \mathbb{Z}_p^*, T \in \mathbb{G}_1$, and the random bits used by $\mathcal{B}$.

Definition 2.4. Let $\mathcal{G}$ be a bilinear group generator. We say that the $\mathcal{P} - DBDH$ holds for $\mathcal{G}$ if, for all probabilistic polynomial-time PPT algorithms $\mathcal{A}$, the function $Adv_{\mathcal{A}}^{\mathcal{P}-DBDH}(\lambda)$ is a negligible function of λ.

Chapter 3

Literature Review

In this chapter, we present a review of the research literature related to secure and confidential data sharing in the cloud computing. We first introduce a type of public-key encryption, namely attribute-based encryption (*ABE*) scheme, which is used to enable secure data sharing in the cloud. Then, we present two types of *ABE* scheme: key-policy attribute-based encryption (*KP-ABE*) and ciphertext-policy attribute-based encryption (*CP-ABE*) schemes and compare both of these schemes in terms of the size of keys, ciphertext, and computational overhead. We then describe an advanced cryptographic technique known as proxy re-encryption (*PRE*) scheme and introduce the different types of this scheme.

3.1 Attribute-based Encryption

The concept of attribute-based encryption (*ABE*) was proposed by Sahai and Waters [8] as an extension of identity-based encryption (*IBE*) scheme [24], to express and enforce complex access control policies in both a simple and a flexible way to achieve the privacy-preserving goal. In their scheme (called fuzzy *IBE* (*FIBE*)) identities are viewed as a set of descriptive attributes where a data owner can encrypt a message to all users who have a certain set of attributes and a user can decrypt a message when at least d attributes overlap between the encrypted data and his private key. Although this scheme can prevent the collusion attacks, it only supports threshold access policies, which limits its applicability to large systems due to the lack of expressibility.

In an *ABE* system, both ciphertexts and user's private keys are associated with an attribute set. This system allows any user to decrypt a ciphertext if and only if the set of attributes in his private key matches the

attributes of the ciphertext. Based on whether the access policy is embedded in the private key or the ciphertext, *ABE* schemes can be classified into two categories: key-policy attribute-based encryption (*KP-ABE*) scheme [11] and ciphertext-policy attribute-based encryption (*CP-ABE*) scheme [9].

3.1.1 Properties of Attribute-based Encryption Schemes

Some fundamental properties of a well-designed *ABE* scheme are listed as follows [25]:

1) ***Data confidentiality:*** the data is encrypted by the data owner before outsourcing to the cloud. Therefore, unauthorized users cannot learn any information about the encrypted data.

2) ***Fine-grained access control:*** by providing different private keys for users, the system authority is able to restrict user access rights to specific resources. This property achieves flexible access control, even for users in the same group, their access rights are not the same.

3) ***Scalability:*** the number of authorized users cannot affect the performance of the scheme. Therefore, the scheme can deal with the case that the number of authorized users increases dynamically.

4) ***User/attribute revocation:*** if a user quits the system, the scheme can revoke his access right. Similarly, attribute revocation is inevitable.

6) ***Collusion resistance:*** if users combine their private keys they should not be able to decrypt ciphertexts which they could not decrypt individually. Because each private key is generated with a random seed, combining keys cannot create a new meaningful key.

3.1.2 Key-policy Attribute-based Encryption Scheme

The idea of key-policy attribute-based encryption (*KP-ABE*) system for fine-grained access control over encrypted data was proposed by Goyal *et al.*[11]. A fine-grained access control system allows that different authorized users can decrypt different pieces of encrypted data based on the access policy. In *KP-ABE*, where access policy is built in the user's private key, a user can decrypt the message if and only if the encrypted data with attributes satisfies the access policy of user's private key.

Although Goyal *et al.*'s *KP-ABE* scheme can achieve fine-grained access control and more flexibility to control users than *ABE* scheme, the access

structure in their scheme is a monotonic access structure which cannot express the negative attribute to exclude the participants with whom the data owner does not want to share data [25].

To address this issue, Ostrovsky *et al.* [26] presented the first *KP-ABE* scheme with a non-monotonic access structure by adopting the idea from the broadcast revocation scheme of Naor and Pinkas [27] to the *ABE* scheme of Goyal *et al.* [11]. In particular, their construction can handle any access structure that can be represented by a boolean formula involving *AND*, *OR*, *NOT*, and threshold operations. For example, if a teacher in the department of information management wants to share the data with students, he will define a set of attributes in the ciphertext and there is an access structure $\{MIS^1 \bigwedge Student\}$ in student's private key. But the teacher does not want graduates to access this data, he adds "$NOT\, graduate$" to the access structure: $\{MIS \bigwedge Student \bigwedge NOT\, graduate\}$. So the access structure can let data not be accessed by graduates. However, their mechanism increases the private key size by a multiplicative factor of $\log n$, where n is the maximum number of attributes and adds encryption and decryption overhead at the same time, since there are many negative attributes in the encrypted data and each attribute adds a negative word to describe it but they are useless for decrypting the ciphertexts.

Then Lewko *et al.* [28] improved the initial construction of Ostrovsky *et al.* [26] by using a revocation system with short key instead of Naor and Pinkas [27] scheme and designed the most efficient non-monotonic *KP-ABE* scheme, where the key storage is significantly more efficient than previous solutions.

In the above *KP-ABE* schemes, the size of the ciphertext increases linearly with the number of attributes in the system. Therefore, to cope with the ciphertext overhead, Attrapadung *et al.* [29] proposed the first non-monotonic *KP-ABE* scheme with constant size ciphertexts. To achieve this goal, they applied the technique of Lewko *et al.* [28] to their efficient identity-based revocation mechanism and combined it with the monotonic *KP-ABE* which is derived from a certain class of identity-based broadcast encryption (*IBBE*) scheme. Although the ciphertext size of their scheme is constant and decryption just need 3 pairing operations, but the private keys have a quadratic size in the number of attributes. Table 3.1 compares efficiency among available expressive *KP-ABE* schemes that support non-monotonic access structures.

[1] Management Information System

Scheme	Private key size	Ciphertext overhead	Decryption cost
Ostrovsky *et al.* [26]	$O(t \cdot \log n)$	$O(\overline{n})$	$O(t)$
Lewko *et al.* [28]	$O(t)$	$O(\overline{n})$	$O(t)$
Attrapadung *et al.* [29]	$O(t \cdot n)$	3	3

n: the maximum number of attributes in the system, $\overline{n}$: the number of attributes for a ciphertext, t: the number of attributes in an access structure for a key.

Table 3.1: Comparison of non-monotonic *KP-ABE* schemes

Formal Definition of KP-ABE Scheme

A *KP-ABE* scheme consists of four fundamental algorithms Setup, Encrypt, KeyGen, and Decrypt with the following functions:

- $(PK, MSK) \leftarrow$ **Setup** (sp). This algorithm is run by the trust authority (TA), which takes as input a security parameter sp and outputs a public key PK and a master secret key MSK.

- $CT \leftarrow$ **Encrypt** (PK, γ, M). This algorithm is run by a data owner who takes as input the public key PK, a set of attributes γ, and a message M to output a ciphertext CT associated with the set of attributes γ.

- $SK \leftarrow$ **KeyGen** $(MSK, PK, \mathbb{A})$. This algorithm is run by the TA. It takes as input the master secret key MSK, the public key PK, and the access control structure $\mathbb{A}$. It outputs a private key SK associated with the access control structure $\mathbb{A}$.

- $M \leftarrow$ **Decrypt** (SK, CT). This algorithm is run by the user. It takes as input the private key SK and the ciphertext CT. It outputs either a message M if $\mathbb{A}(\gamma) = 1$ or the distinguished symbol $\perp$ if $\mathbb{A}(\gamma) = 0$.

3.1.3 Ciphertext-policy Attribute-based Encryption Scheme

Although *KP-ABE* scheme can achieve fine-grained access control and reduces the communication overhead, the data owner has no control over who has access to his data, except by his choice of a set of attributes for the data. Therefore, the data owner must trust the key-issuer to issue the appropriate keys for granting or denying access to the authorized users. This mechanism is unsuitable in some distributed systems where a user should only have access to data if he has a certain set of credentials or attributes. A straightforward method for enforcing such policies is to employ a trusted

server for storing the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised.

To address this problem, Bethencourt *et al.* [9] proposed the first ciphertext policy attribute-based encryption (*CP-ABE*) scheme based on Goyal *et al.*'s scheme [11]. In *CP-ABE* scheme, access policy is built in the encrypted data and a user's private key is associated with a set of attributes. A user will be able to decrypt a ciphertext if and only if the attribute set in his private key satisfies the access policy of encrypted data. For example in healthcare application, if a patient encrypts his medical record under the access policy $I = (Physician \wedge Policlinico\ of\ Milan) \vee (Insurance\ company\ A \wedge Healthinsurance\ department)$. A user who wants to decrypt this file must be a physician in "*Policlinico of Milan*" or he must be an employee in the health insurance department of a particular insurance company. Therefore, by using *CP-ABE* technique the data owner is able to decide who should or should not have access to his data and does not require to know the exact identities of all authorized users. Moreover, the encrypted data can be kept confidential even if the storage server is untrusted. Bethencourt *et al.* [9] also proposed a novel construction for preventing collusion users attacks. In the technique used by Sahai and Waters [8], collusion resistance is insured by using a secret-sharing scheme and embedding independently chosen secret shares into each private key. Because of the independence of the randomness used in each invocation of the secret sharing scheme, collusion-resistance follows. However, in Bethencourt *et al.*'s scheme [9], user's private keys are associated with sets of attributes instead of access structures over them, and so secret sharing schemes do not apply. Instead, they propose a novel private key randomization technique that uses a new two-level random masking methodology. While Bethencourt *et al.*'s construction [9] is very expressive, but its security proof is in the generic bilinear group model an artificial model which assumes the attacker needs to access an oracle in order to perform any group operations. However, it is desirable to propose a scheme which is provably secure under a more standard assumption.

To address this issue, Cheung *et al.* [30] presented the first formal chosen-ciphertext (*CCA*) security proof for *CP-ABE* under the Decisional Bilinear Diffie-Hellman (*DBDH*) assumption, in which access structures are *AND* gates on positive and negative attributes. Existing *ABE* schemes [8, 11, 9] involve some form of threshold secret sharing construction. In [8, 11], shares of a system master secret are embedded into user's private keys, while in [9] shares of the randomness in an encryption are embedded into ciphertext components. However, Cheung *et al.* [30] break from this tradition and show

that by separating threshold secret sharing from the *CP-ABE* primitive, they can obtain simple and efficient schemes that are provably secure under standard complexity assumptions. Furthermore, threshold access policies can be re-introduced in an independent mechanism; namely, one can construct shares of a message using a standard secret sharing scheme and then encrypt each share independently using *CP-ABE*. Cheung *et al.* [30] also used a "*don't care*" element to indicate the attribute which does not appear in the *AND* gate. Therefore, their scheme treats negation and "*don't care*" in a more streamlined fashion. Moreover, their scheme does not involve any secret sharing construction, therefore no exponentiations are necessary in their Decrypt algorithm, while the Bethencourt *et al.*'s Decrypt algorithm requires "d" (threshold value) exponentiations in order to perform polynomial interpolation. Although Cheung *et al.*'s scheme [30] improves the security proof in Bethencourt *et al.*'s [9], the ciphertext and private key size and Encryption/Decryption time increase linearly with the total number of attributes in the system. In contrast, the ciphertext and private key size and Encryption/Decryption time of Bethencourt *et al.*'s scheme [9] are linear in the size of the access structure. Therefore, Cheung *et al.*'s scheme [30] is less efficient than Bethencourt *et al.*'s scheme [9].

Nishide *et al.* [31] improved the security of Cheung *et al.*'s scheme [30] by proposing a recipient-anonymous *CP-ABE* scheme, where an access structure associated with ciphertext is hidden and a decryptor cannot obtain the information about the ciphertext policy. For example, suppose a company wants to hire certain qualified people who satisfy the policy the company specified and the policy may contain the useful information about the company's business strategy. The company can post a message encrypted by Nishide *et al.*'s *CP-ABE* scheme on a public bulletin board to seek applications. By doing so, the company can keep the important policy confidential. Since the policy is hidden, the rival companies cannot know what kind of policy the company used to hire its employees. Nishide *et al.* [31] also generalized their idea to adopt the access structures used in Cheung *et al.*'s scheme [30] to the multi-valued attribute setting where an attribute can take multiple values. They proved the security of their construction based on the Decisional Bilinear Diffie-Hellman (*DBDH*) assumption and the Decision Linear assumption.

In order to design *CP-ABE* scheme with flexible strategy under *DBDH* assumption, Goyal *et al.* [32] and Liang *et al.* [33] presented bounded tree structure.

Goyal *et al.* [32] designed a bounded ciphertext-policy attribute-based encryption (*BCP-ABE*) scheme having a secure proof based on a standard

theoretic assumption and supporting advanced access structures. Their construction can support access structures which can be represented by a bounded size access tree with threshold gates as its nodes. The bound on the size of the access tree is chosen at the time of the system setup and is represented by a tuple *(d, num)*, where "d" represents the maximum depth of the access tree and "*num*" represents the maximum number of children each non-leaf node of the tree might have. Each access tree satisfies these upper bounds on the size can be dynamically chosen by the encryptor. Furthermore, they extended their construction to support non-monotonic access policies which can express any access formula with bounded polynomial size (including the *AND*, *OR* and *threshold* operations). The drawback of their scheme is that the depth of the access tree "d" under which messages can be encrypted is defined in the setup phase. Thus, the message sender is restricted to use only an access tree which has the depth $d' \leqslant d$. However, constructing a more efficient *CP-ABE* scheme based on number-theoretic assumptions was left as an important open question.

Liang *et al.* [33] improved the *BCP-ABE* scheme of Goyal *et al.* [32] and presented an efficient *BCP-ABE* scheme (called BCP_1). The security proof of BCP_1 is reduced to *DBDH* assumption in the standard model. By eliminating redundant steps, their scheme provided faster Encrypt/Decrypt algorithm and shortened the length of public key, private key, and ciphertext compared with Goyal *et al.*'s scheme [32]. Moreover, they proposed a provably secure *BCP-ABE* scheme (called BCP_2) in the standard model under chosen ciphertext secure notion by adopting one-time signature technique.

Later, Ibraimi *et al.* [34] used the general access tree structure to eliminate the boundary constraints in [32, 33]. Firstly, they presented a new technique to construct a *CP-ABE* scheme without using Shamir's threshold secret sharing technique [35]. Their proposed scheme is appropriate for resource-constrained devices since calculating polynomial interpolations to construct the secret is computationally expensive. In their scheme, the encryptor specifies the policy in the encryption phase by using an n-ary access tree which is represented by *AND* and *OR* nodes. Their scheme requires less computation overhead for Encrypt and Decrypt algorithms compared with Cheung *et al.*'s scheme [30]. Secondly, they extended their scheme and presented a second *CP-ABE* scheme which uses Shamir's threshold secret sharing technique [35]. In their modified scheme, the policy can be expressed as an n-ary access tree which consists of *AND*, *OR* and "*of*" nodes. They compared the efficiency of their scheme with Bethencourt *et al.*'s scheme [9] and showed that their scheme requires less computation overhead in the Encrypt and Decrypt algorithms.

In all previous *ABE* schemes [8, 11, 9, 30, 31, 32, 33, 34], the length of the ciphertext and the number of pairing operations depend on the number of attributes. Therefore, Emura *et al.* [36] proposed the first *CP-ABE* scheme with a constant ciphertext length and a constant number of pairing operations. In their scheme, the sum of master keys is used to achieve the constant ciphertext length. The access structure used in their *CP-ABE* is constructed by *AND*-gates on multi-valued attributes, which is a subset of the access structures used in [30, 31]. However, their scheme is inefficient in that the size of public key grows linearly with the number of attributes.

Waters [37] proposed a new methodology for realizing *CP-ABE* systems from a general set of access structures that are efficient, expressive, and provably secure under concrete and non-interactive assumptions. He presented three constructions which provide a trade-off in terms of efficiency and the complexity of assumptions. His constructions allow any encryptor to specify access control in terms of any access formula. His first construction expressed access control by a linear secret sharing scheme (*LSSS*) matrix M over the attributes in the system. This scheme is proven selectively secure under the decisional Parallel Bilinear Diffie-Hellman Exponent (*PB-DHE*) assumption which can be viewed as a generalization of the *BDHE* assumption [38]. In this most efficient system, the ciphertext size and encryption/decryption overhead increase linearly with the complexity of the access formula. His scheme achieves the same performance and functionality as Bethencourt *et al.*'s scheme [9] but under the standard model. In addition, he presented two other constructions which provide performance trade-off to achieve provable security under the (weaker) decisional Bilinear-Diffie-Hellman Exponent (*d-BDHE*) and decisional Bilinear Diffie-Hellman assumptions, respectively.

Previous constructions of *ABE* schemes were only proven to be selectively secure. To achieve adaptive (non-selective) security, Lewko *et al.* [39] used a novel information-theoretic argument to adapt the dual system encryption methodology introduced by Waters [37] to the more complicated structure of *ABE* system. Their *ABE* scheme supports arbitrary monotone access formulas. They constructed their system in composite order bilinear groups, where the order is a product of three primes. This causes less practical efficiency compared with Waters's scheme. They proved the security of their system under three static assumptions used by Lewko and Waters [40].

Until Zhang *et al.* [41] proposed a *CP-ABE* from lattice assumptions, there have been only implementations of *ABE* schemes based on bilinear pairings as mentioned above. Previous to their work no other cryptographic

assumptions than bilinear pairings have been utilized for *ABE* schemes. Their construction is defined on q-ary lattices and supports *AND*-gate access structure on positive and negative attributes. Their proposed scheme has a very strong security proof under the learning with error (*LWE*) assumption. Although their construction seems to be not much efficient, it gives light to the possibility of constructing attribute schemes under other hard problem assumptions (e.g., lattice problems), instead of the pairing-related assumptions. They remain an open problem to obtain a *CP-ABE* scheme that can support more general access structure from lattices.

3.1.3.1 Formal Definition of CP-ABE Scheme

A *CP-ABE* scheme also consists of four fundamental algorithms Setup, Encrypt, KeyGen, and Decrypt with the following functions:

- $(PK, MSK) \leftarrow$ **Setup** (sp). This algorithm is run by the trust authority (TA), which takes as input a security parameter sp and outputs a public key PK and a master secret key MSK.
- $CT \leftarrow$ **Encrypt** $(PK, \mathbb{A}, M)$. This algorithm is run by a data owner who takes as input the public key PK, an access control structure $\mathbb{A}$, and a message M to output a ciphertext CT associated with the access control structure $\mathbb{A}$.
- $SK \leftarrow$ **KeyGen** (MSK, PK, γ). This algorithm is run by the TA. It takes as input the master secret key MSK, the public key PK, and the set of attributes γ. It outputs a private key SK associated with the set of attributes γ.
- $M \leftarrow$ **Decrypt** (SK, CT). This algorithm is run by the user. It takes as input the private key SK and the ciphertext CT. It outputs either a message M if $\mathbb{A}(\gamma) = 1$ or the distinguished symbol $\perp$ if $\mathbb{A}(\gamma) = 0$.

3.1.4 Comparison between ABE schemes

The basic *ABE* scheme, *KP-ABE*, and *CP-ABE* schemes are different in terms of complexity hypothesis, strategic flexibility, and applications. The basic *ABE* scheme, which only supports threshold policy, is suitable for simply policy-required applications. On the other hand, *KP-ABE* and *CP-ABE*, which support complex policies, are appropriate for the applications of fine-grained data sharing.

In *KP-ABE* schemes, the access policy is built into the user's private key; therefore, the data owner cannot choose who decrypts his data. However,

in *CP-ABE* schemes, a ciphertext is associated with an access policy and makes them more suitable for the realistic application compared with *KP-ABE* schemes. In general, *KP-ABE* schemes apply to query applications such as pay-TV system, audit log, targeted broadcast, and database access; while, *CP-ABE* schemes are used for access control applications such as social networking site access and electronic medical system.

The comparison of the keys size and ciphertext and the computational overhead of Encrypt and Decrypt algorithms of different *ABE* schemes are shown in Table 3.2 and Table 3.3, respectively.

Scheme	**Public Key**	**Private Key**	**Ciphertext**
Sahai *et al.* [8]	$n\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$A_U\|\mathbb{G}\|$	$A_C\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Goyal *et al.*[11]	$n\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$A_U\|\mathbb{G}\|$	$A_C\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Bethencourt *et al.* [9]	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2A_U+1)\|\mathbb{G}\|$	$(2A_C+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Cheung *et al.* [30]	$(3n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n+1)\|\mathbb{G}\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Nishide *et al.* [31]	$(2N'+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n+1)\|\mathbb{G}\|$	$(2N'+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Ibraimi *et al.* [34]	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(A_U+1)\|\mathbb{G}\|$	$(A_C+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Emura *et al.* [36]	$(N'+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Waters [37]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(A_U+2)\|\mathbb{G}\|$	$(2A_C+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Lewko *et al.* [39]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(A_U+2)\|\mathbb{G}\|$	$(2A_C+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$

$|\mathbb{G}|$: bit length of element in $\mathbb{G}$, $|\mathbb{G}_T|$: bit length of element in $\mathbb{G}_T$, n: the number of attributes in the system, $N' = \Sigma_{i=1}^{n} n_i$: the total number of possible values for attributes, where n_i is the number of possible values for attribute i, A_C ($|A_C|= A_C$): a set of attributes associated with the ciphertext, A_U ($|A_U|= A_U$): a set of attributes associated with the private key.

Table 3.2: Comparison of the size of keys and ciphertext of *ABE* schemes

Scheme	**Encrypt**	**Decrypt**
Sahai *et al.* [8]	$A_C\mathbb{G} + \mathbb{G}_T$	$A_U P + (A_U + 1)\mathbb{G}_T$
Goyal *et al.*[11]	$A_C\mathbb{G} + \mathbb{G}_T$	$A_U P + (A_U + 1)\mathbb{G}_T$
Bethencourt *et al.* [9]	$(2A_C + 1)\mathbb{G} + 2\mathbb{G}_T$	$(2A_U)P + (2\|S\|+2)\mathbb{G}_T$
Cheung *et al.* [30]	$(n + 1)\mathbb{G} + 2\mathbb{G}_T$	$(n + 1)P + (n + 1)\mathbb{G}_T$
Nishide *et al.* [31]	$(2N' + 1)\mathbb{G} + 2\mathbb{G}_T$	$(3n + 1)P + (3n + 1)\mathbb{G}_T$
Ibraimi *et al.* [34]	$(A_C + 1)\mathbb{G} + 2\mathbb{G}_T$	$(A_U + 1)P + (A_U + 1)\mathbb{G}_T$
Emura *et al.* [36]	$(n + 1)\mathbb{G} + 2\mathbb{G}_T$	$2P + 2\mathbb{G}_T$
Waters [37]	$(4A_C + 1)\mathbb{G} + 2\mathbb{G}_T$	$(2A_U)P + (3A_U)\mathbb{G}_T$
Lewko *et al.* [39]	$(4A_C + 1)\mathbb{G} + 2\mathbb{G}_T$	$(2A_U)P + (3A_U)\mathbb{G}_T$

$\mathbb{G}$: operation in group $\mathbb{G}$, $\mathbb{G}_T$: operation in group $\mathbb{G}_T$, P: paring computation, n: the number of attributes in the system, A_C ($|A_C|= A_C$): a set of attributes associated with the ciphertext, A_U ($|A_U|= A_U$): a set of attributes associated with the private key, $N' = \Sigma_{i=1}^{n} n_i$: the total number of possible values for attributes, where n_i is the number of possible values for attribute i, $|S|$: least number of interior nodes satisfying an access structure (including root node).

Table 3.3: Comparison of computation overhead of *ABE* schemes

From these comparisons, we conclude that Emura *et al.*'s scheme [36] is more efficient compared to other *ABE* schemes since its private key size and ciphertext length and the number of pairing operation in its Decrypt algorithm are independent of the number of attributes. Hence, their Decrypt algorithm requires constant pairing operations. In addition, Bethencourt *et al.*'s scheme [9] has the shortest public key that its size does not depend on the number of attributes. Furthermore, the computation overhead of Encrypt and Decrypt algorithms in Ibraimi *et al.*'s scheme [34] are lower than Waters scheme[37].

According to different access structures, they can be divided into four kinds: threshold structure, tree-based structure, *AND*-gate structure, and *LSSS* matrix structure. The comparison of complexity assumptions, security models and supported access structures of different *ABE* schemes are shown in Table 3.4 and Table 3.5, respectively.

Scheme	**Policy**	**Recipient Anonymity**	**Assumption**	**Model**
Sahai *et al.* [8]	-	No	DBDH	Selective
Goyal *et al.*[11]	Key	No	DBDH	Selective
Bethencourt *et al.* [9]	Ciphertext	No	Generic group	Adaptive
Cheung *et al.* [30]	Ciphertext	No	DBDH	Selective
Nishide *et al.* [31]	Ciphertext	Yes	DBDH, D-linear	Selective
Goyal *et al.* [32]	Ciphertext	No	DBDH	Selective
Liang *et al.* [33]	Ciphertext	No	DBDH	Selective
Ibraimi *et al.* [34]	Ciphertext	No	DBDH	Selective
Emura *et al.* [36]	Ciphertext	No	DBDH	Selective
Waters [37]	Ciphertext	No	PBDHE	Selective
Lewko *et al.* [39]	Ciphertext	No	3P-SDP	Adaptive

DBDH: Decisional Bilinear Diffie-Hellman, $D-linear$: Decision Linear, *PBDHE*: Parallel Bilinear Diffie-Hellman Exponent, $3P-SDP$: Subgroup Decision Problem for 3 Primes.

Table 3.4: Comparison of security proof and some properties of *ABE* schemes

Scheme	**Access Structure**
Sahai *et al.* [8]	Threshold structure
Goyal *et al.* [11]	Tree-based structure without bound
Bethencourt *et al.* [9]	Tree-based structure without bound
Cheung *et al.* [30]	AND-gate on positive and negative attributes with wildcards
Nishide *et al.* [31]	AND-gate on multi-valued attributes with wildcards
Goyal *et al.* [32]	Bounded tree
Liang *et al.* [33]	Bounded tree
Ibraimi *et al.* [34]	Tree-based structure without bound
Emura *et al.* [36]	AND-gates on multi-valued attributes
Waters [37]	LSSS matrix
Lewko *et al.* [39]	LSSS matrix

LSSS: Linear Secret-Sharing Scheme.

Table 3.5: Comparison of access structure of *ABE* schemes

3.2 Proxy Re-encryption Scheme

The concept of proxy re-encryption (PRE) was first proposed by Mambo and Okamoto [42] as a way to support the delegation of decryption rights. Since then, many proxy re-encryption schemes have been proposed in the literature. A seminal paper by Blaze *et al.* [43] proposed a bidirectional PRE scheme (BBS proxy re-encryption scheme) based on ElGamal cryptosystem [44] and introduced the notion of "*re-encryption key*". Using this key, a semi-trusted party called *proxy* can transform a ciphertext that encrypted under user A's (delegator) public key (PK_A) into another ciphertext of the same plaintext that encrypted under user B's (delegatee) public key (PK_B) without revealing the underlying plaintext and user private key. In their scheme, the data owner (part A) chooses a random number $r \in \mathbb{Z}_p^*$ and encrypts the message M using his pair of private\public key ($SK_A = a, PK_A = g^a$) to obtain ciphertext $C_A = (g^r \cdot M, (g^a)^r)$ and computes $b \cdot a^{-1}$ as the re-encryption key ($rk_{A \to B} = b/a = b \cdot a^{-1}$), where b is the user's private key, and then transfers the ciphertext C_A and $rk_{A \to B}$ to the proxy. The proxy uses the $rk_{A \to B}$ to transform C_A to $C_B = (g^r \cdot M, (g^{ar})^{rk_{A \to B}}) = (g^r \cdot M, g^{ar})^{b/a}) = (g^r \cdot M, g^{br})$. When the user (part B) receives the new ciphertext C_B, he decrypts it with his private key b and gets the message: $M = \frac{g^r \cdot M}{(g^{br})^{(1/b)}}$. Although BBS proxy re-encryption scheme is secure against chosen plaintext attacks (CPA); however, it requires pre-sharing private key between parties in order to compute re-encryption key and has *bidirectional* and *transitive* properties and exposed to collusion attacks. As such, the proxy can compute $(rk_{A \to B})^{-1} = a/b$ to obtain the re-encryption key for transforming ciphertexts in the opposite direction, from part B to part A (*bidirectional* property), therefore it is only useful when the trust relationship between involved parties is mutual. Moreover, the proxy can combine the two re-encryption keys $rk_{A \to B} = b/a$ and $rk_{B \to C} = c/b$ that have never agreed on this and gets the valid re-encryption key from part A to part C: $rk_{A \to C} = c/a = (b/a) \cdot (c/b)$ (*transitive* property). Furthermore, if the proxy colludes with one party (e.g, part A) they can recover the private key of the other party ($(a/b) \cdot b = a$) (collusion attack).

To tackle these disadvantages, Ateniese et al. [45] proposed the first unidirectional and collusion resistant proxy re-encryption scheme without requiring pre-sharing between parties, based on bilinear maps ($e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$). In their scheme, the data owner A computes proxy re-encryption key using his private key ($SK_A = a$) and user's public key ($PK_B = g^b$) in the form of $rk_{A \to B} = (g^b)^{1/a} = g^{b/a}$; therefore, the user B does not need to share his private key. Such configuration provides *non-interactive* property since re-

encryption key is created without any interaction with the user. Moreover, the possession of two re-encryption keys, $rk_{A\to B} = g^{b/a}$ and $rk_{B\to C} = g^{c/b}$, the proxy cannot find out $rk_{A\to C} = g^{c/a}$ due to the Decision Diffie-Hellman problem [46] ($non - transitive$ property). Furthermore, their scheme is collusion-resistant, it is hard for the proxy to collude with one of the parties and extracts the private key of another party from $rk_{A\to B} = g^{b/a}$.

The notion of proxy re-encryption approaches have been extended to adopt different cryptographic settings and creates various data encryption techniques such as type-based proxy re-encryption ($TBPRE$), identity-based proxy re-encryption ($IBPRE$) and attribute-based proxy re-encryption ($ABPRE$). The main advantages and disadvantages of the above-mentioned approaches are summarized in Table 3.6.

3.2.1 Properties of Proxy Re-encryption Scheme

Proxy re-encryption schemes are characterized based on different criteria [43, 47], which are listed as follows:

- ***Unidirectionality:*** the proxy is only able to transform a ciphertext C into ciphertext $C^{'}$ in one direction ($C \to C^{'}$) and does not allow a transform from ($C^{'} \to C$).
- ***Non-interactive:*** the re-encryption key can be computed by the key generator without any interaction with the users or the proxy.
- ***Non-transitive:*** the proxy cannot combine provided re-encryption keys to re-delegate decryption rights. If the proxy has two re-encryption keys, $rk_{A\to B}$ and $rk_{B\to C}$, it can not generate the re-encryption key $rk_{A\to C}$ from $rk_{A\to B}$ to $rk_{B\to C}$.
- ***Multi-use:*** the proxy (or proxies) can re-encrypt a ciphertext multiple times (e.g, re-encrypt from A to B, then re-encrypt the result from B to C and etc).
- ***Collusion resistance:*** the proxy and user cannot collude their secrets to generate a new private key also, two users cannot combine their private keys to extend their decryption power.

