Privacy Preserving Revocable Predicate Encryption Revisited

Kwangsu Lee[*] Intae Kim[†] Seong Oun Hwang[‡]

Abstract

Predicate encryption (PE) that provides both the access control of ciphertexts and the privacy of ciphertexts is a new paradigm of public-key encryption. An important application of PE is a searchable encryption system in cloud storage, where it enables a client to securely outsource the search of a keyword on encrypted data without revealing the keyword to the cloud server. One practical issue of PE is to devise an efficient revocation method to revoke a user when the secret key of the user is compromised. Privacy preserving revocable PE (RPE) can provide not only revocation, but also the privacy of revoked users. In this paper, we first define two new security models of privacy preserving RPE: the strongly full-hiding security and the weakly full-hiding security. The strongly full-hiding security provides the full privacy of ciphertexts against outside and inside adversaries, but the weakly full-hiding security provides the full privacy of ciphertexts against an outside adversary who cannot decrypt the challenge ciphertext. Next, we propose a general RPE construction from any PE scheme, and prove its security in the weakly full-hiding security model. Our generic RPE scheme is efficient since the number of ciphertext elements is not proportional to the number of users in a receiver set. Additionally, our RPE scheme can support polynomial-size circuits if a recently proposed FE scheme for polynomial-size circuits is used as an underlying PE scheme.

Keywords: Public-key encryption, Predicate encryption, Revocation, Privacy, Adaptive security.

1 Introduction

Predicate encryption (PE) is a special type of public-key encryption (PKE) and the concept of PE was introduced by Boneh and Waters [11]. PKE is suitable for traditional one-to-one communication environments since only the receiver who was specified by a sender can decrypt the ciphertext of the sender. PE, by contrast, can be used for more complex communication environments of today since any user who has a secret key for specific ciphertexts can decrypt the ciphertexts [11, 16]. In PE, a ciphertext is associated with an attribute x and a secret key is associated with a predicate f, and a user who has a secret key associated with f can decrypt a ciphertext associated with x if $f(x) = 1$. Furthermore, PE provides not only the message hiding property but also the attribute hiding property.

One important application of PE is the searches on encrypted data in outsourced storage. For example, the outsourced storage of a hospital can store the ciphertext of patient's health information by attaching a keyword x to the ciphertext. A researcher can request a secret key for his research area f to an authorized center and receives a secret key associated with f. After that, the researcher can access to a limited set of ciphertexts in the outsourced storage if $f(x) = 1$. In this case, the researcher cannot access to other

[*]Korea University, Korea. Email: `guspin@korea.ac.kr`. This work was partially done at Columbia University.
[†]Hongik University, Korea. Email: `ittaros.kim@gmail.com`.
[‡]Hongik University, Korea. Email: `sohwang@hongik.ac.kr`.

ciphertexts such that $f(x) \neq 1$. Therefore, PE not only protects the privacy of patients, but also enables researchers to use the health information of patients.

One difficulty of using the PE schemes in real applications is the lack of an efficient revocation method that can revoke a user whose secret key is compromised. For example, if the secret key of a researcher in the outsourced storage of the hospital is leaked, then the secret key of the researcher should be revoked to prevent an adversary from using the secret key to access the information of patients. Thus revocation is a very important mechanism to preserve the whole security of a system. In PKE, an efficient revocation method was easily achieved by using the certificate revocation list (CRL) since a sender can check the validity of a receiver's certificate before he sends a ciphertext. However, it is not easy to apply the revocation method of PKE to PE since there is no user's certificate in PE. In identity-based encryption (IBE), Boneh and Franklin [8, 9] proposed a revocation method such that an authority regularly issues the secret key of a user for an identity and a current time period. However, this method does not provide an efficient revocation method in PE since the authority should be online to generate the secret keys of users and a secure channel between the authority and a user should be established. Another revocation method is for an authority to periodically publish secret key update information [6]. Though this approach can solve the previous problems, it still requires unrevoked users to update their secret keys regularly.

To revoke a user without updating the user's secret key, we may consider the sender-local revocation method such that the authority periodically publishes a revoked user list and a sender constructs a ciphertext only for the non-revoked users by using the current revoked user list of the authority. Nieto et al. [21] proposed revokable PE (RPE) schemes by adapting this method. Their first scheme supports both the attribute hiding property and the message hiding property, but their second scheme additionally supports the privacy of revoked users. Their privacy preserving RPE scheme is appealing since it try to hide all information of ciphertexts. However, their privacy preserving RPE scheme is only secure in a restricted security model, and it can only be constructed from a specific PE scheme. Therefore, very natural, but interesting questions arise:

> Can we build an RPE scheme that provides the stronger privacy of revoked users? Can we construct an RPE scheme from any PE scheme?

1.1 Our Contributions

In this paper, we address these questions and answer them in positive directions.

New Security Models for RPE. We first introduce two new security models for the privacy preserving RPE schemes. In these security models, we divide adversaries into two types: an outside adversary who can not decrypt the challenge ciphertext since he cannot obtain a secret key for the decryption of the ciphertext and an inside adversary who can decrypt the ciphertext since he can obtain a secret key for the decryption of the ciphertext. The first security model, called *strongly full-hiding* security, is the best achievable security model of privacy preserving RPE schemes. It provides adaptive security against outside and inside adversaries. By considering the insider adversary, we guarantee that even a valid receiver who has a secret key for a predicate f and an index u cannot obtain the partial information of an attribute x and a revoked user set R in the ciphertext except that his predicate of PE matches (e.g. $f(x) = 1$) and he does not belongs to the revoked set (e.g. $u \notin R$). Although this security model is the best possible one, it is currently only possible to construct a privacy preserving RPE scheme directly encoding the receiver set $S = \mathcal{N} \setminus R$ into the attribute in a PE scheme if the PE scheme supports subset predicates where $\mathcal{N}$ is the set of all user indexes. In this case, the number of ciphertext elements is proportional to the size of the receiver set S. The second security model, called *weakly full-hiding* security, provides adaptive security against an outside adversary, but it does

Table 1: The comparison of privacy preserving RPE schemes

Scheme	Security	Predicate	SK Size	CT Size
NMS [21]	wFH-limits	Inner product	$O(l+\log N_{max})$	$O(l+r\log(N_{max}/r))$
Ours with IPE [22]	wFH	Inner product	$O(\log N_{max})$	$O(l\cdot r\log(N_{max}/r))$
Ours with FE [13]	wFH	Circuit	$O(\lvert f\rvert\cdot\log N_{max})$	$O(\lvert x\rvert\cdot r\log(N_{max}/r))$

wFH = weakly full-hiding security, wFH-limits = weakly full-hiding security with limitations
N_{max} = the number number of users, r = the maximum number of revoked users.

not provide the privacy of a revoked user set R in a ciphertext against an inside adversary. We should note that this security model still provides the privacy of an attribute x in a ciphertext against an inside adversary. By weakening the privacy against the inside adversary, we have an opportunity to build an efficient privacy preserving RPE scheme such that the number of ciphertext elements is not proportional to the size of the receiver set S. Compared to our weakly full-hiding security model, the security model of Nieto et al. [21] has several restrictions such that an inside adversary was not considered at all and the condition of challenge revoked sets R_0, R_1 that are submitted by an adversary is severely limited. Therefore, our weakly full-hiding security model is the right one for efficient and privacy preserving RPE schemes.

