Secure Data Sharing in Cloud Using an Efficient Inner-Product Proxy Re-Encryption Scheme

Masoomeh Sepehri[1], Alberto Trombetta[2] and Maryam Sepehri[1]

[1]*Department of Computer Science, University of Milan, Milan, Italy*
[2]*Department of Computer Science and Communication, University of Insubria,Varese, Italy*
E-mail: masoomeh.sepehri@unimi.it; maryam.sepehri@unimi.it; alberto.trombetta@uninsubria.it

Received 1 December 2017; Accepted 6 December 2017;
Publication 9 January 2018

Abstract

With the ever-growing production of data coming from multiple, scattered, highly dynamical sources (like those found in *IoT* scenarios), many providers are motivated to upload their data to the cloud servers and share them with other persons with different purposes. However, storing data on cloud imposes serious concerns in terms of data confidentiality and access control. These concerns get more attention when data is required to be shared among multiple users with different access policies. In order to update access policy without making re-encryption, we propose an efficient inner-product proxy re-encryption scheme that provides a proxy server with a transformation key with which a delegator's ciphertext associated with an attribute vector can be transformed to a new ciphertext associated with delegatee's attribute vector set. Our proposed policy updating scheme enables the delegatee to decrypt the shared data with its own key without requesting a new decryption key. We experimentally analyze the efficiency of our scheme and show that our scheme is adaptive attribute-secure against chosen-plaintext under standard Decisional Linear (*D-Linear*) assumption.

Keywords: Attribute-based cryptography, Secure data sharing, Fine-grained access control, Proxy re-encryption.

Journal of Cyber Security, Vol. 6_3, 339–378.
doi: 10.13052/jcsm2245-1439.635

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 [33], 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 [12]. 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. Hence, many cryptographic-based approaches have been proposed and among them, attribute-based encryption (*ABE*) schemes [28] look very promising since they bind fine-grained access control policies to the data and they do not require an access control manager to check the access policies in real time. In *ABE* scheme, data is encrypted based on the set of attributes (key-policy *ABE* [10]) or according to an access control policy over attributes (ciphertext-policy *ABE* [2]), 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. Furthermore, the *ABE* scheme is more strongly resistant to collusion attacks than traditional public key encryption schemes [2]. Although access control mechanisms based on *ABE* schemes present advantages regarding reduced communication, storage management and provide a fine-grained access control, they are not suitable for scenarios in which data must be shared among different parties with different access policies. A straightforward 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. These issues can be addressed

by the adopting an attribute-based proxy re-encryption (*ABPRE*) scheme that delegates 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 as well as the user's private key cannot be revealed to the proxy. Although *ABPRE* approaches preserve the privacy of shared data among users with different access policies, they do not sufficiently protect the attributes associated with the ciphertexts. For example, in a healthcare scenario medical data require a high degree of privacy since they are accessed by many parties such as patients or staffs (e.g. doctors, nurses, care practitioners, etc.,) from a 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 [14] 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 where a ciphertext conceal the plaintext. Informally, in the attribute-hiding, the secrecy of challenge attributes $\overrightarrow{x}_0$ and $\overrightarrow{x}_1$ is ensured against the adversary having private key $\overrightarrow{v}$ if the compatibility condition $< \overrightarrow{x}_0, \overrightarrow{v} > = < \overrightarrow{x}_1, \overrightarrow{v} >$ holds (here, $< \overrightarrow{x}, \overrightarrow{v} >$ denotes the standard inner product). In this work, we construct an adaptive secure attribute-hiding scheme through a proxy re-encryption method obtained with a non-trivial modification of a well known inner-product-based, attribute-based encryption scheme proposed by Park [26]. Unlike the existing scheme [32], we formally show that our proposal is adaptively secure for attribute-hiding (attribute-hiding in the sense of the definition by Katz *et al.* [14]).

The work is organized as follows: in Section 2 we present and analyze the related works; in Section 3 we introduce the definitions of inner-product encryption *IPE* and inner-product proxy re-encryption *IPPRE* and their security definitions; in Section 4 we present cryptographic primitives and complexity assumptions which are used in our proposed protocol; in Section 5 we provide a high-level view of the system in which we deploy in our scheme; in Section 6 we provide a detailed description of our proposed scheme and security proof based on standard game-based techniques; in Section 7 we show that our scheme can be efficiently implemented; finally, in Section 8 we draw some conclusions and propose some lines for future work.

2 Related Work

Proxy re-encryption scheme. Recently, several research papers have been developed for secure data sharing in the cloud [27, 30, 31]. Most of these works have adopted proxy re-encryption (*PRE*) scheme which was first proposed by Mambo and Okamoto [24] as a way to support the delegation of decryption rights. A seminal paper by Blaze *et al*. [3] proposed a bidirectional *PRE* scheme (called *BBS*) based on El-Gamal scheme [8] and introduced the notion of "*re-encryption key*". Using this key, a semi-trusted proxy server transforms a ciphertext encrypted under the delegator's public key into another ciphertext of the same plaintext encrypted under delegate's public key without revealing the underlying plaintext and user private key. 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* property i.e., re-encryption key can be used to transform ciphertext from the delegator to the delegatee and vice versa, therefore it is only useful when the trust relationship between involved parties is mutual. Moreover, the scheme is not suitable for group communication since the proxy has to preserve n re-encryption key for n group members. Furthermore, *BBS* proxy re-encryption scheme exposed to collusion attacks, if the proxy colludes with one party they can recover the private key of the other party. To tackle these disadvantages, Ateniese *et al*. [1] proposed the first unidirectional and collusion resistant proxy re-encryption scheme without requiring pre-sharing between parties, based on bilinear maps.

Although proxy re-encryption techniques enable secure data sharing among different users in the cloud, they do not enforce fine-grained access control policies on the shared data. To address this issue, the traditional *PRE* approach has been extended with functionalities taken from attribute-based encryption (*ABE*) scheme in which both ciphertexts and user's private keys are associated with an attribute set and a user can decrypt a ciphertext only if the set of attributes in his private key match the attributes associated to the ciphertext [2, 10, 28].

Attribute-based proxy re-encryption scheme. An attribute-based proxy re-encryption (*ABPRE*) scheme was first proposed by Guo *et al*. [13] based on an (key-policy) attribute-based encryption scheme [10] and a general proxy re-encryption scheme. 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, without leakages about the plaintext and user

private key. It has been proven that the security of the proposed scheme in the standard model based on decisional bilinear Diffie-Hellman (*DBDH*) assumption.

By adopting identity-based proxy re-encryption [11] to the construction of ciphertext-policy attribute-based encryption (*CP-ABE*) scheme [7], Liang *et al.* [21] presented the first ciphertext-policy attribute-based proxy re-encryption (*CP-ABPRE*) scheme. In this 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. Their scheme satisfies *multi-hop* 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.* [22] 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-interactivity* and *multi-hop.* Moreover, their scheme has two new properties: (i) *re-encryption control:* the encryptor can decide whether the ciphertext can be re-encrypted or not, and (ii) *extra access control:* the proxy can add extra access policy to the ciphertext during re-encryption process.

The computation cost of the previous *ABPRE* schemes is according to the number of attributes in the system, which implies huge computational overhead. To address this issue, based on Emura *et al.'s* [9] *CP-ABE* scheme which has a constant ciphertext length, Seo *et al.* [29] presented a *CP-ABPRE* scheme with 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.