3.2.2 Type-based Proxy Re-encryption Scheme

The concept of type-based proxy re-encryption scheme ($TBPRE$) was proposed by [48] to address the inefficiency issues of traditional proxy re-encryption schemes in practical applications. In this scheme, a delegator

decrypts his message based on his public key and the message type which is used to identify the message subset. Then the delegator categorizes his ciphertexts into different subsets and delegates the decryption right of each subset to a specific delegatee. The type-based proxy re-encryption enables the delegator to implement fine-grained policies with one key pair without any additional trust on the proxy. This scheme has the following promising features:

- Simplifies key management problem since the delegator only needs one key pair.
- The delegator can choose a particular proxy for a specific delegatee, which might be based on the sensitiveness of the delegation. Compromise of one proxy key will only affect one subset of messages.

3.2.3 Identity-based Proxy Re-encryption Scheme

Identity-based encryption (*IBE*) [24] scheme allows a data owner encrypts a message using the user's identity as the public key without exchanging private or public keys and without keeping key directories, which simplifies key management and reduces transmission overhead.

In identity-based proxy re-encryption scheme [49] a proxy with re-encryption key can translate a ciphertext computed under one identity into a new ciphertext under another identity. This scheme ensures that no reasonable set of colluding key holders will obtain an advantage against non-colluding users.

Scheme	**Advantages**	**Disadvantages**
Proxy re-encryption	Secure against plaintext attack	Collusion problem and plaintext attack
Type-based proxy re-encryption	Semantic security and ciphertext privacy control	Encoding operations over encrypted messages is not possible
Identity-based proxy re-encryption	Secure against an adaptive chosen plaintext attack	Difficult to find efficient constructions for multiuse *CCA*-secure
Attribute-based proxy re-encryption	Fine-grained access control on encrypted data	Average efficiency and flexibility

CCA: Chosen Ciphertext Attack.

Table 3.6: Comparison of different proxy re-encryption schemes

3.2.4 Attribute-based Proxy Re-encryption Scheme

Attribute-based proxy re-encryption (*ABPRE*) scheme is one of the proxy cryptography, which extends the traditional proxy re-encryption to the attribute-based encryption scheme. The *ABPRE* scheme delegates the re-encryption capability to the proxy who can re-encrypt a ciphertext to the new ciphertext with the same plaintext by using the re-encryption key. The *ABPRE* scheme has some advantages: (i) the proxy executes the re-encryption operation, which reduces the computation overhead of the data owner, (ii) the authorized user just uses his private key to decrypt the encrypted data, and he does not require to store an additional decryption key for deciphering, and (iii) the sensitive information and the decryption key cannot be revealed to the proxy in re-encryption procedure.

The concept of attribute-based proxy re-encryption (*ABPRE*) was first introduced by Guo *et al.* [50] based on key-policy attribute-based encryption (*KP-ABE*) scheme [11] and a general proxy re-encryption scheme. Furthermore, they proved the security of their scheme in the standard model based on *DBDH* assumption. Under this scheme, a semi-trusted proxy server transforms a ciphertext associated with a set of attributes into a new ciphertext associated with different attributes set, but not vice versa, while preserving the privacy of the underlying plaintext and user private key. This scheme can be used for sharing encrypted data between users and supports fine-grained access control, but the general proxy re-encryption scheme cannot do it, so the proposed scheme can be thought as an improvement of general traditional proxy re-encryption scheme.

Generally speaking, a ciphertext-policy attribute-based encryption (*CP-ABE*) is more appropriate for proxy-based techniques than *KP-ABE* because in *CP-ABE* the encryptor uses the access policy in the ciphertext and has control over the user access.

The seminal paper of Liang *et al.* [51] proposed the first ciphertext-policy attribute-based proxy re-encryption (*CP-ABPRE*) scheme by adopting identity-based proxy re-encryption [49] to the construction of *CP-ABE* scheme [30]. In Liang's scheme, a proxy transforms a ciphertext generated under an access policy to another one corresponding to the same plaintext but to a different access policy. Liang *et al.*'s scheme satisfies *multi-use* property and supports only access policies with *AND*-gates on positive and negative attributes (NOT). However, in this scheme, the size of the ciphertext increases linearly with the number of attributes in the system.

Later, Luo *et al.* [52] presented a ciphertext-policy *ABPRE*, which supports *AND*-gates access policies on multi-value attributes, negative attributes,

and wildcards (which means the attributes don't appear in the *AND*-gates, therefore they are not considered in decryption algorithm). Their scheme satisfies the properties of *PRE*, such as *unidirectionality*, *non-interactive*, and *multi-use*. Moreover, their scheme provides two improvements: (i) *re-encryption control* enabling the encryptor to decide whether the ciphertext can be re-encrypted or not, and (ii) *extra access control* that lets the proxy to add its own access policy to the ciphertext during re-encryption process. Furthermore, their scheme can be modified to have constant ciphertext size in original encryption and constant number of pairing computations in original decryption process, by removing wildcards in the access policy.

Yu *et al.* proposed two different policy types of *ABPRE* scheme [53, 54]. The scheme presented in [53] addressed the user's attribute revocation issue by integrating the technique of proxy re-encryption with the *CP-ABE* scheme proposed by Cheung *et al.* [30]. They considered practical application scenarios of data sharing, in which semi-trusted servers are always available for providing various types of content services. In their scheme, the authority delegates most laborious tasks of user's attribute revocation to proxy servers without leaking any confidential information to them. Whenever an attribute revocation event occurs, the authority redefines the master secret key components for involved attributes. Corresponding public key components are updated accordingly. Then, data will be encrypted with the new public key and user private keys should be updated accordingly for data access. For this purpose, the authority generates several proxy re-encryption keys for updating master secret key components and transmits them to the proxy servers. By using these proxy re-encryption keys, the proxy servers are able to securely update user private keys to the latest version for all users except for the one to be revoked. This removes the involved attributes from that users attribute set since their corresponding private key components no longer comply with the new master secret key. The proxy re-encryption keys also allow the proxy servers to re-encrypt existing ciphertexts stored on them to the latest version without disclosing any plaintext information. This construction enables the authority to freely revoke any user's attribute at any time while placing a minimal load on it. Their proposed scheme is provably secure against chosen ciphertext attacks under the *DBDH* assumption.

The scheme in [54] combined techniques of *KP-ABE* [11], proxy re-encryption [43], and lazy re-encryption [55] in order to guarantee scalability, data confidentiality, and fine-grained access control in cloud computing. To achieve fine-grained access control, they used *KP-ABE* scheme. However, this construction, if deployed alone, would introduce heavy computation

overhead and cumbersome online burden towards the data owner, as he is in charge of all the operations of data/user management. Specifically, such an issue is mainly caused by the operation of user revocation, which inevitably requires the data owner to re-encrypt all the data files accessible to the leaving user, or even needs the data owner to stay on-line to update private keys for users. To resolve this challenging issue and make the construction suitable for cloud computing, they combined *PRE* with *KP-ABE* and enable the data owner to delegate most of the computation-intensive operations to cloud servers without disclosing the underlying file contents. Such a construction allows the data owner to control access to his data with a minimal overhead in terms of computation effort and on-line time, and thus fits well into the cloud environment. Data confidentiality is also achieved since cloud servers are not able to learn the plaintext of any data in their construction. For further reducing the computation overhead on cloud servers and thus saving the data owner's investment, they took advantage of the lazy re-encryption technique and allowed cloud servers to aggregate computation tasks of multiple system operations. The computation complexity on cloud servers is either proportional to the number of system attributes, or linear to the size of the user access structure, which is independent to the number of users in the system. Thus, scalability is achieved. Although their proposed scheme is secure under standard cryptographic models, it is not secure against collusion attack of a revoked user in the system and cloud server.

Do *et al.* [56] proposed a key-policy *ABPRE* scheme to prevent the collusion attack of the Yu *et al.*'s scheme [54]. They created a trust authority called the privilege manager group. Then, they divided the data stored on the cloud servers into a header and a body and stored to the trust authority and the cloud server, respectively. Moreover, they introduced a data access privilege management model using type-based proxy re-encryption scheme for mobile cloud environments. According to the level that the data owner trusts the user group, the data is divided into the whole part or parts, and type information is granted to each and is encrypted. Through this, the data owner can selectively delegate decryption right for decrypting the whole or part of the data to the user group. Furthermore, they detailed a scenario involving privacy-preserving data sharing in health cloud environments and analyze security against collusion attack.

The computation cost of the previous *ABPRE* schemes is according to the number of attributes in the system, which implies huge computational overhead. Based on Emura *et al.*'s [36] *CP-ABE* scheme which has a constant ciphertext length, Seo *et al.* [57] presented a *CP-ABPRE* scheme with

a constant number of pairing operations, which reduced significantly the computational cost and ciphertext length compared to previous *ABPRE* schemes. They reduced the number of pairing operation by using an exponential operation which can easily calculate the summation of the exponent. Therefore, they calculated the exponent and then computed the pairing operation just once. Their scheme can be adapted to various applications including e-mail forwarding and distributed file systems.

Li [58] presented a new *CP-ABPRE* scheme, which supports *LSSS* matrix access structure on multi-value attributes, negative attributes, and wildcards. Their scheme satisfies some properties of *PRE*, such as *unidirectionality*, *non-interactive*, and *multi-use*. Moreover, this scheme supports other three properties: (i) *re-encryption control* allows the encryptor to decide whether the ciphertext can be re-encrypted or not, (ii) *extra access control* allows the proxy to add extra access policy to the ciphertext during re-encryption process, and (iii) *secret key security* provides a guarantee for the delegator, even if the proxy and all delegatees collude, they can not recover his master secret key. Furthermore, they described the security model called Selective-Policy Model for their *CP-ABPRE* scheme.

The aforementioned *CP-ABPRE* schemes are only secure against selective chosen-plaintext attacks (*CPA*). The *CPA* security might not be sufficient enough in an open network since it only achieves the very basic requirement from an encryption scheme, which only allows an encryption to be secure against "passive" adversaries. Nevertheless, in a real network scenario, there might exist "active" adversaries trying to tamper an encryption in transit and next observing its decryption such that to obtain useful information related to the underlying data. Accordingly, a *CP-ABPRE* system being secure against chosen-ciphertext attacks (*CCA*) is needed. Because *CCA* security not only helps the system to prevent the above subtle attacks but also enables the system to be further developed. Especially, when the *CP-ABPRE* is implemented within a large protocol/system, where a much wider array of attacks are possible. Moreover, the expressiveness of access policy is another critical factor for a practical *CP-ABPRE* system. Most of the previous *CP-ABPRE* schemes only support *AND*-gates access structure on (multi-valued) positive and negative attributes. This limits their practical use. Therefore, it is desirable to propose a *CP-ABPRE* system supporting more expressive and flexible access policy. To tackle these issues, Liang *et al.* [59] proposed the first secure *CP-ABPRE* scheme against selective chosen-ciphertext attacks (*CCA*) with supporting any monotonic access structures. They chose the Waters's *CP-ABE* scheme [37] as a basic building block of their scheme because the construction of the Waters's *ABE*

scheme is able to convert their scheme to be an *ABE* Key Encapsulation in the random oracle model. Moreover, the Waters's *ABE* scheme utilizes *LSSS* to support any monotonic access formula for ciphertexts. It is a desirable property for *CP-ABPRE* systems when being implemented in practice. Despite their scheme is constructed in the random oracle model, it can be proved *CCA* secure under the decisional q-parallel bilinear Diffie-Hellman exponent (q-parallel *BDHE*) assumption.

However, a *CP-ABPRE* system with selective security limits an adversary to choose an attack target before playing security game. This might not scale in practice because a realistic adversary can adaptively choose his attack target upon attacking a cryptosystem. Therefore, an adaptively *CCA* secure *CP-ABPRE* scheme is needed in most of the practical network applications. Thus, Liang *et al.* [60] formalized the notion of adaptive *CCA* security for *CP-ABPRE* systems and proposed the first adaptively *CCA*-secure *CP-ABPRE* scheme by integrating the dual system encryption technology with selective proof technique. Compared to the selective *CPA* security notion, their new notion enables an adversary to commit to a target access policy in the challenge phase and gains access to re-encryption and decryption oracles additionally. Their scheme supports any monotonic access structure such that users are allowed to fulfill more flexible delegation of decryption rights. Although their scheme is built in the composite order bilinear group, it is proven adaptively *CCA* secure in the standard model without jeopardizing the expressiveness of access policy. However, their scheme demands a number of paring operations that implies huge computational overheads.

Li *et al.* [61] proposed an efficient and adaptively secure *CP-ABPRE* scheme basing on Waters' dual system encryption technology [62]. This scheme is constructed in composite order bilinear groups and supports any monotone access structure. They proved that their scheme was secure under the complexity assumptions of the subgroup decision problem for 3 primes (*3P-SDP*). Compared with the existing schemes, their scheme requires a constant number of paring operations in Re-encryption and Decryption phases, which reduces the computational overhead.

3.2.4.1 Formal Definition of ABPRE Scheme

An *ABPRE* scheme consists of six fundamental algorithms Setup, KeyGen, Encrypt, Re-KeyGen, Re-Encrypt, and Decrypt with the following functions:

- $(PK, MSK) \leftarrow$ **Setup** (sp). This algorithm is run by the trust authority (TA), which takes as input a security parameter sp and outputs a

public key PK and a master secret key MSK.

- $SK \leftarrow$ **KeyGen** $(M\text{SK}, L_1)$. This algorithm is run by the TA. It takes as input the master secret key MSK and the set of attributes L_1. It outputs a private key SK associated with the set of attributes L_1.

- $CT \leftarrow$ **Encrypt** (PK, M, W_1). This algorithm is run by the data owner who takes as input the public key PK, a message, and a set of attributes W_1 to output a ciphertext CT associated with the set of attributes W_1.

- $RK_{L_1 \to L_2} \leftarrow$ **Re-KeyGen** (SK, L_2). This algorithm is run by the TA. It takes as input the private key SK and a set of attributes L_2. It outputs a re-encryption key $RK_{L_1 \to L_2}$ that can be used to transform a ciphertext that could be decrypted by SK into a ciphertext encrypted with L_2.

- $CT^{'} \leftarrow$ **Re-Encrypt** $(RK_{L_1 \to L_2}, CT)$. This algorithm is run by the proxy server. It takes as input a re-encryption key $RK_{L_1 \to L_2}$ and a ciphertext CT to output the transformed ciphertext $CT^{'}$.

- $M \leftarrow$ **Decrypt** (SK, CT). This algorithm is run by the user. It takes as input the ciphertext CT and the private key SK and returns the message M if a set of attributes in the user's private key matches with the attributes of the ciphertext; otherwise, it returns the distinguished symbol $\perp$.

3.2.4.2 Comparison between ABPRE schemes

In the section 3.2.4, we surveyed two various access policy attribute-based proxy re-encryption schemes: key policy and ciphertext policy and analyzed these schemes. In this section, we list the comparisons of them by some criteria.

The comparison results of different $ABPRE$ schemes are summarized in Tables 3.7- 3.10. The comparison of the keys size and ciphertext and the computation overhead of Encrypt, Decrypt, Re-Encrypt, and Re-Decrypt algorithms of different $ABPRE$ schemes are shown in Table 3.7 and Table 3.8, respectively. In addition, the comparison of complexity assumptions, security models, and supported access structures of different $ABPRE$ schemes are shown in Table 3.9 and Table 3.10, respectively.

From Tables 3.7-3.10, we can draw the following conclusions. As shown in Table 3.7, the size of public/private key and ciphertext increases linearly

Scheme	Public Key	Private Key	Ciphertext
Liang *et al.* [51]	$(6n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n+1)\|\mathbb{G}\|$	$(2n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Luo *et al.* [52]	$(N'+2n+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4n+1)\|\mathbb{G}\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Seo *et al.* [57]	$(3n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|+3n\|\mathbb{Z}_p^*\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{Z}_p^*\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Li [58]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(A_U+2)\|\mathbb{G}\|$	$(2A_C+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Liang *et al.* [59]	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|+6Hash$	$(A_U+2)\|\mathbb{G}\|$	$(2A_C+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Liang *et al.* [60]	$(n+6)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(A_U+3)\|\mathbb{G}\|$	$(2A_C+5)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Li *et al.* [61]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(A_U+2)\|\mathbb{G}\|$	$(2A_C+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$

$|\mathbb{G}|$: bit length of element in $\mathbb{G}$, $|\mathbb{G}_T|$: bit length of element in $\mathbb{G}_T$, n: the number of attributes in the system, $N' = \Sigma_{i=1}^{n} n_i$: the total number of possible values for attributes, where n_i is the number of possible values for attribute i, $|\mathbb{Z}_p^*|$: bit length of element in finite field $\mathbb{Z}_p^*$, A_C ($|A_C| = A_C$): a set of attributes associated with the ciphertext, A_U ($|A_U| = A_U$): a set of attributes associated with the private key.

Table 3.7: Comparison of the size of keys and ciphertext of ABPRE schemes

with the number of attributes. On the other hand, the computation overhead of the schemes in [51] and [52] increases linearly with the number of attributes, as shown in Table 3.8.

Also in these schemes, the number of pairing operations increases with the number of attributes, resulting in high computation overhead for Decrypt, Re-Encrypt, and Re-Decrypt algorithms. Promising results have been achieved by Seo *et al.* [57] whose scheme requires a constant number of pairing operations for Decrypt, Re-Encrypt, and Re-Decrypt algorithms and, therefore reduces the computation cost compared to [51] and [52]. Li *et al.*'s scheme [61] also requires a constant number of pairing operations for Decrypt, Re-Encrypt, and Re-Decrypt algorithms. Their scheme greatly reduces the computation overhead compared to Seo *et al.*'s scheme [57].

From Tables 3.9 and 3.10, we can see that Liang *et al.* [51], Luo *et al.* [52], and Seo *et al.* [57] proposed their schemes based on the *CP-ABE* in which the ciphertext is associated with *AND*-gates access structure. However, the access policy in these schemes is not flexible enough because it only supports *AND* operation on attributes. However, the ciphertext policy in Li [58], Liang *et al.* [59], Liang *et al.* [60], and Li *et al.* [61] schemes is *LSSS* matrix access structure which supports any monotonic access formula. Different from Li [58] and Liang *et al.* [59] schemes, the schemes of Liang *et al.* [60] and Li *et al.* [61] are adaptively secure.

Scheme	**Encrypt**	**Decrypt**	**Re-Encrypt**	**Re-Decrypt**
Liang *et al.* [51]	$(n+2)\mathbb{G}+2\mathbb{G}_T$	$(n+2)P+2\mathbb{G}_T$	$(n+1)P+\mathbb{G}_T$	$(n+3)P+4\mathbb{G}_T$
Luo *et al.* [52]	$(n+2)\mathbb{G}+2\mathbb{G}_T$	$(2n)P+3\mathbb{G}_T$	$(2n+1)P+(n+1)\mathbb{G}_T$	$(2n+1)P+5\mathbb{G}_T$
Seo *et al.* [57]	$(n+2)\mathbb{G}+2\mathbb{G}_T$	$(3n+2)\mathbb{G}+2P+2\mathbb{G}_T$	$(3n)\mathbb{G}+2P+\mathbb{G}_T$	$(3n)\mathbb{G}+3P+4\mathbb{G}_T$
Li [58]	$(4A_C+2)\mathbb{G}+2\mathbb{G}_T$	$(2A_U+1)P+3\mathbb{G}_T$	$(4A_U)\mathbb{G}+(2A_U+1)P+(3A_U)\mathbb{G}_T$	$(2A_U+1)P+(3A_U+2)\mathbb{G}_T$
Liang *et al.* [59]	$(4A_C+2)\mathbb{G}+\mathbb{G}_T$	$(2A_U+1)P+(3A_U)\mathbb{G}_T$	$(2A_U+2)P+(3A_U+1)\mathbb{G}_T$	$(2A_U+2)P+(3A_U)\mathbb{G}_T$
Liang *et al.* [60]	$(4A_C+4)\mathbb{G}+2\mathbb{G}_T$	$(2A_U+1)P+(2A_U+1)\mathbb{G}_T$	$(2A_U+2)P+(2A_U+2)\mathbb{G}_T$	$(2A_U+3)P+(2A_U+4)\mathbb{G}_T$
Li *et al.* [61]	$(4A_C+2)\mathbb{G}+2\mathbb{G}_T$	$(4A_U-1)\mathbb{G}+2P+2\mathbb{G}_T$	$(4A_U-1)\mathbb{G}+2P+2\mathbb{G}_T$	$(4A_U-1)\mathbb{G}+3P+4\mathbb{G}_T$

$\mathbb{G}$: operation in group $\mathbb{G}$, $\mathbb{G}_T$: operation in group $\mathbb{G}_T$, P: paring computation, n: the number of attributes in the system, A_C ($|A_C|= A_C$): a set of attributes associated with the ciphertext, A_U ($|A_U|= A_U$): a set of attributes associated with the private key, $N' = \Sigma_{i=1}^{n} n_i$: the total number of possible values for attributes, where n_i is the number of possible values for attribute i.

Table 3.8: Comparison of computation overhead of ABPRE schemes

From the above analysis, we can conclude that Li *et al.*'s scheme [61] is more efficient and secure than previous *ABPRE* schemes.

Scheme	Policy	Assumption	Model
Liang *et al.* [51]	Ciphertext	ADBDH & CTDH	Selective
Luo *et al.* [52]	Ciphertext	DBDH	Selective
Yu *et al.* [53]	Ciphertext	DBDH	Selective
Yu *et al.* [54]	Key	-	Selective
Do *et al.* [56]	Key	-	Selective
Seo *et al.* [57]	Ciphertext	ADBDH & CTDH	Selective
Li [58]	Ciphertext	DPBDHE	Selective
Liang *et al.* [59]	Ciphertext	DPBDHE	Selective
Liang *et al.* [60]	Ciphertext	DPBDHE	Adaptive
Li *et al.* [61]	Ciphertext	3P-SDP	Adaptive

$ADBDH$: Augment Decisional Bilinear Diffie-Hellman, $CTDH$: Complex Triple Diffie-Hellman, $DBDH$: Decisional Bilinear Diffie-Hellman, $DPBDHE$: Decisional q-Parallel Bilinear Diffie-Hellman Exponent, $3P-SDP$: Subgroup Decision Problem for 3 Primes.

Table 3.9: Comparison of security proof and some properties of ABPRE schemes

Scheme	Access Structure
Liang *et al.* [51]	AND-gates on positive and negative attributes with wildcards
Luo *et al.* [52]	AND-gates on multi-valued and negative attributes with
Yu *et al.* [53]	AND gates on positive and negative attributes with wildcards
Seo *et al.* [57]	AND gates on positive and negative attributes with
Li [58]	LSSS matrix access structure which supports any monotonic access formula
Liang *et al.* [59]	LSSS matrix access structure which supports any monotonic access formula
Liang *et al.* [60]	LSSS matrix access structure which supports any monotonic access formula
Li *et al.* [61]	LSSS matrix access structure which supports any monotonic access formula

$LSSS$: Linear Secret-Sharing Scheme

Table 3.10: Comparison of access structure of ABPRE schemes

Chapter 4

Secure Data Sharing with *ABE* in Cloud Computing

Cloud computing is a cost-efficient paradigm providing seemingly unlimited data storage and computing resources application scenarios in which vast amounts of data are collected and have to be properly accessed and processed. Cloud data sharing has found many practical applications in healthcare where patients are willing to share their personal health records among a group of care practitioners to improve availability and coverage of remote assistance [63]. Due to the high sensitivity of medical records, outsourcing raises strong confidentiality and privacy concerns since this information is shared among different entities and can be extracted by intentional attackers such as legitimate insurance, financing or governmental organizations.

A straightforward solution is to encrypt data (i.e., *payload-hiding*) before outsourcing it to the cloud to ensure confidentiality, but this manner causes much computational overhead. However, existing cryptographic techniques do not support access control models where different users have different access rights to the shared data. To achieve the encryption-based support of access control policies [64], an attractive solution is to adopt attribute-based encryption (*ABE*) [8], which enables fine-grained access control over encrypted data using access policies and associates attributes with private keys and ciphertexts. In *ABE* scheme, a data owner (e.g., a patient) can share his data with users (e.g., doctors, nurses, care practitioners, etc.,) only if a set of attributes in the user's private key matches with the attributes of his ciphertext. Several research papers have been devoted to implementing fine-grained access control via *ABE* cryptosystems [8, 11]. However, access control mechanisms based on *ABE* raise two challenges: (i) they do not sufficiently protect the attributes associated with the ciphertexts, potentially

jeopardizing the data owner's privacy, and (ii) do not support updating attribute sets without decrypting the encrypted data, making policy updates extremely inefficient.

To address the first challenge, a variety of solutions have been proposed [65, 1] adopting the inner-product encryption (*IPE*) scheme [10], where a ciphertext corresponding to an attribute vector $\vec{x}$ can be decrypted by any key $SK_{\vec{v}}$ such that $< \vec{x}, \vec{v} >= 0$. Inner-product encryption schemes generally offer the *attribute-hiding* property, meaning that it is not possible to determine the vector with which the ciphertext is encrypted. However, current *IPE* schemes do not support efficient access policy changes. To address the second challenge, proxy re-encryption (*PRE*) scheme offers a good solution by allowing a semi-trusted proxy to transform a ciphertext into a new ciphertext without leaking any information about the encrypted data and user's private key, as discussed in Section 3.2.

To address the above-mentioned issues, in this chapter we enhance *ABE* scheme and combine it with *PRE* scheme. We proposed two proxy-based inner-product schemes: (i) a novel inner-product proxy re-encryption (*IP-PRE*) scheme, and (ii) an efficient inner-product proxy re-encryption (*E-IPPRE*) scheme. Both schemes guarantee secure data sharing over untrusted cloud service provider.

The remainder of this chapter is organized as follows. In Section 4.1, we first introduce the predicate encryption inner-product scheme. Section 4.2 presents an overview of our system. Then, we present two new *IPPRE* schemes in Sections 4.3 and 4.4. Finally, we present the summary of this Chapter in Section 4.5.

4.1 Predicate Encryption Inner-product

Attribute-based encryption (*ABE*) schemes have desirable functionality (see Section 3.1.1), but a common limitation of these schemes is that the ciphertexts do not conceal their corresponding attributes set and therefore they do not guarantee *attribute-hiding* property. For example, in a *CP-ABE* scheme, a user who cannot decrypt can still learn the access policy associated with the ciphertext. For applications where the access policy must also be kept secret, this is undesirable.

Katz *et al.* [10] introduced the notion of predicate encryption (*PE*) as a generalized (fine-grained) notion of public key encryption that allows one to encrypt a message as well as attributes. In predicate encryption scheme, a ciphertext associated with attribute set $I \in \Sigma$ can be decrypted by a private key SK_f corresponding to the predicate $f \in \mathcal{F}$ if and only if $f(I) = True$.

Security notion of predicate encryption is considered in two aspects: *payload-hiding* and *attribute-hiding*. Generally, *attribute-hiding* requires that a ciphertext conceal the associated attributes as well as the plaintext so that no information about attributes is revealed during the decryption process, while *payload-hiding* only requires that a ciphertext conceal the plaintext.

Katz *et al*. [10] also presented a special type of predicate encryption for a class of predicates called *inner-product encryption* (*IPE*), which is obtained by having each attribute correspond to a vector $\vec{x}$ and each predicate $f_{\vec{v}}$ correspond to a vector $\vec{v}$, where $f_{\vec{v}}(\vec{x}) = 1$ iff $< \vec{x}, \vec{v} >= 0$ (here, $< \vec{x}, \vec{v} >$ denotes the standard inner-product). The inner-product predicates represent a wide class of predicates that includes an equality test, disjunctions or conjunctions of equality tests, and, more generally, arbitrary *CNF* or *DNF* formulas (*ABE* schemes) and polynomial evaluations. Therefore, realizing a predicate encryption supporting inner-product is the main goal of predicate encryption, because such schemes can achieve high flexibility in terms of access control.

Following Katz *et al*., a hierarchical *IPE* scheme was proposed by Okamoto *et al*. [65], which used n-dimensional vector spaces in prime order bilinear groups and achieves full security under the standard model. In [39], Lewko *et al*. showed a fully secure *IPE* scheme based on composite bilinear groups resulting low practical efficiency. Although these *IPE* constructions achieve *attribute-hiding* property, the security of their schemes is not under well-known standard assumptions. A different work of Park [1] presented an efficient *IPE* scheme supporting the *attribute-hiding* property. His scheme is based on prime order bilinear groups and secure against the well-known Decision Bilinear Diffie-Hellman (*BDH*) and Decision Linear assumptions.

Chen *et al*. [66] improved the scheme of Katz *et al*. [10] to implement a fully secure *ABE* scheme with constant size ciphertexts. Their scheme supports threshold access policies with constant operations independent of the number of attributes. Chen *et al*.'s approach was improved by Backes *et al*. [67] who proposed an efficient inner-product proxy re-encryption scheme with constant size ciphertext adopting Green *et al*.'s technique [68]. Their scheme requires a constant number of pairing operations encryption/decryption and the length of ciphertext is also independent of the length of the attributes' vector. Compared to the original Liang *et al*. [51] scheme, Backes *et al*.'s method is more suitable cloud storage systems because the private key is no longer needed generating a re-encryption key. However, like the method of Chen *et al*. [66], Backes *et al*.'s scheme does not hide the attributes used in encryption procedure.

Kim *et al*. [2] proposed an efficient *IPE* scheme to compute inner-product

operations using exponentiation instead of pairing computations; therefore, their scheme requires constant pairing operations decryption with shorter sizes of the public key, private key, and ciphertext as well as reduces the time needed key generation. Furthermore, Kim *et al.*'s scheme supports the evaluations of polynomials, disjunctions, conjunctions, and threshold.

4.1.1 Inner-Product Encryption

In this section, we formally define the syntax of inner-product encryption (*IPE*) and inner-product proxy re-encryption (*IPPRE*) schemes and their security properties. Our *IPE* definition follows the general framework of that given in [1]. Throughout this section, we consider the general case where Σ denotes an arbitrary set of attribute vectors and $\mathcal{F}$ denotes an arbitrary set of predicates involving inner-products over Σ.

Definition 4.1. An inner-product predicate encryption scheme (*IPE*) for the class of predicates $\mathcal{F}$ over the set of attributes Σ consists of PPT algorithms Setup, KeyGen, Encrypt, and Decrypt such that:

- $\mathsf{Setup}_{\mathsf{IPE}}$: takes as input a security parameter λ and a positive dimension n of vectors. It outputs a public key PK and a master secret key MSK.

- $\mathsf{KeyGen}_{\mathsf{IPE}}$: takes as input a public key PK, a master secret key MSK, and a predicate vector $\vec{v} \in \mathcal{F}$. It outputs a private key $SK_{\vec{v}}$ associated with vector $\vec{v}$.

- $\mathsf{Encrypt}_{\mathsf{IPE}}$: takes as input a public key PK, an attribute vector $\vec{x}$, and a message $M \in \mathcal{M}$. It outputs a corresponding ciphertext $CT_{\vec{x}}$.

- $\mathsf{Decrypt}_{\mathsf{IPE}}$: takes as input a private key $SK_{\vec{v}}$ and the ciphertext $CT_{\vec{x}}$. It outputs either a massage M if $f_{\vec{v}}(\vec{x}) = 1$, i.e., $< \vec{x}, \vec{v} >= 0$, or the distinguished symbol $\perp$ if $f_{\vec{v}}(\vec{x}) = 0$.