Generic Construction of RPE. Next we propose a generic privacy preserving RPE scheme that provides the weakly full-hiding security from any PE scheme. That is, we can instantiate our RPE scheme from a hidden vector encryption (HVE) scheme [11, 18, 25], an inner-product encryption (IPE) scheme [16, 22–24], or a functional encryption (FE) scheme for polynomial-size circuits [13] since it uses an underlying PE scheme as a black-box way. Note that if the underlying PE scheme provides selective security (or adaptive security), then our RPE scheme also provides selective and weakly full-hiding security (or adaptive and weakly full-hiding security). In our generic RPE scheme, the ciphertext consists of $O(r\log(N_{max}/r))$ number of the PE ciphertexts and the secret key consists of $(\log N_{max})$ number of the PE secret keys where N_{max} is the maximum number of users in the system and r is the number of revoked users. The comparison of RPE schemes is given in Table 1. Additionally, we can improve the efficiency of the decryption algorithm of our RPE scheme by using an anonymous hint system [5, 19].

1.2 Related Work

Functional Encryption. Functional encryption (FE) is a new paradigm of PKE and includes IBE [8, 9], hierarchical IBE (HIBE) [14], attribute-based encryption (ABE) [15, 27], and PE [10, 11, 16]. In FE, a ciphertext is associated with an attribute x and a secret key is associated with a function f. A user who has a private key for f can decrypt a ciphertext for x if $f(x) = 1$. PE includes anonymous IBE (AIBE) that supports equality queries [1, 7], hidden vector encryption (HVE) that supports conjunctive queries [11, 18, 28], and IPE that supports evaluations of polynomials [3, 16, 22–24].

Revocation in Functional Encryption. As mentioned, a simple revocation method for IBE was proposed by Boneh and Franklin [8, 9]. Their revocation method is for an authorized center to issue a user's secret key by attaching the current date to the identity of a user. If a user is revoked, then the authority center stops issuing the secret key of the user for a next date. However, this method has the problem of scalability since the authority should be online and all users are required to update their secret keys regularly. Additionally, this method is not applicable to ABE and PE since the private key of ABE or PE is associated with multiple attributes and the revocation of one attribute affects many users who have the same attribute.

To solve the scalability problem of previous approach, Boldyreva et al. [6] proposed another revocation method such that an authority publicly broadcasts key update information to non-revoked users. In this method, each user at first obtains a (long-term) secret key from the authority, and a sender constructs a ciphertext by encrypting a message with time information. After that, the authority regularly broadcasts key update information that contains the updated secret keys of non-revoked users. If a user is not revoked, then he can decrypt the ciphertext after deriving a (short-term) decryption key from his private key and the key update information. If a user is revoked, then he cannot derive a decryption key. This method can be applicable not only to IBE, but also to ABE and PE. Recently, Sahai et al. [26] proposed revocable-storage ABE schemes that can be used for cloud storage by extending this method. One disadvantage of this method is that non-revoked users should update theirs secret keys periodically.

A sender-local revocation method can revoke users without affecting the secret keys of non-revoked users. In this method, the authorized center first posts a revocation list, and then a sender constructs a ciphertext by directly including revocation information in a ciphertext. Attrapadung and Imai [4] proposed a broadcast ABE scheme. Recently, Nieto et al. [21] proposed revocable PE schemes to support revocation in PE.

Anonymous Broadcast Encryption. Anonymous broadcast encryption is a special type of broadcast encryption such that the revoked user set that is included in a ciphertext is also hidden from an adversary [5]. Privacy preserving RPE can be used for anonymous broadcast encryption since the encryption algorithm of RPE takes the revoked user set as an input and privacy preserving RPE hides the information of the revoked user set. Barth et al. [5] proposed the first private broadcast encryption scheme that has linear size of ciphertexts and proved its security under random oracle model. Libert et al. [19] proposed anonymous broadcast encryption schemes and proved their security without random oracles. To improve the efficiency of anonymous broadcast encryption, Fazio and Perera [12] proposed an outsider-anonymous broadcast encryption scheme with sublinear size of ciphertexts by weakening the security model of anonymity.

2 Preliminaries

In this section, we first define PE and its security model, and then we introduce the subset cover framework.

2.1 Predicate Encryption

As mentioned, PE is a special type of PKE where a ciphertext is associated with an attribute x and a secret key is associated with a predicate f [11]. In PE, a user who has a secret key with a predicate f that is given from an authority can decrypt a ciphertext with an attribute x if $f(x) = 1$ and additional information in the ciphertext except $f(x)$ is not revealed to an adversary. The syntax of PE is defined as follows:

Definition 2.1 (Predicate Encryption). *A predicate encryption (PE) scheme for the class $\mathcal{F}$ of predicates over the set Σ of attributes consists of four PPT algorithms **Setup**, **GenKey**, **Encrypt**, and **Decrypt**, which are defined as follows:*

Setup(1^λ). *The setup algorithm takes as input a security parameter 1^λ and outputs a public key PK and a master secret key MK.*

GenKey(f, MK, PK). *The key generation algorithm takes as input a predicate $f \in \mathcal{F}$ and the master secret key MK, and outputs a secret key SK_f.*

Encrypt(x, M, PK). *The encryption algorithm takes as input an attribute $x \in \Sigma$, a message $M \in \mathcal{M}$, and the public key PK, and outputs a ciphertext CT.*

Decrypt(CT, SK_f, PK). *The decryption algorithm takes as input a ciphertext CT and a secret key SK_f, and outputs a message M or the distinguished symbol $\perp$.*

The correctness property of PE is defined as follows: For all PK, MK generated by ***Setup****, all $f \in \mathcal{F}$, any SK_f generated by* ***GenKey****, all $x \in \Sigma$, and any $M \in \mathcal{M}$, it is required that*

- *If $f(x) = 1$, then* ***Decrypt***$($***Encrypt***$(x, M, PK), SK_f, PK) = M$.
- *If $f(x) = 0$, then* ***Decrypt***$($***Encrypt***$(x, M, PK), SK_f, PK) = \perp$ *with all but negligible probability.*

The second condition of the correctness property in PE is not a trivial one to satisfy since the decryption algorithm of PE cannot easily check whether $f(x) = 0$ or not. That is, the decryption algorithm cannot obtain x from a ciphertext because of anonymity. One possible relaxation is to use a computational condition instead of a statistical condition. For a computational condition, we can use weak robustness of Abdalla et al. [2].