Most of the previous *CP-ABPRE* schemes [21, 22, 29, 36] 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 this issue, Li [18] presented a new *CP-ABPRE* scheme using matrix access structure policy which supports any monotonic access formula. Their scheme satisfies the properties of both *PRE* and *CP-ABPRE* schemes, such

as *unidirectionality, non-interactivity, multi-hop, re-encryption control, extra access control* and *secret key security* providing a guarantee for the delegator such that if the proxy and all delegatees collude, they can not recover his master secret key. Moreover, they described the security model called Selective-Policy Model for their *CP-ABPRE* scheme based on [21].

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. Therefore, a *CP-ABPRE* system being secure against chosen-ciphertext attacks (*CCA*) is needed to prevent the above subtle attacks and enables the system to be further developed. To address this issue, based on the Waters's *CP-ABE* scheme [35], Liang *et al*. [19] proposed the first secure *CP-ABPRE* scheme against selective chosen-ciphertext attacks (*CCA*) which supports any monotonic access structures. Moreover, They constructed their proposed scheme in the random oracle model and they showed that their scheme can be proven *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. Therefore, an adaptively *CCA* secure *CP-ABPRE* scheme is needed in most of the practical network applications. Thus, Liang *et al*. [20] proposed the first adaptively *CCA*-secure *CP-ABPRE* scheme by integrating the dual system encryption technology with selective proof technique. Their scheme supports any monotonic access structure such that users are allowed to fulfill more flexible delegation of decryption rights. This scheme is proven adaptively *CCA* secure in the standard model without loss of expressiveness on access policy. However, their scheme demands a number of paring operations that implies huge computational overheads.

Recently, Li *et al*. [17] proposed an efficient and adaptively secure *CP-ABPRE* scheme basing on Waters' dual system encryption technology [34]. 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 (3*P-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.

Predicate encryption scheme. Although the attribute-based proxy re-encryption schemes have desirable functionality, they do not guarantee attribute-hiding property i.e., a ciphertext conceals the associated attributes as well as the plaintext so that no information about attributes is revealed during the decryption process. Therefore, to preserve the confidentiality of the attributes associated with the ciphertext, a seminal paper of Katz *et al.* [14] 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 $f(I) = True$. Katz *et al.* [14] also presented a special type of predicate encryption for a class of predicates called *inner-product encryption (IPE).* In *IPE,* both ciphertext and private key are associated with vector $\overrightarrow{x}$ and $\overrightarrow{v}$ respectively and the ciphertext can be decrypted by the private key $SK_{\overrightarrow{v}}$ if and only if $< \overrightarrow{x}, \overrightarrow{v} >= 0$ (here, $< \overrightarrow{x}, \overrightarrow{v} >$ denotes the standard inner-product). Their method represents a wide class of predicates including conjunction and disjunction formulas and polynomial evaluations.

Later, Okamoto *et al.* [25] proposed a hierarchical predicate encryption (*HPE*) scheme for inner-product encryption. They used *n*-dimensional vector spaces in prime order bilinear groups and achieves full security under the standard model. In [16], 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 properties, the security of their schemes is not under well-known standard assumptions. A different work of Park [26] presented an efficient *IPE* scheme supporting the attribute-hiding property. Their scheme is based on prime order bilinear groups and secure against the well-known Decision Bilinear Diffie-Hellman (*BDH*) and Decision Linear assumptions.

3 Definitions

In this section, we formally define the syntax of inner-product encryption (*IPE*) and inner-product proxy re-encryption (*IPPRE*) and their security properties. Our *IPE* definition follows the general framework of that given in [26]. 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 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:

$\text{Setup}_{\text{IPE}}$: 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*.

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

$\text{Encrypt}_{\text{IPE}}$: takes as input a public key *PK,* an attribute vector $\overrightarrow{x}$ and a message $M \in \mathcal{M}$. It outputs a corresponding ciphertext $CT_{\overrightarrow{x}} \leftarrow (PK, \overrightarrow{x}, M)$.

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

Definition 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 $\overrightarrow{v}$, the challenger gives $SK_{\overrightarrow{v}}$ to $\mathcal{A}$.
Challenge. For a challenge query $(X_0, X_1, \overrightarrow{x}_0, \overrightarrow{x}_1)$, subject to the following restriction:

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

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

Phase 2. The adversary $\mathcal{A}$ is allowed to adaptively issues polynomial number of key queries. For a private key query $\overrightarrow{v}$, 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 $Adv_{\mathcal{A}}^{\text{IPE-AH}}(\lambda) = Pr[b = b^{'}] - \frac{1}{2}$. 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 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 $\overrightarrow{x}$ into a new ciphertext encrypting the same plaintext but associated with a different vector $\overrightarrow{w}$, 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:

$\text{Setup}_{\text{IPPRE}}$: 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*.

$\text{Encrypt}_{\text{IPPRE}}$: takes as input the public key *PK,* a vector $\overrightarrow{x} \in \Sigma$ of attributes and a message $M \in \mathcal{M}$ to output a ciphertext $CT_{\overrightarrow{x}}$.

$\text{KeyGen}_{\text{IPPRE}}$: takes as input the master secret key *MSK,* the public key *PK* and a predicate vector $\overrightarrow{v} \in \mathcal{F}$. It outputs a private key $SK_{\overrightarrow{v}}$ associated with vector $\overrightarrow{v}$.

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

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

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

From here on, we use the terms *Level-1* (*L1*) and *Level-2* (*L2*) to denote ciphertexts obtained as the output of Encrypt and Re-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: Decrypt (KeyGen ($MSK, PK, \overrightarrow{v}$), Encrypt ($PK, \overrightarrow{x}, M$)) $= M$;

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

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

Definition 4 (Attribute-Hiding for Level-1 Ciphertexts (*AH-L1*)). An inner-product proxy re-encryption (*IPPRE*) 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 Setup (λ, *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 $\overrightarrow{v}$, the challenger gives $SK_{\overrightarrow{v}} \xleftarrow{R}$ KeyGen (*MSK, PK,* $\overrightarrow{v}$) to $\mathcal{A}$, where *R* indicates that $SK_{\overrightarrow{v}}$ is randomly selected from KeyGen according to its distribution.

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

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

Challenge. For a challenge query $(\overrightarrow{x}_0, \overrightarrow{x}_1, M_0, M_1)$ under the condition that:

– Any private key query $\overrightarrow{v}$ and re-encryption key query $(\overrightarrow{v}_l, \overrightarrow{w}_l)$, 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 $< \overrightarrow{v}, \overrightarrow{x}_0 >=< \overrightarrow{v}, \overrightarrow{x}_1 >= 0$ and $< \overrightarrow{v}_l, \overrightarrow{x}_0 >=< \overrightarrow{v}_l, \overrightarrow{x}_1 >= 0$ in the case that $< \overrightarrow{v}, \overrightarrow{w}_l >= 0$.