Definition 4.2. An inner-product predicate encryption scheme for predicate $\mathcal{F}$ over attributes Σ is attribute-hiding secure against adversary $\mathcal{A}$ under chosen-plaintext attacks is given as follows:

Setup. The challenger runs the Setup algorithm and it gives the public key PK to the adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ is allowed to adaptively issue a polynomial

number of key queries. For a private key query $\vec{v}$, the challenger gives $SK_{\vec{v}}$ to $\mathcal{A}$.

Challenge. For a challenge query $(X_0, X_1, \vec{x}_0, \vec{x}_1)$, subject to the following restriction:

1. $< \vec{v}, \vec{x_0} >=< \vec{v}, \vec{x_1} >\neq 0$ for all private key queries $\vec{v}$, or
2. two challenge messages are equal, i.e., $X_0 = X_1$, and any private key query $\vec{v}$ satisfies $< \vec{v}, \vec{x_0} >=< \vec{v}, \vec{x_1} >$.

The challenger flips a random $b \in \{0, 1\}$ and computes the corresponding ciphertext as $CT_{\vec{x}_b} \leftarrow \mathsf{Encrypt}\,(PK, \vec{x}_b, X_b)$. It then gives $CT_{\vec{x}_b}$ to the adversary.

Phase 2. The adversary $\mathcal{A}$ is allowed to adaptively issues polynomial number of key queries. For a private key query $\vec{v}$, subject to the aforementioned restrictions.

Finally, $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$ for b and wins the game if $b = b'$. An advantage $\mathcal{A}$ in attacking *IPE* is defined as $Adv_{\mathcal{A}}^{\mathsf{IPE\text{-}AH}}(\lambda) = Pr[b = b'] - \frac{1}{2}$. Therefore, an *IPE* scheme is *attribute-hiding* if all polynomial-time adversaries have at most negligible advantage in the above game. If the restriction 1 in challenge is allowed for $\mathcal{A}$, an *IPE* scheme is *payload-hiding* if all polynomial-time adversaries have at most negligible advantage in the game.

Definition 4.3. An inner-product proxy encryption (*IPPRE*) scheme creates a re-encryption key *ReKey* that gives the possibility of transforming a ciphertext associated with a vector $\vec{x}$ into a new ciphertext encrypting the same plaintext but associated with a different vector $\vec{w}$, while maintaining the confidentiality of the underlying plaintext. *IPPRE* scheme for the class of predicates $\mathcal{F}$ over $n-$dimensional vectors Σ for message space $\mathcal{M}$, consists of seven *PPT* algorithms Setup, Encrypt, KeyGen, Re-KeyGen, Re-Encrypt, and Decrypt such that:

- $\mathsf{Setup}_{\mathsf{IPPRE}}$: takes as input a security parameter λ and a dimension n of vectors. It outputs a public key PK and a master secret key MSK.
- $\mathsf{Encrypt}_{\mathsf{IPPRE}}$: takes as input the public key PK, a vector $\vec{x} \in \Sigma$ of attributes and a message $M \in \mathcal{M}$ to output a ciphertext $CT_{\vec{x}}$.
- $\mathsf{KeyGen}_{\mathsf{IPPRE}}$: takes as input the master secret key MSK, the public

key PK, and a predicate vector $\vec{v} \in \mathcal{F}$. It outputs a private key $SK_{\vec{v}}$ associated with vector $\vec{v}$.

- $\mathsf{Re\text{-}KeyGen}_{\mathsf{IPPRE}}$: takes as input the master secret key MSK and two vectors $\vec{v}$ and $\vec{w}$. It outputs a re-encryption key $RK_{\vec{v},\vec{w}}$ that transforms a ciphertext that could be decrypted by $SK_{\vec{v}}$ into a ciphertext encrypted with vector $\vec{w}$.

- $\mathsf{Re\text{-}Encrypt}_{\mathsf{IPPRE}}$: takes as input a re-encryption key $RK_{\vec{v},\vec{w}}$ and a ciphertext $CT_{\vec{x}}$ to output a re-encrypted ciphertext $CT'_{\vec{x}}$.

- $\mathsf{Decrypt}_{\mathsf{IPPRE}}$: takes as input the ciphertext $CT_{\vec{x}}$ and the private key $SK_{\vec{v}}$. It outputs either a message M if $f_{\vec{v}}(\vec{x}) = 1$, i.e., $< \vec{x}, \vec{v} >= 0$, or the distinguished symbol $\perp$ if $f_{\vec{v}}(\vec{x}) = 0$.

From here on, we use the terms *Level-1 (L1)* and *Level-2 (L2)* to denote ciphertexts obtained as the output of $\mathsf{Encrypt}$ and $\mathsf{Re\text{-}Encrypt}$ algorithms, respectively.

Correctness. The correctness property requires to decrypt the ciphertext by the appropriate private key. More precisely, for the two levels *L1* and *L2* we have:

L1: $\mathsf{Decrypt}\,(\mathsf{KeyGen}\,(MSK, PK, \vec{v}), \mathsf{Encrypt}\,(PK, \vec{x}, M)) = M$;

L2: $\mathsf{Decrypt}\,(\mathsf{KeyGen}\,(MSK, PK, \vec{v'}), \mathsf{Re\text{-}Encrypt}\,(\mathsf{Re\text{-}KeyGen}\,(\mathsf{KeyGen}\,(MSK, PK, \vec{v}), \vec{w}), CT_{\vec{x}})) = M$,

where $\vec{x}$ satisfies $\vec{v}$, $\vec{w}$ satisfies $\vec{v'}$, MSK is a master secret key, PK is a public key, $CT_{\vec{x}}$ is a ciphertext related to message M and an attribute vector $\vec{x}$.

4.1.2 Security Model IPPRE

The *IPPRE* protocol should satisfy the following security requirements in adaptive security game:

- **Attribute-hiding Security of Original Ciphertext:** an original ciphertext (*Level-1*) of a message M with vector $\vec{x}$ releases no information regarding $(M, \vec{x})$ against a user not in possession of matching private key $SK_{\vec{v}}$ such that $< \vec{x}, \vec{v} >= 0$ or having a matching key pair of a re-encryption key and a private key $(RK_{\vec{v},\vec{w}}, SK_{\vec{v'}})$ with $< \vec{x}, \vec{v} >= 0$ and $< \vec{v'}, \vec{w} >= 0$. It also releases no information regarding $\vec{x}$ against a

user in possession of matching private key $SK_{\vec{v}}$ except that $< \vec{x}, \vec{v} >= 0$ or a matching key pair $(RK_{\vec{v},\vec{w}}, SK_{\vec{v'}})$ except that $< \vec{x}, \vec{v} >= 0$ and $< \vec{v'}, \vec{w} >= 0$.

- **Attribute-hiding Security of Re-encrypted Ciphertext:** a re-encrypted ciphertext (*Level-2*) of a message M, original vector $\vec{x}$, and re-encryption key $RK_{\vec{v},\vec{w}}$ with vector $\vec{w}$ release no information regarding $(M, \vec{x}, \vec{v}, \vec{w})$ against a user not in possession of a matching private key $SK_{\vec{v'}}$ for $\vec{w}$, and no information regarding $\vec{w}$ against a user in possession of a matching private key $SK_{\vec{v'}}$ except that $< \vec{v'}, \vec{w} >= 0$.

Definition 4.4 (Attribute-hiding for Level-1 Ciphertexts (AH-L1)). An inner-product proxy re-encryption scheme, predicate $\mathcal{F}$ over vectors Σ is attribute-hiding secure *Level-1* against adversary $\mathcal{A}$ under chosen-plaintext attacks (*CPA*) if for all probabilistic polynomial-time *PPT*, the advantage of $\mathcal{A}$ in the following security game Γ is negligible in the security parameter.

Setup. The challenger $\mathcal{B}$ runs $\mathsf{Setup}(\lambda, n)$ algorithm and gives the public key PK to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ adaptively makes a polynomial number of queries as:

(a) **Private key query:** for a private key query $\vec{v}$, the challenger gives $SK_{\vec{v}} \xleftarrow{R} \mathsf{KeyGen}\ (MSK, PK, \vec{v})$ to $\mathcal{A}$, where R indicates that $SK_{\vec{v}}$ is randomly selected from KeyGen according to its distribution.

(b) **Re-encryption key query:** for a re-encryption key query with $(\vec{v}, \vec{w})$, the challenger computes $RK_{\vec{v},\vec{w}} \xleftarrow{R} \mathsf{Re\text{-}KeyGen}\ (MSK, \vec{v}, \vec{w})$ where $SK_{\vec{v}} \xleftarrow{R} \mathsf{KeyGen}\ (MSK, PK, \vec{v})$ and gives the re-encryption key to the adversary.

(c) **Re-encryption query:** for a re-encryption query $(\vec{v}, \vec{w}, CT_{\vec{x}})$, $\mathcal{B}$ computes the re-encryption key $RK_{\vec{v},\vec{w}} \xleftarrow{R} \mathsf{Re\text{-}KeyGen}\ (MSK, \vec{v}, \vec{w})$, where $SK_{\vec{v}} \xleftarrow{R} \mathsf{KeyGen}(MSK, PK, \vec{v})$ and $CT'_{\vec{x}} \xleftarrow{R} \mathsf{Re\text{-}Encrypt}(PK, RK_{\vec{v},\vec{w}}, CT_{\vec{x}}, \vec{w})$.

Challenge. For a challenge query $(\vec{x}_0, \vec{x}_1, M_0, M_1)$ under the condition that: - Any private key query $\vec{v}$ and re-encryption key query $(\vec{v}_l, \vec{w}_l)$, for $l = 1, \ldots, p_1$ where p_1 is the maximum number of private key queries requested by the adversary, $M_0 = M_1$ if $< \vec{v}, \vec{x}_0 >=< \vec{v}, \vec{x}_1 >= 0$ and $< \vec{v}_l, \vec{x}_0) >= < \vec{v}_l, \vec{x}_1) >= 0$ in the case that $< \vec{v}, \vec{w}_l >= 0$.

The challenger $\mathcal{B}$ samples a random bit $b \xleftarrow{U} \{0,1\}$, where U indicates that b is uniformly selected from $\{0,1\}$ and gives $CT_{\vec{x}_b} \xleftarrow{R} \mathsf{Encrypt}\ (PK, \vec{x}_b, M_b)$ to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ may continue to request private key queries, re-encryption key queries, and re-encryption queries subject to the same restrictions as before and the condition for the re-encryption queries.

> **Re-encryption Query:** for a re-encryption query of the form $(\vec{v}_t, \vec{w}_t, CT_t)$, for $t = 1, \ldots, p_2$ where p_2 is the maximum number of re-encrypted queries, under the condition that $M_0 = M_1$ if $< \vec{v}_t, \vec{x}_0 >=< \vec{v}_t, \vec{x}_1 >= 0$ and $< \vec{v'}, \vec{w}_t >= 0$ for any decryption key query for $\vec{v'}$ if $CT_t = CT_{\vec{x}_b}$.
>
> The challenger computes $RK_{\vec{v}_t,\vec{w}_t} \xleftarrow{R} \mathsf{Re\text{-}KeyGen}\ (MSK, \vec{v}_t, \vec{w}_t)$ and $CT'_{\vec{w}_t} \xleftarrow{R} \mathsf{Re\text{-}Encrypt}\ (PK, RK_{\vec{v}_t,\vec{w}_t}, CT_t)$, and it gives $CT'_{\vec{w}_t}$ to the adversary.

Guess. $\mathcal{A}$ outputs a bit b' and succeeds if $b' = b$.

Hence, we define the advantage $\mathcal{A}$ as $Adv_{\mathcal{A}}^{\mathsf{AH\text{-}L1}}(\lambda) := \mathsf{Pr}[b = b'] - \frac{1}{2}$. The *IPPRE* scheme is attribute-hiding *Level-1* ciphertext if all polynomial-time adversaries have at most negligible advantage in the above game.

Definition 4.5 (Attribute-hiding for Level-2 Re-encrypted Ciphertexts (AH-L2)). An inner-product proxy re-encryption scheme, predicate $\mathcal{F}$ over vectors Σ is attribute-hiding secure *Level-2* against adversary $\mathcal{A}$ under chosen-plaintext attacks (*CPA*) if for all probabilistic polynomial-time *PPT*, the advantage of $\mathcal{A}$ in the following security game Γ is negligible in the security parameter.

Setup, Phase 1. These algorithms are defined as the same as those we defined in Definition 4.4, respectively.

Challenge. Upon receiving the query $(\vec{x_0}, \vec{x_1}, M_0, M_1, \vec{v_0}, \vec{v_1}, \vec{w_0}, \vec{w_1})$ from the adversary with the restrictions that $(M_0, \vec{x_0}, \vec{v_0}) = (M_1, \vec{x_1}, \vec{v_1})$ if $< \vec{v'}, \vec{w_0} >=< \vec{v'}, \vec{w_1} >= 0$, for any private key query $\vec{v'}$, the challenger $\mathcal{B}$ samples a random bit $b \xleftarrow{R} \{0,1\}$ and gives:
$CT'_{\vec{w}_b} \xleftarrow{R} \mathsf{Re\text{-}Encrypt}(PK, \mathsf{Re\text{-}KeyGen}(PK, \mathsf{KeyGen}(PK, SK, \vec{v_b}), \vec{w_b}), \mathsf{Encrypt}($ $PK, \vec{x_b}, M_b))$. Then the challenger gives the result to the adversary.

Phase 2. The adversary $\mathcal{A}$ may continue to request private key queries,

re-encryption key queries, and re-encryption queries under the restrictions we mentioned in challenge phase.

Guess. $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ b and wins the game $b = b'$.

We define the advantage of $\mathcal{A}$ as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PAH\text{-}L2}}(\lambda) := \mathsf{Pr}[b = b'] - \frac{1}{2}$. Hence, the scheme is predicate- and attribute-hiding for re-encrypted ciphertexts if all polynomial-time adversaries have at most negligible advantage in the above game. To prove this statement for each run of the game, we define a variable $s_{M,\vec{x},\vec{v}} := 0$ if $(M_0, \vec{x_0}, \vec{v_0}) \neq (M_1, \vec{x_1}, \vec{v_1})$ for challenge $(M_l, \vec{x_l}, \vec{v_l})$ for $l = 0, 1$ and $s_{M,\vec{x},\vec{v}} := 1$, otherwise.

4.2 An Overview of our System

In this section, we first describe our problem statement and then we give a general overview of our system model that our protocols are built upon.

4.2.1 Problem Statement

For the sake of conciseness and clarity, we demonstrate a simplified version of secure data sharing scenario in the healthcare environment in which patient's data is collected through multiple medical sensing devices. Consider a patient (data owner) who is willing to store and share his medical records with a physician (user) via the cloud. Before data outsourcing, the (medical sensing device on behalf of a) patient encrypts its own data M under a set of associated attributes $I_{patient}$ (i.e. $Enc_{I_{patient}}(M_{patient})$), where $I_{patient}$ indicates access privilege on the patient's data. Then, the physician who satisfies $I_{patient}$ can access to the patient's data using its own private key related to an attribute set. Now suppose some cooperation is required during patient's treatment and the physician needs to share the patient's medical data with a group of care professionals in another hospital holding a different access policy $I'_{patient}$. Hence the problem is how a proxy is allowed to translate patient's data encrypted under access policy $I_{patient}$ to the one under access policy $I'_{patient}$ in an efficient way without revealing the patient's data and its corresponding attributes.

4.2.2 System Model

In this section, we present a streamlined version of secure and private data sharing system in a healthcare environment to show how to deploy an inner-

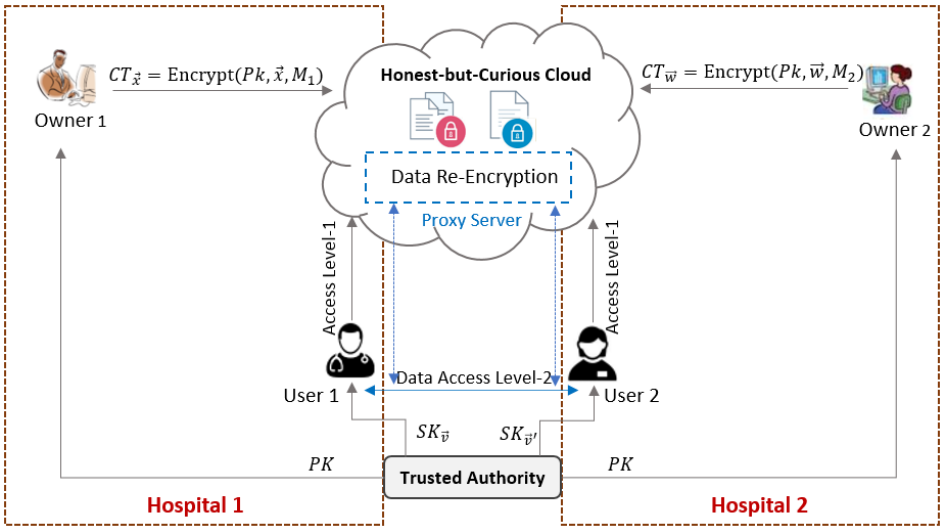


Figure 4.1: The proposed healthcare data sharing system model.

product proxy re-encryption scheme in such real-world scenarios. A inner-product proxy re-encryption-based data sharing healthcare system including five entities *Data Owner*, *Authorized Users Owner*, *Cloud Storage Server*, *Trust Authority*, and the *Proxy Server* works as follows:

Initialization. This step is run by a *Trust Authority (TA)* who is responsible for key issuing and attribute management. As shown in Figure 4.1, the authority first generates master secret key MSK and public key PK and then distributes PK and access policy A_i to each data owner i (e.g., Owner 1 from hospital 1 and Owner 2 from hospital 2). It also generates private keys for *Authorized Users* (User 1 (e.g., a group of care practitioners) and User 2 (e.g., specialist)).

Data Upload. This step is run at data owner side. Consider that the owner 1 from hospital 1 is willing to store and share its medical records via the *Cloud Storage Server* in such a way that only care practitioners from hospital 1 can have access. The owner 1 encrypts its own data (e.g., message M_1) under a set of associated attributes A_1 (e.g., $\mathsf{Encrypt}_{A_1}(M_1)$), where A_1 indicates access privilege on the owner 1's data. In a similar way, the owner 2 uploads its encrypted message (M_2).

Data Access. This step is run between the authorized users and the cloud server (*Level-1*) or between authorized users through the cloud using a proxy

server (*Level-2*).

1. *Level-1.* User 1 who satisfies A_1 can access to the owner 1's data using its own private key associated with a vector $\vec{v}$.

2. *Level-2.* There are some situations in which the user 1 needs to share the owner 1's medical data with the user 2 from hospital 2 who is able to decrypt only the ciphertexts associated with an access policy A_2 (attribute vector $\vec{w}$ in Figure 4.1), but not the access policy A_1 (attribute vector $\vec{x}$ in Figure 4.1). In this case, a *Proxy Server* is used to translate the data encrypted with access policy A_1 to the one under access policy A_2 in an efficient way without revealing the data (payload-hiding property) and its corresponding attributes (attribute-hiding property).

In our system model, we assume that the cloud and proxy server are honest-but-curious i.e., they will correctly execute the protocol, and will not deny services to the authorized users. But they are curious to learn information about data contents.

4.3 A Novel Inner-product Proxy Re-encryption Scheme

Here, we address the challenge of secure data sharing between users with different access policies in cloud computing by introducing a novel inner-product encryption (*IPE*) scheme adopting proxy re-encryption method. We modified the *IPE* scheme presented in [1] to construct a secure inner-product proxy re-encryption scheme which is called *IPPRE*. The proposed scheme delegates the re-encryption capability to a semi-trusted proxy who transforms a ciphertext under an access policy to the new ciphertext with the same plaintext but under another access policy. The proxy learns nothing about the underlying message, the associated attributes in the ciphertext, and the private key. The proposed *IPPRE* scheme satisfies attribute-hiding, fine-grained access control, supporting dynamic environment, collusion resistance, unidirectionality,non-interactivity, and multi-use properties.

4.3.1 The Main Construction

In this section, we construct our *IPPRE* scheme in detail. The scheme consists of six algorithms namely Setup, KeyGen, Encrypt, Re-KeyGen, Re-Encrypt, and Decrypt. We describe our construction with considering the

following assumptions:

Assumptions:

- some positive integer n, $\Sigma = (\mathbb{Z}_p^*)^n$ is the set of attributes,
- a vector $\vec{v} = (v_1, \cdots, v_n) \in \Sigma$, each component v_i belong to the set $\mathbb{Z}_p^*$, and
- a message $M \in \mathcal{M}$ and a vector $\vec{v}$, each $\vec{v}$ belongs to Σ and $\mathcal{M} = \mathbb{G}_T$.

4.3.1.1 Proposed IPPRE Scheme

$(PK, MSK) \leftarrow$ **Setup** (λ, n). On input a security parameter $\lambda \in \mathbb{Z}^+$ and the number of attributes n, Setup algorithm runs $init(\lambda)$ [1] to get the tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$. It then picks a random generator $g \in \mathbb{G}$, random exponents $\delta_1, \delta_2, \theta_1, \theta_2,\ \{w_{1,i}\}_{i=1}^n, \{t_{1,i}\}_{i=1}^n, \{f_{1,i}, f_{2,i}\}_{i=1}^n,\ \{h_{1,i}, h_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p^*$. It also picks a random $g_2 \in \mathbb{G}$ and a random $\Omega \in \mathbb{Z}_p^*$ to obtain $\{w_{2,i}\}_{i=1}^n, \{t_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p^*$ under constraints that:

$$\Omega = \delta_1 w_{2,i} - \delta_2 w_{1,i} \quad , \quad \Omega = \theta_1 t_{2,i} - \theta_2 t_{1,i}.$$

For $i = 1, \ldots, n$, the *setup* algorithm first computes:

$$\begin{array}{llll} W_{1,i} = g^{w_{1,i}}, & W_{2,i} = g^{w_{2,i}}, & T_{1,i} = g^{t_{1,i}}, & T_{2,i} = g^{t_{2,i}}, \\ F_{1,i} = g^{f_{1,i}}, & F_{2,i} = g^{f_{2,i}}, & H_{1,i} = g^{h_{1,i}}, & H_{2,i} = g^{h_{2,i}}, \end{array}$$

and then sets:

$$U_1 = g^{\delta_1}, \quad U_2 = g^{\delta_2}, \quad V_1 = g^{\theta_1}, \quad V_2 = g^{\theta_2}, \quad g_1 = g^{\Omega}, \quad \Lambda = e(g, g_2).$$

Finally, the Setup algorithm outputs the public key PK (including $(p, \mathbb{G}, \mathbb{G}_T, e)$) and master secret key MSK as:

$$PK = (g,\ g_1,\ \{W_{1,i}, W_{2,i}, F_{1,i}F_{2,i}\}_{i=1}^n,\ \{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\}_{i=1}^n, \{U_i, V_i\}_{i=1}^2,\ \Lambda) \in \mathbb{G}^{8n+6} \times \mathbb{G}_{\mathbb{T}},$$

$$MSK = (\{w_{1,i}, w_{2,i}, t_{1,i}, t_{2,i}, f_{1,i}, f_{2,i}, h_{1,i}, h_{2,i}\}_{i=1}^n,\ \{\delta_i, \theta_i\}_{i=1}^2,\ g_2) \in \mathbb{Z}_p^{8n+4} \times \mathbb{G}.$$

[1] *init* is an algorithm that takes as input a security parameter 1^n and outputs a tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order p and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map.

$SK_{\vec{v}} \leftarrow$ **KeyGen** $(MSK, PK, \vec{v})$. On input vector $\vec{v} = (v_1, \ldots, v_n) \in (\mathbb{Z}_p^*)^n$, public key PK and master secret key MSK, the algorithm randomly picks exponents $\lambda_1, \lambda_2, \{r_i\}_{i=1}^n, \{\phi_i\}_{i=1}^n$ in $\mathbb{Z}_p^*$ to output the private key as:

$$SK_{\vec{v}} = (K_A,\ K_B,\ \{K_{1,i}, K_{2,i}\}_{i=1}^n,\ \{K_{3,i}, K_{4,i}\}_{i=1}^n) \in \mathbb{G}^{4n+2}, \text{ where}$$

$$\begin{aligned} &\{K_{1,i} = g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}, \quad K_{2,i} = g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\}_{i=1}^n, \\ &\{K_{3,i} = g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}, \quad K_{4,i} = g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\}_{i=1}^n, \end{aligned}$$

$$K_A = g_2 \textstyle\prod_{i=1}^n K_{1,i}{}^{-f_{1,i}} K_{2,i}{}^{-f_{2,i}} K_{3,i}{}^{-h_{1,i}} K_{4,i}{}^{-h_{2,i}},\ K_B = \prod_{i=1}^n g^{-(r_i+\phi_i)}.$$

$CT_{\vec{x}} \leftarrow$ **Encrypt** $(PK, \vec{x}, M)$. On input vector $\vec{x} = (x_1, \ldots, x_n) \in (\mathbb{Z}_p^*)^n$, a message $M \in \mathbb{G}_T$ and the public key PK, the algorithm selects random exponents $s_1, s_2, s_3, s_4 \in \mathbb{Z}_p^*$ to get ciphertext $CT_{\vec{x}}$ as follows:

$$\begin{aligned} &CT_{\vec{x}} = (A,\ B,\ \{C_{1,i} = W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3}, C_{2,i} = W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\}_{i=1}^n, \\ &\{C_{3,i} = T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_i s_4}, C_{4,i} = T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_i s_4}\}_{i=1}^n,\ D) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}}, \end{aligned}$$

where we define each component of $CT_{\vec{x}}$ as follows, $1 \leq i \leq n$:

$$\begin{aligned} &A = g^{s_2}\ ,\ B = g_1^{s_1} = g^{s_1 \Omega},\ D = \Lambda^{-s_2} M, \\ &C_{1,i} = g^{w_{1,i} s_1} g^{f_{1,i} s_2} g^{\delta_1 x_i s_3}, \quad C_{2,i} = g^{w_{2,i} s_1} g^{f_{2,i} s_2} g^{\delta_2 x_i s_3} \\ &C_{3,i} = g^{t_{1,i} s_1} g^{h_{1,i} s_2} g^{\theta_1 x_i s_4}, \quad C_{4,i} = g^{t_{2,i} s_1} g^{h_{2,i} s_2} g^{\theta_2 x_i s_4}. \end{aligned}$$

In this step, the ciphertext is associated with the attribute vector $\vec{x}$ such that it reveals nothing about $\vec{x}$ to a computationally bounded adversary. It uses random elements $\{W_{1,i}^{s_1}\ ,\ W_{2,i}^{s_1}\ ,\ T_{1,i}^{s_1}\ ,\ T_{2,i}^{s_1}\}$ to mask each component x_i of a vector $\vec{x}$. For instance, the ciphertext $C_{1,i}$ is in the form $W_{1,i}^{s_1}\, F_{1,i}^{s_2}\, U_1^{x_i s_3}$, which is not easily tested even if we use prime order groups equipped with a symmetric bilinear map. If we omit $W_{1,i}^{s_1}$, the resulting term $F_{1,i}^{s_2}\, U_1^{x_i s_3}$ is enough for hiding x_i component, however, for the case that $x_i = 0$ in $\mathbb{Z}_p^*$, the term becomes $F_{1,i}^{s_2}$ that can be tested as $e(A, F_{1,i}) \stackrel{?}{=} e(g, C_{1,i})$ using bilinear maps.

$RK_{\vec{v},\vec{w}} \leftarrow$ **Re-KeyGen** $(MSK, \vec{v}, \vec{w})$. The algorithm first calls the KeyGen algorithm and picks a random $d \in \mathbb{Z}_p$ to compute g_2^d and $g_2^{d\delta_2}$, $g_2^{-d\delta_1}$, $g_2^{d\theta_2}$, $g_2^{-d\theta_1}$. It then calls the Encrypt algorithm to encrypt g_2^d under the vector $\vec{w}$ using Encrypt $(PK, \vec{w}, g_2^d)$ and outputs $CT_{\vec{w}}$.

To compute the re-encryption key, the Re-KeyGen algorithm picks random exponents $\lambda_1^{'}$, $\lambda_2^{'}$, $\{r_i^{'}\}_{i=1}^n$, $\{\phi_i^{'}\}_{i=1}^n$ in $\mathbb{Z}_P^*$ and computes $RK_{\vec{v},\vec{w}}$, $1 \leq i \leq n$ as:

$$K'_A = g_2 \prod_{i=1}^{n} K_{1,i}^{'\,-f_{1,i}} K_{2,i}^{'\,-f_{2,i}} K_{3,i}^{'\,-h_{1,i}} K_{4,i}^{'\,-h_{2,i}},\ K'_B = \prod_{i=1}^{n} g^{-(r'_i+\phi'_i)},$$

where we have:

$$K'_{1,i} = g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} g_2^{d\delta_2}, \quad K'_{2,i} = g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} g_2^{-d\delta_1},$$
$$K_{3,i}^{'} = g^{-\theta_2 \phi'_i} g^{\lambda'_2 v_i t_{2,i}} g_2^{d\theta_2}, \quad K'_{4,i} = g^{\theta_1 \phi'_i} g^{-\lambda'_2 v_i t_{1,i}} g_2^{-d\theta_1}.$$

The Re-KeyGen algorithm with the inputs vectors $\vec{v}, \vec{w}$ consists of two parts: a modified decryption key vector $\vec{v}$ and a ciphertext encrypted with vector $\vec{w}$. The modified decryption key differs from a normal decryption key: in the decryption procedure, a normal decryption key combines with elements of the ciphertext to recover the binding factor that is used for hiding the message (e.g., $e(g, g_2)^{-s_2}$); the modified decryption key instead produces the product of the blinding factor with another new binding factor. This new blinding factor can only be removed with the combination of a group element encrypted in the Re-KeyGen algorithm (e.g., g_2^d) and the element $B = g^{s_1 \Omega}$ in the ciphertext. Therefore, the *Level-2 access* of the Decrypt algorithm consists of the original blinded message, the product with the new blinding factor obtained by decrypting the original ciphertext with the modified decryption key in the Re-Encrypt algorithm, the element B from the original ciphertext and the ciphertext component of the Re-KeyGen algorithm.