PE provides not only the message hiding property (indistinguishability), but also the attribute hiding property (anonymity). The security model of PE was defined by Boneh and Waters [11]. The security of PE is defined as follows:

Definition 2.2 (Attribute-Hiding)**.** *The security notion of attribute-hiding under a chosen plaintext attack is defined in terms of the following experiment between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$:*

1. ***Setup****: $\mathcal{C}$ runs* ***Setup***(1^λ) *to generate a public key PK and a master secret key MK, and it gives PK to $\mathcal{A}$.*
2. ***Phase I****: $\mathcal{A}$ may adaptively request a polynomial number of secret keys for any predicates $f_1, \ldots, f_{q_1} \in \mathcal{F}$, and $\mathcal{C}$ gives the corresponding secret keys $SK_{f_1}, \ldots, SK_{f_{q_1}}$ to $\mathcal{A}$ by running* ***GenKey***(f_i, MK, PK).
3. ***Challenge****: $\mathcal{A}$ outputs challenge attributes $x_0, x_1 \in \Sigma$ and challenge messages $M_0, M_1 \in \mathcal{M}$ subject to the following restrictions:*
   - *For all predicate f_i of secret key queries, it is required that $f_i(x_0) = f_i(x_1)$.*
   - *If there is f_i in secret key queries such that $f_i(x_0) = f_i(x_1) = 1$, then it is required that $M_0 = M_1$.*

   $\mathcal{C}$ chooses a random bit b and gives the ciphertext CT to $\mathcal{A}$ by running ***Encrypt***(x_b, M_b, PK).
4. ***Phase II****: $\mathcal{A}$ may continue to request secret keys for additional predicates $f_{q_1+1}, \ldots, f_q$ subject to the same restrictions as before, and $\mathcal{C}$ gives the corresponding secrets keys to $\mathcal{A}$.*
5. ***Guess****: Finally $\mathcal{A}$ outputs a bit b'.*

The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv}_{\mathcal{A}}^{PE,AH}(\lambda) = |\Pr[b = b'] - \frac{1}{2}|$ where the probability is taken over all the randomness of the experiment. A PE scheme is (adaptively) attribute-hiding under a chosen plaintext attack if for all PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the above experiment is negligible in the security parameter λ.

2.2 Subset Cover Framework

The subset cover (SC) framework, introduced by Naor, Naor, and Lotspiech [20], is a general methodology to construct an efficient revocation system. They constructed revocation systems such that a center can encrypt a message for non-revoked users by combining the SC framework with a symmetric-key encryption scheme. The original SC framework consists of a subset-assignment part and a key-assignment part. We define the SC scheme by only including the subset-assignment part.

Definition 2.3 (Subset Cover)**.** *A subset cover (SC) scheme for the set $\mathcal{N} = \{1,\dots,N_{max}\}$ of users consists of four PPT algorithms **Setup**, **Assign**, **Cover**, and **Match**, which are defined as follows:*

***Setup**(N_{max}). The setup algorithm takes as input the maximum number N_{max} of users and outputs a collection $\mathcal{S}$ of subsets $S_1,\dots,S_w$ where $S_i \subseteq \mathcal{N}$.*

***Assign**($\mathcal{S},u$). The assigning algorithm takes as input the collection $\mathcal{S}$ and a user index $u \in \mathcal{N}$, and outputs a private set $PV_u = \{S_{j_1},\dots,S_{j_d}\}$ that is associated with the user index u.*

***Cover**($\mathcal{S},R$). The covering algorithm takes as the collection $\mathcal{S}$ and a revoked set $R \subset \mathcal{N}$ of users, and it outputs a covering set $CV_R = \{S_{i_1}\dots,S_{i_m}\}$ that is a partition of the non-revoked users $\mathcal{N} \setminus R$ into disjoint subsets $S_{i_1},\dots,S_{i_m}$, that is, they are disjoint, and it holds that $\mathcal{N} \setminus R = \bigcup_{k=1}^{m} S_{i_k}$.*

***Match**(CV_R, PV_u). The matching algorithm takes as input a covering set $CV_R = \{S_{i_1},\dots,S_{i_m}\}$ and a private set $PV_u = \{S_{j_1},\dots,S_{j_d}\}$. It outputs $(S_{i_k}, S_{j_{k'}})$ such that $S_{i_k} \in CV_R$, $u \in S_{i_k}$, and $S_{j_{k'}} \in PV_u$, or it outputs $\perp$.*

*The correctness property of SC is defined as follows: For all $\mathcal{S}$ generated by **Setup**, all PV_u generated by **Assign**, and any R, it is required that:*

- *If $u \notin R$, then $\textbf{Match}(\textbf{Cover}(\mathcal{S},R),PV_u) = (S_{i_k},S_{j_{k'}})$.*
- *If $u \in R$, then $\textbf{Match}(\textbf{Cover}(\mathcal{S},R),PV_u) = \perp$.*

We use the complete subtree (CS) scheme of Naor et al. [20] for our schemes. Before presenting the CS scheme, we define a full binary tree.

Full Binary Tree. A full binary tree $\mathcal{BT}$ is a tree data structure where each node except the leaf nodes has two child nodes. Let N be the number of leaf nodes in $\mathcal{BT}$. The number of all nodes in $\mathcal{BT}$ is $2N-1$. For any index $0 \le i < 2N-1$, we denote by v_i a node in $\mathcal{BT}$. We assign the index 0 to the root node and assign other indices to other nodes by using breadth-first search. That is, if a node v has an index i, then the index of its left child node is $2i+1$ and the index of its right child node is $2i+2$, while the index of its parent node (if any) is $\lfloor \frac{i-1}{2} \rfloor$. The depth of a node v_i is the length of the path from the root node to the node. The root node is at depth zero. The depth of $\mathcal{BT}$ is the depth of a leaf node.