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 $CT_{\overrightarrow{x}_b} \xleftarrow{R}$ Encrypt $(PK, \overrightarrow{x}_b, M_b)$ 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 $(\overrightarrow{v}_t, \overrightarrow{w}_t, CT_t)$, 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 $< \overrightarrow{v}_t, \overrightarrow{x}_0 >=< \overrightarrow{v}_t, \overrightarrow{x}_1 >= 0$ and $< \vec{v'}, \overrightarrow{w}_t >= 0$ for any decryption key query for $\vec{v'}$ if $CT_t = CT_{\overrightarrow{x}_b}$.
>
> The challenger computes $RK_{\overrightarrow{v}_t, \overrightarrow{w}_t} \xleftarrow{R}$ Re-KeyGen $(MSK, \overrightarrow{v}_t, \overrightarrow{w}_t)$ and $CT'_{\overrightarrow{w}_t} \xleftarrow{R}$ Re-Encrypt $(PK, RK_{\overrightarrow{v}_t, \overrightarrow{w}_t}, CT_t)$, and it gives $CT'_{\overrightarrow{w}t}$ to the adversary.

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

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

Definition 5 (Attribute-Hiding for Level-2 Re-encrypted Ciphertexts (*AH-L2*)). An inner-product proxy re-encryption (*IPPRE*) 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, respectively.

Challenge: Upon receiving the query $(\overrightarrow{x}_0, \overrightarrow{x}_1, M_0, M_1, \overrightarrow{v}_0, \overrightarrow{v}_1, \overrightarrow{w}_0, \overrightarrow{w}_1)$ from the adversary with the restrictions that $(M_0, \overrightarrow{x}_0, \overrightarrow{v}_0) = (M_1, \overrightarrow{x}_1, \overrightarrow{v}_1)$

if $< \vec{v'}, \overrightarrow{w}_0 >=< \vec{v'}, \overrightarrow{w}_1 >= 0$, for any private key query $\vec{v'}$, the challenger $\mathcal{B}$ samples a random bit $b \xleftarrow{R} \{0,1\}$ and gives:

$CT'_{\overrightarrow{w}_b} \xleftarrow{R}$ Re-Encrypt(*PK*, Re-KeyGen(*PK*, KeyGen(*PK*, *SK*,$\overrightarrow{v}_b$),$\overrightarrow{w}_b$), Encrypt (*PK*, $\overrightarrow{x}_b, M_b$)). 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 $\text{Adv}_{\mathcal{A}}^{\text{PAH-L2}}(\lambda) := \Pr[b = b'] - \frac{1}{2}$. 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 $s_{M,\overrightarrow{x},\overrightarrow{v}} := 0$ if $(M_0, \overrightarrow{x}_0, \overrightarrow{v}_0) \neq (M_1, \overrightarrow{x}_1, \overrightarrow{v}_1)$ for challenge $(M_l, \overrightarrow{x}_l, \overrightarrow{v}_l)$ for $l = 0, 1$ and $s_{M,\overrightarrow{x},\overrightarrow{v}} := 1$, otherwise.

4 Cryptographic Background and Complexity Assumptions

In this section, we define Bilinear Map following the notation in [5] and review some general assumptions we use in Section 6 to prove the security of our construction.

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) $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a function that has the following properties [5]:

1. *Bilinear:* a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ 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 $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

Decisional Bilinear Diffie-Hellman (DBDH) Assumption [5]. 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 [4]. 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 6. 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.

5 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 *IPPRE* scheme in such real-world scenarios. A *IPPRE*-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 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., Encrypt$_{A_1}$ (M_1)), where A_1 indicates access privilege on the owner l'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*).

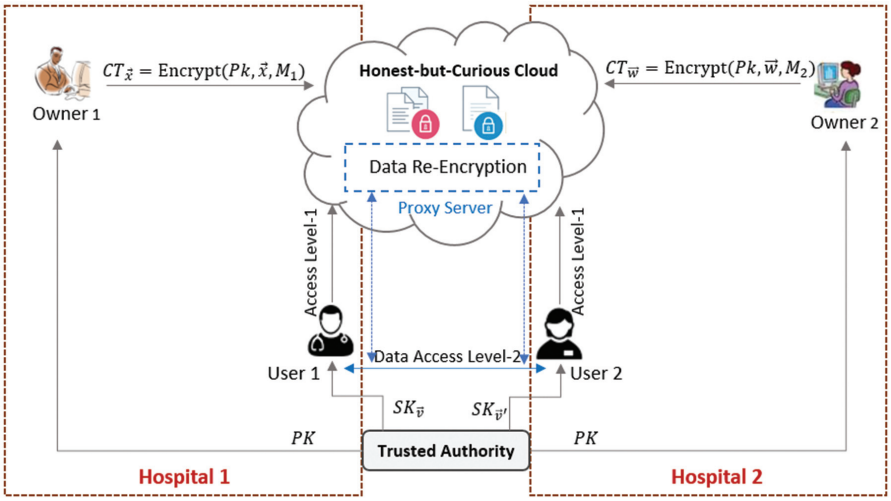


Figure 1 *IPPRE*-based data sharing healthcare system model.

1. *Level-1*. User 1 who satisfies A_1 can access to the owner l's data using its own private key associated with a vector $\overrightarrow{v}$.
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 $\overrightarrow{w}$ in Figure 1), but not the access policy A_1 (attribute vector $\overrightarrow{x}$ in Figure 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.

6 The Main Construction

In this section, we construct our *IPPRE* scheme in detail and give intuition about our proof. The scheme consists of six algorithms namely Setup, Encrypt, KeyGen, Decrypt, Re-KeyGen, Re-Encrypt. 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 $\overrightarrow{v} = (v_1, \ldots, v_n) \in \Sigma$, each component v_i belong to the set $\mathbb{Z}_p^*$, and
- a message $M \in \mathcal{M}$ and a vector $\overrightarrow{v}$, each $\overrightarrow{v}$ belongs to Σ and $\mathcal{M} = \mathbb{G}_T$.

6.1 The Construction

(*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*(λ)[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:

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

For *i* = 1, . . . , *n,* the Setup algorithm first computes:

$$W_{1,i} = g^{w_{1,i}}, \quad W_{2,i} = g^{w_{2,i}}, \quad T_{1,i} = g^{t_{1,i}}, \quad T_{2,i} = g^{t_{2,i}},$$
$$F_{1,i} = g^{f_{1,i}}, \quad F_{2,i} = g^{f_{2,i}}, \quad H_{1,i} = g^{h_{1,i}}, \quad H_{2,i} = g^{h_{2,i}},$$

and then sets:

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

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

$$\begin{aligned} PK = (&g, g_1, \{W_{1,i}, W_{2,i}, F_{1,i}, F_{2,i}\}_{i=1}^n, \{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\}_{i=1}^n, \\ &\{U_i, V_i\}_{i=1}^2, \Lambda) \in \mathbb{G}^{8n+6} \times \mathbb{G}_T \\ MSK = (&\{w_{1,i}, w_{2,i}, t_{1,i}, t_{2,i}, f_{1,i}, f_{2,i}, h_{1,i}, h_{2,i}\}_{i=1}^n, \{\delta_i, \theta_i\}_{i=1}^2, g_2) \\ &\in \mathbb{Z}_p^{8n+4} \times \mathbb{G}. \end{aligned}$$

[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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map.