$CT'_{\vec{x}} \leftarrow$ **Re-Encrypt** $(RK_{\vec{v},\vec{w}}, CT_x)$. On input a re-encryption key $RK_{\vec{v},\vec{w}}$ and CT_x, the algorithm outputs $CT'_{\vec{x}} = (A, B, CT_{\vec{x}}, \hat{CT}, D)$, where:

$$A = g^{s_2},\ B = g_1^{s_1} = g^{s_1 \Omega},\ D = \Lambda^{-s_2} M,\ , CT_{\vec{w}} = \mathsf{Encrypt}\,(PK, \vec{w}, g_2^d)$$

computing $\hat{CT}$, the algorithm checks if the attributes list in $RK_{\vec{v},\vec{w}}$ satisfies the attributes set of $CT_{\vec{x}}$, if not, returns $\perp$; otherwise, $1 \leq i \leq n$, it calculates the following pairings to output $\hat{CT}$:

$$\prod_{i=1}^{n} e(C_{1,i}, K_{1,i}^{'}) \cdot e(C_{2,i}, K_{2,i}^{'}) \cdot e(C_{3,i}, K_{3,i}^{'}) \cdot e(C_{4,i}, K_{4,i}^{'}),$$

where we have:

$$e(C_{1,i}, K_{1,i}^{'}) = e(g^{w_{1,i} s_1} g^{f_{1,i} s_2} g^{\delta_1 x_i s_3}, g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} g_2^{d\delta_2})$$
$$e(C_{2,i}, K_{2,i}^{'}) = e(g^{t_{2,i} s_1} g^{h_{2,i} s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1})$$
$$e(C_{3,i}, K_{3,i}^{'}) = e(g^{t_{2,i} s_1} g^{h_{2,i} s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1})$$
$$e(C_{4,i}, K_{4,i}^{'}) = e(g^{t_{2,i} s_1} g^{h_{2,i} s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1}).$$

Hence, we expand the above formula as follows:

$$\prod_{i=1}^{n} e(C_{1,i}, K'_{1,i}) \cdot e(C_{2,i}, K'_{2,i}) \cdot e(C_{3,i}, K'_{3,i}) \cdot e(C_{4,i}, K'_{4,i})$$

$$= \prod_{i=1}^{n} e(g^{w_{1,i}s_1} g^{f_{1,i}s_2} g^{\delta_1 x_i s_3}, g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} g_2^{d\delta_2}) \cdot e(g^{w_{2,i}s_1} g^{f_{2,i}s_2} g^{\delta_2 x_i s_3},$$
$$g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} g_2^{-d\delta_1}) \cdot e(g^{t_{1,i}s_1} g^{h_{1,i}s_2} g^{\theta_1 x_i s_4}, g^{-\theta_2 \phi'_i} g^{\lambda'_1 v_i t_{2,i}} g_2^{d\theta_2})$$
$$\cdot e(g^{t_{2,i}s_1} g^{h_{2,i}s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1})$$

$$= \prod_{i=1}^{n} e(g^{w_{1,i}s_1}, g^{-\delta_2 r'_i}) \cdot e(g^{f_{1,i}s_2}, g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} g_2^{d\delta_2}) \cdot e(g^{\delta_1 x_i s_3}, g^{\lambda'_1 v_i w_{2,i}})$$
$$\cdot e(g^{w_{1,i}s_1}, g_2^{d\delta_2}) \cdot e(g^{w_{2,i}s_1}, g^{\delta_1 r'_i}) \cdot e(g^{f_{2,i}s_2}, g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} g_2^{-d\delta_1})$$
$$\cdot e(g^{\delta_2 x_i s_3}, g^{-\lambda'_1 v_i w_{1,i}}) \cdot e(g^{w_{2,i}s_1}, g_2^{-d\delta_1}) \cdot e(g^{t_{1,i}s_1}, g^{-\theta_2 \phi'_i})$$
$$\cdot e(g^{h_{1,i}s_2}, g^{-\theta_2 \phi'_i} g^{\lambda'_2 v_i t_{2,i}} g_2^{d\theta_2}) \cdot e(g^{\theta_1 x_i s_4}, g^{\lambda'_2 v_i t_{2,i}}) \cdot e(g^{t_{1,i}s_1}, g_2^{d\theta_2})$$
$$\cdot e(g^{t_{2,i}s_1}, g^{\theta_1 \phi'_i}) \cdot e(g^{h_{2,i}s_2}, g^{\theta_1 \phi'_i} g^{-\lambda'_2 v_i t_{1,i}} g_2^{-d\theta_1}) \cdot e(g^{\theta_2 x_i s_4}, g^{-\lambda'_1 v_i t_{1,i}})$$
$$\cdot e(g^{t_{2,i}s_1}, g_2^{-d\theta_1})$$

$$= \prod_{i=1}^{n} e(g^{-\delta_2 w_{1,i}}, g^{r'_i s_1}) \cdot e(g^{s_2}, (g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} g_2^{d\delta_2})^{f_{1,i}}) \cdot e(g,g)^{\lambda'_1 \delta_1 w_{2,i} x_i v_i s_3}$$
$$\cdot e(g^{w_{1,i}s_1}, g_2^{d\delta_2}) \cdot e(g^{\delta_1 w_{2,i}}, g^{r'_i s_1}) \cdot e(g^{s_2}, (g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} g_2^{-d\delta_1})^{f_{2,i}})$$
$$\cdot e(g,g)^{-\lambda'_1 \delta_2 w_{1,i} x_i v_i s_3} \cdot e(g^{w_{2,i}s_1}, g_2^{-d\delta_1}) \cdot e(g^{-\theta_2 t_{1,i}}, g^{\phi'_i s_1})$$
$$\cdot e(g^{s_2}, (g^{-\theta_2 \phi'_i} g^{\lambda'_2 v_i t_{2,i}} g_2^{d\theta_2})^{h_{1,i}}) \cdot e(g,g)^{\lambda'_2 \theta_1 t_{2,i} x_i v_i s_4} \cdot e(g^{t_{1,i}s_1}, g_2^{d\theta_2})$$
$$\cdot e(g^{\theta_1 t_{2,i}}, g^{\phi'_i s_1}) \cdot e(g^{s_2}, (g^{\theta_1 \phi'_i} g^{-\lambda'_2 v_i t_{1,i}} g_2^{-d\theta_1})^{h_{2,i}}) \cdot e(g,g)^{-\lambda'_2 \theta_2 t_{1,i} x_i v_i s_4}$$
$$\cdot e(g^{t_{2,i}s_1}, g_2^{-d\theta_1})$$

$$= \prod_{i=1}^{n} e(g^{\delta_1 w_{2,i} - \delta_2 w_{1,i}}, g^{r'_i s_1}) \cdot e(g^{\theta_1 t_{2,i} - \theta_2 t_{1,i}}, g^{\phi'_i s_1})$$
$$\cdot e(g^{s_2}, K'^{\,f_{1,i}}_{1,i} K'^{\,f_{2,i}}_{2,i} K'^{\,h_{1,i}}_{3,i} K'^{\,h_{2,i}}_{4,i})$$
$$\cdot e(g,g)^{[\lambda'_1(\delta_1 w_{2,i} - \delta_2 w_{1,i}) s_3 + \lambda'_2(\theta_1 t_{2,i} - \theta_2 t_{1,i}) s_4] x_i v_i} \cdot e(g^{-\delta_1 w_{2,i} + \delta_2 w_{1,i}}, g_2^{ds_1})$$
$$\cdot e(g^{-\theta_1 t_{2,i} + \theta_2 t_{1,i}}, g_2^{ds_1})$$
$$= e(g^{\Omega s_1}, \prod_{i=1}^{n} g^{(r'_i + \phi'_i)}) \cdot e(g^{s_2}, \prod_{i=1}^{n} K'^{\,f_{1,i}}_{1,i} K'^{\,f_{2,i}}_{2,i} K'^{\,h_{1,i}}_{3,i} K'^{\,h_{2,i}}_{4,i})$$
$$\cdot e(g,g)^{\Omega(\lambda'_1 s_3 + \lambda'_2 s_4) < \vec{x}, \vec{v} >} \cdot e(g^{-\Omega}, g_2^{ds_1}).$$

Finally, the algorithm outputs $\hat{CT}$ to obtain:

$$\hat{CT} = e(A, K_A^{'}) \cdot e(B, K_B^{'}) \cdot \prod_{i=1}^{n} e(C_{1,i}, K_{1,i}^{'}) \cdot e(C_{2,i}, K_{2,i}^{'}) \cdot e(C_{3,i}, K_{3,i}^{'}) \cdot e(C_{4,i}, K_{4,i}^{'})$$

$$= e(g^{s_2} g_2 \prod_{i=1}^{n} K_{1,i}^{'\,-f_{1,i}} K_{2,i}^{'\,-f_{2,i}} K_{3,i}^{'\,-h_{1,i}} K_{4,i}^{'\,-h_{2,i}}) \cdot e(g^{\Omega s_1}, \prod_{i=1}^{n} g^{-(r_i' + \phi_i')})$$
$$\cdot e(g^{\Omega s_1}, \prod_{i=1}^{n} g^{(r_i' + \phi_i')}) \cdot e(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{'\,f_{1,i}} K_{2,i}^{'\,f_{2,i}} K_{3,i}^{'\,h_{1,i}} K_{4,i}^{'\,h_{2,i}})$$
$$\cdot e(g^{-\Omega}, g_2^{ds_1}) \cdot e(g, g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \vec{x}, \vec{v} >}$$
$$= e(g^{s_2}, g_2) \cdot e(g, g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \vec{x}, \vec{v} >} \cdot e(g^{-\Omega}, g_2^{ds_1}).$$

$M \leftarrow$ **Decrypt** $(CT_{\vec{x}}, SK_{\vec{v}})$. On input the ciphertext $CT_{\vec{x}}$ and a private key $SK_{\vec{v}}$, the algorithm proceeds differently according to two *Level-1* or *Level-2 access*:

1. **Level-1 access.** If $CT_{\vec{x}}$ is an original well-formed ciphertext, then algorithm decrypts $CT_{\vec{x}} = (A, B, \{C_{1,i}, C_{2,i}\}_{i=1}^{n},\ \{C_{3,i}, C_{4,i}\}_{i=1}^{n},\ D = e(g, g_2)^{-s_2} M)$ using the private key $SK_{\vec{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^{n}, \{K_{3,i}, K_{4,i}\}_{i=1}^{n})$ to output message M:

   $$M \leftarrow D \cdot e(A, K_A) \cdot e(B, K_B) \cdot \prod_{i=1}^{n} e(C_{1,i}, K_{1,i}) \cdot e(C_{2,i}, K_{2,i}) \cdot e(C_{3,i}, K_{3,i}) \cdot e(C_{4,i}, K_{4,i}).$$

   In this step, the masking elements used in Encrypt algorithm have to be canceled out. To this purpose, the proposed scheme generates two relative pairing values, a positive and a negative in order to be removed at the end. This can be checked by the following equality:

   $$e(C_{1,i}\ ,\ K_{1,i})\ .\ e(C_{2,i}\ ,\ K_{2,i}) = e(g^{w_{1,i} s_1} g^{f_{1,i} s_2} g^{\delta_1 x_i s_3},\ g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}})$$
   $$\cdot e(g^{w_{2,i} s_1} g^{f_{2,i} s_2} g^{\delta_2 x_i s_3},\ g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}),$$

   where both $e(g^{w_{1,i} s_1}\ ,\ g^{\lambda_1 v_i w_{2,i}})$ and $e(g^{\delta_1 x_i s_3}\ ,\ g^{-\delta_2 r_i})$ are canceled out. Additionally, we need to remove $e(g^{w_{1,i} s_1}\ ,\ g^{-\delta_2 r_i})\ .\ e(g^{w_{2,i} s_1}\ ,\ g^{\delta_1 r_i})$ that are changed into one pairing as $e(g^{\Omega s_1}\ ,\ g^{r_i})$. This value is also eliminated by the additional computation of $e(B, K_B)$ in the decryption procedure.

   Correctness. Assume the ciphertext $CT_{\vec{x}}$ is well-formed the vector $\vec{x} = x_1, \ldots, x_n$. Then, we have:

   $$D \cdot e(A, K_A) \cdot e(B, K_B) \cdot \prod_{i=1}^{n} e(C_{1,i}, K_{1,i}) \cdot e(C_{2,i}, K_{2,i}) \cdot e(C_{3,i}, K_{3,i})$$
   $$\cdot e(C_{4,i}, K_{4,i})$$
   $$= e(g, g_2)^{-s_2} M \cdot e(g, g_2)^{s_2} \cdot e(g, g)^{\Omega(\lambda_1 s_3 + \lambda_2 s_4) < \vec{x}, \vec{v} >}$$
   $$= M \cdot e(g, g)^{\Omega(\lambda_1 s_3 + \lambda_2 s_4) < \vec{x}, \vec{v} >}.$$

It is worth noting that the term $e(g, g_2)^{s_2}$ is generated from the pairing computation of $e(A, K_A) = e(g_2 \prod_{i=1}^{n} K_{1,i}{}^{-f_{1,i}}{}_{2,i}{}^{-f_{2,i}} K_{3,i}{}^{-h_{1,i}} K_{4,i}{}^{-h_{2,i}})$. Thus, the output of the above result is M if $< \vec{x}, \vec{v} >= 0$ in $\mathbb{Z}_p^*$. If $< \vec{x}, \vec{v} > \neq 0$ in $\mathbb{Z}_p^*$, then there is only such case that $\lambda_1 s_3 + \lambda_2 s_4 = 0$ in $\mathbb{Z}_p^*$ with probability at most $1/p$, as in the predicate-only *IPE* scheme.

2. **Level-2 access** (from here on referred to as Re-Decrypt). If $CT_{\vec{x}}$ is a re-encrypted well-formed ciphertext, then it is of the form $CT'_{\vec{x}} = (A, B, CT_{\vec{w}}, \hat{CT}, D = e(g, g_2)^{-s_2} M)$. The algorithm first decrypts $CT_{\vec{w}}$ using $SK_{\vec{v'}}$ as above to obtain ${g_2}^d$ as Decrypt $(SK_{\vec{v'}}, CT_{\vec{w}}) \rightarrow {g_2}^d$.

   Then, it calculates: $\bar{CT} = e(B, {g_2}^d) = e(g^{s_1 \Omega}, {g_2}^d)$ and obtains the message as $M \leftarrow D \cdot \hat{CT} \cdot \bar{CT}$.

   The *Level-2 access* of the Decrypt algorithm consists of the original blinded message, the product with the new blinding factor obtained by decrypting the original ciphertext with the modified decryption key in the Re-Encrypt algorithm, the element B from the original ciphertext and the ciphertext component of the Re-KeyGen algorithm. To decrypt a re-encrypted ciphertext of *Level-2 access*, the proposed scheme first decrypts the ciphertext component of the Re-KeyGen algorithm to obtain the group element, then combines this group element with the element B from the original ciphertext to use the result removing both the original blinding factor of the message and the new binding factor introduced by the Re-Encrypt algorithm. Finally, the message is recovered if the vector $\vec{x}$ associated with the ciphertext and the vector $\vec{v}$ associated with the private key m orthogonal vectors (e.g., $< \vec{x}, \vec{v} >= 0$).

   Correctness. To verify the correctness, we compute $D \cdot \hat{CT} \cdot \bar{CT}$ as:

$$
\begin{aligned}
& e(g, g_2)^{-s_2} M \cdot e(g^{s_2}, g_2) \cdot e(g, g)^{\Omega(\lambda'_1 s_3 + \lambda'_2 s_4) < \vec{x}, \vec{v} >} \cdot e(g^{-\Omega}, g_2^{ds_1}) \\
& \cdot e(g^{s_1 \Omega}, {g_2}^d) \\
& = e(g, g_2)^{-s_2} M \cdot e(g, g_2)^{s_2} \cdot e(g, g)^{\Omega(\lambda'_1 s_3 + \lambda'_2 s_4) < \vec{x}, \vec{v} >} \cdot e(g, g_2)^{-s_1 \Omega d} \\
& \quad \cdot e(g, g_2)^{s_1 \Omega d} \\
& = M \cdot e(g, g)^{\Omega(\lambda_1{}' s_3 + \lambda_2{}' s_4) < \vec{x}, \vec{v} >}.
\end{aligned}
$$

   The result outputs M if $< \vec{x}, \vec{v} >= 0$ in $\mathbb{Z}_p^*$. If $< \vec{x}, \vec{v} > \neq 0$ in $\mathbb{Z}_p^*$, then there is only such case that $(\lambda'_1 s_3 + \lambda'_2 s_4) = 0$ in $\mathbb{Z}_p^*$ with probability at most $1/p$, as in the predicate-only *IPE* scheme.

4.3.2 Proof of Security

Here, we describe a mechanism to show that our proposed scheme achieves the security requirements according to the definitions stated in Section 4.1. For *Level-1* and *Level-2* ciphertext challenge, an adversary may request private key, re-encryption key, and re-encryption queries by choosing vectors $(\vec{x}_0, \vec{x}_1, \vec{w}_0, \vec{w}_1)$ at the beginning of the security game. For instance, in the case of *Level-1 access*, the adversary outputs two vectors $\vec{x}_0, \vec{x}_1$ and queries corresponding to a vector $\vec{v}$ such that $< \vec{v}, \vec{x}_0 >=< \vec{v}, \vec{x}_1 >= 0$, where $M_0 = M_1$. The adversary goal is to decide which one of the two vectors is associated with the challenge ciphertext. In the case of *Level-2 access*, the adversary outputs challenge vectors $\vec{x}_0, \vec{x}_1$ along with $\vec{w}_0, \vec{w}_1$ for re-encryption keys. The adversary goal is to decide which one of the two vectors $\vec{w}_0, \vec{w}_1$ is associated with the re-encrypted query.

- To prove the *Level-1 access*, similarly to [10] we suppose that our encryption system contains two parallel sub-systems. That is, a challenge ciphertext will be encrypted with respect to one vector in the first subsystem and a different vector in a second sub-system. Let $(\vec{a}, \vec{b})$ denote a ciphertext encrypted using $\vec{0}$ vector (that is orthogonal to everything) in an intermediate game to prove indistinguishably when encrypting to $\vec{x_0}$ corresponding to $(\vec{x}_0, \vec{x}_0)$ and when encrypting to $\vec{x}_1$ corresponding to $(\vec{x}_1, \vec{x}_1)$ as:

  $$(\vec{x}_0, \vec{x}_0) \approx (\vec{x}_0, \vec{0}) \approx (\vec{x}_0, \vec{x}_1) \approx (\vec{0}, \vec{x}_1) \approx (\vec{x}_1, \vec{x}_1).$$

  This structure allows us to use a simulator (challenger) that will essentially work in one subsystem without knowing what is happening in the other one [10]. It determines whether a sub-system encrypts the given vector or the zero vector. Details of this proof are given in [1].

- To prove the *Level-2 access*, we apply game transformation proof [69] with multiple sequences of games whose aim are to change components of the challenge ciphertext to independent ones from challenge bit b (random form). In the following, we discuss it in details.

In the following, we show that the proposed *IPPRE* scheme is predicate- and attribute-hiding re-encrypted ciphertext (*Level-2*) against chosen-plaintext attacks provided the underlying *IPE* scheme under the Decision Linear assumption holds in $\mathbb{G}$.

Proof of Theorem 1 (PAH-L2: Predicate- and Attribute-hiding Re-encrypted ciphertext)
We consider two cases in the proof of Theorem 1 according to the value of $s_{M,\vec{x},\vec{v}}$ mentioned in the Definition 4.5. This value holds the following claims:

- For any private key query $\vec{v'}$, the variable $s_{M,\vec{x},\vec{v}} = 0$ when it holds $< \vec{v'}, \vec{w}_0 >=< \vec{v'}, \vec{w}_1 > \neq 0$.
- For any private key query $\vec{v'}$, the variable $s_{M,\vec{x},\vec{v}} = 1$ when it holds $< \vec{v'}, \vec{w}_0 >=< \vec{v'}, \vec{w}_1 >$.

Theorem 1. The *IPPRE* scheme is predicate- and attribute-hiding for re-encrypted ciphertexts against chosen-plaintext attacks provided underlying *IPE* scheme is fully attribute-hiding. For any adversary $\mathcal{A}$, there exist probabilistic machines $\epsilon_{1-1}, \epsilon_{1-2}, \epsilon_{2-1}$, and ϵ_{2-2} whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ:

$$Adv_{\mathcal{A}}^{\mathsf{PAH\text{-}L2}}(\lambda) \leq Adv_{\epsilon_{1-1}}^{\mathsf{IPE\text{-}AH}}(\lambda) + Adv_{\epsilon_{1-2}}^{\mathsf{IPE\text{-}AH}}(\lambda) + \frac{1}{2}(Adv_{\epsilon_{2-1}}^{\mathsf{IPE\text{-}AH}}(\lambda) + Adv_{\epsilon_{2-2}}^{\mathsf{IPE\text{-}AH}}(\lambda)).$$

Proof. We execute a preliminary game transformation from Game 0 (original game in Definition 4.5) to Game $0'$, which is the same as Game 0 except flip a coin $\tau_{M,\vec{x},\vec{v}}$ before setup, and the game is aborted at the final step if $\tau_{M,\vec{x},\vec{v}} \neq s_{M,\vec{x},\vec{v}}$. Hence, the advantage of Game $0'$ is a half of that in Game 0. The value $\tau_{M,\vec{x},\vec{v}}$ is chosen independently from $s_{M,\vec{x},\vec{v}}$, and therefore the probability that the game is aborted is $\frac{1}{2}$ that is $Adv_{\mathcal{A}}^{(0')}(\lambda) = \frac{1}{2} \cdot Adv_{\mathcal{A}}^{\mathsf{PAH\text{-}L2}}(\lambda)$. Moreover, $\mathsf{Pr}[\mathcal{A}\ wins] = \frac{1}{2}(\mathsf{Pr}(\mathcal{A}\ wins|\tau_{M,\vec{x},\vec{v}} = 0) + (\mathsf{Pr}(\mathcal{A}\ wins|\tau_{M,\vec{x},\vec{v}} = 1))$ in Game $0'$. Hence, we have:

$$Adv_{\mathcal{A}}^{\mathsf{PAH\text{-}L2}}(\lambda) \leq Adv_{\epsilon_{1-1}}^{\mathsf{IPE\text{-}AH}}(\lambda) + Adv_{\epsilon_{1-2}}^{\mathsf{IPE\text{-}AH}}(\lambda)\frac{1}{2}(Adv_{\epsilon_{2-1}}^{\mathsf{IPE\text{-}AH}}(\lambda) + Adv_{\epsilon_{2-2}}^{\mathsf{IPE\text{-}AH}}(\lambda)).$$

Therefore, to show how our scheme is predicate- and attribute-hiding for re-encrypted ciphertext under the *D-Linear* assumption, we consider the two cases as below:

Proof of Theorem 1 in the case $\tau_{M,\vec{x},\vec{v}=0}$

Lemma 1. The proposed *IPPRE* scheme is predicate- and attribute-hiding for re-encrypted ciphertexts against chosen-plaintext attack in the case $\tau_{M,\vec{x},\vec{v}}$ under the attribute-hiding underlying *IPE* scheme.

For any adversary $\mathcal{A}$, there exist probabilistic mechanisms ϵ_{1-1} and ϵ_{1-2}, whose running times are essentially the same as that of $\mathcal{A}$ such that for any

security parameter λ in the case $\tau_{M,\vec{x},\vec{v}=0}$:

$$\mathsf{Pr}[\mathcal{A}_{wins}|\tau_{m,\vec{x},\vec{v}}=0]-\tfrac{1}{2}\leq \mathsf{Adv}^{IPE-AH}_{\epsilon_{1-1}}+\mathsf{Adv}^{IPE-AH}_{\epsilon_{1-2}}.$$

The aim is that $CT_{\vec{w}}$ is changed to a ciphertext with random attribute and random attribute message. We apply the game transformation consisting of three games Game $0'$, Game 1, and Game 2. In Game 1, the $CT_{\vec{w}_b}$ under vector $\vec{w}_b$ is changed to $CT_{\vec{r}} = \mathsf{Encrypt}\,(PK, \vec{r}, R)$ where $\vec{r}$ is chosen uniformly random from Σ and random value $R \in \mathbb{G}_T$.
In the case $\tau_{M,\vec{x},\vec{v}=0}$, the adversary does not request private key query $\vec{v}$ such that $<\vec{x_b},\vec{v}>=0$. Hence, $CT_{\vec{x_b}}$ is changed to $\mathsf{Encrypt}_{IPE}(PK,\vec{r},R)$ by using the attribute-hiding security underlying *IPE* scheme.

Proof of Lemma 1. In order to prove the Lemma 1, we consider the following games. We only describe the components which are changed in the other games.

Game $0'$**.** Same as Game 0 except that flip a coin $\tau_{M,\vec{x},\vec{v}} \xleftarrow{U} \{0,1\}$ before setup, and the game is aborted if $\tau_{M,\vec{x},\vec{v}} \neq S_{M,\vec{x},\vec{v}}$. We consider the case with $\tau_{M,\vec{x},\vec{v}}=0$ and rely to the challenge query $(\vec{x_0},\vec{x_1},M_0,M_1,\vec{v_0},\vec{v_1},\vec{w_0},\vec{w_1})$ as the following:

$$\begin{aligned}
&(\boldsymbol{A}=g^{s_2},\ \boldsymbol{B}={g_1}^{s_1}=g^{s_1\Omega},\ \boldsymbol{\Lambda}=e(g,g_2)^{-s_2},\\
&\boldsymbol{CT}_{\vec{x}_b}=\mathsf{Encrypt}\,(PK,\vec{x_b},M_b)\\
&\quad=(g^{s_2},\ g_1^{s_1},\ \{W_{1,i}^{s_1}\cdot F_{1,i}^{s_2}\cdot U_1^{x_{ib}s_3},W_{2,i}^{s_1}\cdot F_{2,i}^{s_2}\cdot U_2^{x_{ib}s_3}\}_{i=1}^n,\\
&\quad\{T_{1,i}^{s_1}\cdot H_{1,i}^{s_2}\cdot V_1^{x_{ib}s_4},T_{2,i}^{s_1}\cdot H_{2,i}^{s_2}\cdot V_2^{x_{ib}s_4}\}_{i=1}^n,\ \Lambda^{-s_2}M_b)\in\mathbb{G}^{4n+2}\times\mathbb{G}_{\mathbb{T}},\\
&\boldsymbol{CT}_{\vec{w}_b}=\mathsf{Encrypt}\,(PK,\vec{w_b},{g_2}^d)\\
&\quad=(g^{s_2},\ g_1^{s_1},\ \{W_{1,i}^{s_1}\cdot F_{1,i}^{s_2}\cdot U_1^{w_{ib}s_3},W_{2,i}^{s_1}\cdot F_{2,i}^{s_2}\cdot U_2^{w_{ib}s_3}\}_{i=1}^n,\\
&\quad\{T_{1,i}^{s_1}\cdot H_{1,i}^{s_2}\cdot V_1^{w_{ib}s_4},T_{2,i}^{s_1}\cdot H_{2,i}^{s_2}\cdot V_2^{w_{ib}s_4}\}_{i=1}^n,\ \Lambda^{-s_2}{g_2}^d)\in\mathbb{G}^{4n+2}\times\mathbb{G}_{\mathbb{T}},\\
&\hat{\boldsymbol{CT}}=e(g^{s_2},g_2)\cdot e(g,g)^{\Omega(\lambda_1's_3+\lambda_2's_4)<\vec{x_b},\vec{v_b}>}\cdot e(g^{-\Omega},g_2^{ds_1})).
\end{aligned}$$

Game 1**.** Game 1 is the same as Game $0'$ except that the reply to challenge query for $(\vec{x_0},\vec{x_1},M_0,M_1,\vec{v_0},\vec{v_1},\vec{w_0},\vec{w_1})$ is as follows:

$$\begin{aligned}
&\boldsymbol{CT}_{\vec{r}}=\mathsf{Encrypt}\,(PK,\vec{r},{g_2}^{d'})\\
&\quad=(g^{s_2},\ g_1^{s_1},\ \{W_{1,i}^{s_1}\cdot F_{1,i}^{s_2}\cdot U_1^{r_is_3},W_{2,i}^{s_1}\cdot F_{2,i}^{s_2}\cdot U_2^{r_is_3}\}_{i=1}^n,\\
&\quad\{T_{1,i}^{s_1}\cdot H_{1,i}^{s_2}\cdot V_1^{r_is_4},T_{2,i}^{s_1}\cdot H_{2,i}^{s_2}\cdot V_2^{r_is_4}\}_{i=1}^n,\ \Lambda^{-s_2}{g_2}^{d'})\in\mathbb{G}^{4n+2}\times\mathbb{G}_{\mathbb{T}},\\
&\hat{\boldsymbol{CT}}=e(g^{s_2},g_2)\cdot e(g,g)^{\Omega(\lambda_1's_3+\lambda_2's_4)<\vec{x_b},\vec{v_b}>}\cdot e(g^{-\Omega},{g_2}^{d's}),
\end{aligned}$$

where $\vec{r} = \{r_0, \ldots, r_n\} \xleftarrow{U} \mathcal{F}$ and $d' \xleftarrow{U} \mathbb{Z}_p^*$.

Game 2. Game 2 is the same as Game 1 except that the reply to challenge query for $(\vec{x_0}, \vec{x_1}, M_0, M_1, \vec{v_0}, \vec{v_1}, \vec{w_0}, \vec{w_1})$ is as follows:

$$\begin{aligned}\boldsymbol{CT}_{\vec{u}} &= \mathsf{Encrypt}(PK, \vec{u}, M_b) \\ &= (g^{s_2},\ g_1^{s_1},\ \{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{u_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{u_i s_3}\}_{i=1}^n, \\ &\quad \{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{u_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{u_i s_4}\}_{i=1}^n,\ \Lambda^{-s_2} M_b) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}},\end{aligned}$$

$$\hat{\boldsymbol{CT}} = e(g^{s_2}, g_2) \cdot e(g, g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \vec{u}, \vec{u'} >} \cdot e(g^{-\Omega}, g_2^{d s_1})),$$

where $\vec{u}, \vec{u'} \xleftarrow{U} \mathcal{F}$. We note that $\vec{u}$ and $\vec{u'}$ are chosen uniformly and independent from $\vec{x_b}$ and $\vec{v_b}$, respectively. $CT_{\vec{w}}$ is generated as in Game 1.

Let $\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)$ and $\mathsf{Adv}_{\mathcal{A}}^{(2)}(\lambda)$ be the advantages of $\mathcal{A}$ in Games $0', 1$ and 2, respectively. We will use three lemmas (Lemmas $2, 3, 4$) that evaluate the gaps between pairs of neighboring games. From these lemmas we obtain:

$$\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) \leq |\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)| + |\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2)}(\lambda)| + \mathsf{Adv}_{\mathcal{A}}^{(2)}(\lambda) \leq$$

$$\mathsf{Adv}_{\beta_{1-1}}^{IPE-AH}(\lambda) + \mathsf{Adv}_{\beta_{1-2}}^{IPE-AH}(\lambda).$$

Lemma 2. For any adversary $\mathcal{A}$, there exists a probabilistic machine β_{1-1} and β_{1-2}, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ:

$$|\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)| \leq \mathsf{Adv}_{\beta_{1-1}}^{IPE-AH}(\lambda) + \mathsf{Adv}_{\beta_{1-2}}^{IPE-AH}(\lambda).$$

Proof of Lemma 2. We construct probabilistic machines β_{1-1} and β_{1-2} against the fully-attribute-hiding security using an adversary $\mathcal{A}$ in a security game (Game $0'$ or Game 1) as a block box. To this purpose, we consider the intermediate game Game $1'$ that is the same as Game $0'$ except that $CT_{\vec{w}}$ of the reply the challenge re-encrypted ciphertext is of the form of Game 1. Hence, to prove that $|\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1')}(\lambda)| \leq \mathsf{Adv}_{\beta_{1-1}}^{IPE-AH}(\lambda)$, we construct a probabilistic machine β_{1-1} against the fully attribute-hiding security using the adversary $\mathcal{A}$ in a security game (Game $0'$ or Game $1'$) as block box as follows:

1. β_{1-1} plays a role of the challenger in the security game against the adversary $\mathcal{A}$.

2. β_{1-1} generates a public and private key and provides $\mathcal{A}$ with the public key and keeps the private key as details are stated in Section 4.3.1:

$$PK = (g,\ g_1,\ \{W_{1,i}, W_{2,i}, F_{1,i}F_{2,i}\}_{i=1}^{n},\ \{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\}_{i=1}^{n},\ \{U_i, V_i\}_{i=1}^{2},\ \Lambda),$$

$$MSK = (\{w_{1,i}, w_{2,i}, t_{1,i}, t_{2,i}, f_{1,i}, f_{2,i}, h_{1,i}, h_{2,i}\}_{i=1}^{n},\ \{\delta_i, \theta_i\}_{i=1}^{2},\ g_2).$$

3. When a private key query is issued for a vector $\vec{v}$, $\beta-1$ computes a normal form decryption key and provides $\mathcal{A}$ with $SK_{\vec{v}} = (K_A,\ K_B,\ \{K_{1,i}, K_{2,i}\}_{i=1}^{n},\ \{K_{3,i}, K_{4,i}\}_{i=1}^{n})$.