For any node $v_i \in \mathcal{BT}$, L is defined as a label that is a fixed and unique string. The label of each node in the tree is assigned as follows: Each edge in the tree is assigned with 0 or 1 depending on whether the edge is connected to its left or right child node. The label L of a node v_i is defined as the bitstring obtained by reading all the labels of edges in the path from the root node to the node v_i. Note that we assign a special empty string to the root node as a label. We define $ID(i)$ be a mapping from the index i of a node v_i to a label L. Note that there is a simple mapping between the index i and the label L of a node v_i such that $i = (2^d-1) + \sum_{j=0}^{d-1} 2^j L[j]$ where d is the depth of v_i. We also use $ID(v_i)$ as $ID(i)$ if there is no ambiguity.

For any node $v_i \in \mathcal{BT}$, $\mathcal{T}_i$ is defined as a subtree that is rooted at v_i and S_i is defined as the set of leaf nodes in $\mathcal{T}_i$. We also define $ID(\mathcal{T}_i) = ID(v_i)$ and $ID(S_i) = ID(\mathcal{T}_i)$. For the tree $\mathcal{BT}$ and a subset R of leaf nodes, $ST(\mathcal{BT}, R)$ is defined as the Steiner Tree induced by the set R and the root node, that is, the minimal subtree of $\mathcal{BT}$ that connects all the leaf nodes in R and the root node. we simply denote $ST(\mathcal{BT}, R)$ by $ST(R)$.

Complete Subtree Scheme. The CS scheme uses a full binary tree to define the collection $\mathcal{S}$ of subsets. The CS scheme is described as follows:

CS.Setup(N_{max}): This algorithm takes as input the maximum number N_{max} of users. Let $N_{max} = 2^d$ for simplicity. It first sets a full binary tree $\mathcal{BT}$ of depth d. Each user is assigned to a different leaf node in $\mathcal{BT}$. The collection $\mathcal{S}$ of CS is $\{S_i : v_i \in \mathcal{BT}\}$. Recall that S_i is the set of all the leaves in the subtree $\mathcal{T}_i$. It outputs the full binary tree $\mathcal{BT}$.

CS.Assign($\mathcal{BT}, u$): This algorithm takes as input the tree $\mathcal{BT}$ and a user index $u \in \mathcal{N}$. Let v_u be the leaf node of $\mathcal{BT}$ that is assigned to the index u. Let $(v_{j_0}, v_{j_1}, \ldots, v_{j_d})$ be the path from the root node $v_{j_0} = v_0$ to the leaf node $v_{j_d} = v_u$. It sets $PV_u = \{S_{j_0}, \ldots, S_{j_d}\}$, and outputs the private set PV_u.

CS.Cover($\mathcal{BT}, R$): This algorithm takes as input the tree $\mathcal{BT}$ and a revoked set R of users. It first computes the Steiner tree $ST(R)$. Let $\mathcal{T}_{i_1}, \ldots \mathcal{T}_{i_m}$ be all the subtrees of $\mathcal{BT}$ that hang off $ST(R)$, that is all subtrees whose roots $v_{i_1}, \ldots v_{i_m}$ are not in $ST(R)$ but adjacent to nodes of outdegree 1 in $ST(R)$. It outputs a covering set $CV_R = \{S_{i_1}, \ldots, S_{i_m}\}$.

CS.Match(CV_R, PV_u): This algorithm takes input as a covering set $CV_R = \{S_{i_1}, \ldots, S_{i_m}\}$ and a private set $PV_u = \{S_{j_0}, \ldots, S_{j_d}\}$. It finds a subset S_k such that $S_k \in CV_R$ and $S_k \in PV_u$. If there is such a subset, it outputs (S_k, S_k). Otherwise, it outputs $\perp$.

Lemma 2.4 ([20]). *Let N_{max} be the number of leaf nodes in a full binary tree and r be the size of a revoked set. In the CS scheme, the size of a private set is $O(\log N_{max})$ and the size of a covering set is at most $r \log(N_{max}/r)$.*

3 Revocable Predicate Encryption

In this section, we define privacy preserving RPE and its new security models.

3.1 Definition

Revocable predicate encryption (RPE), introduced by Nieto et al. [21], is a variation of PE that has additional functionality such that a sender can specify a revoked user set in a ciphertext. In RPE, a user obtains a secret key associated with a predicate f and an index u from an authorized center. The authorized center posts a revocation list that contains the list of revoked user's indexes. After that, a sender constructs a ciphertext for an attribute x and a revoked user set R. If a receiver has a secret key associated with a predicate f and an index u such that $(f(x) = 1) \wedge (u \notin R)$, then he can decrypt the ciphertext associated with the attribute x and the revoked user set R.

Definition 3.1 (Revocable Predicate Encryption). *A revocable predicate encryption (RPE) scheme for the class $\mathcal{F}$ of predicates over a set Σ of attributes and the set $\mathcal{N} = \{1, \ldots, N_{max}\}$ of users consists of four PPT algorithms **Setup**, **GenKey**, **Encrypt**, and **Decrypt**, which are defined as follows:*

Setup($1^\lambda, N_{max}$): *The setup algorithm takes as input a security parameter* 1^λ *and the maximum number* N_{max} *of users, and then it outputs a public key PK and a master secret key MK.*

GenKey(f, u, MK, PK): *The key generation algorithm takes as input a predicate* $f \in \mathcal{F}$*, a user index* $u \in \mathcal{N}$*, and the master secret key MK, and then it outputs a secret key* $SK_{f,u}$.

Encrypt(x, R, M, PK): *The encryption algorithm takes as input an attribute* $x \in \Sigma$*, a revoked set* $R \subseteq \mathcal{N}$ *of users, a message* $M \in \mathcal{M}$*, and the public key PK, and then it outputs a ciphertext CT.*

Decrypt($CT, SK_{f,u}, PK$): *The decryption algorithm takes as input a ciphertext CT and a secret key* $SK_{f,u}$*, and outputs a message M or the distinguished symbol* $\perp$.

The correctness property of RPE is defined as follows: For all PK, MK generated by ***Setup****, all* $f \in \mathcal{F}$*, all* $u \in \mathcal{U}$*, any* $SK_{f,u}$ *generated by* ***GenKey****, all* $x \in \Sigma$*, any* $R \subseteq \mathcal{U}$*, and any* $M \in \mathcal{M}$*, it is required that:*

- *If* $(f(x) = 1) \wedge (u \notin R)$*, then* $\textbf{Decrypt}(\textbf{Encrypt}(x, R, M, PK), SK_{f,u}, PK) = M$.
- *If* $(f(x) = 0) \vee (u \in R)$*, then* $\textbf{Decrypt}(\textbf{Encrypt}(x, R, M, PK), SK_{f,u}, PK) = \perp$ *with all but negligible probability.*

We can use weak robustness of Abdalla et al. [2] for the second condition of the correctness property.