$SK_{\overrightarrow{v}} \leftarrow$ **KeyGen** $(MSK, PK, \overrightarrow{v})$. On input vector $\overrightarrow{v} = (v_1, \ldots, v_n) \in (\mathbb{Z}_p^*)^n$, 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:

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

$$\{K_{1,i} = g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}, \quad K_{2,i} = g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\}_{i=1}^n,$$
$$\{K_{3,i} = g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}, \quad K_{4,i} = g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\}_{i=1}^n,$$
$$K_A = g_2 \prod_{i=1}^n K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-h_{1,i}} K_{4,i}^{-h_{2,i}}, K_B = \prod_{i=1}^n g^{-(r_i+\phi_i)}.$$

$CT_{\overrightarrow{x}} \leftarrow$ **Encrypt** $(PK, \overrightarrow{x}, M)$. On input vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in (\mathbb{Z}_p^*)^n$, 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 $CT_{\overrightarrow{x}}$ as the follows:

$$CT_{\overrightarrow{x}} = (A, B, \{C_{1,i} = W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3}, C_{2,i} = W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\}_{i=1}^n,$$
$$\{C_{3,i} = T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_i s_4}, C_{4,i} = T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_i s_4}\}_{i=1}^n, \Lambda^{-s_2} M) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}}.$$

Where we define each component of $CT_{\overrightarrow{x}}$ as the following, $1 \leq i \leq n$:

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

In this step, random elements $\{W_{1,i}^{s_1}, W_{2,i}^{s_1}, T_{1,i}^{s_1}, T_{2,i}^{s_1}, \}$ are used to mask each component x_i of a vector $\overrightarrow{x}$. 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 $F_{2,i}^{s_2}$ that can be tested as $e(A, F_{1,i}) \stackrel{?}{=} e(g, C_{1,i})$ using bilinear maps.

$RK_{\overrightarrow{v},\overrightarrow{w}} \leftarrow$ **Re-KeyGen** $(MSK, \overrightarrow{v}, \overrightarrow{w})$. 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 $\overrightarrow{w}$ using Encrypt $(PK, \overrightarrow{w}, g_2^d)$ and outputs $CT_{\overrightarrow{w}}$.

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 $RK_{\overrightarrow{v},\overrightarrow{w}}$, $1 \leq i \leq n$ 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 $\overrightarrow{v}$, $\overrightarrow{w}$ consists of two parts: a modified decryption key vector $\overrightarrow{v}$ and a ciphertext encrypted with vector $\overrightarrow{w}$. 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.

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

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

computing $\hat{CT}$, the algorithm checks if the attributes list in $RK_{\overrightarrow{v},\overrightarrow{w}}$ satisfies the attributes set of $CT_{\overrightarrow{x}}$ 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:

$$e(C_{1,i}, K'_{1,i}) = e(g^{w_{1,i}s_1} g^{f_{1,i}s_2} g^{\delta_1 x_i s_3}, g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} g_2^{d\delta_2})$$

$$e(C_{2,i}, K'_{2,i}) = e(g^{t_{1,i}s_1} g^{h_{2,i}s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1})$$

$$e(C_{3,i}, K'_{3,i}) = e(g^{t_{2,i}s_1} g^{h_{2,i}s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1})$$

$$e(C_{4,i}, K'_{4,i}) = e(g^{t_{2,i}s_1} g^{h_{2,i}s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1}).$$

Hence, we expand the above formula as follows:

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

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

$$\cdot e(g^{w_{2,i}s_1} g^{f_{2,i}s_2} g^{\delta_2 x_i s_3}, g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} g_2^{-d\delta_1})$$

$$\cdot e(g^{t_{1,i}s_1} g^{h_{1,i}s_2} g^{\theta_1 x_i s_4}, g^{-\theta_2 \phi'_i} g^{\lambda'_1 v_i t_{2,i}} g_2^{d\theta_2})$$

$$\cdot e(g^{t_{2,i}s_1} g^{h_{2,i}s_2} g^{\theta_2 x_i s_4}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1})$$

$$= \prod_{i=1}^{n} e(g^{w_{1,i}s_1}, g^{-\delta_2 r'_i}) \cdot e(g^{f_{1,i}s_2}, g^{-\delta_2 r'_i} g^{\lambda'_1 v_i w_{2,i}} g_2^{d\delta_2})$$

$$\cdot e(g^{\delta_1 x_i s_3}, g^{\lambda'_1 v_i w_{2,i}}) \cdot e(g^{w_{1,i}s_1}, g_2^{d\delta_2}) \cdot e(g^{w_{2,i}s_1}, g^{\delta_1 r'_i})$$

$$\cdot e(g^{f_{2,i}s_2}, g^{\delta_1 r'_i} g^{-\lambda'_1 v_i w_{1,i}} g_2^{-d\delta_1}) \cdot e(g^{\delta_2 x_i s_3}, g^{-\lambda'_1 v_i w_{1,i}}) \cdot e(g^{w_{2,i}s_1}, g_2^{-d\delta_1})$$

$$\cdot e(g^{t_{1,i}s_1}, g^{-\theta_2 \phi'_i}) \cdot e(g^{h_{1,i}s_2}, g^{-\theta_2 \phi'_i} g^{\lambda'_2 v_i t_{2,i}} g_2^{d\theta_2}) \cdot e(g^{\theta_1 x_i s_4}, g^{\lambda'_2 v_i t_{2,i}})$$

$$\cdot e(g^{t_{1,i}s_1}, g_2^{d\theta_2}) \cdot e(g^{t_{2,i}s_1}, g^{\theta_1 \phi'_i}) \cdot e(g^{h_{2,i}s_2}, g^{\theta_1 \phi'_i} g^{-\lambda'_1 v_i t_{1,i}} g_2^{-d\theta_1})$$

$$\cdot e(g^{\theta_2 x_i s_4}, g^{-\lambda'_1 v_i t_{1,i}}) \cdot e(g^{t_{2,i}s_1}, g_2^{-d\theta_1})$$

$$= \prod_{i=1}^{n} e(g^{-\delta_2 w_{1,i}}, g^{r_i^{'} s_1}) \cdot e(g^{s_2}, (g^{-\delta_2 r_i^{'}} g^{\lambda_1^{'} v_i w_{2,i}} g_2^{d\delta_2})^{f_{1,i}})$$

$$\cdot e(g,g)^{\lambda_1^{'}\delta_1 w_{2,i} x_i v_i s_3} \cdot e(g^{w_{1,i} s_1}, g_2^{d\delta_2}) \cdot e(g^{-\delta_1 w_{2,i}}, g^{r_i^{'} s_1})$$

$$\cdot e(g^{s_2}, (g^{\delta_1 r_i^{'}} g^{-\lambda_1^{'} v_i w_{1,i}} g_2^{d\delta_1})^{f_{2,i}}) \cdot e(g,g)^{\lambda_1^{'}\delta_2 w_{1,i} x_i v_i s_3}$$

$$\cdot e(g^{w_{2,i} s_1}, g_2^{-d\delta_1}) \cdot e(g^{-\theta_2 t_{1,i}}, g^{\phi_i^{'} s_1}) \cdot e(g^{s_2}, (g^{-\theta_2 \phi_i^{'}} g^{\lambda_1^{'} v_i t_{2,i}} g_2^{d\theta 2})^{h_{1,i}})$$

$$\cdot e(g,g)^{\lambda_2^{'}\theta_1 t_{2,i} x_i v_i s_4} \cdot e(g^{t_{1,i} s_1}, g_2^{d\theta_2}) \cdot e(g^{\theta_1 t_{2,i}}, g^{\phi_i^{'} s_1})$$