4. When a re-encryption key query is issued for $(\vec{v}, \vec{w})$, β_{1-1} computes a normal form re-encryption key $RK_{\vec{v},\vec{w}} = (K_A^{'},\ K_B^{'},\ \{K_{1,i}^{'}, K_{2,i}^{'}\}_{i=1}^{n}, \{K_{3,i}^{'}, K_{4,i}^{'}\}_{i=1}^{n})$ along with $CT_{\vec{w}} = (PK, \vec{w}, {g_2}^d)$.

5. When a re-encryption query is issued for $(\vec{v}, \vec{w}, CT_{\vec{x}})$, the challenger β_{1-1} computes a normal form of re-encryption $CT_{\vec{x}}^{'}$ and provides $\mathcal{A}$ with $CT_{\vec{x}}^{'} = (A, B, CT_{\vec{w}}, \hat{CT}, D)$.

6. When a challenge query is issued for $(\vec{x_0}, \vec{x_1}, M_0, M_1, \vec{v_0}, \vec{v_1}, \vec{w_0}, \vec{w_1})$, β_{1-1} picks a bit $b \xleftarrow{U} \{0,1\}$ and computes $CT_{\vec{x}}$, $CT_{\vec{w}}$, $CT_{\vec{x}}^{'}$. The β_{1-1} submits $(X_b := {g_2}^d, X_{(1-b)} := R, \vec{x_b} := \vec{w_b}, \vec{x}_{1-b} := \vec{r})$ to the attribute-hiding challenger underlying *IPE* scheme (see Definition 4.1) where R and $\vec{r}$ are chosen independently uniform. It then receives $CT_{\vec{w_\beta}}$ for $\beta \xleftarrow{U} \{0,1\}$. Finally β_{1-1} provides $\mathcal{A}$ with a challenge ciphertext $CT_b^{'} = (A, B, CT_{\vec{w_b}} = CT_{\vec{w_\beta}}, \hat{CT}, D)$.

7. $\mathcal{A}$ finally outputs b_1. β_{1-1} outputs $\beta = 0$ if $b = b'$, otherwise outputs $\beta = 1$. Since $CT^{'}$ of the challenge re-encrypted ciphertext is of the form Game $0'$ (resp. Game 1 if $\beta = 0$ (resp. $\beta = 1$), the view of $\mathcal{A}$ given by β_{1-1} is distributed as Game $1'$ (resp. Game $0'$) if $\beta = 0$ (resp. $\beta = 1$). Then:

$$|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1')}(\lambda)| \leq |Pr[b = b'] - \tfrac{1}{2}| \leq \mathsf{Adv}_{\beta_{1-1}}^{IPE-AH}(\lambda)$$

In a similar way, we construct a probabilistic machine β_{1-2} against the fully attribute-hiding security using an adversary $\mathcal{A}$ in a security game (Game $1'$ or Game 1) as a block box. Game 1 is the same as Game $1'$ except that $CT_{\vec{w}}$ of the reply to the challenge re-encrypted ciphertext $CT_{\vec{x}}$ where $\vec{r} \xleftarrow{U} \mathcal{F}$. Hence, we have:

$$|\mathsf{Adv}_{\mathcal{A}}^{(1')}(\lambda) - \mathsf{Adv}_{\mathcal{A}|}^{(1)}(\lambda)| \leq \mathsf{Adv}_{\beta_{1-2}}^{IPE-AH}(\lambda).$$

Therefore, we can prove this Lemma by using hybrid argument.

Lemma 3. For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda)$.

Proof of Lemma 3. From the adversary's view, $CT_{\vec{x}}$ of Game 1 and $CT_{\vec{u}}$ of Game 2 where $\vec{u} \xleftarrow{U} \mathcal{F}$ are information-theoretically indistinguishable.

Lemma 4. For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda) = 0$.

Proof of Lemma 4. The value b is independent from adversary's view in Game 2. Hence, $\mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda) = 0$.

Proof of Theorem 1 in the case $\tau_{M,\vec{x},\vec{v}=1}$

Lemma 5. The proposed *IPPRE* scheme is predicate- and attribute-hiding for re-encrypted ciphertexts against chosen-plaintext attack in the case $\tau_{M,\vec{x},\vec{v}=1}$ under the attribute-hiding underlying *IPE* scheme.

For any adversary $\mathcal{A}$, there exist probabilistic mechanisms ϵ_{2-1} and ϵ_{2-2}, whose running times are essentially the same as that of $\mathcal{A}$ such that for any security parameter λ in the case $\tau_{M,\vec{x},\vec{v}=1}$:

$$\mathsf{Pr}[\mathcal{A}_{wins}|\tau_{m,\vec{x},\vec{v}} = 1] - \tfrac{1}{2} \leq \mathsf{Adv}^{IPE-AH}_{\epsilon_{2-1}} + \mathsf{Adv}^{IPE-AH}_{\epsilon_{2-2}}.$$

The aim of game transformation here is that $CT_{\vec{w}_b}$ is changed to ciphertext with opposite attribute $\vec{w}_{(1-b)}$. Again, we employ two games Game $0'$ and Game 1. In Game 1, the $CT_{\vec{w}_b}$ is changed to $\mathsf{Encrypt}\,(PK, \vec{w}_{(1-b)}, {g_2}^d)$, respectively, by using the fully attribute-hiding security of the *IPE* scheme.

Proof of Lemma 5. To prove this lemma, we consider the following games:

Game $0'$. Same as Game 0 except that flip a coin $\tau_{M,\vec{x},\vec{v}} \xleftarrow{U} \{0,1\}$ before setup, and the game is aborted if $\tau_{M,\vec{x},\vec{v}} \neq S_{M,\vec{x},\vec{v}}$. We consider the case with $\tau_{M,\vec{x},\vec{v}} = 1$. Again here we only describe the components which are changed in the other games. The reply to challenge query for $(M, \vec{x}, \vec{v}, \vec{w}_0, \vec{w}_1)$ with

$(M, \vec{x}, \vec{v}) = (M_0, \vec{x}_0, \vec{v}_0) = (M_1, \vec{x}_1, \vec{v}_1)$ is:

$$\begin{aligned}\boldsymbol{CT}_{\vec{w}_b} &= \mathsf{Encrypt}\,(PK, \vec{w}_b, {g_2}^d)\\ &= (g^{s_2},\ g_1^{s_1},\ \{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{w_{ib}s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{w_{ib}s_3}\}_{i=1}^n,\\ &\quad \{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{w_{ib}s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{w_{ib}s_4}\}_{i=1}^n,\ \Lambda^{-s_2} {g_2}^d) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}},\end{aligned}$$

where $d \xleftarrow{U} \mathbb{Z}_p^*$.

Game 1. Game 1 is the same as Game $0'$ except that the reply to the challenge query for $(M, \vec{x}, \vec{v}, \vec{w}_0, \vec{w}_1)$ with $(M, \vec{x}, \vec{v}) = (M_0, \vec{x}_0, \vec{v}_0) = (M_1, \vec{x}_1, \vec{v}_1)$ is:

$$\begin{aligned}\boldsymbol{CT}_{\vec{w}_{1-b}} &= \mathsf{Encrypt}\,(PK, \vec{w}_{1-b}, {g_2}^d)\\ &= (g^{s_2},\ g_1^{s_1},\ \{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{w_{i1-b}s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{w_{i1-b}s_3}\}_{i=1}^n,\\ &\quad \{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{w_{i1-b}s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{w_{i1-b}s_4}\}_{i=1}^n,\ \Lambda^{-s_2} {g_2}^d) \in\\ &\quad \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}}.\end{aligned}$$

Let $\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda)$ and $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)$ be the advantage of $\mathcal{A}$ in Game $0'$ and Game 1, respectively. In order to evaluate the gaps between pairs of neighboring games, we consider the following Lemmas (6 and 7). We have:

$$\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) \leq |\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) + \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) \leq \mathsf{Adv}_{\beta_{2-1}}^{IPE-AH}(\lambda) +$$

$$\mathsf{Adv}_{\beta_{2-2}}^{IPE-AH}(\lambda) + \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda).$$

The proof is completed from the Lemma 7 since:

$$\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) \leq \tfrac{1}{2}(\mathsf{Adv}_{\beta_{2-1}}^{IPE-AH}(\lambda) + \mathsf{Adv}_{\beta_{2-2}}^{IPE-AH}(\lambda)).$$

Lemma 6. For any adversary $\mathcal{A}$, there exists a probabilistic machines β_{2-1} and β_{2-2}, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ:

$$|\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)| \leq \mathsf{Adv}_{\beta_{2-1}}^{IPE-AH}(\lambda) + \mathsf{Adv}_{\beta_{2-1}}^{IPE-AH}(\lambda).$$

The proof of this lemma is similar to the Lemma 2.

Lemma 7. For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) = -\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda)$ The challenge re-encrypted ciphertext for the opposite bit $1-b$ to the challenge bit b and the others components are normal forms in Game 1. Hence, success probability $Pr[Succ_{\mathcal{A}}^{(1)}]$ in Game 1 is $1 - Pr[Succ_{\mathcal{A}}^{(0')}]$, where $Succ_{\mathcal{A}}^{(0')}$ is success probability in Game $0'$. Therefore, we have $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) = -\mathsf{Adv}_{\mathcal{A}}^{(0')}(\lambda)$.

4.3.3 Performance Evaluation

Here, we present our evaluation results of the proposed inner-product proxy re-encryption (*IPPRE*) scheme in terms of computation and communication costs as well as storage overhead. We present both theoretical and the experimental results with the assumption that the total number of attributes in the system is equal to n.

4.3.3.1 Theoretical Results

The computational load, defined in terms of number of computational steps required to perform a given task, can be described in the following terms, depending on the party who is performing the task itself:

- **Computational Load on the Trust Authority.** The trust authority is responsible executing three algorithms: Setup, KeyGen, and Re-KeyGen. In the Setup algorithm, the main computation overhead consists of $(8n + 5)$ exponentiation operations on the group $\mathbb{G}_1$ and one pairing operation $e(g, g_2)$ that can be ignored since it can compute in advance (pre-computed).The main computation overhead of KeyGen algorithm belongs to the private key generation, which consumes $(9n)$ exponentiation operations on the group $\mathbb{G}_2$. The Re-KeyGen algorithm requires $(12n + 2)$ exponentiation operations on the group $\mathbb{G}_1$ encrypting g_2^d and $(13n)$ exponentiation operations on the group $\mathbb{G}_2$ for generating re-encryption key.

- **Computational Load on the Data Owner.** The computational overhead on the side of the data owner is caused by the execution of the Encrypt algorithm, which needs $(12n + 2)$ exponentiation operations on the group $\mathbb{G}_1$ and one exponentiation operations on the group $\mathbb{G}_T$.

- **Computational Load on the Proxy.** The proxy is responsible for transforming the ciphertext by executing the Re-Encrypt algorithm, which requires $(4n + 2)$ pairing operations.

- **Computational Load on the Users.** The computational overhead on the user side is mainly caused by the Decrypt algorithm. According to our protocol, we have two Decrypt algorithms: one decrypting a ciphertext and another decrypting a re-encrypted ciphertext. The computational overhead of the former consists of $(4n + 2)$ pairing operations. The computational overhead of the latter consists of $(4n + 3)$ pairing operations.

- **Communication Load.** The original ciphertext has four parts: $A = g^{s_2}, B = g^{s_1\Omega}, \{C_{1,i}, C_{2,i}\}_{i=1}^{n}$ and $\{C_{3,i}, C_{4,i}\}_{i=1}^{n}$. Each C_i has three elements. The ciphertext contains $(12n + 2)$ $\mathbb{G}_1$ and a (1) $\mathbb{G}_T$ group elements in total. The re-encrypted ciphertext contains $(4n + 2)$ $\mathbb{G}_T$ group elements.

- **Storage Load for Users.** The main storage load of each user is for the private key $SK_{\vec{v}}$, which represents $(9n)$ $\mathbb{G}_2$ group elements in total.

4.3.3.2 Experimental Results

We implemented our scheme in *C* using the Pairing-Based Crypto (*PBC*) library [70]. The experiments were carried out on Ubuntu *16.04 LTS* with *2.60 GHz 8x Intel(R) Core(TM) i7-4720HQ CPU* and *16 GB RAM*.

Using Different Types of Elliptic Curves. The choice of elliptic curve parameters impacts on the credential, signature sizes, and the computational efficiency. We measured the execution time of our scheme on three different types of elliptic curves with 80 bits of security level: SuperSingular (*SS*) curve (type *A*), *MNT* curves (type *D*) and Barreto-Naehrig (*BN*) curve (type *F*), respectively as defined in *PBC* [70]. The parameters of each curve are shown in Table 4.1.

Type of elliptic curve	**SuperSingular**	**MNT159**	**MNT201**	**BN**
Bit length of q	512	159	201	158
Bit length of r	160	158	181	158
Embedding Degree	2	6	6	12
Curve	$y^2 = x^3 + x$	$y^2 = x^3 + ax + b$	$y^2 = x^3 + ax + b$	$y^2 = x^3 + b$

Table 4.1: Curve Parameters

Elliptic curves are classified into two categories: symmetric bilinear group ($e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T, \mathbb{G}_1 = \mathbb{G}_2$) and asymmetric bilinear group ($e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T, \mathbb{G}_1 \neq \mathbb{G}_2$). To achieve fast pairing computation, elliptic curves from symmetric bilinear groups with small embedding degree are chosen. On the other hand, elliptic curves from asymmetric bilinear groups with high embedding degree offer a good operation for short group element size. For a symmetric bilinear group, we selected the SuperSingular curve over a prime finite field with embedding degree of 2 and the base field size of $\mathbb{G}$ equal to 512 bits. Then for asymmetric bilinear groups, we considered two *MNT* curves (namely *MNT159* and *MNT201*) with embedding

degree of 6 and one *BN* curve with embedding degree of 12 and the base 2 and 3, the execution time of each algorithm of our scheme considering an increasing number of attributes from 5 to 30 over 100 runs. The execution time of each algorithm increases linearly with the number of attributes according to its computational overhead (see Section 4.3.3.1).

The computational overhead of Encrypt algorithm is dominated by exponentiation operation on group $\mathbb{G}_1$ and therefore *MNT159* curve and *SS* curve respectively with smaller and larger base field size of $\mathbb{G}_1$[2] have the best and worst encryption performance. As shown in Table 4.2, for 5 attributes, the Encrypt algorithm takes about 19ms under *MNT159* curve and 41ms under *SS* curve. On the other hand, the computational overhead of the KeyGen and the Re-KeyGen algorithms is dominated by exponentiation operation on group $\mathbb{G}_2$. As we can see from Table 4.3, the execution time of the KeyGen and the Re-KeyGen algorithms under *BN* curve that has smaller base field size of $\mathbb{G}_2$[3] among other curves is more efficient.

Curve	**SS**			**MNT159**		
Attribute.num	**5**	**10**	**30**	**5**	**10**	**30**
Encrypt	41.6	78.6	234	19	33.8	91.5
Keygen	54.6	108.5	318.7	157	308.7	935
Decrypt	13.4	24.9	69.5	33.2	61.5	173.6
Re-encrypt	13.5	24.8	71.1	33	61.6	176.3
Re-keygen	131.1	240.9	715.5	273.5	509.7	1497.3
Re-Decrypt	17.2	28.9	73.9	47.2	76.6	188.9

Table 4.2: Average execution time (ms) of each algorithm of the proposed *IPPRE* scheme on eliptic curves *SS* and *MNT159*

The embedding degree of elliptic curves directly influences the size of $\mathbb{G}_T$ and increases the complexity of pairing computation. Therefore, the *SS* curve with embedding degree of 2 has the best execution time for the algorithms Re-Decrypt, Decrypt, and Re-Decrypt among other curves, as we can see in Tables 4.2. While, the *BN* curve with embedding degree 12 has higher execution time for the Re-Decrypt, Decrypt, and Re-Decrypt algorithms. Specifically, from the Tables 4.2 and 4.3 we can see that, in the case of 5 attributes the Decrypt and Re-Decrypt algorithms take less than 18ms

[2]The base field size of $\mathbb{G}_1$ *MNT159*, *BN*, *MNT201* and *SS* curves are 159 bits, 160 bits, 201 bits and 512 bits, respectively [70].

[3]The base field size of $\mathbb{G}_2$ *BN*, *MNT159*, *SS* and *MNT201* curves are 320 bits, 477 bits, 512 bits and 603 bits, respectively [70].

for the *SS* curve and less than 63ms for *MNT* curves, while for 10 attributes these algorithms take about 29ms for *SS* curve and less than 100ms for *MNT* curves.

Curve	**MNT201**			**BN**		
Attribute.num	**5**	**10**	**30**	**5**	**10**	**30**
Encrypt	25	45.4	123.7	34	48.9	106
Keygen	205	403.4	1219	40.2	79.4	241.2
Decrypt	42.8	80	235.6	367.4	691.4	2082.6
Re-encrypt	43.7	80.6	231.2	367.3	700.3	2025.4
Re-keygen	359.7	668.2	1960.9	87.8	153.5	447.4
Re-Decrypt	63.1	100.7	250.4	380.5	711.8	2036.8

Table 4.3: Average execution time (ms) of each algorithm of the proposed *IPPRE* scheme on eliptic curves *MNT201* and *BN*

4.3.4 Discussion of the Proposed *IPPRE* Scheme

We adopted a proxy re-encryption technique to address the problem of data sharing, where data collected for instance through IoT devices and encrypted by an IoT gateway, according to a given access policy, is shared with other medical staff holding different access policies. We analyzed the protocol in terms of performance and also we tested the execution time of each algorithm on different types of elliptic curves. While our proposed protocol is a first step towards a promising direction in terms of security and efficiency, it is impractical for adopting in IoT environment as it requires $O(n)$ computation for decryption with large sized public parameters, private key, and the ciphertext. To this purpose, in the next section, we introduce an efficient $IPPER$ scheme, *E-IPPRE* that requires constant pairing computations with short private key and ciphertext.

4.4 An Efficient Inner-product Proxy Re-encryption Scheme

In this section, we present an efficient variant of *IPPRE* scheme developing an inner-product encryption model presented in [2].

The scheme updates the attribute vectors via proxy server without interaction with the involved data owners. The proxy holds a re-encryption key to update all ciphertexts encrypted according to attribute vector $\vec{x}$ into

ciphertexts encrypted according to attribute vector $\vec{w}$. Our efficient inner-product proxy re-encryption (*E-IPPRE*) scheme is faster in decryption than the proposed *IPPRE* since only the attribute is computed as the exponent of a group for supporting attribute-hiding feature, but not the predicate. Hence, the *E-IPPRE* scheme requires a constant number of pairings (as the result three pairing computations) in decryption. According to the fact that each predicate is not computed as the exponent of a group, *E-IPPRE* scheme results in the shorter size of the private key and the time needed for key generation compared to the first version of our *IPPRE* protocol with the same level of security.

4.4.1 The E-IPPRE Framework

Our *E-IPPRE* is composed of six algorithms namely Setup, KeyGen, Encrypt, Re-KeyGen, Re-Encrypt, and Decrypt. Following, we describe each algorithm in details.

4.4.1.1 Scheme

We are given a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ over a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ of prime order p with respective generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$. The size of p is determined by security parameter. The proposed scheme works as follows:

$(PK, M\text{SK}) \leftarrow$ **Setup** (λ, n). Given a security parameter λ and n number of attributes, the *TA* picks random values $(\alpha, \beta, \gamma, a_1, \cdots, a_n, z) \in (\mathbb{Z}_p^*)^{n+4}$ and sets:

$$g_1 = g^{\gamma}, g_{2,1} = g^{a_1}, \cdots, g_{2,n} = g^{a_n}, g_0 = g^z \in \mathbb{G}_1$$

$$h_1 = h^{\gamma}, h_{2,1} = h^{a_1}, \cdots, h_{2,n} = h^{a_n}, h^{\alpha\beta} \in \mathbb{G}_2$$

to output the public key PK and master secret key MSK as:

$$PK = (g, g_0, g_1, g_{2,1}, \cdots, g_{2,n}, Y = e(g, h^{\alpha\beta})) \in \mathbb{G}_1^{n+3} \times \mathbb{G}_T,$$

$$M\text{SK} = (h^{\alpha\beta}, h, h_1, h_{2,1}, \cdots, h_{2,n}) \in \mathbb{G}_2^{n+3}.$$

$SK_{\vec{v}} \leftarrow$ **KeyGen** $(M\text{SK}, PK, \vec{v})$. Let $\vec{v} \in \mathcal{F}$ be the predicate the user who requests the corresponding private key $SK_{\vec{v}}$. Given the master secret key MSK and user's predicate $\vec{v}$ with PK, *TA* chooses random values $r, R \in \mathbb{Z}_p^*$.

It then computes the user's private key $SK_{\vec{v}} = (K_1, K_2, K_3, \vec{v}) \in \mathbb{G}_2^3 \times (\mathbb{Z}_p^*)^n$ as:

$$K_1 = h^{\alpha\beta} \prod_{i=1}^{n} (h_{2,i})^{v_i r} \, h_1^R \;\; , \;\; K_2 = h^r \;\; , \;\; K_3 = h^R.$$

$CT_{\vec{x}} \leftarrow$ **Encrypt** $(PK, \vec{x}, M)$. To encrypt a message $M \in \mathbb{G}_T$ under the attribute $\vec{x} \in \Sigma$, the data owner chooses a random $s \in \mathbb{Z}_p^*$ and outputs the ciphertext $CT = (C_0, C_1, C_{2,i}, C_3) \in \mathbb{G}_T \times \mathbb{G}_1^{n+2}$ for $1 \leqslant i \leqslant n$ as follows:

$$C_0 = M \cdot Y^s \;\; , \;\; C_1 = g^s \;\; , \;\; C_{2,i} = g_0^{x_i s} \, (g_{2,i})^s \;\; for \; 1 \leqslant i \leqslant n \;\; , \;\; C_3 = g_1^s.$$

$(RK_{\vec{v},\vec{w}}, \mathbb{C}) \leftarrow$ **Re-KeyGen** $(MSK, \vec{v}, \vec{w})$. On input MSK, the attribute vector $\vec{v}$, and $\vec{w}$, the TA outputs a re-encryption key $RK_{\vec{v},\vec{w}}$ for $\vec{w}$. The algorithm outputs $(RK_{\vec{v},\vec{w}}, \mathbb{C})$ as the following:

a. First, it picks a random value $d \in \mathbb{Z}_p^*$ and sets $h^{\alpha\beta d}$, where α, β are value randoms defined in the Setup algorithm. It then encrypts $h^{\alpha\beta d}$ under the vector $\vec{w}$ to obtain $\mathbb{C} \leftarrow$ Encrypt $(PK, \vec{w}, [h^{\alpha\beta d}])$ where $[h^{\alpha\beta d}]$ denotes the encoding of $h^{\alpha\beta d}$ as an element of $\mathbb{G}_{\mathbb{T}}$.

b. Then, it picks random values $r', R' \in \mathbb{Z}_p^*$ to compute the re-encryption key $RK_{\vec{v},\vec{w}} = \{RK_1, RK_2, RK_3\}$, where RK_1, RK_2 and RK_3 are computed as:

$$RK_1 = h^{\alpha\beta} \prod_{i=1}^{n} ((h_{2,i})^{v_i} h^{\frac{\alpha\beta d}{r'}})^{r'} h_1^{R'} \;\; , \;\; RK_2 = h^{r'} \;\; , \;\; RK_3 = h^{R'}.$$

$CT_{\vec{w}} \leftarrow$ **Re-Encrypt** $((RK_{\vec{v},\vec{w}}, \mathbb{C}), CT_{\vec{x}})$. On input $RK_{\vec{v},\vec{w}}$, $\mathbb{C}$ and the ciphertext $CT_{\vec{x}} = (C_0, C_1, C_{2,i}, C_3)$, it computes:

a. In the first step, the algorithm computes $\hat{\mathbb{C}}$ as follows:

$$\hat{\mathbb{C}} = \frac{e(\prod_{i=1}^{n}(C_{2,i})^{v_i}, RK_2)\cdot e(C_3, RK_3)}{e(C_1, RK_1)}$$

$$= e(\prod_{i=1}^{n}(C_{2,i})^{v_i}, RK_2)\cdot e(C_3, RK_3)\cdot e(C_1, RK_1)^{-1}$$

$$= e(\prod_{i=1}^{n}(g_0^{x_i s}(g_{2,i})^{s})^{v_i}, h^{r'})\cdot e(g_1^{s}, h^{R'})\cdot e(g^{s}, h^{\alpha\beta}\prod_{i=1}^{n}((h_{2,i})^{v_i}h^{\frac{\alpha\beta d}{r'}})^{r'}h_1^{R'})^{-1}$$

$$= e(\prod_{i=1}^{n}(g^{z x_i v_i s}g^{a_i v_i s}), h^{r'})\cdot e(g^{\gamma s}, h^{R'})\cdot e(g^{s}, h^{\alpha\beta}\prod_{i=1}^{n}h^{a_i v_i r'}h^{\alpha\beta d}h^{\gamma R'})^{-1}$$

$$= e(g^{zs<\vec{x},\vec{v}>}g^{<\vec{a},\vec{v}>s}, h^{r'})\cdot e(g^{\gamma s}, h^{R'})\cdot e(g^{s}, h^{\alpha\beta}h^{<\vec{a},\vec{v}>r'}h^{\alpha\beta d}h^{\gamma R'})^{-1}$$

$$= e(g^{zs<\vec{x},\vec{v}>}, h^{r'})\cdot e(g^{<\vec{a},\vec{v}>s}, h^{r'})\cdot e(g^{\gamma s}, h^{R'})\cdot e(g^{s}, h^{\alpha\beta})^{-1}$$

$$\cdot e(g^{s}, h^{<\vec{a},\vec{v}>r'})^{-1}\cdot e(g^{s}, h^{\alpha\beta d})^{-1}\cdot e(g^{s}, h^{\gamma R'})^{-1}$$

$$= e(g,h)^{zsr'<\vec{x},\vec{v}>}\cdot e(g,h)^{<\vec{a},\vec{v}>sr'}\cdot e(g,h)^{\gamma sR'}\cdot e(g,h)^{-s\alpha\beta}\cdot e(g,h)^{-<\vec{a},\vec{v}>sr'}$$

$$\cdot e(g,h)^{-s\alpha\beta d}\cdot e(g,h)^{-\gamma sR'} = e(g,h)^{zsr'<\vec{x},\vec{v}>}\cdot e(g,h)^{-s\alpha\beta}\cdot e(g,h)^{-s\alpha\beta d}.$$

b. In the second step, the algorithm outputs the re-encrypted ciphertext $CT_{\vec{w}} = (C_0',\ C_1',\ \hat{\mathbb{C}},\ C_3')$, where $C_0' = C_0,\ C_1' = C_1,\ C_3' = \mathbb{C}$ and $\hat{\mathbb{C}}$ as computed in Step a.

$M \leftarrow$ **Decrypt** $(SK_{\vec{v}}, CT_{\vec{x}})$. On input the private key $SK_{\vec{v}}$ and ciphertext $CT_{\vec{x}}$, the algorithm proceeds differently according to *Level-1/ Level-2 access*:

Level-1 access. If $CT_{\vec{x}}$ is an original well-formed ciphertext, then algorithm decrypts ciphertext $CT_{\vec{x}} = (C_0, C_1, C_{2,i}, C_3)$ using the private key $SK_{\vec{v}} = (K_1, K_2, K_3, \vec{v})$ to obtain message M as:

$$M \leftarrow C_0 \cdot \frac{e(\prod_{i=1}^{n}(C_{2,i})^{v_i}, K_2)\cdot e(C_3, K_3)}{e(C_1, K_1)} \in \mathbb{G}_T$$

Correctness. To see that correctness holds, let $CT_{\vec{x}}$ and $SK_{\vec{v}}$ be as above.

Then:

$C_0 \cdot e(\prod_{i=1}^{n}(C_{2,i})^{v_i}, K_2) \cdot e(C_3, K_3) \cdot e(C_1, K_1)^{-1}$

$= M \cdot e(g,h)^{\alpha\beta s} \cdot e(\prod_{i=1}^{n}(g_0^{x_i s}(g_{2,i})^s)^{v_i}, h^r) \cdot e(g_1^s, h^R) \cdot e(g^s, h^{\alpha\beta}\prod_{i=1}^{n}(h_{2,i})^{v_i r} h_1^R)^{-1}$

$= M \cdot e(g,h)^{\alpha\beta s} \cdot e(\prod_{i=1}^{n}(g^{z x_i v_i s} g^{a_i v_i s}), h^r) \cdot e(g^{\gamma s}, h^R) \cdot e(g^s, h^{\alpha\beta}\prod_{i=1}^{n} h^{a_i v_i r} h^{\gamma R})^{-1}$

$= M \cdot e(g,h)^{\alpha\beta s} \cdot e(g^{zs<\vec{x},\vec{v}>} g^{<\vec{a},\vec{v}>s}, h^r) \cdot e(g^{\gamma s}, h^R) \cdot e(g^s, h^{\alpha\beta} h^{<\vec{a},\vec{v}>r} h^{\gamma R})^{-1}$

$= M \cdot e(g,h)^{\alpha\beta s} \cdot e(g^{zs<\vec{x},\vec{v}>}, h^r) \cdot e(g^{<\vec{a},\vec{v}>s}, h^r) \cdot e(g^{\gamma s}, h^R) \cdot e(g^s, h^{\alpha\beta})^{-1}$

$\cdot e(g^s, h^{<\vec{a},\vec{v}>r})^{-1} \cdot e(g^s, h^{\gamma R})^{-1}$

$= M \cdot e(g,h)^{\alpha\beta s} \cdot e(g,h)^{zsr<\vec{x},\vec{v}>} \cdot e(g,h)^{<\vec{a},\vec{v}>sr} \cdot e(g,h)^{\gamma sR} \cdot e(g,h)^{-s\alpha\beta}$

$\cdot e(g,h)^{-<\vec{a},\vec{v}>sr} \cdot e(g,h)^{-\gamma sR} = M \cdot e(g,h)^{zsr<\vec{x},\vec{v}>}.$

Hence, if $<\vec{x},\vec{v}>=0$, then we can recover the message M.

Level-2 access (from here on referred to as Re-Decrypt). If $CT_{\vec{w}}$ is a re-encrypted well-formed ciphertext, then it is of the form $CT_{\vec{w}} = (C_0', C_1', \hat{\mathbb{C}}, C_3')$. The algorithm first decrypts C_3' using $SK_{\vec{v'}}$ to obtain $h^{\alpha\beta d}$ as:
$h^{\alpha\beta d} \leftarrow \mathsf{Decrypt}(SK_{\vec{v'}}, C_3')$.

Then, it computes $\bar{\mathbb{C}} = e(C_1', h^{\alpha\beta d}) = e(g^s, h^{\alpha\beta d}) = e(g,h)^{s\alpha\beta d}$ to recover the message $M \leftarrow C_0' \cdot \hat{\mathbb{C}} \cdot \bar{\mathbb{C}}$.