3.2 Security Model

The security model of RPE was introduced by Nieto et al. [21]. They introduced two security models: the attribute-hiding (AH) security that does not provide the privacy of the set R of revoked users, and the full-hiding (FH) security that provides the privacy of the set R of revoked users. In this paper, we only consider the security model that provides the privacy of the set R. The full-hiding (FH) security of Nieto et al. is a restricted security model that limits the capability of an adversary. In this paper, we first introduce the strongly full-hiding security that does not limit the capability of the adversary.

Strongly Full-Hiding. The strongly full-hiding security provides the attribute hiding property, the revocation set hiding property, and the message hiding property against not only an outside adversary who cannot decrypt the challenge ciphertext, but also an inside adversary who can decrypt the challenge ciphertext. In this security model, an adversary may adaptively obtain a secret key associated with a predicate f and an index u. After that, the adversary outputs the challenge attributes x_0, x_1, the challenge revocation sets R_0, R_1, and the challenge messages M_0, M_1 with restrictions such that the adversary cannot request a secret key that trivially can distinguish the challenge ciphertext. That is, if an adversary requested a secret key for a predicate f and an index u that can decrypt the challenge ciphertext such that $(f(x_0) = 1) \wedge (u \notin R_0)) = (f(x_1) = 1) \wedge (u \notin R_1)) = 1$, then the adversary should output the challenge messages M_0, M_1 such that $M_0 = M_1$.

Definition 3.2 (Strongly Full-Hiding)**.** *The security notion of strongly full-hiding under a chosen plaintext attack is defined in terms of the following experiment between a challenger* $\mathcal{C}$ *and a PPT adversary* $\mathcal{A}$:

1. ***Setup***: $\mathcal{C}$ *runs* $\textbf{Setup}(1^\lambda, N_{max})$ *to generate a public key PK and a master secret key MK, and it gives PK to* $\mathcal{A}$.
2. ***Phase I***: $\mathcal{A}$ *may adaptively request a polynomial number of secret keys for any predicate* $f \in \mathcal{F}$ *with a user index* $u \in \mathcal{U}$*, and then* $\mathcal{C}$ *gives the corresponding secret key* $SK_{f,u}$ *to* $\mathcal{A}$ *by running* $\textbf{GenKey}(f, u, MK, PK)$.

3. ***Challenge***: *$\mathcal{A}$ outputs challenge attributes $x_0, x_1 \in \Sigma$, challenge sets $R_0, R_1 \subseteq \mathcal{U}$ of revoked users of equal length, and challenge messages $M_0, M_1 \in \mathcal{M}$ subject to the following restrictions:*

   - *For all predicate f_i with a user index u_i of secret key queries, it is required that $((f_i(x_0) = 1) \wedge (u_i \notin R_0)) = ((f_i(x_1) = 1) \wedge (u_i \notin R_1))$.*
   - *If there is f_i with u_i in secret key queries such that $((f_i(x_0) = 1) \wedge (u_i \notin R_0)) = ((f_i(x_1) = 1) \wedge (u_i \notin R_1)) = 1$, then it is required that $M_0 = M_1$.*

   *$\mathcal{C}$ chooses a random bit b and gives the ciphertext CT to $\mathcal{A}$ by running **Encrypt**(x_b, R_b, M_b, PK).*

4. ***Phase II***: *$\mathcal{A}$ may continue to request secret keys for additional predicates with user indexes subject to the same restrictions as before, and $\mathcal{C}$ gives the corresponding secrets keys to $\mathcal{A}$.*

5. ***Guess***: *Finally $\mathcal{A}$ outputs a bit b'.*

The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv}_{\mathcal{A}}^{RPE,sFH}(\lambda) = |\Pr[b = b'] - \frac{1}{2}|$ where the probability is taken over all the randomness of the experiment. A RPE scheme is strongly full-hiding under a chosen plaintext attack if for all PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the above experiment is negligible in the security parameter λ.

Weakly Full-Hiding. Even though the strongly full-hiding security provides the full privacy of the revoked user set, it is not easy to devise an efficient RPE scheme that satisfies this strong security. Thus we propose another security model, namely the weakly full-hiding security, by weakening the strongly full-hiding security. The weakly full-hiding security provides the same security level against an outside adversary. However, this security does not provide the privacy of the revoked user set against an inside adversary. That is, an inside adversary may obtain partial information about the revoked user set. Therefore, it is required that the inside adversary outputs the challenge sets R_0, R_1 and the challenge message M_0, M_1 such that $R_0 = R_1$ and $M_0 = M_1$ in this weaker security model.

Definition 3.3 (Weakly Full-Hiding)**.** *The security notion of weakly full-hiding under a chosen plaintext attack is defined in terms of the following experiment between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$: The experiment is almost the same as the experiment of strongly full-hiding notion except that the restrictions in the challenge step are replaced as follows:*

- *For all predicate f_i with a user index u_i of secret key queries, it is required that $((f_i(x_0) = 1) \wedge (u_i \notin R_0)) = ((f_i(x_1) = 1) \wedge (u_i \notin R_1))$.*
- *If there is f_i with u_i in secret key queries such that $((f_i(x_0) = 1) \wedge (u_i \notin R_0)) = ((f_i(x_1) = 1) \wedge (u_i \notin R_1)) = 1$, then it is required that $R_0 = R_1$ and $M_0 = M_1$.*

The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv}_{\mathcal{A}}^{RPE,wFH}(\lambda) = |\Pr[b = b'] - \frac{1}{2}|$ where the probability is taken over all the randomness of the experiment. A RPE scheme is weakly full-hiding under a chosen plaintext attack if for all PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the above experiment is negligible in the security parameter λ.

Remark 3.4. *The full-hiding (FH) security of Nieto et al. [21] is weaker than the weakly full-hiding security of this paper. In their security model, the security against an inside adversary was not considered at all and the challenge sets R_0, R_1 of revoked users are severely limited such that the size of covering sets CV_{R_0}, CV_{R_1} should be equal where the covering set is a set of nodes in a tree and this can be derived in the complete subtree scheme. Note that even if the size of the challenge sets R_0 and R_1 is equal, the size of covering sets CV_{R_0} and CV_{R_1} can be different.*

The main advantage of the weakly full-hiding security is that it gives an opportunity to build an efficient RPE scheme such that the number of ciphertexts is not proportional to the number of receiver users. In the next section, we will show that it is possible to build an efficient and general RPE scheme that provides this security from any PE scheme. As mentioned, the disadvantage of this security is that it provides the privacy of revoked users against outside adversaries, but does not provides the privacy of revoked users against inside adversaries.