$$\cdot e(g^{s_2}, (g^{\theta_1 \phi_i^{'}} g^{-\lambda_1^{'} v_i t_{1,i}} g_2^{-d\theta 1})^{h_{2,i}}) \cdot e(g,g)^{-\lambda_2^{'}\theta_2 t_{1,i} x_i v_i s_4}$$

$$\cdot e(g^{t_{2,i} s_1}, g_2^{-d\theta_1}) \cdot$$

$$= \prod_{i=1}^{n} e(g^{\delta_1 w_{2,i} - \delta_2 w_{1,i}}, g^{r'_i s_1}) \cdot e(g^{\theta_1 t_{2,i} - \theta_2 t_{1,i}}, g^{\phi' i s_1})$$

$$\cdot e(g^{s_2}, K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}} K_{3,i}^{'h_{1,i}} K_{4,i}^{'h_{2,i}})$$

$$\cdot e(g,g)^{[\lambda_1^{'}(\delta_1 w_{2,i} - \delta_2 w_{1,i}) s_3 + \lambda_2^{'}(\theta_1 t_{2,i} - \theta_2 t_{1,i}) s_4] x_i v_i}$$

$$\cdot e(g^{-\delta_1 w_{2,i} + \delta_2 w_{1,i}}, g_2^{ds_1}) \cdot e(g^{-\theta_1 t_{2,i} + \theta_2 t_{1,i}}, g_2^{ds_1})$$

$$= e(g^{\Omega s_1}, \prod_{i=1}^{n} g^{(r_i^{'} + \phi_i^{'})}) \cdot e(g^{s_2}, \prod_{i=1}^{n} K_{1,i}^{'f_{1,i}} K_{2,i}^{'f_{2,i}} K_{3,i}^{'h_{1,i}} K_{4,i}^{'h_{2,i}})$$

$$\cdot e(g,g)^{\Omega(\lambda_1^{'} s_3 + \lambda_2^{'} s_4) < \vec{x}, \vec{v} >} \cdot e(g^{-\Omega}, g_2^{ds_1}).$$

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

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

$$\cdot e(C_{3,i}, K_{3,i}^{'}) \cdot e(C_{4,i}, K_{4,i}^{'})$$

$$= e(g^{s_2} g_2 \prod_{i=1}^{n} {K'_{1,i}}^{-f_{1,i}} {K'_{2,i}}^{-f_{2,i}} {K'_{3,i}}^{-h_{1,i}} {K'_{4,i}}^{-h_{2,i}}) \cdot e(g^{\Omega s_1}, \prod_{i=1}^{n} g^{-(r'_i+\phi'_i)})$$

$$\cdot e(g^{\Omega s_1}, \prod_{i=1}^{n} g^{(r'_i+\phi'_i)}) \cdot e(g^{s_2}, \prod_{i=1}^{n} {K'_{1,i}}^{f_{1,i}} {K'_{2,i}}^{f_{2,i}} {K'_{3,i}}^{h_{1,i}} {K'_{4,i}}^{h_{2,i}})$$

$$\cdot e(g^{-\Omega}, g_2^{ds_1}) \cdot e(g,g)^{\Omega(\lambda'_1 s_3 + \lambda'_2 s_4) < \overrightarrow{x}, \overrightarrow{v} >}$$

$$= e(g^{s_2}, g_2) \cdot e(g,g)^{\Omega(\lambda'_1 s_3 + \lambda'_2 s_4) < \overrightarrow{x}, \overrightarrow{v} >} \cdot e(g^{-\Omega}, g_2^{ds_1}).$$

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

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

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

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

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

It is worth noting that the term $e(g,g_2)^{s_2}$ is generated from the pairing computation $e(A, K_A) = e(g_2 \prod_{i=1}^{n} K_{1,i}{}^{-f_{1,i}}{}_{2,i}^{-f_{2,i}} K_{3,i}{}^{-h_{1,i}} K_{4,i}{}^{-h_{2,i}})$. Thus, the output of the above result is *M* if $< \overrightarrow{x}, \overrightarrow{v} >= 0$ in $\mathbb{Z}_p^*$. If $< \overrightarrow{x}, \overrightarrow{v} > \neq 0$ 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 $CT_{\overrightarrow{x}}$ is a re-encrypted well-formed ciphertext, then it is of the form $CT'_{\overrightarrow{x}} = (A, B, CT_{\overrightarrow{w}}, \hat{CT}, D = e(g, g_2)^{-s_2} M)$. The algorithm first decrypts $CT_{\overrightarrow{w}}$ using $(SK_{\overrightarrow{v'}}$ as above to obtain g_2^d as Decrypt $(SK_{\overrightarrow{v'}}, CT_{\overrightarrow{w}}) \rightarrow g_2^d$.

   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 $\overrightarrow{x}$ associated with the ciphertext and the vector $\overrightarrow{v}$ associated with the private key m orthogonal vectors (e.g., $< \overrightarrow{x}, \overrightarrow{v} >= 0$).

Correctness.

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

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

The result outputs M if $<\overrightarrow{x},\overrightarrow{v}>=0$ in $\mathbb{Z}_p^*$. If $<\overrightarrow{x},\overrightarrow{v}>\neq 0$ 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.

Level-1 access. If $CT_{\overrightarrow{x}}$ is an original well-med ciphertext, then algorithm decrypts $CT_{\overrightarrow{x}} = (A,B,\{C_{1,i},C_{2,i}\}_{i=1}^n,\{C_{3,i},C_{4,i}\}_{i=1}^n, D=e(g,g_2)^{-s_2}M)$ using the private key:

$SK_{\overrightarrow{v}}=(K_A,K_B,\{K_{1,i},K_{2,i}\}_{i=1}^n,\{K_{3,i},K_{4,i}\}_{i=1}^n)$ to output message M:

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

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:

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

where both $e(g^{w_{1,i}s_1},g^{\lambda_1 v_i w_{2,i}})$ and $e(g^{\delta_{1,i}x_i s_3},g^{-\delta_2 r_i})$ are canceled out. Additionally, we need to remove $e(g^{w_{1,i}s_1},g^{-\delta_2 r_i}) \cdot e(g^{w_{2,i}s_1},g^{\delta_1 r_i})$ 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 $CT_{\overrightarrow{x}}$ is well-formed the vector $\overrightarrow{x}=x_1,\ldots,x_n$. Then, we have:

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

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 $< \overrightarrow{x}, \overrightarrow{v} >= 0$ in $\mathbb{Z}_p^*$. If $< \overrightarrow{x}, \overrightarrow{v} > \neq 0$ 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.

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 $\overrightarrow{x}$ associated with the ciphertext and the vector $\overrightarrow{v}$ associated with the private key m orthogonal vectors (e.g., $< \overrightarrow{x}, \overrightarrow{v} >= 0$).

Correctness.