Correctness. To see that correctness holds, we have:

$C_0' \cdot \hat{\mathbb{C}} \cdot \bar{\mathbb{C}} = M \cdot e(g,h)^{s\alpha\beta} \cdot e(g,h)^{zsr'<\vec{x},\vec{v}>} \cdot e(g,h)^{-s\alpha\beta} \cdot e(g,h)^{-s\alpha\beta d}$
$\cdot e(g,h)^{s\alpha\beta d} = M \cdot e(g,h)^{zsr'<\vec{x},\vec{v}>}.$

If $<\vec{x},\vec{v}>=0$, then we can get the message M.

4.4.2 Security Results

In this section, we prove the security of our scheme under *DBDH* assumption.

Definition 4.6. The proposed scheme is selectively attribute-hiding secure against chosen-plaintext attacks (*CPA*) in the standard model under the *DBDH* assumption, if for all *PPT* adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following security game Γ_b is negligible in the security parameter.

- **Initialization.** $\mathcal{A}$ outputs two challenge attribute vectors $\vec{x_0^*}$ and $\vec{x_1^*} \in \Sigma$ to the challenger $\mathcal{B}$.

- **Setup.** $\mathcal{B}$ runs Setup (λ, n) algorithm and gives a public key PK to $\mathcal{A}$.

- **Phase 1.** $\mathcal{A}$ makes a polynomial number of queries as a block box as follows:

  (a) **Private key oracle ($\mathcal{O}_{KeyGen}$):** $\mathcal{A}$ submits an attribute vector $\vec{v}$, and the challenger returns $SK_{\vec{v}} \xleftarrow{R} \mathsf{KeyGen}\ (MSK, PK, \vec{v})$ to $\mathcal{A}$ if only $< \vec{x_0^*}, \vec{v} >=< \vec{x_1^*}, \vec{v} > \neq 0$ or $< \vec{x_0^*}, \vec{v} >=< \vec{x_1^*}, \vec{v} >$. Otherwise it outputs $\perp$.

  (b) **Re-encryption key oracle ($\mathcal{O}_{Re\text{-}KeyGen}$):** $\mathcal{A}$ submits $\vec{v}$ with new vector $\vec{w}$ and the challenger computes $RK_{\vec{v},\vec{w}} \xleftarrow{R} \mathsf{Re\text{-}KeyGen}$ $(MSK, \vec{v}, \vec{w})$, where $SK_{\vec{v}} \xleftarrow{R} \mathsf{KeyGen}\ (MSK, PK, \vec{v})$ if $< \vec{x_0^*}, \vec{v} >= < \vec{x_1^*}, \vec{v} > \neq 0$ or $< \vec{x_0^*}, \vec{v} >=< \vec{x_1^*}, \vec{v} >$. Otherwise it outputs $\perp$.

  (c) **Re-encryption oracle ($\mathcal{O}_{Re\text{-}Encrypt}$):** $\mathcal{A}$ submits $\vec{v}$ and $\vec{w'}$, and the ciphertext $CT_{\vec{x}}$ under an attribute vector $\vec{x}$, if $< \vec{x_0^*}, \vec{v} >= < \vec{x_1^*}, \vec{v} > \neq 0$ or $< \vec{x_0^*}, \vec{v} >=< \vec{x_1^*}, \vec{v} >$. Otherwise it outputs $\perp$.

- **Challenge.** For challenge query $(\vec{x_0^*}, \vec{x_1^*}, M_0, M_1)$ where M_0 and M_1 are equal in length. It is required that $M_0 = M_1$ if any private key on $\vec{v}$ satisfying the condition $< \vec{x_0^*}, \vec{v} >=< \vec{x_1^*}, \vec{v} >= 0$ the challenger randomly samples a bit $b \in \{0, 1\}$ and gives $\mathrm{CT}_{\vec{x_b^*}} \xleftarrow{R} \mathsf{Encrypt}(PK, M_b, \vec{x_b^*})$ and sends $\mathrm{CT}_{\vec{x_b^*}}$ to $\mathcal{A}$ where $\vec{x_b^*}$ is hidden.

- **Phase 2.** $\mathcal{A}$ may continue to request private key queries, re-encryption key queries and re-encryption queries subject to the same restrictions as before.

- **Guess.** $\mathcal{A}$ outputs a bit b' and succeeds if $b' = b$. Hence, we define the advantage $\mathcal{A}$ as:

$$Adv_{\mathcal{A}}^{IND\text{-}sAH\text{-}CPA}(\lambda) := |Pr[b' = b] - \tfrac{1}{2}|.$$

The *IPPRE* scheme is attribute-hiding (AH) if all polynomial-time adversaries have at most negligible advantage in the above game.

4.4.2.1 Proof of Security

Our proof proceeds by a sequence of games starting with the actual scheme stated in Definition 4.6. To this purpose, we argue that the games are indistinguishable to the adversary while preserving attribute-hiding properties.

Theorem 1. The proposed scheme is the *IND-sAH-CPA* model with an adversary ϵ_{IND} such that $\epsilon_{IND} \leq \epsilon_{DBDH}$.

Lemma 1. Let $\mathcal{A}$ be an adversary playing the *IND-sAH-CPA* attack game. Then, there exists an algorithm $\mathcal{B}$ solving *DBDH* problem such that:

$$|Pr[\mathcal{A}^{\Gamma_{b,0}} = 0] - Pr[\mathcal{A}^{\Gamma_{b,1}} = 0]| \leq Adv_{\mathcal{B}}^{DBDH},$$

where $\Gamma_{b,0}$ and $\Gamma_{b,1}$ are the games according to the chosen bit b that we will define as follows:

Let $C = (A, B, C_1, \ldots, C_n, D) \in \mathbb{G}_T \times \mathbb{G}_1^{n+2}$ denote the challenge ciphertext given to the adversary during two real attacks (Γ_0, Γ_1). Additionally, let R be a random element of $\mathbb{G}_T$.

- Game $\Gamma_{0,0}$: This game is the original security game, where the challenge attribute and message are $\vec{x_0}$ and M_0, respectively. $C = (A, B, C_1, \ldots, C_n, D)$.
- Game $\Gamma_{0,1}$: In this game, the element A of the ciphertext is changed to a random element R of $\mathbb{G}_T$. But the challenge attribute and message are the same as for $\Gamma_{0,0}$. Hence, $C = (R, B, C_1, \ldots, C_n, D)$.
- Game $\Gamma_{1,1}$: This game is almost the same as $\Gamma_{0,1}$ except that the challenge attribute and the message are $\vec{x_1}$ and M_1, respectively. $C = (R, B, C_1, \ldots, C_n, D)$.
- Game $\Gamma_{1,0}$: This game is almost the same as $\Gamma_{0,0}$ except that the challenge attribute and the message are $\vec{x_1}$ and M_1, respectively. $C = (A, B, C_1, \ldots, C_n, D)$.

$\Gamma_{0,0}$ and $\Gamma_{1,0}$ are the same as games Γ_0 and Γ_1 in Definition 4.6, respectively. Therefore,

$$Adv_{\mathcal{B}}{}^{IND-sAH-CPA}(\lambda) \leq |Pr[\mathcal{A}^{\Gamma_{b,0}} = 0] - Pr[\mathcal{A}^{\Gamma_{b,1}} = 0]|.$$

Proof of Lemma 1. Suppose $\mathcal{A}$ has the advantage ϵ in distinguishing games $\Gamma_{b,0}$ from $\Gamma_{b,1}$. We build an algorithm $\mathcal{B}$ that solves the *DBDH* problem in asymmetric pairing. $\mathcal{B}$ is given as input a random 7-tuple $(g, g^a, g^c, h, h^a, h^b, T)$ that is either sampled from $\mathcal{P}_{\mathcal{A}}$, where $T = e(g,h)^{abc}$ or from $\mathcal{R}_{\mathcal{A}}$ where T is uniform and independent in $\mathbb{G}_T$. Algorithm $\mathcal{B}$'s goal is to output "1" if $T = e(g,h)^{abc}$ and "0" otherwise. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective attribute game as follows:

- **Initialization.** $\mathcal{A}$ begins the selective attribute game by outputting two attribute vectors $\vec{x_0}, \vec{x_1} \in \Sigma$ that it intends to attack.
- **Setup.** $\mathcal{B}$ generates the system's parameters randomly choosing $z', \gamma', \delta, a'_1, \ldots, a'_n \in \mathbb{Z}_p^*$ to define the following parameters:

$$\vec{x} = \vec{x_b}, \quad g_0 = g^{z'} g^{a\delta}, \quad g_1 = g^{\gamma'},$$

$$g_{2,1} = g^{-a\delta x_1} g^{a'_1}, \ldots, g_{2,n} = g^{-a\delta x_n} g^{a'_n}, \quad Y = e(g^a, g^h),$$

  where $\alpha = a, \beta = b, \gamma = \gamma'$ and $a_1 = -a\delta x_1 + a'_1, \ldots, a_n = -a\delta x_n + a'_n$, $\mathcal{B}$ sends to $\mathcal{A}$ the pubic key $PK = (g, g_0, g_1, g_{2,1}, \ldots, g_{2,n}, Y)$ and keeps the corresponding master secret key $MSK = (h^{ab}, h, h_1 = h^{\gamma'}, h_{2,1} = h^{-a\delta x_1} h^{a'_1}, \ldots, h_{2,n} = h^{-a\delta x_n} h^{a'_n})$. Note that h^{ab} is unknown to $\mathcal{B}$.
- **Phase 1.** $\mathcal{A}$ issues a polynomial number of queries, once at a time.
  - **Private key query ($\mathcal{O}_{\boldsymbol{KeyGen}(\vec{v})}$):** suppose $\mathcal{A}$ requests for a private key corresponding to vector $\vec{v}$. We only consider the case where $< \vec{x_0^*}, \vec{v} > = < \vec{x_1^*}, \vec{v} > \neq 0$ according to our definition on security model, the message M_0 is equal to M_1 only if $< \vec{x_0^*}, \vec{v} > = < \vec{x_1^*}, \vec{v} >$. As a result, this game is the same as the game in Definition 4.6 and there is no difference between advantages in these two games for the adversary $\mathcal{A}$.

    In the case that $< \vec{x_0^*}, \vec{v} > = < \vec{x_1^*}, \vec{v} > \neq 0$, $\mathcal{B}$ picks randomly $r, R \in \mathbb{Z}_p^*$, and computes:

$$k_1 = \prod_{i=1}^{n} (h^{-a\delta x_i} h^{a'_i})^{v_i r} h^{\frac{a'_i v_i b}{\delta I}} h^{\gamma' R}, \ k_2 = h^r h^{b\frac{1}{\gamma I}}, \ k_3 = h^R,$$

    with $I = < \vec{x}, \vec{v} >$.

    The generated $SK_{\vec{v}} = (k_1, k_2, k_3, \vec{v})$ is a valid private key for $\vec{v}$.

To see this, let's consider $\tilde{r} = r + \frac{b}{\beta I} \in \mathbb{Z}_p^*$. We have:

$$\prod_{i=1}^{n} (h^{-a\delta x_i} h^{a'_i})^{v_i r} h^{\frac{a'_i v_i b}{\delta I}}$$

$$= \prod_{i=1}^{n} h^{-a\delta x_i v_i r} h^{ab\delta x_i v_i \frac{1}{\delta I}} h^{-ab\delta x_i v_i \frac{1}{\delta I}} h^{a'_i v_i r} h^{\frac{a'_i v_i b}{\delta I}}$$

$$= h^{ab} \prod_{i=1}^{n} (h^{-a\delta x_i} h^{a'_i})^{v_i (r + b\frac{I}{\delta I})} = h^{ab} \prod_{i=1}^{n} (h_{2,i})^{v_i \tilde{r}}.$$

Following the definition of KeyGen algorithm, the private key of vector $\vec{v}$ is defined as $(k_1, k_2, k_3, \vec{v})$ where $k_1 = h^{ab} \prod_{i=1}^{n} (h_{2,i})^{v_i \tilde{r}} {h_1}^R$, $k_2 = h^{\tilde{r}}$, *and* $k_3 = h^R$ with uniform and independent $\tilde{r}, R \in \mathbb{Z}_p^*$. The generated key $(k_1, k_2, k_3, \vec{v})$ matches the definition and is sent to $\mathcal{A}$ as a valid private key for the vector $\vec{v}$.

– **Re-encryption key query ($\mathcal{O}_{\textbf{\textit{Re-KeyGen}}\,(\vec{v},\vec{w})}$):** $\mathcal{A}$ submits $\vec{v}$ and an attribute vector $\vec{w}$ in a re-encryption key query. Again, we consider only the case where $< \vec{x_0^*}, \vec{v} > = < \vec{x_1^*}, \vec{v} >$. In this case $\mathcal{B}$ sends $\vec{v}$ to $\mathcal{O}_{KeyGen}$ oracle and obtains $SK_{\vec{v}} \xleftarrow{R} (k_1, k_2, k_3, \vec{v})$. In order to obtain a re-encryption key corresponding to $\vec{w}$, $\mathcal{B}$ randomly chooses $r', R' \in \mathbb{Z}_p^*$ and computes $RK_{\vec{v},\vec{w}}$ as:

$$RK_1 = h^{ab} \prod_{i=1}^{n} ((h_{2,i})^{r_i} h^{\frac{\alpha\beta d}{r'}})^{r'} {h_1}^{R'}, RK_2 = h^r, RK_3 = h^{R'}.$$

Moreover, it computes $\mathrm{CT}_{\vec{w}}$ corresponding to ciphertext $[h^{\alpha\beta d}]$, that is $\mathrm{CT}_{\vec{w}} = \mathsf{Encrypt}\,(PK, [h^{\alpha\beta d}], \vec{w})$. Finally, $\mathcal{B}$ sends the re-encryption key along with $\mathrm{CT}_{\vec{w}}$ to $\mathcal{A}$.

– **Re-encryption oracle ($\mathcal{O}_{\textbf{\textit{Re-Encrypt}}\,(\vec{v},\vec{w'},CT_{\vec{x}})}$):** for the same reason as in $\mathcal{O}_{KeyGen\,(\vec{v})}$ and $\mathcal{O}_{Re\text{-}encrypt\,(\vec{v},\vec{w})}$, we only consider the case where $< \vec{x_0^*}, \vec{v} > \neq < \vec{x_1^*}, \vec{v} >$, then $\mathcal{B}$ submits $< \vec{w'}, \vec{v} >$ to the re-encryption key query $\mathcal{O}_{Re\text{-}encrypt\,(\vec{v},\vec{w'})}$ and received the re-encryption key $RK_{\vec{v},\vec{w'}}$. Then, $\mathcal{B}$ proceeds according to the corresponding algorithm.

- **Challenge.** When $\mathcal{A}$ decides that Phase 1 is over, it sends $\mathcal{B}$ two messages M_0 and $M_1 \in \mathbb{G}_T$. Then, $\mathcal{B}$ picks a random bit b and responds with the following ciphertext:

$CT_{\vec{x}_b} = (M_b.T, g^c, (g^c)^{z' x_1 + a'_1}, \ldots, (g^c)^{z' x_n + a'_n}, (g^c)^{\gamma'})$.

Let $z = z' + a\delta$ and $s = c$. Since $(g^c)^{z'x_i+a'_i} = (g^c)^{z'x_i+a\delta x_i-a\delta x_i+a'_i} = (g^c)^{z'x_i+a\delta x_i}(g^c)^{-a\delta x_i+a'_i} = {g_0}^{x_i s}(g_{2,i})^s$ for $1 \neq i \neq n$, we have:
$CT_{\vec{x_b}} = (M_b.T, g^s, {g_0}^{x_1 s}(g_{2,1})^s, \ldots, {g_0}^{x_n s}(g_{2,n})^s, {g_1}^s)$, which is a valid encryption of M_b with attribute vector $\vec{x_b}$.

- If $\mathcal{A}$ sends two messages, M_0 and M_1 for *Level-2* challenge ciphertext with a challenge vectors $\vec{x_b}, \vec{x_b^*} = (x_1, x_2, \ldots, x_n)$ and $\vec{y}$. $\mathcal{B}$ randomly choose $b \in \{0, 1\}$ and a random $d \in \mathbb{Z}_N$ to output the ciphertext $CT_{\vec{w}} = (M_b.T, g^s, \hat{\mathbb{C}}, CT_{\vec{x}})$ along with $\mathbb{C}$ (the ciphertext of $[h^{abd}]$). Then, $\mathcal{B}$ computes $\mathbb{C}$ and $\hat{\mathbb{C}}$ as the following:

$$\mathbb{C} = (T \cdot [h^{abd}], g^c, (g^c)^{z'x_1+a'_1}, \ldots, (g^c)^{z'x_n+a'_n}, (g^c)^{\gamma'}),$$

$$\hat{\mathbb{C}} = \frac{e(\prod_{i=1}^{n}((g^c)^{z'x_i+a'_i})^{v_i} \cdot e(g^{c\gamma'}, h^{\tilde{R}})}{e(g^c, h^{ab}\prod_{i=1}^{n}(h_{2,i})^{v_i\tilde{r}}{h_1}^{\tilde{R}})^{-1}}$$

$$= e(g,h)^{cz\tilde{r}<\vec{x_b},\vec{v}>}.e(g,h)^{-abc}.e(g,h)^{-cabd}.$$

For the ciphertext $\mathbb{C}$, $\mathcal{B}$ simplicity sets $s = c$ and $z = z' + a\delta$. Therefore, for $1 \leq i \leq n$, we have $\mathbb{C} = ([h^{abd}] \cdot T, g^s, {g_0}^{x_1 s}(g_{2,1})^s, \ldots, {g_0}^{x_n s}(g_{2,n})^s, {g_1}^s)$ that is a valid ciphertext of $[h^{abd}]$ with an attribute vector $\vec{x_b^*}$. Since $< \vec{x_b}, \vec{v} > \neq 0$ for any private key query $\vec{v}$, the $e(g,h)^{cz\tilde{r}<\vec{x_b},\vec{v}>}$ is a random value in the adversary view.

- **Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $b = b'$, then $\mathcal{B}$ outputs "1" meaning that $T = e(g,h)^{abc} = e(g, h^{ab})^c$, i.e., when $\mathcal{B}$'s 7 tuple input is sampled from $\mathcal{P}_A$, then C is valid encryption of M under the attribute $\vec{x}$ initially chosen by the adversary. Thus $\mathcal{A}$ is playing game $\Gamma_{b,0}$. Otherwise it outputs "0" meaning that T is uniform and independent in $\mathbb{G}_T$ i.e., when $\mathcal{B}$'s 7-tuple input is sampled from $R_{\mathcal{A}}$, then $C = (B, C_1, \ldots, C_n, D)$ for a random R. In this case $\mathcal{A}$ playing game $\Gamma_{b,1}$.

  Hence, if $\mathcal{A}$ has an advantage ϵ in distinguishing game $\Gamma_{b,0}$ from game $\Gamma_{b,1}$, then $\mathcal{B}$ has the same advantage ϵ against *DBDH*.

4.4.3 Performance Evaluation

This section presents our evaluation results corresponding to the proposed efficient inner-product proxy re-encryption (*E-IPPRE*) scheme and the comparison with previous methods in terms of computation and communication overhead. We present both theoretical results and the experimental results.

4.4.3.1 Theoretical Results

Here, we analyze and compare our *E-IPPRE* scheme with previous attribute-based proxy re-encryption and inner-product encryption schemes [51, 52, 57, 10, 65, 67] in terms of the size of keys and ciphertexts, and computational overhead as the results can be seen respectively in Tables 4.4 and 4.5.

To the best of our knowledge, Backes' [67] scheme is the only one based on inner-product proxy re-encryption method. As shown in Table 4.4, their scheme provides a shorter ciphertext compared to the others because the length of ciphertext is independent of the length of attribute vector. Our protocol achieves a shorter private key size that is independent of the length of attribute vector.

As shown in Table 4.5, the computation overhead of the schemes in [51] and [52] increases linearly with the number of attributes. In these schemes the number of pairing operations increases with the number of attributes, resulting high computational overhead for the Decrypt and Re-Decrypt algorithms. Promising results have been achieved by Seo *et al.* [57] whose scheme requires a constant number of pairing operations for the Decrypt and Re-Decrypt algorithms and, therefore, reduces the computational cost compared to [51] and [52]. The inner-product scheme proposed in [10] is based on bilinear groups with a composite order ($N = pqr$). The length of its keys and ciphertexts are three times larger than others in low efficiency. The scheme of Okamoto *et al.* [65] introduced additional overhead due to the fact that it is based on dual pairing vector spaces. Therefore, its Encrypt and Decrypt algorithms take $O(n^2)$ and $O(n)$ pairing computations, respectively. Our proposed scheme requires a constant number of pairing operation for the Decrypt, Re-Encrypt, and Re-Decrypt algorithms.

As indicated in Table 4.5, the computation overhead of Backes *et al.*'s Encrypt algorithm is slightly better than the one of our scheme. However, Backes *et al.*'s scheme does not provide the attribute-hiding property.

From the above comparison in terms of storage and computational overhead, we conclude that our scheme is more efficient and secure compared to the ones shown in Table 4.4 and Table 4.5 because the private key size and the number of pairing operations are independent of the length of attribute vector. Hence, our Decrypt, Re-Encrypt, and Re-Decrypt algorithms require fixed pairing operations, which is more appropriate addressing the challenge of secure data sharing among multiple users in critical applications such as healthcare.

Scheme	**Public Key**	**Private Key**	**Ciphertext**
Liang *et al.* [51]	$(6n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n+1)\|\mathbb{G}\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Luo *et al.* [52]	$(N+2n+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4n+1)\|\mathbb{G}\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Seo *et al.* [57]	$(3n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|+3n\|\mathbb{Z}_p^*\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{Z}_p^*\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Katz *et al.* [10]	$(2n+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n+1)\|\mathbb{G}\|$	$(2n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Okamoto *et al.* [65]	$(n+2)(n+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+3)\|\mathbb{G}\|$	$(n+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Backes *et al.* [67]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+1)\|\mathbb{G}\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|+n\|\mathbb{Z}_p^*\|$
Ours	$(n+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+n\|\mathbb{Z}_p^*\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$

$|\mathbb{G}|$: bit length of element in $\mathbb{G}$, $|\mathbb{G}_\mathbb{T}|$: bit length of element in $\mathbb{G}_\mathbb{T}$, n: the number of attributes, N: the total number of possible values attributes, $|\mathbb{Z}_p^*|$: bit length of element in finite field $\mathbb{Z}_p^*$.

Table 4.4: Comparison of the size of keys and ciphertext

Scheme	Encrypt	Decrypt	Re-Encrypt	Re-Decrypt
Liang *et al.* [51]	$(n+2)E+E_T$	$(n+2)P$	$(n+1)P$	$(n+3)P$
Luo *et al.* [52]	$(n+2)E+E_T$	$(2n)P$	$(2n+1)P$	$(2n+1)P$
Seo *et al.* [57]	$(n+2)E+E_T$	$(3n+2)E+2P$	$(3n)E+2P$	$(3n)E+3P$
Katz *et al.* [10]	$(4n+1)E+E_T$	$(2n+1)P$	–	–
Okamoto *et al.* [65]	$(n+2)(n+3)E+E_T$	$(n+3)P$	–	–
Backes *et al.* [67]	$(n+3)E+E_T$	$nE+2P$	$(n-1)E+2P$	$nE+5P$
Ours	$(2n+2)E+E_T$	$nE+3P$	$nE+3P$	$nE+7P$

E: exponentiation in $\mathbb{G}$, E_T: exponentiation in $\mathbb{G}_\mathbb{T}$, P: pairing computation, n: number of attributes.

Table 4.5: Comparison of computation overhead

4.4.3.2 Experimental Results

In order to show the practical viability of our approach, we implemented our scheme in *C* using Pairing-Based Crypto (*PBC*) library [70]. The experiments permed on an Ubuntu machine with *16 GB RAM* and *Intel i7-4720HQ 2.60 GHz CPU*.

We measured the execution time of our scheme on 3 different types of elliptic curves with 80 bits of security level: SuperSingular (*SS*) curve (type *A*), *MNT* curves (type *D*) and Barreto-Naehrig (*BN*) curve (type *F*) [70] (parameters of each curve has indicated in Table 4.1). Table 4.6 shows the execution time of each algorithm of our scheme considering an increasing number of attributes from 5 to 30 over 100 runs. As shown in Table 4.6, the execution time of each algorithm increases linearly with the number of attributes.

The computation overhead of Encrypt algorithm is dominated by exponentiation operation on group $\mathbb{G}_1$. Hence, the elliptic curves with small based field size of $\mathbb{G}_1$ are more efficient than others and this results in shorter ciphertext size. As Table 4.6 indicates the curves *MNT159* and *SS* with the smaller and larger base field size of $\mathbb{G}_1$[4], respectively have the best and the worst encryption performance. More precisely, the time to encrypt five attributes using *MNT159* and *SS* curves is 7.9ms and 11ms, respectively. However, the computation overhead of KeyGen and Re-KeyGen algorithms is dominated by exponentiation operation on group $\mathbb{G}_2$. Table 4.6 shows that *BN* curve with the smaller base field size of $\mathbb{G}_2$[5] among other curves has better performance in terms of the execution time.

As we can see in Table 4.6, *SS* curve with embedding degree 2 and *BN* curve with embedding degree 12 have the lower and higher execution time, respectively for the algorithms Re-Eecrypt, Decrypt and Re-Decrypt, since the embedding degree of elliptic curves directly influences the size of $\mathbb{G}_\mathbb{T}$ and increases the complexity of pairing computation. Specifically, the Decrypt and Re-Decrypt algorithms take about 9ms and 12.6ms for the *SS* curve and about 9.2ms and 23.2ms respectively for the *MNT159* curve in the case of five attributes. For ten attributes these algorithms take about 11.5ms and 18.7ms for *SS* curve and 15ms and 25.3ms for *MNT159* curve.

[4]The base field size of $\mathbb{G}_1$ *MNT159*, *BN*, *MNT201* and *SS* curves are 159 bits, 160 bits, 201 bits and 512 bits, respectively [70].

[5]The base field size of $\mathbb{G}_2$ *BN*, *MNT159*, *SS* and *MNT201* curves are 320 bits, 477 bits, 512 bits and 603 bits, respectively [70].

Curve	SS			MNT159			MNT201			BN		
Attribute.num	**5**	**10**	**30**	**5**	**10**	**30**	**5**	**10**	**30**	**5**	**10**	**30**
Encrypt	11	20.3	50.3	7.9	11.8	27.5	9.7	14.8	34.8	10.2	18	42.6
KeyGen	5.5	6.8	12.2	12.9	14.3	19.8	16.4	17.8	23.2	4.5	5.8	11.3
Decrypt	9	11.5	20.5	9.2	15	28	12.6	15.6	28.9	50.9	52.9	62.1
Re-Encrypt	8.9	11.4	20.4	9.2	15	28.1	12.6	15.7	29	50.2	52.7	61.4
Re-KeyGen	21.1	30.8	70.3	34.2	38.4	59.6	44.7	51.5	77.3	14.5	19.7	40.8
Re-Decrypt	12.6	18.7	43	23.2	25.3	34.9	31	34.1	48	66.3	68.6	77

Table 4.6: Execution Time (ms)

4.5 Summary

In this chapter, we proposed two secure proxy-based techniques adopting inner-product encryption method (*IPE*) to share data among users with different ciphertext vector set. Our proposed schemes provide the proxy server with a transformation key with which a ciphertext associated with an attribute vector can be transformed to a new ciphertext associated with a different attribute vector. We showed that the proposed schemes preserve the confidentiality of the message and the attributes associated with the outsourced ciphertexts. We also showed that the second protocol *E-IPPRE* is more efficient than the first proposed *IPPRE* since in *E-IPPRE* only the attribute is computed as an exponent and not the predicate, this reduces the number of pairing computations. Therefore, adopting *E-IPPRE* scheme delivers a fast attribute vector update for sharing data. It requires a constant number of pairing operations for Decrypt, Re-Encrypt, and Re-Decrypt algorithms and ensures a short size of the public key, private key, and ciphertext. This advantage makes the scheme most efficient and practical compared to other proxy-based schemes in terms of computation and communication time.

Chapter 5

Secure Data Sharing in the Internet of Things

The Internet of things (IoT) is a modern computing and communication technology consisting a wide range of heterogeneous network devices (things/ objects) that allows us to generate and share data more easily and faster. In the IoT environment, billions devices interact with each other and co-operate with other devices and sensors to create new and innovative applications/services ranging from smart metering to remote health monitoring. The goal of IoT is to enable things to be connected anytime, anywhere with anything and anyone ideally using any path/network and any service [71].

According to the Cisco Internet Business Solutions Group ($IBSG$) [72], by 2020 around 50 billion IoT devices will be installed to the Internet, that is more than 6 devices for every human on the earth. Note that this estimate dose not take into account rapid advances in Internet or device technology. Therefore, by having billions of connected devices in the Internet, the future IoT will shift toward autonomous cyber-physical environments where the Internet and users are closely integrated and the operations of connected things are controlled by computer-based algorithms.

However, this massive connectivity in future will generate a huge amount of data that requires high performance computing capabilities and storage infrastructure for storing and real-time analytics/processing in an efficient and cost-effective way. All these requirements are not possible with the resource-constrained IoT devices. Therefore, cloud-based IoT (called as *Cloud* IoT) has emerged as a new paradigm that enables the resource-constrained IoT devices to be connected to the cloud through the Internet in order to get the benefits from cloud such as unlimited storage and processing capabilities with scalability and on-demand accessibility from anywhere.

The convergence of cloud and Internet of Things will provide new application services to end-users in the various areas such as smart homes, smart cities, smart grids, smart agriculture, smart transportation, smart medical and healthcare systems to improve all aspects of people's life. For instance, the *Cloud IoT* can be used in healthcare application where patients whose health status requires close attention can be constantly monitored using *IoT* devices. This requires sensors to collect physiological information of patients and uses gateway and cloud to analyze and store the information, and then send the analyzed data wirelessly to care practitioners for future analysis and remote monitoring. The major objective of this technique is to improve quality of patients care who need permanent monitoring to avoid unnecessary healthcare costs and the right medical support at the right time.

Moreover, the *Cloud IoT* enables sharing of data among things and users in order to achieve particular goal. In such a sharing environment, the information may contain personal and sensitive data that it is necessary to support anonymity and confidentiality of them. Therefore, the satisfaction of privacy and security requirements plays a fundamental role in the *Cloud IoT* environment. These requirements include data confidentiality, access control, security and privacy policies [73] that can be achieved using cryptographic encryption techniques such as public key encryption. However, traditional public key infrastructure is not appropriate for applications of *Cloud IoT* where data is shared among different entities, due to its drawbacks related to the key management and distribution. Therefore, to manage secure data sharing in these applications a flexible and fine-grained access control scheme should be used. Attribute-based encryption (*ABE*) scheme is a promising encryption method, which allows a data owner to encrypt his data without knowing the identity of all users and binds a fine-grained access control policy to his encrypted data to define access right over the data. In this scheme, the data is encrypted once regardless of the number of users in the system and only authorized users with the desired attributes are able to decrypt the data. Due to the expensive operations of *ABE* scheme, a pure *ABE* is not suitable for the resource-constrained *IoT* devices.

This issue can be dealt with by using a technique called *hybrid encryption*. Basically, a hybrid encryption scheme uses public key encryption techniques to derive a symmetric key that is then used to encrypt the data using standard symmetric key techniques [74]. Therefore, this technique provides the performance of the public key encryption scheme and the benefit of the symmetric key encryption scheme while enables strong security and low computational complexity. Hence, hybrid encryption techniques with high security, fast speed, and low memory requirements are more suitable for the

Cloud IoT environments.