4 Weakly Full-Hiding RPE

In this section, we propose an efficient general RPE scheme by using any PE scheme, and prove that it is secure in the weakly full-hiding security model if the underlying PE scheme is secure in the adaptively attribute-hiding security model.

The main idea of our efficient general RPE scheme is to combine a PE scheme with the CS scheme[1]. To provide the privacy of revoked users, we hide the identifier of the CS scheme by using the attribute hiding property of PE. Additionally, we use a dummy identifier to hide the size of covering subsets of the CS scheme since two covering subsets of the CS scheme have different sizes even if the number of revoked users is the same. Though this scheme provide the weakly full-hiding security, it does not provide the strongly full-hiding security since an inside adversary can easily obtain partial information of the revoked set through the decryption process.

4.1 Construction

Let $\mathcal{F}$ be a class of predicates over Σ. Suppose there exists a PE scheme for the class of predicates $\mathcal{P} = \{P_{f,u} \mid f \in \mathcal{F} \text{ and } u \in \Sigma\}$ corresponding to the conjunction of some predicate query and an equality query where

$$P_{f,u}(x,v) = \begin{cases} 1 & \text{if } (f(x)=1) \wedge (u=v), \\ 0 & \text{otherwise.} \end{cases}$$

We construct an RPE scheme for the class of predicates $\mathcal{F}$ using the PE scheme for the class of predicates $\mathcal{P}$. Let $\mathcal{N} = \{1,\ldots,N_{max}\}$ be the set of all users in a system where $N_{max} = 2^d$ is the maximum number of users, and R be the revoked set of users where $|R| = r$. Our RPE scheme for the class of predicates $\mathcal{F}$ is described as follows:

RPE.Setup$(1^\lambda, N_{max})$**:** This algorithm takes as input a security parameter 1^λ and the maximum number N_{max} of users.

1. It first obtains PK_{PE} and MK_{PE} by running **PE.Setup**(1^λ). It also obtains a full binary tree $\mathcal{BT}$ by running **CS.Setup**(N_{max}).
2. It outputs a public key as $PK = (PK_{PE}, \mathcal{BT})$ and a master secret key $MK = MK_{PE}$.

RPE.GenKey(f, u, MK, PK)**:** This algorithm takes as input a predicate $f \in \mathcal{F}$, a user index $u \in \mathcal{N}$, the master secret key $MK = MK_{PE}$, and the public key PK. It proceeds as follows:

[1]The overall strategy of our RPE scheme is similar to the scheme of Nieto et al. [21], but the security of their scheme is only valid against an outsider adversary. Additionally the scheme of Nieto et al. does not hide the size of a cover in the ciphertext. Note that two covers CV_{R_0} and CV_{R_1} can have different sizes even if revoked sets R_0 and R_1 have the same size.

1. It first obtains a private set $PV_u = \{S_{j_0}, \ldots, S_{j_d}\}$ by running **CS.Assign**$(\mathcal{BT}, u)$. Let $L_k = ID(S_{j_k})$ for all $0 \le k \le d$.
2. For $0 \le k \le d$, it sets a new predicate $P_{f,L_k} \in \mathcal{P}$ and computes a subkey $SK_{PE,k}$ by running **PE.GenKey**$(P_{f,L_k}, MK_{PE}, PK_{PE})$.
3. It outputs a secret key by implicitly including f and u as $SK_{f,u} = \big(SK_{PE,0}, \ldots, SK_{PE,d}\big)$.

RPE.Encrypt(x, R, M, PK)**:** This algorithm takes as input an attribute $x \in \Sigma$, a revoked set of users $R \subseteq \mathcal{N}$, a message $M \in \mathcal{M}$, and the public key $PK = (PK_{PE}, \mathcal{BT})$. Let $\tilde{L}$ be a dummy identifier and $\tilde{M}$ be a random message. It proceeds as follows:

1. It first obtains a covering set $CV_R = \{S_{i_1}, \ldots, S_{i_m}\}$ by running **CS.Cover**$(\mathcal{BT}, R)$. Let $L_k = ID(S_{i_k})$ for all $1 \le k \le m$. It sets $W = \lceil r \log(N_{max}/r) \rceil$. Note that $m \le r \log(N_{max}/r) \le W$ by the claim of Naor et al. in [20].
2. For $1 \le k \le m$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((x, L_k), M, PK_{PE})$.
3. For $m+1 \le k \le W$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((x, \tilde{L}), \tilde{M}, PK_{PE})$.
4. It outputs a ciphertext as $CT = \big(CT_{PE,\pi(1)}, \ldots, CT_{PE,\pi(W)}\big)$ where π is a random permutation over $\{1, \ldots, W\}$.

RPE.Decrypt$(CT, SK_{f,u}, PK)$**:** This algorithm takes as input a ciphertext $CT = (CT_{PE,1}, \ldots, CT_{PE,W})$, a secret key $SK_{f,u} = (SK_{PE,0}, \ldots, SK_{PE,d})$, and the public key PK. It proceeds as follows:

1. For $0 \le k_1 \le d$ and $1 \le k_2 \le W$, it computes M by running **PE.Decrypt**$(CT_{PE,k_2}, SK_{PE,k_1}, PK_{PE})$ and returns M if $M \ne \perp$. Note that it cannot simply run **CS.Math** to find a matching pair for decryption since CV_R is not given in CT and CT contains the random permutation of $\{CT_{PE,k}\}$.
2. If it fails to obtain $M \ne \perp$ during the above iteration, then it outputs $\perp$.

4.2 Correctness

The correctness of the RPE scheme easily follows from the correctness of the PE and CS schemes. We consider two cases. If $(f(x) = 1) \wedge (u \notin R)$ in RPE, then we have $(f(x) = 1) \wedge (L_{k_1} = L_{k_2})$ since there exits a tuple $(S_{i_{k_2}}, S_{j_{k_1}})$ by the correctness of the CS scheme. Thus we obtain **RPE.Decrypt**$(CT, SK_{f,u}) = M$ since **PE.Decrypt**$(CT_{PE,k_2}, SK_{PE,k_1}) = M$. If $(f(x) = 0) \vee (u \in R)$ in RPE, then we have $(f(x) = 0) \vee (L_{k_1} \ne L_{k_2})$ for all k_1 and k_2 since there is no tuple $(S_{i_{k_2}}, S_{j_{k_1}})$ by the correctness of the CS scheme. Thus we obtain **RPE.Decrypt**$(CT, SK_{f,u}) = \perp$ since **PE.Decrypt**$(CT_{PE,k_2}, SK_{PE,k_1}) = \perp$ for all j and k by the correctness of the PE scheme.