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

$$e(g, g_2)^{-s_2} M \cdot e(g^{s_2}, g_2) \cdot e(g, g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \overrightarrow{x}, \overrightarrow{y} >} \cdot e(g^{-\Omega}, g_2^{ds_1})$$
$$\cdot e(g^{s_1 \Omega}, g_2^d) = e(g, g_2)^{-s_2} M \cdot e(g, g_2)^{s_2} \cdot e(g, g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \overrightarrow{x}, \overrightarrow{y} >}$$
$$\cdot e(g, g_2)^{-s_1 \Omega d} \cdot e(g, g_2)^{s_1 \Omega d} = M \cdot e(g, g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \overrightarrow{x}, \overrightarrow{y} >}.$$

The result outputs M if $< \overrightarrow{x}, \overrightarrow{v} >= 0$ in $\mathbb{Z}_p^*$. If $< \overrightarrow{x}, \overrightarrow{v} > \neq 0$ in $\mathbb{Z}_p^*$, then there is only such case that $(\lambda_1^{'} s_3 + \lambda_2^{'} s_4)$ in $\mathbb{Z}_p^*$ with probability at most $1/p$, as in the predicate-only *IPE* scheme.

6.2 Proof of Security

Here, we describe a mechanism to show that our proposed scheme achieves the security requirements according to the definitions stated in the Section 3. For *Level-1* and *Level-2* ciphertext challenge, an adversary may request private key, re-encryption key and re-encryption queries by choosing vectors $(\overrightarrow{x}_0, \overrightarrow{x}_1, \overrightarrow{w}_0, \overrightarrow{w}_1)$ at the beginning of the security game. For instance, in the case of *Level-1* access, the adversary outputs two vectors $\overrightarrow{x}_0, \overrightarrow{x}_1$ and queries corresponding to a vector $\overrightarrow{v}$ such that $< \overrightarrow{v}, \overrightarrow{x}_0 >=< \overrightarrow{v}, \overrightarrow{x}_1 >= 0$ 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 $\overrightarrow{x}_0, \overrightarrow{x}_1$ along with $\overrightarrow{w}_0, \overrightarrow{w}_1$ for re-encryption keys. The adversary goal is to decide which one of the two vectors $\overrightarrow{w}_0, \overrightarrow{w}_1$ is associated with the re-encrypted query.

To prove the *Level-1* access, similarly to [14] 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 $(\overrightarrow{a}, \overrightarrow{b})$ denote a ciphertext encrypted using $\vec{0}$ vector (that is orthogonal to everything) in intermediate game to prove indistinguishably when encrypting to $\overrightarrow{x}_0$ corresponding to $(\overrightarrow{x}_0, \overrightarrow{x}_0)$ and when encrypting to $\overrightarrow{x}_1$ corresponding to $(\overrightarrow{x}_1, \overrightarrow{x}_1)$ as:

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

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 [14]. It determines whether a sub-system encrypts the given vector or the zero vector. Details of this proof is given in [26].

To prove the *Level-2* access, we apply game transformation proof [15] with a multiple sequence 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 $s_{M,\overrightarrow{x},\overrightarrow{v}}$ mentioned in the Definition 5. This value holds the following claims:

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

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.

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

Proof. We execute a preliminary game transformation from Game 0 (original game in Definition 5) to Game $0^{'}$, which is the same as Game 0 except flip a coin $\tau_{M,\overrightarrow{x},\overrightarrow{v}}$ before setup, and the game is aborted at the final step if $\tau_{M,\overrightarrow{x},\overrightarrow{v}} \neq s_{M,\overrightarrow{x},\overrightarrow{v}}$. Hence, the advantage of Game $0^{'}$ is a half of that in Game 0. The value $\tau_{M,\overrightarrow{x},\overrightarrow{v}}$ is chosen independently from $s_{M,\overrightarrow{x},\overrightarrow{v}}$, and therefore the probability that the game is aborted is $\frac{1}{2}$ that is $Adv_{\mathcal{A}}^{(0^{'})}(\lambda) = \frac{1}{2} \cdot Adv_{\mathcal{A}}^{\text{PAH-L2}}(\lambda)$. Moreover, $\Pr[\mathcal{A}\ wins] = \frac{1}{2}(\Pr(\mathcal{A}\ wins | \tau_{M,\overrightarrow{x},\overrightarrow{v}} = 0) + (\Pr(\mathcal{A}\ wins | \tau_{M,\overrightarrow{x},\overrightarrow{v}} = 1))$ in Game $0^{'}$.

Hence, we have:

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

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

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

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

For any adversary $\mathcal{A}$, there exists 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 $\tau_{M,\overrightarrow{x},\overrightarrow{v}=0}$.

$$\Pr[\mathcal{A}_{wins}|\tau_{M,\overrightarrow{x},\overrightarrow{v}} = 0] - \frac{1}{2} \leq \mathrm{Adv}_{\epsilon_{1-1}}^{IPE\text{-}AH} + \mathrm{Adv}_{\epsilon_{1-2}}^{IPE\text{-}AH}.$$

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 $\overrightarrow{w}_b$ is changed to $CT_{\vec{r}} = \text{Encrypt } (PK, \vec{r}, R)$ where $\overrightarrow{r}$ chosen uniformly random from Σ and random value $R \in \mathbb{G}_T$.

In the case $\tau_{M,\overrightarrow{x},\vec{v}} = 0$, 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 $\text{Encrypt}_{IPE}\ (PK, \vec{r}, R)$ 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 $\tau_{M,\vec{x},\vec{v}} \overset{U}{\longleftarrow} \{0,1\}$ 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:

$$\begin{aligned}
&(\boldsymbol{A} = g^{s_2}, \boldsymbol{B} = {g_1}^{s_1} = g^{s_1\Omega}, \Lambda = e(g,g_2)^{-s_2},\\
&\boldsymbol{C}T_{\vec{x}_b} = \text{Encrypt } (PK, \vec{x}_b, M_b)\\
&\quad = (g^{s_2}, {g_1}^{s_1}, \{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_{ib}s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_{ib}s_3}\}_{i=1}^n,\\
&\qquad \{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_{ib}s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_{ib}s_4}\}_{i=1}^n, \Lambda^{-s_2} M_b) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}},\\
&\boldsymbol{C}T_{\vec{w}_b} = \text{Encrypt } \left(PK, \vec{w}_b, g_2^d\right)\\
&\quad = (g^{s_2}, {g_1}^{s_1}, \{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{w_{ib}s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{w_{ib}s_3}\}_{i=1}^n,\\
&\qquad \left\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{w_{ib}s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{w_{ib}s_4}\right\}_{i=1}^n, \Lambda^{-s_2} g_2^d\Big) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}},
\end{aligned}$$

$$\hat{\boldsymbol{C}T} = e(g^{s_2}, g_2) \cdot e(g,g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \vec{x}_b, \vec{v}_b >} \cdot e(g^{-\Omega}, g_2^{ds_1})).$$

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

$$\begin{aligned}
\boldsymbol{C}T_{\vec{r}} &= \text{Encrypt}\left(\boldsymbol{PK}, \vec{r}, g_2^{d'}\right) \\
&= (g^{s_2}, {g_1}^{s_1}, \{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{r_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{r_i s_3}\}_{i=1}^n, \\
&\quad \{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{r_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{r_i s_4}\}_{i=1}^n, \Lambda^{-s_2} {g_2}^{d'}) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}}, \\
\hat{\boldsymbol{C}T} &= e(g^{s_2}, g_2) \cdot e(g,g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \vec{x}_b, \overrightarrow{v}_b >} \cdot e(g^{-\Omega}, g_2^{d' s}),
\end{aligned}$$