The main goals of this chapter are to make the expensive *ABE* operations affordable to resource-constrained *IoT* devices and enhance the encryption speed by providing a secure cryptographic-based access control mechanism using a hybrid encryption scheme. We combine the flexibility and expressiveness of the proposed *E-IPPRE* (see Section 4.4) with the efficiency of symmetric key encryption technique and propose a light inner-product proxy re-encryption (*L-IPPRE*) to guarantee secure data sharing between different entities in the *Cloud IoT* environment.

The remainder of this chapter is structured as follows. Section 5.1 reviews some existing techniques regarding to employ *ABE* scheme on resource-constrained *IoT* devices. In Section 5.2, we present a proposed architecture in the *Cloud IoT* environment. Then, in Section 5.3, we propose a light inner-product proxy re-encryption scheme. Finally, Section 5.4 introduces a use case that the proposed protocol can apply on it.

5.1 ABE on Resource-constrained IoT Devices

With the advance of *IoT* devices, the feasibility of cryptographic techniques (e.g., attribute-based encryption) on resource-constrained devices has become an important issue in the context of privacy-preserving access control mechanisms. Attribute-based encryption (*ABE*) scheme is a promising technique that can be used to satisfy the privacy and security of user's data confidentiality and fine-grained access control in the *IoT*. However, its cost and complexity makes its implementation hard to employ on resource-constrained *IoT* devices. Therefore, obtaining acceptable performance in the Internet of Things by using *ABE* scheme is a great issue that recently has been studied by the research community [75, 76, 9].

In [75, 76], the authors measured the performance of *ABE* scheme on smartphones. Wang *et al.* [75] evaluated the performance of two major types of *ABE* schemes, key-policy attribute-based encryption (*KP-ABE*) [11] and ciphertext-policy attribute-based encryption (*CP-ABE*) [9], on a laptop and Android smartphone devices. They implemented these two *ABE* schemes with the Java library and analyzed their results in terms of execution time, data overhead, energy consumption, and *CPU*/memory usage. They concluded that the performance of *ABE* scheme is unacceptable on mobile platforms even for the lowest security level (mainly in terms of execution time and energy consumption).

Later, Ambrosing *et al.* [76] showed that this conclusion depends on the specific implementation provided in [75]. They presented an implementa-

tion of *KP-ABE* [11] and *CP-ABE* [9] schemes as a *C* library for Android smartphones and considered a comparative analysis similar to Wang *et al.*'s [75]. This implementation is considerably faster and provides better performance in terms of execution time, energy consumption and *CPU*/memory usage compared to [75]. Their resulting execution time for *KP-ABE* and *CP-ABE* schemes is significantly lower than obtained results in [75]. Their results prove that achieving acceptable performance for the *ABE* operations on Android smartphones and similar devices is feasible.

A seminal paper presented by Green *et al.* [77], improved the efficiency adopting an attribute-based encryption method for resource-constrained devices by outsourcing expensive decryption operations to the cloud. In their scheme, the ciphertexts are stored in the cloud and each user has two keys: a transformation key and an ElGamal-style key. The proposed scheme delegates the re-encryption capability to a semi-trusted proxy that can transform any ciphertext satisfied by user's attributes/access policy into an ElGamal-style ciphertext using the transformation key. The proxy learns nothing about the underlying plaintext and the associated attributes in the ciphertext. The ElGamal-style ciphertext is then transmitted to the user who can decrypt it with his ElGamal-style key. This manner incurs less computational overhead at the user side so that user can save significantly on both bandwidth and decryption time, without increasing the number of transmissions. Although this scheme increases the performance of *ABE* scheme on resource-constrained devices, it does not address computational load reduction for the encryption operation.

Later, Asim *et al.* [78] proposed a new *CP-ABE* scheme for resource-constrained devices with encryption and decryption outsourcing capabilities. The proposed scheme uses two independent semi-trusted proxies: one for outsourcing computationally expensive encryption operations (Proxy A) and another for outsourcing decryption operations (Proxy B). In the encryption process, the data owner creates a partial ciphertext which consists of an encrypted message, the encrypted private key for the policy creation and an access policy. Then this partial ciphertext is outsourced to the Proxy A, who encrypts the encrypted message according to the given access policy. In this way, the proxy learns nothing about the partially ciphertext. The decryption process is realized by deploying the idea of Green *et al.*'s scheme [77]. A user who wants to decrypt a ciphertext outsources the ciphertext and his transformation key to the Proxy B, who checks if the attributes in user's key satisfy the access policy associated with ciphertext, if yes, returns a partially decrypted ElGamal-style ciphertext that can be further decrypted by the user. This proposed scheme reduces significantly the computational

overhead at the data owner and user sides and it is suitable for applications where both the data owner and users are using resource-constrained devices (e.g., mobile devices).

Touati *et al.* [79] presented a cooperative approach for reducing the computational overhead of *ABE* operations on resource-constrained sensor nodes in the *IoT* environment by delegating expensive operations of encryption process to the unconstrained trusted neighbor nodes called "assistant nodes". For each sensor, there are at least two unconstrained devices in its neighborhood which assist the sensor during the encryption process. When a sensor wants to encrypt a message, it looks for its powerful neighbor nodes and sends to them the expensive operations and shares pairwise keys with them. Each assistant node shares pairwise keys with the remote server, then computes the blinding factor and some operations and encrypts them with the shared key. The encrypted blinding factor and other encrypted operations are sent to the sensor and server, respectively. When the sensor receives a response from the assistant nodes, it decrypts the blinding factor and uses it for hiding message and sends the result to the server after encrypted it with the shared key. On the other hand, when the server receives the encrypted operations from the assistant nodes, it decrypts them with the shared key and applies them to compute other operations. Finally, the ciphertext is constructed by using all these parts. Although, this proposed scheme reduces the computational overhead and energy consumption in the encryption process of *ABE* scheme, but data could be disclosed to unauthorized entities if some of the assistant nodes have been compromised.

5.2 Proposed Architecture in Cloud-based IoT

In this section, we provide a high-level view of our proposed architecture to address the challenge of secure data sharing between users with different access policies in the cloud-based *IoT*. The proposed architecture is scalable and able to store a large amount of data generated by sensors. Since these data are sensitive, we propose a new security mechanism basing on *ABE* scheme to guarantee data confidentiality, flexible and fine-grained access control, and scalable key management in the cloud-based *IoT*.

In order to achieve the aforementioned objectives, we propose the architecture described in Figure 5.1. This architecture including six entities *Data Source Devices*, *IoT Gateway Device*, *Data Storage Server*, *Trust Authority*, *Proxy Server*, and *Data Consumer Devices* works in four phases as follows:

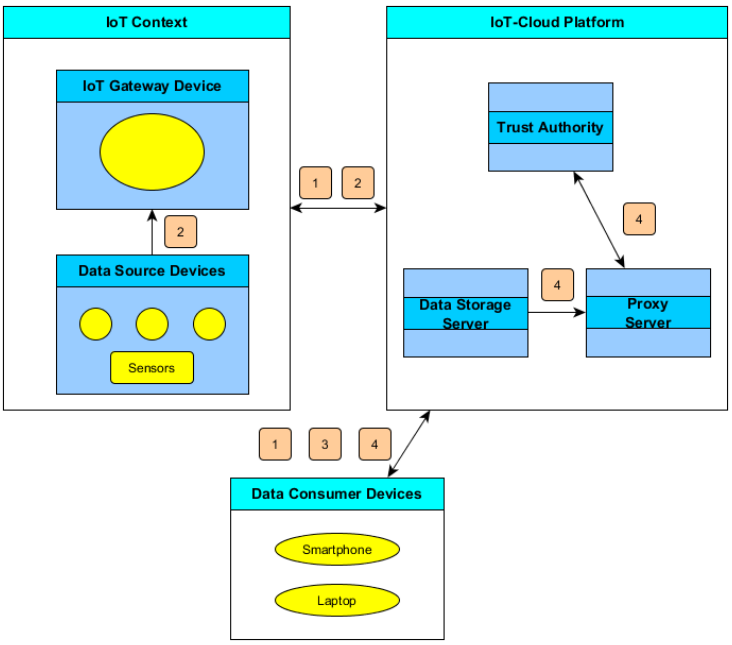


Figure 5.1: Overview of our proposed architecture

- **Phase 1. Initial configuration**

  This phase (Figure 5.1- ①) is run by *Trust Authority* who is responsible for attributes management and key issuing (e.g., private keys for *Data Consumer Devices* and re-encryption key for the *Proxy Server*). In this phase, the *IoT Gateway Device* and *Data Consumer Devices* communicate with the *Trust Authority* to obtain both public parameters and their private key, respectively. Firstly, the *Trust Authority* generates "master secret key" and "public key" by running the Setup algorithm, then distributes the public key and a set of attributes to the *IoT Gateway Device*, so that this entity can perform the Encrypt algorithm for protecting its data. Next, the *Trust Authority* runs the KeyGen algorithm to generate private keys for authorized data consumers based on their associated attributes and the "master secret key", then sends the private keys to the *Data Consumer Devices*.

- **Phase 2. Encrypt and Store Information**

  This phase (Figure 5.1- ②) is run by *IoT Gateway Device*, when the *Data Source Devices* are willing to store and share their data to the *IoT-Cloud Platform*. Since the *Data Source Devices* (e.g., sensors) are resource constraints in terms of energy consumption, processing, and

memory, they cannot directly perform expensive hybrid cryptography operations for protecting their data. Therefore, they sense data and send it to the powerful *IoT Gateway Device*. This entity encrypts the data using a hybrid encryption scheme which combines a key encapsulation mechanism (*KEM*) with a symmetric key encryption scheme (*SKE*). For this purpose, first, a key encapsulation mechanism generates a random "session key" underlying the *ABE* system and drives a 128-bit symmetric key by hashing the generated "session key" using the underlying key derivation function (*KDF*). Then, this mechanism uses the *ABE* scheme to "encapsulate" (encrypt) the generated symmetric key using the set of attributes and the public key provided by the *Trust Authority* in Phase 1. Next, an efficient data encapsulation mechanism (*DEM*) runs a symmetric key encryption scheme (e.g., *AES*) to encrypt the actual data using the generated symmetric key. Finally, both the "encapsulated symmetric key" and the "*AES*-encrypted data" form a ciphertext and are stored in the *Data Storage Server*.

- **Phase 3. Recover Ciphertext**

  In this phase (Figure 5.1- ③), the *Data Consumer Devices* communicate with the *Data Storage Server* to obtain the ciphertext. If the set of attributes associated with the "encrypted symmetric key" and the set of attributes associated with the "data consumer's private key" form orthogonal, the *Data Consumer Device* can recover the "session key". Then, this party recovers the 128-bit symmetric key by hashing the obtained "session key" using the same key derivation function applied in Phase 2. Finally, the *Data Consumer Device* is able to decrypt the original data using this symmetric key.

- **Phase 4. Re-encrypt Ciphertext**

  In some cases, it is necessary to share encrypted data among different data consumers with different access policy in the cloud-based *IoT*. To achieve this goal, in this phase (Figure 5.1- ④), the *Trust Authority* generates a re-encryption key for the *Proxy Server* who transforms a ciphertext associated with a set of attributes into a new ciphertext with the same plaintext but under a different attributes set without revealing the underlying plaintext and data consumer's private key. Then, the re-encrypted ciphertext will be sent to another data consumer who has different access policy. Finally, this data consumer can recover the message with his private key.

5.2.1 Assumptions

Our proposed architecture relies on the following assumptions:

- we assume that the *Data Storage Server* and *Proxy Server* are honest-but-curious. On one hand, the honest servers always follow their tasks to execute the required operations; on the other hand, the curious servers try to gain any information about the user's sensitive data. Note that the honesty feature is assumed to ensure service availability and data integrity, but not for the confidentiality of sensitive data.
- we assume that communication channels between all involved entities are secured by a security protocol such as *SSL* [1].
- we assume that *Data Source Devices* (e.g., sensors) are resource constraints and are not able to generate the keys and encrypt their data.
- we assume that the *IoT Gateway Device* is a powerful entity to perform expensive *ABE* operations.
- we assume that the *Trust Authority* is a fully trusted party.

5.3 Light Inner-product Proxy Re-encryption Scheme

In this section, we present a formal description of our light inner-product proxy re-encryption (*L-IPPRE*) scheme in the cloud-based *IoT*, which is attribute-hiding under the Asymmetric Decisional Bilinear Diffie-Hellman (*DBDH*) and $\mathcal{P}$-Asymmetric Decisional Bilinear Diffie-Hellman ($\mathcal{P}$-*DBDH*) assumptions (see Section 2.5). The scheme encrypts a message M as well as the attributes $\vec{x} \in \Sigma$, where we assume that $\Sigma = (\mathbb{Z}_p)^n$ for some positive integer n and $M \in \mathcal{M}$, where $\mathcal{M} = \mathbb{G}_T$.

5.3.1 The L-IPPRE Framework

Our proposed light inner-product proxy re-encryption (*L-IPPRE*) scheme for cloud-base *IoT* environment has six algorithms: Setup, KeyGen, Encrypt, Re-KeyGen, Re-Encrypt, and Decrypt.

[1] Secure Sockets Layer

5.3.1.1 Scheme

We are given a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ over a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ of prime order p with respective generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$. The size of p is determined by security parameter. The proposed scheme works as follows:

$(PK, M\text{SK}) \leftarrow$ **Setup** (λ, n). This algorithm is run by *Trust Authority*. On input a security parameter λ and the number of attribute n. It picks a random $(\alpha, \beta, a_1, \cdots, a_n, z) \in (\mathbb{Z}_p^*)^{n+4}$, and sets:

$$g_{1,1} = g^{a_1}, \cdots, g_{1,n} = g^{a_n}, g_0 = g^z \in \mathbb{G}_1,$$

$$h_{1,1} = h^{a_1}, \cdots, h_{1,n} = h^{a_n}, h^{\alpha\beta} \in \mathbb{G}_2.$$

Then, the Setup algorithm outputs the public key PK and master secret key MSK as:

$$PK = (g, g_0, g_{1,1}, \cdots, g_{1,n}, Y = e(g, h^{\alpha\beta})) \in \mathbb{G}_1^{n+2} \times \mathbb{G}_T,$$

$$M\text{SK} = (h^{\alpha\beta}, h, h_{1,1}, \cdots, h_{1,n}) \in \mathbb{G}_2^{n+2}.$$

$SK_{\vec{v}} \leftarrow$ **KeyGen** $(M\text{SK}, PK, \vec{v})$. This algorithm is run by the *Trust Authority*. Let $\vec{v} \in \mathcal{F}$ be the predicate for the *Data Consumer Device* which requests for the corresponding private key $SK_{\vec{v}}$ from the *Trust Authority*. Given the master secret key MSK and the *Data Consumer Device*'s predicate $\vec{v}$ with the public key PK, this entity chooses a random value $r \in \mathbb{Z}_p^*$. It then computes the *Data Consumer Device*'s private key $SK_{\vec{v}} = (K_1, K_2, \vec{v}) \in \mathbb{G}_2^2 \times (\mathbb{Z}_p^*)^n$ as:

$$K_1 = h^{\alpha\beta} \prod_{i=1}^{n} (h_{1,i})^{v_i r}, \quad K_2 = h^r.$$

$(CT_{\vec{x}}, CT_M) \leftarrow$ **Encrypt** $(PK, \vec{x}, M)$. This algorithm is run by the *IoT Gateway Device*. On input the public key PK, a vector $\vec{x} = (x_1, \ldots, x_n) \in (\mathbb{Z}_p^*)^n$, and a message $M \in \mathbb{G}_T$. The algorithm outputs a hybrid ciphertext $(CT_{\vec{x}}, CT_M)$ by combining a key encapsulation mechanism (KEM) with a symmetric key encryption (SKE) scheme as the following:

- **Key encapsulation mechanism (*KEM*).** This mechanism runs the KEM-Encryption algorithm to generate a 128-bit symmetric key and a

ciphertext $CT_{\vec{x}}$. The encryption algorithm $(K, CT_{\vec{x}}) \leftarrow \mathsf{KEM\text{-}Encrypt}\ (PK, \vec{x})$ takes the public key PK, along with the vector $\vec{x}$ as input, and outputs a key/ciphertext pair $(K, CT_{\vec{x}})$. First, this algorithm picks a random value $s \in \mathbb{Z}_p^*$ and computes the session key as: $D = Y^s = e(g,h)^{\alpha\beta s}$. Then, uses an underlying key derivation function (KDF) to derive a 128-bit symmetric key K by hashing the generated "session key" as: $K = KDF\,(e(g,h)^{\alpha\beta s})$. Next, it encapsulates the generated symmetric key by computing the ciphertext $CT_{\vec{x}} = (C_1, C_{2,i})\ for\ 1 \leqslant i \leqslant n$, as follows:

$$C_1 = g^s, \quad C_{2,i} = g_0^{x_i s}\ (g_{1,i})^s\ \ for\ 1 \leqslant i \leqslant n.$$

- **Symmetric key encryption scheme (*SKE*).** This scheme uses a symmetric key encryption scheme (e.g., AES) and encrypts the message M under the generated symmetric key. It runs the encryption algorithm $CT_M \leftarrow \mathsf{SKE\text{-}Encrypt}\ (K, M)$ and obtains the ciphertext CT_M.

$(RK_{\vec{v},\vec{w}}\ , \mathbb{C}) \leftarrow$ **Re-KeyGen** $(M\mathrm{SK}, \vec{v}, \vec{w})$. This algorithm is run by the *Trust Authority*. On input the master secret key MSK, the vector $\vec{v}$, and a vector $\vec{w}$. The algorithm outputs $(RK_{\vec{v},\vec{w}}\ , \mathbb{C})$ as the following:

a. First, it picks a random value $d \in \mathbb{Z}_p^*$ and sets $h^{\alpha\beta d}$, where α, β are value randoms defined in the Setup algorithm. It then encrypts $h^{\alpha\beta d}$ with the vector $\vec{w}$ to obtain $\mathbb{C} \leftarrow \mathsf{Encrypt}\ (PK, \vec{w}, [h^{\alpha\beta d}])$, where $[h^{\alpha\beta d}]$ denotes the encoding of $h^{\alpha\beta d}$ as an element of $\mathbb{G}_{\mathbb{T}}$.

b. Then, it calls the KeyGen algorithm and picks a random value $r' \in \mathbb{Z}_p^*$ to compute the re-encryption key $RK_{\vec{v},\vec{w}} = \{RK_1, RK_2\}$, where RK_1 and RK_2 are computed as below:

$$RK_1 = h^{\alpha\beta} \prod_{i=1}^{n} ((h_{1,i})^{v_i}\ h^{\frac{\alpha\beta d}{r'}})^{r'}, \quad RK_2 = h^{r'}.$$

$CT_{\vec{w}} \leftarrow$ **Re-Encrypt** $((RK_{\vec{v},\vec{w}}\ , \mathbb{C}), (CT_{\vec{x}}\ , CT_M))$. This algorithm is run by the *Proxy Server*. On input $(RK_{\vec{v},\vec{w}}\ , \mathbb{C})$ and the hybrid ciphertext $(CT_{\vec{x}}\ , CT_M) = ((C_1, C_{2,i}), CT_M)$, it computes:

a. In the first step, the algorithm computes $\hat{\mathbb{C}}$ as follows:

$$\hat{\mathbb{C}} = \frac{e(C_1, RK_1)}{e(\prod_{i=1}^{n}(C_{2,i})^{v_i}, RK_2)}$$

$$= e(g^s, h^{\alpha\beta}\prod_{i=1}^{n}((h_{1,i})^{v_i}h^{\frac{\alpha\beta d}{r'}})^{r'}) \cdot e(\prod_{i=1}^{n}(g_0^{x_i s}(g_{1,i})^s)^{v_i}, h^{r'})^{-1}$$

$$= e(g^s, h^{\alpha\beta}\prod_{i=1}^{n}h^{a_i v_i r'}h^{\alpha\beta d}) \cdot e(\prod_{i=1}^{n}(g^{z x_i v_i s}g^{a_i v_i s}), h^{r'})^{-1}$$

$$= e(g^s, h^{\alpha\beta}h^{(\vec{a},\vec{v})r'}h^{\alpha\beta d}) \cdot e(g^{zs(\vec{x},\vec{v})}g^{(\vec{a},\vec{v})s}, h^{r'})^{-1}$$

$$= e(g^s, h^{\alpha\beta}) \cdot e(g^s, h^{(\vec{a},\vec{v})r'}) \cdot e(g^s, h^{\alpha\beta d}) \cdot e(g^{zs(\vec{x},\vec{v})}, h^{r'})^{-1} \cdot e(g^{(\vec{a},\vec{v})s}, h^{r'})^{-1}$$

$$= e(g,h)^{\alpha\beta s} \cdot e(g,h)^{sr'(\vec{a},\vec{v})} \cdot e(g,h)^{\alpha\beta ds} \cdot e(g,h)^{-zsr'(\vec{x},\vec{v})} \cdot e(g,h)^{-sr'(\vec{a},\vec{v})}$$

$$= e(g,h)^{\alpha\beta s} \cdot e(g,h)^{\alpha\beta ds} \cdot e(g,h)^{-zsr'(\vec{x},\vec{v})}$$

b. In the second step, the algorithm outputs the re-encrypted ciphertext $CT_{\vec{w}} = (C'_0,\ C'_1,\ \hat{\mathbb{C}},\ C'_3)$, where $C'_0 = CT_M,\ C'_1 = C_1,\ C'_3 = \mathbb{C}$ and $\hat{\mathbb{C}}$ as computed in Step *a*.

$M \leftarrow$ **Decrypt** $(SK_{\vec{v}}, (CT_{\vec{x}}\ , CT_M))$. This algorithm is run by the *Data Consumer Devices*. On input the private key $SK_{\vec{v}}$ and the hybrid ciphertext $CT = (CT_{\vec{x}}\ , CT_M)$, the algorithm proceeds differently according to *Level-1* or *Level-2 access*:

- **Level-1 access.** If CT is an original well-formed ciphertext, then algorithm decrypts ciphertext $CT = (CT_{\vec{x}} = (C_1, C_{2,i}), CT_M)$ using the private key $SK_{\vec{v}} = (K_1, K_2, \vec{v})$ to output message M:

$$M \leftarrow \frac{e(C_1, K_1)}{e(\prod_{i=1}^{n}(C_{2,i})^{v_i}, K_2)} \in \mathbb{G}_T.$$

Correctness. To see that correctness holds, let CT and $SK_{\vec{v}}$ be as above. The Decrypt algorithm decrypts the message M as the following:

a. In the first step, it runs the decryption algorithm $K \leftarrow$ KEM-Decrypt $(SK_{\vec{v}}, CT_{\vec{x}})$ which takes the private key $SK_{\vec{v}}$ and the ciphertext $CT_{\vec{x}}$ as input and outputs the 128-bit symmetric key as below:

$$e(C_1, K_1) \cdot e(\prod_{i=1}^{n}(C_{2,i})^{v_i}, K_2)^{-1}$$

$$= e(g^s, h^{\alpha\beta}\prod_{i=1}^{n}(h_{1,i})^{v_i r}) \cdot e(\prod_{i=1}^{n}(g_0^{x_i s}(g_{1,i})^s)^{v_i}, h^r)^{-1}$$

$$= e(g^s, h^{\alpha\beta}\prod_{i=1}^{n}h^{a_i v_i r}) \cdot e(\prod_{i=1}^{n}(g^{z x_i v_i s} g^{a_i v_i s}), h^r)^{-1}$$

$$= e(g^s, h^{\alpha\beta} h^{(\vec{a},\vec{v})r}) \cdot e(g^{zs(\vec{x},\vec{v})} g^{(\vec{a},\vec{v})s}, h^r)^{-1}$$

$$= e(g,h)^{\alpha\beta s} \cdot e(g,h)^{(\vec{a},\vec{v})sr} \cdot e(g,h)^{-zsr(\vec{x},\vec{v})} \cdot e(g,h)^{-(\vec{a},\vec{v})sr}$$

$$= e(g,h)^{\alpha\beta s} \cdot e(g,h)^{-zsr(\vec{x},\vec{v})}.$$

If $(\vec{x},\vec{v}) = 0$, then $e(g,h)^{\alpha\beta s}$, which is equal to the "session key" from the Encrypt algorithm. Then, it uses the same key derivation function applied in the Encrypt algorithm to derive a 128-bit symmetric key K by hashing the "session key" as: $K = KDF(e(g,h)^{\alpha\beta s})$.

b. In the second step, the message M will be decrypted under the generated 128-bit symmetric key using the decryption algorithm $M \leftarrow \mathsf{SKE\text{-}Decrypt}(K, CT_M)$.

- **Level-2 access** (from here on referred to as Re-Decrypt). If $CT_{\vec{w}}$ is a re-encrypted well-formed ciphertext, then it is of the form $CT_{\vec{w}} = (C'_0, C'_1, \hat{\mathbb{C}}, C'_3)$. The algorithm first decrypts C'_3 using $SK_{\vec{v}'}$ to obtain $h^{\alpha\beta d}$ as:

$$h^{\alpha\beta d} \leftarrow \mathsf{Decrypt}(SK_{\vec{v}'}, C'_3).$$

Then, it computes $\bar{\mathbb{C}} = e(C'_1, h^{\alpha\beta d}) = e(g^s, h^{\alpha\beta d}) = e(g,h)^{\alpha\beta ds}$ to recover the message as:

$$M \leftarrow \frac{\hat{\mathbb{C}}}{\bar{\mathbb{C}}}$$

Correctness. The Decrypt algorithm decrypts the message M as the following:

a. In the first step, the 128-bit symmetric key recovers as:

$$\begin{aligned}\hat{\mathbb{C}} \cdot \bar{\mathbb{C}} &= e(g,h)^{\alpha\beta s} \cdot e(g,h)^{\alpha\beta ds} \cdot e(g,h)^{-zsr'(\vec{x},\vec{v})} \cdot e(g,h)^{-\alpha\beta ds} \\ &= e(g,h)^{\alpha\beta s} \cdot e(g,h)^{-zsr'(\vec{x},\vec{v})}.\end{aligned}$$

If $(\vec{x}, \vec{v}) = 0$, then $e(g,h)^{\alpha\beta s}$, which is equal to the "session key" from the Encrypt algorithm. Then, it uses the same key derivation function applied in the Encrypt algorithm to derive a 128-bit symmetric key K by hashing the "session key" as: $K = KDF\left(e(g,h)^{\alpha\beta s}\right)$.

b. In the second step, the message M will be decrypted under the generated 128-bit symmetric key using the decryption algorithm $M \leftarrow \mathsf{SKE\text{-}Decrypt}\ (K, CT_M)$.

5.3.2 Security Analysis

In this section, we analyze the security of the proposed light inner-product proxy re-encryption scheme.

5.3.2.1 Security Model

Our architecture for sharing data between different users with different access policy in the *Cloud IoT* environment is composed of many devices (e.g., *Data Source Devices*, *IoT Gateway Device*, and *Data Consumer Devices*), *Data Storage Server*, *Trust Authority*, and *Proxy Server*. We assume that the communication channels between thesis entities are secured by a security protocol such as *SSL*. Although *SSL* guarantees data confidentiality and integrity during outsourcing data, we have to encrypt data at the users'devices because we assume that the servers (e.g., *Data Storage Server* and *Proxy Server*) are honest-but-curious. Moreover, we consider that the *Data Storage Server* and *Proxy Server* might collude with some malicious or revoked users for illegally accessing data. In our architecture, the *Trust Authority* is responsible for key issuing and access policies management, Therefore, we consider that this entity is trusted and secured. Finally, we assume that the *IoT Gateway Device* and *Data Consumer Devices* have a public/private key pair and the public key can be easily obtained by other entities.

5.3.2.2 Security Services

Our architecture guarantees fine-grained access control, integrity, and confidentiality during outsourcing data with the secure *SSL* protocol.

The data is encrypted by a randomly generated 128-bit symmetric key using a standard symmetric key encryption scheme (e.g., *AES*) and this key is encrypted by the proposed public key encryption scheme, *E-IPPRE*

scheme, which has been proved secure under *DBDH* assumption in Section 4.4.2. Then the *Data consumer Device* decrypts the generated 128-bit symmetric key, using the *E-IPPRE* scheme, and decrypts the data using this key. The combination of encrypted techniques provides the efficiency of symmetric key encryption scheme with the performance of public key encryption scheme. The public key encryption techniques rely on expensive mathematical computations making them more inefficient and slow. On the other hand, the symmetric key encryption techniques provide faster encryption/decryption processes, hight security, and low memory requirements. Therefore, we improve the security and performance of our proposed *L-IPPRE* scheme by combining the proposed *E-IPPRE* scheme with the symmetric encryption scheme (e.g., AES).

Especially, the proposed *L-IPPRE* scheme is resistant against collusion attacks and ensures that encrypted data cannot be accessed by unauthorized users. From this, we deduce that the random symmetric key is confidential and can be accessed only by authorized users. Consequently, the data confidentiality is guaranteed by the standard symmetric encryption security.

5.3.3 Performance Evaluation

In this section, we evaluate the performance of the proposed lightweight inner-product proxy re-encryption scheme and compares it with the previous proposed protocols, *IPPRE* and *E-IPPRE* in terms of computation and communication overheads.

5.3.3.1 Theoretical Results

We theoretically analyze *L-IPPRE* scheme and compare it with the previous proposed protocols (i.e., *IPPRE* and *E-IPPRE*) in terms of the size of keys and ciphertext and computation overhead.

As shown in Tables 5.1 and 5.2, all result values despond on the number of attributes n in the system. Due to this purpose, we can not make communication size, encryption time, and decryption time a constant. Hence, the best way to improve performance is to make the coefficient of n as small as possible.

As the comparison results indicate from Table 5.2, the *IPPRE* scheme requires at least four times higher computation overhead than *E-IPPRE* and *L-IPPRE*. Moreover, the *IPPRE* scheme requires $(4n + 2)$ pairing computations, which is not suitable for practical scenarios, while *E-IPPRE* and *L-IPPRE* protocols need a constant number of pairing operations. On the

other hand, the *L-IPPRE* scheme is relatively efficient than *E-IPPRE* because it needs just two pairing operations for Decrypt and Re-Encrypt algorithms. As indicated in Table 5.1, *L-IPPRE* has the shorter size of the public key, the private key, and the ciphertext compared to the other proposed schemes. Observing these results, *L-IPPRE* scheme is considered as an efficient one where almost all coefficient of the values are *one* and the private key size and the number of pairing operations are independent of the number of attributes, it needs only *two* pairing operations in the decryption process.

Scheme	**Public key**	**Private Key**	**Cipertext**
IPPRE	$(8n+6)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4n+2)\|\mathbb{G}\|$	$(4n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
E-IPPRE	$(n+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+n\|\mathbb{Z}_p^*\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
L-IPPRE	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+n\|\mathbb{Z}_p^*\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$

$|\mathbb{G}|$: Bit length of element in $\mathbb{G}$, $|\mathbb{G}_\mathbb{T}|$: Bit length of element in $\mathbb{G}_\mathbb{T}$, n: number of attributes,N: the total number of possible values for attributes, $|\mathbb{Z}_p^*|$: Bit length of element in finite field $\mathbb{Z}_p^*$.