4.3 Security Analysis

We prove the weakly full-hiding security of our RPE scheme by using the adaptively attribute-hiding security of the underlying PE scheme. Note that if the underly PE scheme only provides the selectively attribute-hiding security, then our RPE scheme just provides the selectively and weakly full-hiding security such that the adversary commits the challenge attributes x_0, x_1 before he receives a public key.

Theorem 4.1. *The above RPE scheme is weakly full-hiding under a chosen plaintext attack if the underlying PE scheme is adaptively attribute-hiding under a chosen plaintext attack. That is, for any PPT adversary $\mathcal{A}$ for the above RPE scheme, there exists a PPT algorithm $\mathcal{B}$ for the PE scheme such that $\boldsymbol{Adv}_{\mathcal{A}}^{RPE,wFH}(\lambda) \le 2W \cdot \boldsymbol{Adv}_{\mathcal{B}}^{PE,AH}(\lambda)$ where $W = \lceil r \log(N_{max}/r) \rceil$.*

Proof. To prove the security of the RPE scheme, we divide the behavior of an adversary as two types: Type-I and Type-II. The Type-I adversary behaves like an outside adversary who cannot request a secret key that can decrypt the challenge ciphertext. The Type-II adversary behaves like an inside adversary who can request a secret key that can decrypt the challenge ciphertext. The two types of adversaries are formally defined as follows:

Type-I. The Type-I adversary $\mathcal{A}_I$ only requests a secret key with a predicate f_i and an index u_i for all i such that $((f_i(x_0) = 1) \wedge (u_i \notin R_0)) = ((f_i(x_1) = 1) \wedge (u_i \notin R_1)) = 0$.

Type-II. The Type-II adversary $\mathcal{A}_{II}$ requests at least one secret key with a predicate f_i and an index u_i such that $((f_i(x_0) = 1) \wedge (u_i \notin R_0)) = ((f_i(x_1) = 1) \wedge (u_i \notin R_1)) = 1$. In this case, the output of $\mathcal{A}_{II}$ in the challenge step should satisfy the requirements of $R_0 = R_1$ and $M_0 = M_1$.

Let T_I, T_{II} be the event such that an adversary behaves like the Type-I, Type-II adversary respectively. In Lemma 4.2, we show that a Type-I adversary can be used for attacking the PE scheme. In Lemma 4.4, we show that a Type-II adversary can be used for attacking the PE scheme too. Therefore we have

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{A}}^{RPE,wFH}(\lambda) &\leq \Pr[T_I] \cdot \mathbf{Adv}_{\mathcal{A}_I}^{RPE,wFH}(\lambda) + \Pr[T_{II}] \cdot \mathbf{Adv}_{\mathcal{A}_{II}}^{RPE,wFH}(\lambda) \\
&\leq \Pr[T_I] \cdot 2W \cdot \mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda) + (1 - \Pr[T_I]) \cdot W \cdot \mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda) \\
&\leq 2W \cdot \mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda).
\end{aligned}$$

This completes our proof. □

4.3.1 Type-I Adversary

Lemma 4.2. *If there is a Type-I adversary $\mathcal{A}_I$ for the RPE scheme, then there is a PPT algorithm B for the PE scheme such that $\boldsymbol{Adv}_{\mathcal{A}_I}^{RPE,wFH}(\lambda) \leq 2W \cdot \boldsymbol{Adv}_{\mathcal{B}}^{PE,AH}(\lambda)$ where $W = \lceil r \log(N_{max}/r) \rceil$.*

Proof. The main idea of this proof is that a Type-I adversary (i.e. an outside adversary) cannot distinguish the changes of the challenge ciphertext from the correct encryption to the encryption of a random attribute, a dummy set, and a random message since the Type-I adversary cannot request a secret key query that can decrypt the challenge ciphertext. Thus, we first changes the challenge ciphertext from the encryption of an attribute x_0, a revoked set R_0, and a message M_0 to the encryption of a random attribute $\tilde{x}$, a dummy set $\tilde{R}$, and a random message $\tilde{M}$ through hybrid games. Next, we also similarly change the challenge ciphertext from the encryption of a random attribute $\tilde{x}$, a dummy set $\tilde{R}$, and a random message $\tilde{M}$ to the encryption of an attribute x_1, a revoked set R_1, and a message M_1 through hybrid games. Therefore, if the adversary cannot distinguish these changes of hybrid games, then he cannot distinguish the changes from the encryption of x_0, R_0, and M_0 to the encryption of x_1, R_1, and M_1.

We first define a sequence of hybrid games $\mathbf{G}_0^0, \ldots, \mathbf{G}_W^0, \mathbf{G}_W^1, \ldots, \mathbf{G}_0^1$. The games $\mathbf{G}_0^0$ and $\mathbf{G}_0^1$ will be the original security game except that the challenge bit is fixed to 0 and 1 respectively. The games $\mathbf{G}_W^0$ and $\mathbf{G}_W^1$ will be an ideal game such that an adversary has no advantage. Let $\tilde{x}$ be a random attribute, $\tilde{L}$ be a dummy identifier, and $\tilde{M}$ be a random message. For $\gamma \in \{0, 1\}$, the games are defined as follows:

Game $\mathbf{G}_0^\gamma$. This game is the original security game in Definition 3.3 except that the challenge bit b is fixed to γ.

Game $\mathbf{G}_h^\gamma$. We define a new game $\mathbf{G}_h^\gamma$ for $1 \le h \le W-1$. This game is almost identical to $\mathbf{G}_{h-1}^\gamma$ except that the ciphertext component $CT_{PE,h}$ is the encryption of $(\tilde{x}, \tilde{L})$ and $\tilde{M}$, instead of the encryption of (x_b, L_h) and M_b. The challenge bit b is set as γ and the challenge ciphertext is constructed as follows:

1. It first obtains a covering set $CV_{R_b} = \{S_{i_1}, \dots, S_{i_{m_b}}\}$ by running $\mathbf{CS.Cover}(\mathcal{BT}, R_b)$. Let $L_k = ID(S_{i_k})$ for all k. Note that $m_0, m_1 \le W$ since $|R_0| = |R_1| = r$.
2. If $1 \le h \le m_b$, then it proceeds the follows steps.
   (a) For $1 \le k \le h$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((\tilde{x}, \tilde{L}), \tilde{M}, PK_{PE})$.
   (b) For $h+1 \le k \le m_b$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((x_b, L_k), M_b, PK_{PE})$.
   (c) For $m_b+1 \le k \le W$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((x_b, \tilde{L}), \tilde{M}, PK_{PE})$.
3. Otherwise (that is, if $m_b + 1 \le h \le W$), it proceeds the follows steps.
   (a) For $1 \le k \le h$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((\tilde{x}, \tilde{L}), \tilde{M}, PK_{PE})$.
   (b) For $h+1 \le k \le W$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((x_b, \tilde{L}), \tilde{M}, PK_{PE})$.
4. It outputs a ciphertext as $CT = \big(CT_{PE,\pi(1)}, \dots, CT_{PE,\pi(W)}\big)$ where π is a random permutation.