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 the follows:

$$\begin{aligned}
\boldsymbol{C}T_{\vec{u}} &= \text{Encrypt}\,(\boldsymbol{PK}, \vec{u}, M_b) \\
&= (g^{s_2}, {g_1}^{s_1}, \{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{u_i s_3}, W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{u_i s_3}\}_{i=1}^n, \\
&\quad \{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{u_i s_4}, T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{u_i s_4}\}_{i=1}^n, \Lambda^{-s_2} M_b) \in \mathbb{G}^{4n+2} \times \mathbb{G}_{\mathbb{T}}, \\
\hat{\boldsymbol{C}T} &= e(g^{s_2}, g_2) \cdot e(g,g)^{\Omega(\lambda_1' s_3 + \lambda_2' s_4) < \vec{u}, \vec{u'} >} \cdot e(g^{-\Omega}, g_2^{ds_1})),
\end{aligned}$$

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 $\text{Adv}_{\mathcal{A}}^{(0')}(\lambda)$, $\text{Adv}_{\mathcal{A}}^{(1)}(\lambda)$ and $\text{Adv}_{\mathcal{A}}^{(2)}(\lambda)$ 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:

$$\begin{aligned}
\text{Adv}_{\mathcal{A}}^{(0')}(\lambda) &\leq |\text{Adv}_{\mathcal{A}}^{(0')}(\lambda) - \text{Adv}_{\mathcal{A}}^{(1)}(\lambda)| + |\text{Adv}_{\mathcal{A}}^{(1)}(\lambda) - \text{Adv}_{\mathcal{A}}^{(2)}(\lambda)| \\
&\quad + \text{Adv}_{\mathcal{A}}^{(2)}(\lambda) \leq \text{Adv}_{\beta_{1-1}}^{IPE\text{-}AH}(\lambda) + \text{Adv}_{\beta_{1-2}}^{IPE\text{-}AH}(\lambda).
\end{aligned}$$

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

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 $|\mathrm{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda) - \mathrm{Adv}_{\mathcal{A}}^{(1^{'})}(\lambda)| \leq \mathrm{Adv}_{\beta_{1-1}}^{IPE\text{-}AH}(\lambda)$, 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 secret key and provides $\mathcal{A}$ with the public key and keeps the secret key as details are stated in Section 6.

$$\begin{aligned} \textit{PK} = (g, g_1, \{W_{1,i}, W_{2,i}, F_{1,i}F_{2,i}\}_{i=1}^{n}, \{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\}_{i=1}^{n}, \\ \{U_i, V_i\}_{i=1}^{2}, \Lambda) \\ \textit{MSK} = (\{w_{1,i}, w_{2,i}, t_{1,i}, t_{2,i}, f_{1,i}, f_{2,i}, h_{1,i}, h_{2,i}\}_{i=1}^{n}, \{\delta_i, \theta_i\}_{i=1}^{2}, g_2). \end{aligned}$$

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 $CT_{\vec{w}} = (\textit{PK}, \vec{w}, {g_2}^d)$.

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 $b \overset{U}{\leftarrow} \{0, 1\}$ 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 1) where *R* and $\vec{r}$ are chosen independently uniform. It then receives $CT_{\vec{w}\beta}$ for $\beta \overset{U}{\leftarrow} \{0, 1\}$. 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, $|\mathrm{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathrm{Adv}_{\mathcal{A}}^{(1')}(\lambda)| \leq |Pr[b = b^{'}] - \frac{1}{2}|\mathrm{Adv}_{\beta_{1-1}}^{IPE\text{-}AH}(\lambda)$.

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 $\vec{r} \xrightarrow{U} \mathcal{F}$. Hence, we have $|\mathrm{Adv}_{\mathcal{A}}^{(1^{'})}(\lambda) - \mathrm{Adv}_{\mathcal{A}|}^{(1)}(\lambda)| \leq \mathrm{Adv}_{\beta_{1-2}}^{IPE\text{-}AH}(\lambda)$. Therefore, we can prove this Lemma by using hybrid argument.

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

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

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

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

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 exists 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 $\tau_{M,\vec{x},\vec{v}=1}$.

$$\Pr[\mathcal{A}_{wins}|T_{M,\vec{x},\vec{v}} = 0] - \frac{1}{2} \leq \mathrm{Adv}_{\epsilon_{2-1}}^{IPE\text{-}AH} + \mathrm{Adv}_{\epsilon_{2-2}}^{PAH\text{-}AH}.$$

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 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:

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

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:

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

Let $\text{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda)$ and $\text{Adv}_{\mathcal{A}}^{(1)}(\lambda)$ 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:

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

The proof is completed from the Lemma 7 since $\text{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda) \leq \frac{1}{2}(\text{Adv}_{\beta_{2-1}}^{IPE\text{-}AH}(\lambda) + \text{Adv}_{\beta_{2-2}}^{IPE\text{-}AH}(\lambda))$.

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

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

Lemma 7. For any adversary $\mathcal{A}$, $|\mathrm{Adv}_{\mathcal{A}}^{(1)}(\lambda) = -\mathrm{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda)$. 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 $\mathrm{Adv}_{\mathcal{A}}^{(1)}(\lambda) = -\mathrm{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda)$.

7 Experimental Evaluation

In this section, 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 a total number of attributes in the system is equal to *n*.

7.1 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 of the Trusted Authority.** The trusted authority has to execute 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 computed 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 for 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: $A = g^{s_1}$, $B = g^{s_1\Omega}, \{C_{1,i}, C_{2,i}\}_{i=1}^{n}$ 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.

7.2 Experimental Results

We implemented our scheme in *C* using the Pairing-Based Crypto (*PBC*) library [23]. 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* [23]. The parameters of each curve are shown in Table 1.

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

Table 1 Curve Parameters

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	$y^2 = x^3 + x$	$y^2 = x^3 + ax + b$	$y^2 = x^3 + ax + b$	$y^2 = x^3 + b$

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 *MNT*159 and *MNT*201) 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 7.1).

The computational overhead of Encrypt algorithm is dominated by exponentiation operation on group $\mathbb{G}_1$ and therefore *MNT*159 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 3, for 5 attributes, the Encrypt algorithm takes about 19ms under *MNT*159 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 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.

The embedding degree of elliptic curves directly influences the size of $\mathbb{G}_T$ and increases the complexity of pairing computation. Therefore, the

Table 2 Average execution time (ms) of each algorithm of the proposed *IPPRE* scheme on eliptic curves *SS* and *MNT159*

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

[2]The base field size of $\mathbb{G}_1$ *MNT*159, *BN*, *MNT*201 and *SS* curves are 159 bits, 160 bits, 201 bits and 512 bits, respectively [23].

[3]The base field size of $\mathbb{G}_2$ *BN*, *MNT*159, *SS* and *MNT*201 curves are 320 bits, 477 bits, 512 bits and 603 bits, respectively [23].

Table 3 Average execution time (ms) of each algorithm of the proposed *IPPRE* scheme on eliptic curves *MNT201* and *BN*

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

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 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 2 and 3 we can see that, in the case of 5 attributes the Decrypt and Re-Decrypt algorithms take less than 18ms 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.

8 Conclusions

In this paper, we extend Park's inner-product encryption method [26]. We present a new inner-product proxy re-encryption (*IPPRE*) scheme for sharing data among users with different access policies via the proxy server. The proposed scheme allows updating attribute sets without re-encryption, making policy updates extremely efficient. We fulfill the security model for our *IPPRE* and show that the scheme is adaptive attribute-secure against chosen plaintext under standard Decisional Linear (*D-Linear*) assumption. Moreover, we test the execution time of each algorithm of our protocol on different types of elliptic curves. The execution times hint that our approach is the first step towards a promising direction. As a future work, we plan to show how to apply our scheme to achieve a multi-authority version [6].