Table 5.1: Comparison of the size of keys and ciphertext

Scheme	**Encrypt**	**Decrypt**	**Re-Encrypt**	**Re-Decrypt**
IPPRE	$(12n+2)E+E_T$	$(4n+2)P$	$(4n+2)P$	$(4n+3)P$
E-IPPRE	$(2n+2)E+E_T$	$nE+3P$	$nE+3P$	$nE+7P$
L-IPPRE	$(2n+1)E+E_T$	$nE+2P$	$nE+2P$	$nE+6P$

E: exponentiation in $\mathbb{G}$, E_T: exponentiation in $\mathbb{G}_\mathbb{T}$, P: pairing computation, n: number of attributes.

Table 5.2: Comparison of computation overhead

5.3.3.2 Experimental Results

We implemented our scheme in *C* using the Pairing-Based Crypto (*PBC*) library [70]. The experiments were carried out on Ubuntu 16.04 LTS with *2.60 GHz 8x Intel(R) Core(TM) i7-4720HQ CPU* and *16 GB RAM.*

We tested the execution time of our *L-IPPRE* on 3 different types of elliptic curves with 80 bit of security level: SuperSingular (*SS*) curve (type *A*), *MNT* curves (type *D*) and Barreto-Naehrig (*BN*) curve (type *F*) [70] (parameters of each curve has indicated in Table 4.1). Table 5.3 shows the execution time of each algorithm of our scheme considering an increasing

Curve	SS			MNT159			MNT201			BN		
Attribute.num	**5**	**10**	**30**	**5**	**10**	**30**	**5**	**10**	**30**	**5**	**10**	**30**
Encrypt	9	18	45.2	6.5	9.5	25	7	12	32	8.2	16	40
KeyGen	2.8	4	9.4	5.4	6.5	12	6.5	7.6	13.4	2.4	3.7	9
Decrypt	5.2	9.5	17	7	10	18.8	10.7	12.9	25.2	32.6	34.7	43.5
Re-Encrypt	6.3	9	16	7	10	18.8	9.4	12	25	32.2	35.5	43.2
Re-KeyGen	17.2	27	66.3	25.9	32.2	51.9	34.7	40.6	67	12	17.6	38
Re-Decrypt	10.4	16.6	38	20.7	22.3	30.7	28	30.7	44	49	50.2	58

Table 5.3: Execution Time (ms)

number of attributes from 5 to 30 over 100 runs. As shown in Table 5.3, the execution time of each algorithm increases linearly with the number of attributes.

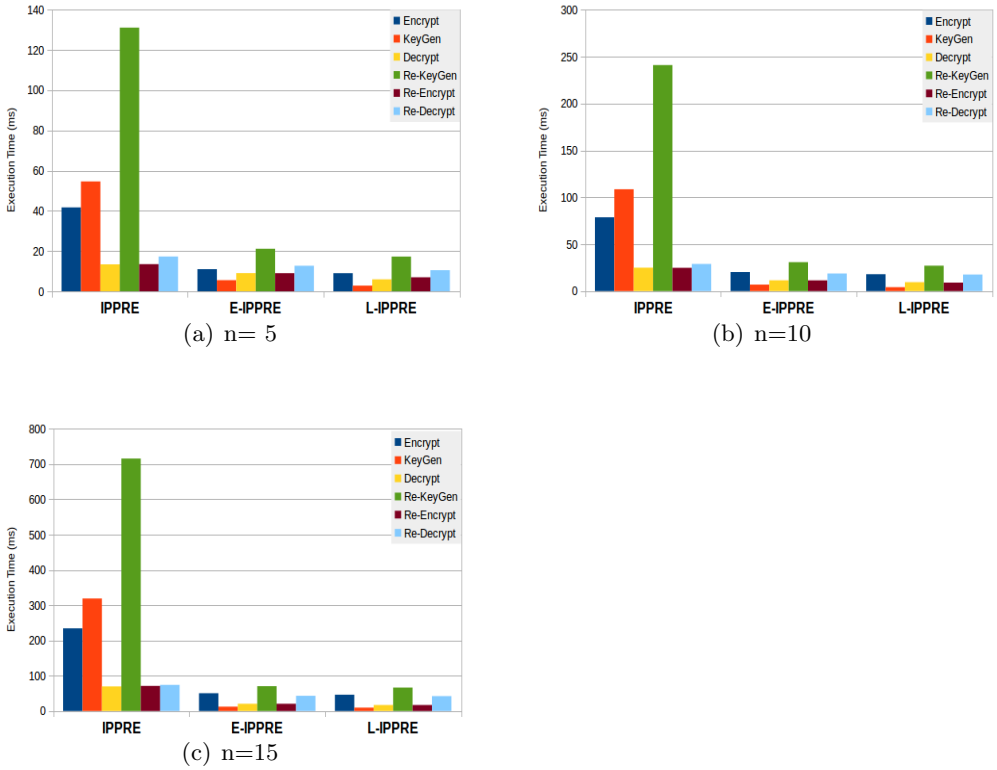
According to the benchmark of *PBC* library [70], elliptic curves with the group size $l = 512$ and the embedding degree $k = 2$ results in the fastest bilinear pairing as compared to those with $k > 2$ for *SS* curves. The case is on the opposite for *MNT* curves. Based on our simulation, the execution time of one pairing operation is about 1.41ms for the *SS* curve with $l = 512$ and $k = 2$, while for the *MNT159* curve with $l = 159$ and $k = 6$ and *BN* curve with $l = 160$ and $k = 12$ it takes about 2.8ms and 16ms, respectively. Therefore, we believe that *SS* curve is suitable for our proposed protocol.

As we can see in Table 5.3, for five attributes, the Decrypt and Re-Decrypt algorithms take about 5ms and 11ms for the *SS* curve and about 7ms and 22ms respectively for the *MNT159* curve. For ten attributes, these algorithms take about 9ms and 17ms for the *SS* curve and 10ms and 24ms for the *MNT159* curve.

In Figure 5.2, we compare the computation time of *IPPRE*, *E-IPPRE*, and *L-IPPRE* schemes for each algorithm on *SS* curve with 80 bit of security level, where we assume n is set to 5, 10, 15 attributes, respectively. As we can see in Figure 5.2, the *IPPRE* protocol takes a significant amount of time for Decrypt and Re-Decrypt algorithms because it requires $(4n + 2)$ and $(4n+3)$ pairing computations, respectively. While these algorithms take less time for *E-IPPRE* and *L-IPPRE* protocols due to require constant pairing operations in their Decrypt and Re-Decrypt algorithms (according to Table 5.2). The *IPPRE* requires about three times more time than *L-IPPRE*. Our the lightweight *IPPRE* scheme takes the shortest amount of time in its Decrypt, Re-Encrypt, and Re-Decrypt algorithms due to low coefficient values of the leading terms and constant pairing operations.

From Figure 5.2 and Table 5.3, we can conclude that our lightweight *IPPRE* scheme is the most efficient and practical compared to *IPPRE* and

E-IPPRE by comparison in terms of communication and computation overhead.



(a) n= 5 (b) n=10

(c) n=15

Figure 5.2: Comparison of the computation time between three proposed protocols

5.4 Use Case: EVOTION

The above architecture has been specifically conceived to address privacy and security issues for sharing data in the *EVOTION*[2] project.

The goal of *EVOTION* is to develop an integrated platform incorporating a big data (Hearing Aids (*HA*), sensors, mobile application) to collect and analyze data related to hearing loss (*HL*) patients to improve and manage *HL* treatments including: (i) to increase *HA* usage efficiency; (ii) to improve access of *HA* users to audiology care services and reduce the cost; (iii) to improve the wider health system for the perspective of *HL*, the safety of *HL* patients, policies for improving access to *HA* related health services for *HL* patients. However, the use of mobile health technologies arises sig-

[2]A research European proposal for big data supporting pubic health policies.

nificant security concerns related to confidentiality and privacy of personal health data.

5.4.1 Data repository

There are different data sources for feeding *HL* data into *EVOTION* framework as shown in Figure 5.3 [3]:

- **Existing Clinical Repositories:** including personal, medical, and occupational data already available from the clinical partners of the *EVOTION* framework.
- **Enhanced Hearing Aids (*HAs*):** enabling the capture and provision of *HA* usage-related data (e.g., rating of *HA* ease or difficulty of use in different listening conditions, frequency and type adjustments of *HA* controls).
- **Wearable device:** supporting the collection of real-time contextual *HA* user physiological data (e.g., heart rate, blood pressure, skin conductance).
- **A mobile application:** with components supporting the acquisition and transmission of behavioral (e.g., recording of *HA* user daily activities such as participation in conversations, watching TV), contextual (e.g., *HA* user's location), cognitive (e.g., verbal reaction time) data as well as the notification and acceptance/rejection of decisions by the *HA* user and/or their carers (decision selection component), and the execution of periodic audiological and cognitive to collect the related data (audiological and cognitive test components).

5.4.2 System Model Overview

Here, we extend our system model to *EVOTION* framework for adopting *L-IPPRE* scheme presented for IoT environments. As shown in Figure 5.4, the system model is composed of the following entities:

- **HL Patient**, who is equipped with *EVOTION HAs*, sensors, and a smartphone which acts as "IoT Gateway" and has *EVOTION* mobile application.

[3] This figure has borrowed from EVOTION Manual Report.

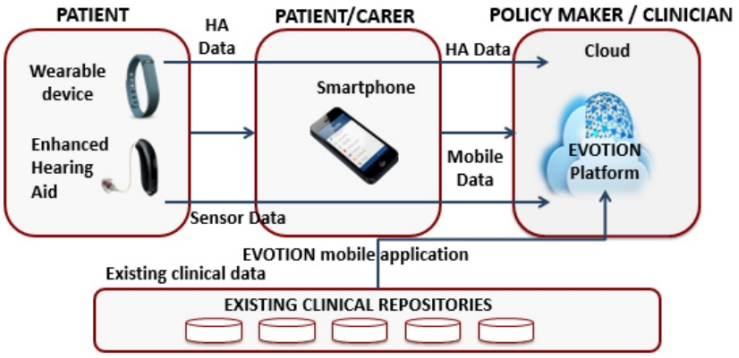


Figure 5.3: Physical architecture of EVOTION

- **Clinicians**, who are able to connect to the *EVOTION* platform and access to the *HL Patient*'s data and send existing and periodically update their encrypted reports or diagnostics (medical data).
- **Honest-but-curious Cloud**, which contains different types of data (*HA* usage, noise episodes, audiological, physiological, clinical, and medical data) provided by five different organizations (4 large hospitals and 1 *HA* manufacturer) and real-time data produced by sensors and *HA* used by *HL* patients in the encrypted form.
- **Trust Authority**, which is responsible for key issuing and attributes management.
- **Proxy server**, which enables to re-encrypt the outsource data under a different access policy.

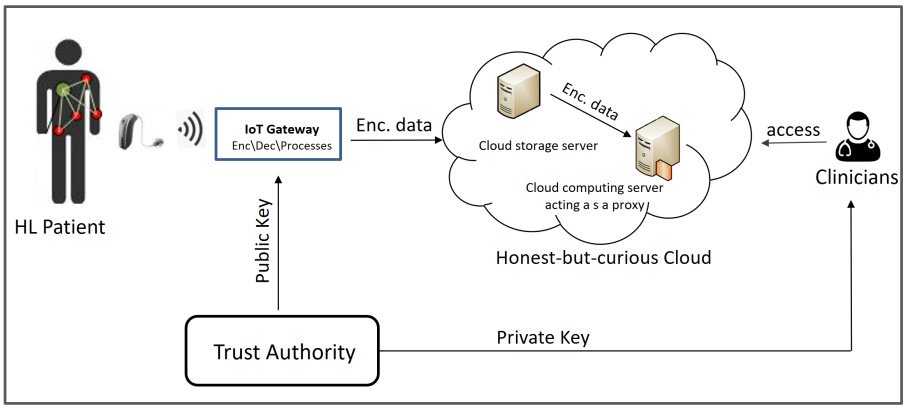


Figure 5.4: The system model developed to *EVOTION* framework

In our system, the *HL patient* is the data owner whose health status is monitored by a group of wearable sensors and *HAs*. The sensors gather sensitive data and send it to an *IoT Gateway* that forwards the data to the *Cloud storage server*. Since the data is sensitive it must be protected from unauthorizes users adopting lightweight encryption techniques (e.g., *L-IPPRE*). The keys of our system are provided by a *Trust Authority*, as key generation center. Each user (*Clinicians*) receives its corresponding key from the *Trust Authority* that holding the master key for the system. The *HL patient* sends data to the *IoT Gateway* (smartphone, *EVOTION* application), which aggregates the periodically collected data. The *Clinicians* act as the data users that provide health-care services for the *HL patient* by querying and retrieving his/her encrypted health data from the *Cloud storage server*. There are some cases where the *Clinicians* need to share the *HL patient*'s medical data with a group of care professionals in another hospital holding a different access policy. In this situation, the *Proxy server* gets a re-encryption key from the *Trust Authority* to update the *Clinicians*' access policy by re-encrypting the original encrypted medical data.

Chapter 6

Conclusion and Future Works

In this chapter, we conclude this thesis by summarizing our contributions and discussing directions for future work.

6.1 Conclusion

The major contribution of this thesis is to address the issue of secure data sharing between users with different access policies while providing the privacy and security of user's data confidentiality and fine-grained access control policies in cloud computing and Internet of Things. We investigated the challenge pertained to this issue and realized that one of the best solutions is to embed secure mechanisms into the data itself and the keys instead of relying on secure communication channels or a trusted third party for enforcing data access policies. We address this concern by designing encryption schemes based on attribute-based encryption (ABE) technique. An ABE scheme provides data confidentiality and enforces fine-grained access control policies over the encrypted data, while at the same time supporting scalability requirement by eliminating the need for sharing private keys between senders and recipients. Thus, Attribute-based encryption schemes have desirable functionality, but a common limitation of these schemes is that the attributes associated with a ciphertext do not conceal; therefore, these schemes do not satisfy the *attribute-hiding* property. According to our knowledge, this issue has not yet been addressed satisfactorily in previous secure data sharing schemes. Therefore, to provide a full-fledged cryptographic basis for secure data sharing on untrusted cloud computing storage, we proposed two inner-product proxy re-encryption schemes (namely inner-

product proxy re-encryption (*IPPRE*) and efficient inner-product proxy re-encryption (*E-IPPRE*)) that are secure under well-known standard assumptions and satisfy the *attribute-hiding* property. In our proposed protocols, we particularly considered practical application scenario, healthcare application where we assume a semi-trusted proxy is available. With this assumption, we combined attribute-based encryption scheme with a cryptographic technique known as proxy re-encryption (*PRE*) and enabled the authority to delegate most laborious tasks to the proxy server.

In the first proposed protocol, we proposed an *IPPRE* scheme derived from a well-known inner-product encryption scheme [1]. This protocol delegates the re-encryption capability to a semi-trusted proxy who transforms a ciphertext under an access policy to a new ciphertext with the same plaintext but under another access policy without learning the underlying plaintext and the associated attributes in the ciphertext as well as the private key. Our proposed scheme is provably secure against chosen-plaintext attacks in the standard model under Decision Bilinear Diffie-Hellman (*DBDH*) and Decision Linear (D-Linear) assumptions. We analyzed the proposed protocol in terms of computation, communication and storage overhead and we also tested the execution time of its algorithms on three different types of elliptic curves and achieved some encouraging experimental results. The execution times hint that our protocol is the first step towards a promising direction. To improve the performance of *IPPRE* protocol in terms of storage, computation, and communication costs, we proposed an efficient inner-product proxy re-encryption scheme (namely *E-IPPRE*) based on the inner-product encryption (*IPE*) method in [2]. In this proposed scheme, the number of pairing operations used for Decrypt, Re-Encrypt, and Re-Decrypt algorithms are constant and do not depend on the number of attributes in the system. Furthermore, it reduces the length of the public key, private key, ciphertext and time needed for key/re-key generation because each predicate is not computed as the exponent of a group element. The proposed protocol is proven selective attribute-secure against chosen-plaintext attacks in the standard model under two assumptions, Asymmetric Decisional Bilinear Diffie-Hellman and $\mathcal{P}$-Asymmetric Decisional Bilinear Diffie-Hellman. The execution time of each algorithm of our protocol was measured on three different types of elliptic curves. The experimental results show that the proposed scheme is the most efficient and practical compared to first protocol in terms of storage, computation, and communication.

The above proposed *IPPRE* protocols support the evaluations of polynomials, disjunctions, conjunctions, *CNF* formulas, *DNF* formulas, and threshold, which are appropriate for applications that require more fine-

grained access control to encrypted data. Moreover, they are suitable for dynamic environments where the access policy for controlling access to the encrypted data changes frequently (e.g. healthcare application). Furthermore, they satisfy attribute-hiding, fine-grained access control, and collision resistance properties.

In the future Internet of Things (IoT) billion devices can be connected with each other and cooperate with other devices to share their data. However, the connected billion devices on the Internet will generate a huge amount of sensitive data that requires privacy and security requirements including data confidentiality, access control, privacy, and enforcement of security and privacy policies. Therefore, to achieve secure data sharing in the IoT environment while preserving the privacy and security of user's data confidentiality, a flexible and fine-grained access control techniques such as public-key encryption schemes (e.g., ABE) should be used. The main challenge, in this case, is that the most existing ABE schemes are expensive for resource-constrained IoT devices due to the expensive pairing operations and the number of such operations increases with the number of attributes in the system. To tackle this issue, we combined the flexibility and expressiveness of the proposed *E-IPPRE* scheme with the efficiency of symmetric key encryption technique and proposed a lightweight inner-product proxy re-encryption scheme (*L-IPPRE*) to guarantee secure data sharing between different entities with different access policies in IoT environment. We tested the execution time of each algorithm of *L-IPPRE* protocol on different types of elliptic curves. The performance evaluation shows that the system complexity in our proposed *L-IPPRE* scheme is reasonable in practical IoT scenarios. To the best of our knowledge, the *L-IPPRE* protocol is the first inner-product proxy re-encryption scheme that provides a secure data sharing mechanism for distributed fine-grained data access control in the IoT.

6.2 Future Works

The results obtained of this thesis point to several interesting directions for secure data sharing on untrusted storage as the future works:

Multi-authority Fine-grained Access Control. The primary attribute-based encryption schemes are based on single trust authority to manage user attributes and their corresponding private keys, which will be vulnerable to security attacks. Recently, some research efforts have been proposed multi-authority schemes supporting fine-grained access control policies [80, 81, 82].

Therefore, one interesting future work would be integrating the proposed protocols with other schemes targeting multiple authorities to avoid collusion among users sharing data.

Combining with Trusted Computing Technologies. In this thesis, the cloud and proxy servers are assumed to be "honest-but-curious" i.e., the cloud and proxy servers execute the pre-defined protocols correctly, but may try to learn as much private information as possible during the protocol execution. In practical application scenarios, it would be desirable to remove this assumption to provide a stronger level of security protection for the cloud and proxy servers. For this purpose, one interesting future work would be combining trust computing techniques with the data access control mechanisms.

Key Management. Key management involves creating, renewing, and managing private keys. With the growing amount of shared data in cloud computing and IoT environments, the data consumers could have an abundant number of keys, which might be difficult to manage cases where keys are lost, stolen, and expired. These cases may compromise security. Therefore, one interesting future work would be integrating proposed protocols with key management schemes in order to further improve performance.

Bibliography

[1] Jong Hwan Park. Inner-product encryption under standard assumptions. *Des. Codes Cryptography*, 58(3):235–257, March 2011.

[2] Intae Kim, Seong Oun Hwang, Jong Hwan Park, and Chanil Park. An efficient predicate encryption with constant pairing computations and minimum costs. *IEEE Trans. Comput.*, 65(10):2947–2958, October 2016.

[3] Danan Thilakanathan, Shiping Chen, Surya Nepal, and Rafael A. Calvo. *Secure Data Sharing in the Cloud*, pages 45–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[4] Hua Guo, Fangchao Ma, Zhoujun Li, and Chunhe Xia. Key-exposure protection in public auditing with user revocation in cloud storage. In *Revised Selected Papers of the 6th International Conference on Trusted Systems - Volume 9473*, INTRUST 2014, pages 127–136, 2015.

[5] W. Itani, A. Kayssi, and A. Chehab. Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 711–716, Dec 2009.

[6] ACL. Access control list. https://en.wikipedia.org/wiki/Access_control_list/.

[7] J. Li, G. Zhao, X. Chen, D. Xie, C. Rong, W. Li, L. Tang, and Y. Tang. Fine-grained data access control systems with user accountability in cloud computing. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pages 89–96, Nov 2010.

[8] Amit Sahai and Brent Waters. *Fuzzy Identity-Based Encryption*, pages 457–473. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[9] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Sympo-*

sium on Security and Privacy, SP '07, pages 321–334, Washington, DC, USA, 2007. IEEE Computer Society.

[10] Jonathan Katz, Amit Sahai, and Brent Waters. *Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products*, pages 146–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[11] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 89–98, New York, NY, USA, 2006. ACM.

[12] Hossein Shafagh, Anwar Hithnawi, Andreas Droescher, Simon Duquennoy, and Wen Hu. Talos: Encrypted query processing for the internet of things. In Junehwa Song, Tarek F. Abdelzaher, and Cecilia Mascolo, editors, *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015, Seoul, South Korea, November 1-4, 2015*, pages 197–210. ACM, 2015.

[13] Arjen K. Lenstra. Key length. contribution to the handbook of information security, 2004.

[14] Michiel Hazewinkel. *Cyclic Group, Encyclopedia of Mathematics*. 2001.

[15] Joseph H Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer, Dordrecht, 2009.

[16] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. *Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs*, pages 119–132. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[17] Dan Boneh and Matt Franklin. *Identity-Based Encryption from the Weil Pairing*, pages 213–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[18] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter ibe and signatures via asymmetric pairings. In *Proceedings of the 5th International Conference on Pairing-Based Cryptography*, Pairing'12, pages 122–140, Berlin, Heidelberg, 2013. Springer-Verlag.

[19] Léo Ducas. Anonymity from asymmetry: New constructions for anonymous hibe. In *Proceedings of the 2010 International Conference on*

Topics in Cryptology, CT-RSA'10, pages 148–164, Berlin, Heidelberg, 2010. Springer-Verlag.

[20] Dan Boneh, Xavier Boyen, and Hovav Shacham. *Short Group Signatures*, pages 41–55. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[21] Dan Boneh and Xavier Boyen. *Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles*, pages 223–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[22] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, CCS '05, pages 320–329, New York, NY, USA, 2005. ACM.

[23] Jong Hwan Park and Dong Hoon Lee. Fully collusion-resistant traitor tracing scheme with shorter ciphertexts. *Des. Codes Cryptography*, 60(3):255–276, September 2011.

[24] Adi Shamir. *Identity-Based Cryptosystems and Signature Schemes*, pages 47–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985.

[25] Cheng-Chi Lee, Pei-Shan Chung, and Min-Shiang Hwang. A survey on attribute-based encryption schemes of access control in cloud environments. *I. J. Network Security*, 15(4):231–240, 2013.

[26] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. *IACR Cryptology ePrint Archive*, 2007:323, 2007.

[27] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *Financial Cryptography, 4th International Conference, FC 2000 Anguilla, British West Indies, February 20-24, 2000, Proceedings*, volume 1962 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2000.

[28] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*, pages 273–285. IEEE Computer Society, 2010.

[29] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. *Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts*, pages 90–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[30] Ling Cheung and Calvin C. Newport. Provably secure ciphertext policy ABE. *IACR Cryptology ePrint Archive*, 2007:183, 2007.

[31] Takashi Nishide, Kazuki Yoneyama, and Kazuo Ohta. *Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures*, pages 111–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[32] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. *Bounded Ciphertext Policy Attribute Based Encryption*, pages 579–591. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[33] Xiaohui Liang, Zhenfu Cao, Huang Lin, and Dongsheng Xing. Provably secure and efficient bounded ciphertext policy attribute based encryption. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 343–352, New York, NY, USA, 2009. ACM.

[34] Luan Ibraimi, Qiang Tang, Pieter Hartel, and Willem Jonker. *Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes*, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[35] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[36] Keita Emura, Atsuko Miyaji, Akito Nomura, Kazumasa Omote, and Masakazu Soshi. *A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length*, pages 13–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[37] Brent Waters. *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*, pages 53–70. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[38] Dan Boneh, Craig Gentry, and Brent Waters. *Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys*, pages 258–275. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[39] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. *Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption*, pages 62–91. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[40] Allison Lewko and Brent Waters. *New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts*, pages 455–479. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[41] Jiang Zhang and Zhenfeng Zhang. A ciphertext policy attribute-based encryption scheme without pairings. In *Proceedings of the 7th International Conference on Information Security and Cryptology*, Inscrypt'11, pages 324–340, Berlin, Heidelberg, 2012. Springer-Verlag.

[42] Masahiro Mambo and Eiji Okamoto. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 80A(1):54–63, 1997.

[43] Matt Blaze, Gerrit Bleumer, and Martin Strauss. *Divertible protocols and atomic proxy cryptography*, pages 127–144. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[44] Taher ElGamal. *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, pages 10–18. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985.

[45] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, February 2006.

[46] Dan Boneh. *The Decision Diffie-Hellman problem*, pages 48–63. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[47] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, February 2006.

[48] Qiang Tang. Type-based proxy re-encryption and its construction. In *Proceedings of the 9th International Conference on Cryptology in India: Progress in Cryptology*, INDOCRYPT '08, pages 130–144, Berlin, Heidelberg, 2008. Springer-Verlag.

[49] Matthew Green and Giuseppe Ateniese. *Identity-Based Proxy Re-encryption*, pages 288–306. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[50] Shanqing Guo, Yingpei Zeng, Juan Wei, and Qiuliang Xu. Attribute-based re-encryption scheme in the standard model. *Wuhan University Journal of Natural Sciences*, 13(5):621–625, 2008.

[51] Xiaohui Liang, Zhenfu Cao, Huang Lin, and Jun Shao. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10-12, 2009*, pages 276–286, 2009.

[52] Song Luo, Jianbin Hu, and Zhong Chen. Ciphertext policy attribute-based proxy re-encryption. In *Proceedings of the 12th International Conference on Information and Communications Security*, ICICS'10, pages 401–415, Berlin, Heidelberg, 2010. Springer-Verlag.

[53] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 261–270, New York, NY, USA, 2010. ACM.

[54] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 534–542, Piscataway, NJ, USA, 2010. IEEE Press.

[55] Mahesh Kallahalla, Erik Riedel, Ram Swaminathan, Qian Wang, and Kevin Fu. Plutus: Scalable secure file sharing on untrusted storage. In *Proceedings of the 2Nd USENIX Conference on File and Storage Technologies*, FAST '03, pages 29–42, Berkeley, CA, USA, 2003. USENIX Association.

[56] Jeong-Min Do, You-Jin Song, and Namje Park. Attribute based proxy re-encryption for data confidentiality in cloud computing environments. In *Proceedings of the 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, CNSI '11, pages 248–251, Washington, DC, USA, 2011. IEEE Computer Society.

[57] Hwajeong Seo and Howon Kim. Attribute-based proxy re-encryption with a constant number of pairing operations. *J. Inform. and Commun. Convergence Engineering*, 10(1):53–60, 2012.

[58] Keying Li. Matrix access structure policy used in attribute-based proxy re-encryption. *CoRR*, abs/1302.6428, 2013.

[59] K. Liang, L. Fang, W. Susilo, and D. S. Wong. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pages 552–559, Sept 2013.

[60] Kaitai Liang, Man Ho Au, Willy Susilo, Duncan S. Wong, Guomin Yang, and Yong Yu. *An Adaptively CCA-Secure Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Cloud Data Sharing*, pages 448–461. Springer International Publishing, Cham, 2014.

[61] Huixian Li and Liaojun Pang. Efficient and adaptively secure attribute-based proxy reencryption scheme. *IJDSN*, 12(5):5235714:1–5235714:12, 2016.

[62] Brent Waters. *Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions*, pages 619–636. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[63] R. Wu, G. J. Ahn, and H. Hu. Secure sharing of electronic health records in clouds. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 711–718, Oct 2012.

[64] Isabelle Hang, Florian Kerschbaum, and Ernesto Damiani. Enki: Access control for encrypted query processing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 183–196, New York, NY, USA, 2015. ACM.

[65] Tatsuaki Okamoto and Katsuyuki Takashima. *Hierarchical Predicate Encryption for Inner-Products*, pages 214–231. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[66] Cheng Chen, Jie Chen, Hoon Wei Lim, Zhenfeng Zhang, Dengguo Feng, San Ling, and Huaxiong Wang. Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In

Proceedings of the 13th International Conference on Topics in Cryptology, CT-RSA'13, pages 50–67, Berlin, Heidelberg, 2013. Springer-Verlag.

[67] Michael Backes, Martin Gagné, and Sri Aravinda Krishnan Thyagarajan. Fully secure inner-product proxy re-encryption with constant size ciphertext. In *Proceedings of the 3rd International Workshop on Security in Cloud Computing*, SCC '15, pages 31–40, New York, NY, USA, 2015. ACM.

[68] Matthew Green, Susan Hohenberger, and Brent Waters. Outsourcing the decryption of ABE ciphertexts. In *20th USENIX Security Symposium, San Francisco, CA, USA, August 8-12, 2011, Proceedings*, 2011.

[69] Yutaka Kawai and Katsuyuki Takashima. Fully-anonymous functional proxy-re-encryption. *IACR Cryptology ePrint Archive*, 2013:318, 2013.

[70] Ben Lynn. Pairing Based Cryptography library. `http://crypto.stanford.edu/pbc/`, June 2007.

[71] Nicolaie Fantana, Till Riedel, Jochen Schlick, Stefan Ferber, Jrgen Hupp, Stephen Miles, Florian Michahelles, and Stefan Svensson. Internet of things - converging technologies for smart environments and integrated ecosystems, 01 2013.

[72] Dave Evans. *The Internet of Things, How the Next Evolution of the Internet Is Changing Everything.* 2011.

[73] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. *Computer Networks*, 76(Supplement C):146 – 164, 2015.

[74] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, January 2004.

[75] Xinlei Wang, Jianqing Zhang, Eve Schooler, and Mihaela Ion. Performance evaluation of attribute-based encryption: Toward data privacy in the iot, 06 2014.

[76] Moreno Ambrosin, Mauro Conti, and Tooska Dargahi. On the feasibility of attribute-based encryption on smartphone devices. In *Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems*, IoT-Sys '15, pages 49–54, New York, NY, USA, 2015. ACM.

[77] Matthew Green, Susan Hohenberger, and Brent Waters. Outsourcing the decryption of abe ciphertexts. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 34–34, Berkeley, CA, USA, 2011. USENIX Association.

[78] M Muhammad Asim, M Milan Petkovic, and T Tanya Ignatenko. Attribute-based encryption with encryption and decryption outsourcing. 2014.

[79] L. Touati, Y. Challal, and A. Bouabdallah. C-cp-abe: Cooperative ciphertext policy attribute-based encryption for the internet of things. In *2014 International Conference on Advanced Networking Distributed Systems and Applications*, pages 64–69, June 2014.

[80] D. Chen, L. Wan, C. Wang, S. Pan, and Y. Ji. A multi-authority attribute-based encryption scheme with pre-decryption. In *2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, pages 223–228, Dec 2015.

[81] Melissa Chase. Multi-authority attribute based encryption. In *Proceedings of the 4th Conference on Theory of Cryptography*, TCC'07, pages 515–534, Berlin, Heidelberg, 2007. Springer-Verlag.

[82] Allison Lewko and Brent Waters. *Decentralizing Attribute-Based Encryption*, pages 568–588. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.