References

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9, 1–30.

[2] J. Bethencourt, A. Sahai, and B. Waters (2007). Ciphertext-policy attribute-based en cryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, *SP '07*, (IEEE Computer Society: Washington, DC, USA), 321–334.

[3] M. Blaze, G. Bleumer, and M. Strauss (1998). *Divertible Protocols and Atomic Proxy Cryptography*. Springer: Berlin, Heidelberg, 127–144.

[4] D. Boneh, X. Boyen, and H. Shacham (2004). *Short Group Signatures*. Springer: Berlin, Heidelberg, pp. 41–55.

[5] D. Boneh and M. Franklin (2001). *Identity-Based Encryption from the Weil Pairing*. Springer: Berlin, Heidelberg, 213–229.

[6] M. Chase (2007). Multi-authority attribute based encryption. In *Proceedings of the 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands.* (ed) Salil P. Vadhan, *Theory of Cryptography*, Lecture Notes in Computer Science, Vol. 4392, (Springer), pp. 515–534.

[7] L. Cheung and Calvin C (2007). Newport. Provably secure ciphertext policy ABE. *IACR Cryptology ePrint Archive*, 2007:183.

[8] T. El Gamal (1985). "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proceedings of CRYPTO 84 on Advances in Cryptology*, New York, NY, USA, (Springer-Verlag New York, Inc.), 10–18.

[9] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi (2009). *A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length*. Springer: Berlin, Heidelberg, 13–23.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters (2006). "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*, (ACM: New York, NY, USA), 89–98.

[11] M. Green and G. Ateniese (2007). *Identity-Based Proxy Re-encryption*. Springer: Berlin, Heidelberg, 288–306.

[12] H. Guo, F. Ma, Z. Li, and C. Xia (2015). "Key-exposure protection in public auditing with user revocation in cloud storage," in *Revised Selected Papers of the 6th International Conference on Trusted Systems, INTRUST 2014*, (LNCS, Vol. 9473), 127–136.

[13] S. Guo, Y. Zeng, J. Wei, and Q. Xu (2008). Attribute-based re-encryption scheme in the standard model. *Wuhan University Journal of Natural Sciences*, 13, 621–625.

[14] J. Katz, A. Sahai, and B. Waters (2008). *Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products*. Springer: Berlin, Heidelberg, 146–162.
[15] Y. Kawai and K. Takashima (2013). Fully-anonymous functional proxy-re-encryption. *IACR Cryptology ePrint Archive*, 2013:318.
[16] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters (2010). *Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption.* Springer: Berlin, Heidelberg, 62–91.
[17] H. Li and L. Pang (2016). Efficient and adaptively secure attribute-based proxy reencryption scheme. *IJDSN*, 12, 5235714:1–5235714:12.
[18] K. Li (2013). Matrix access structure policy used in attribute-based proxy re-encryption. *CoRR.* abs/1302.6428, eprint: arXiv:1302.6428.
[19] K. Liang, L. Fang, W. Susilo, and D. S. Wong (2013). "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, 552–559.
[20] K. Liang, M. H. Au, W. Susilo, D. S. Wong, G. Yang, and Y. Yu (2014). *An Adaptively CCA-Secure Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Cloud Data Sharing*. Springer International Publishing: Cham, 448–461.
[21] X. Liang, Z. Cao, H. Lin, and J. Shao (2009). "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,* 276–286.
[22] S. Luo, J. Hu, and Z. Chen (2010). "Ciphertext policy attribute-based proxy re-encryption," in *Proceedings of the 12th International Conference on Information and Communications Security*, *ICICS'10*, (Springer-Verlag: Berlin, Heidelberg), 401–415.
[23] B. Lynn (2007). *Pairing Based Cryptography Library*. Available at: http://crypto.stanford.edu/pbc/
[24] M. Mambo and E. Okamoto (1997). Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 80A, 54–63.
[25] T. Okamoto and K. Takashima (2009). *Hierarchical Predicate Encryption for Inner-Products*. Springer: Berlin, Heidelberg, 214–231.
[26] J. H. Park (2011). Inner-product encryption under standard assumptions. *Des. Codes Cryptography*, 58, 235–257.

[27] Z. Qin, H. Xiong, S. Wu, and J. Batamuliza (2017). A survey of proxy re-encryption for secure data sharing in cloud computing. *IEEE Transactions on Services Computing*, PP(99), 1–1. doi: 10.1109/TSC.2016.2551238

[28] A. Sahai and B. Waters (2005). *Fuzzy Identity-Based Encryption.* Springer: Berlin, Heidelberg, 457–473.

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

[30] M. Sepehri, S. Cimato, and E. Damiani (2017). "Efficient implementation of a proxy-based protocol for data sharing on the cloud," in *Proceedings of the Fifth ACM International Workshop on Security in Cloud Computing, SCC@AsiaCCS 2017,* Abu Dhabi, United Arab Emirates, April 2, 2017, pp. 67–74.

[31] M. Sepehri, S. Cimato, E. Damiani, and C. Y. Yeuny (2015). "Data sharing on the cloud: A scalable proxy-based protocol for privacy-preserving queries," in *Proceedings of the 7th IEEE International Symposium on Ubisafe Computing in Conjunction with 14th IEEE Conference on Trust, Security and Privacy in Computing and Communications, TrustCom/BigDataSE/ISPA,* Helsinki, Finland, 1357–1362.

[32] M. Sepehri, and A. Trombetta (2017). Secure and Efficient Data Sharing with Atribute-based Proxy Re-encryption Scheme. In *Proceedings of the 12th International Conference on Availability, Reliability and Security.* ACM: Reggio Calabria, Italy, 1–63.

[33] D. Thilakanathan, S. Chen, S. Nepal, and R. A. Calvo (2014). *Secure Data Sharing in the Cloud.* Springer: Berlin, Heidelberg, 45–72.

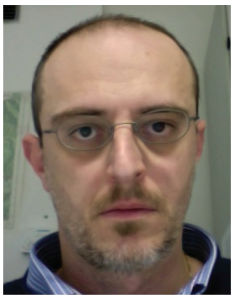
[34] B. Waters (2009). *Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions*. Springer: Berlin, Heidelberg, 619–636.

[35] B. Waters (2011). *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*. Springer: Berlin, Heidelberg, 53–70.

[36] S. Yu, C. Wang, K. Ren, and W. Lou (2010). "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, *ASIACCS '10*, (ACM: New York, NY, USA), 261–270.

Biographies

Masoomeh Sepehri is a Ph.D. candidate in Computer Science at the University of Milan. Her research activity mainly focused on the design of cryptographic protocols for secure data sharing in the cloud computing and Internet of Things.



Alberto Trombetta is Associate Professor at the Department of Scienze Teoriche e Applicate at Insubria University in Varese, Italy. His main research interests include security and privacy issues in data management systems, applied cryptography and trust management.

Maryam Sepehri received the Ph.D. degree in Computer Science from the University of Milan, Italy, in 2014. She is currently pursuing postdoctoral research fellow at the University of Waterloo, Canada. Her research interest includes privacy-preserving query processing over encrypted data.