Game $\mathbf{G}_W^\gamma$. In this game $\mathbf{G}_W^\gamma$, the challenge ciphertext is constructed as follows:

1. For $1 \le k \le W$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((\tilde{x}, \tilde{L}), \tilde{M}, PK_{PE})$.
2. It outputs a ciphertext as $CT = \big(CT_{PE,\pi(1)}, \dots, CT_{PE,\pi(W)}\big)$ where π is a random permutation.

Let $S_{\mathcal{A}_I}^{G_h^\gamma}$ be an event that $\mathcal{A}_I$ outputs 0 in the game $\mathbf{G}_h^\gamma$. We can easily obtain $\Pr[S_{\mathcal{A}_I}^{G_W^0}] = \Pr[S_{\mathcal{A}_I}^{G_W^1}]$ since the challenge ciphertext is not correlated with b. From Claim 4.3, we have that it is hard for the Type-I adversary to distinguish $\mathbf{G}_{h-1}^\gamma$ from $\mathbf{G}_h^\gamma$ if the PE scheme is secure. Thus, we have that

$$\begin{aligned}\big|\Pr[S_{\mathcal{A}_I}^{G_0^0}] - \Pr[S_{\mathcal{A}_I}^{G_0^1}]\big| &= \big|\Pr[S_{\mathcal{A}_I}^{G_0^0}] - \Pr[S_{\mathcal{A}_I}^{G_W^0}] + \Pr[S_{\mathcal{A}_I}^{G_W^1}] - \Pr[S_{\mathcal{A}_I}^{G_0^1}]\big| \\ &\le \sum_{\gamma \in \{0,1\}} \sum_{h=1}^{W} \big|\Pr[S_{\mathcal{A}_I}^{G_{h-1}^\gamma}] - \Pr[S_{\mathcal{A}_I}^{G_h^\gamma}]\big| = 4W \cdot \mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda).\end{aligned}$$

Finally, we have the following equation as

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}_I}^{RPE,wFH}(\lambda) &= \big|\Pr[b=0]\cdot\Pr[b'=b|b=0] + \Pr[b=1]\cdot\Pr[b'=b|b=1] - \frac{1}{2}\big| \\ &= \big|\frac{1}{2}\cdot\Pr[b'=0|b=0] + \frac{1}{2}\cdot(1-\Pr[b'=0|b=1]) - \frac{1}{2}\big| \\ &= \frac{1}{2}\cdot\big|\Pr[S_{\mathcal{A}_I}^{G_0^0}] - \Pr[S_{\mathcal{A}_I}^{G_0^1}]\big| \le 2W \cdot \mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda).\end{aligned}$$

This completes our proof. □

Claim 4.3. *If there is a Type-I adversary that distinguishes between $\mathbf{G}_{I,h-1}^\gamma$ and $\mathbf{G}_{I,h}^\gamma$ with non-negligible advantage, then there is a PPT algorithm $\mathcal{B}$ for the PE scheme such that $\big|\Pr[S_{\mathcal{A}_I}^{G_{h-1}^\gamma}] - \Pr[S_{\mathcal{A}_I}^{G_h^\gamma}]\big| = 2\cdot \mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda)$.*

Proof. The basic idea of this proof is that a simulator can use the challenge oracle of PE to construct the challenge ciphertext component $CT_{PE,h}$ of RPE since the only difference between $\mathbf{G}_{h-1}^\gamma$ and $\mathbf{G}_h^\gamma$ is the challenge ciphertext component $CT_{PE,h}$. Suppose there exists a Type-I adversary $\mathcal{A}_I$ that distinguishes between

$\mathbf{G}_{h-1}^{\gamma}$ and $\mathbf{G}_{h}^{\gamma}$ with non-negligible advantage. A simulator $\mathcal{B}$ that attacks the PE scheme is first given: a challenge public key PK_{PE}. Then $\mathcal{B}$ that interacts with $\mathcal{A}_I$ is described as follows:

Setup: $\mathcal{B}$ first sets $PK = (PK_{PE}, \mathcal{BT})$ and gives PK to $\mathcal{A}_I$.

Phase I: $\mathcal{A}_I$ adaptively requests a secret key for a predicate f and a user index u. $\mathcal{B}$ proceeds the secret key query as follows:

1. It first obtains a private set $PV_u = \{S_{j_0}, \dots, S_{j_d}\}$ by running **CS.Assign**$(\mathcal{BT}, u)$. Let $L_k = ID(S_{j_k})$ for all k.

2. For $0 \leq k \leq d$, it sets a new predicate P_{f,L_k} and requests a secret key $SK_{PE,k}$ for the new predicate P_{f,L_k} to the key generation oracle that simulates **PE.GenKey**.

3. It sets the secret key $SK_{f,u} = (SK_{PE,0}, \dots, SK_{PE,d})$ and gives this to $\mathcal{A}_I$.

Challenge: $\mathcal{A}_I$ outputs challenge attributes $x_0, x_1 \in \Sigma$, challenge revoked sets $R_0, R_1 \subseteq \mathcal{N}$, and challenge messages $M_0, M_1 \in \mathcal{M}$ subject to the restrictions of the Type-I adversary. Let $\tilde{x}$ be a random attribute, $\tilde{L}$ be a dummy identifier, and $\tilde{M}$ be a random message. $\mathcal{B}$ sets $b = \gamma$ and proceeds as follows:

1. It first obtains a covering set $CV_{R_b} = \{S_{i_1}, \dots, S_{i_{m_b}}\}$ by running **CS.Cover**$(\mathcal{BT}, R_b)$.

2. If $1 \leq h \leq m_b$, then it proceeds the following steps.

   (a) For $1 \leq k \leq h-1$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((\tilde{x}, \tilde{L}), \tilde{M}, PK_{PE})$.

   (b) For $k = h$, it gives challenge attributes $x'_0 = (x_b, L_h), x'_1 = (x, \tilde{L})$ and challenge messages $M'_0 = M_b, M'_1 = \tilde{M}$ to the challenge ciphertext oracle that simulates **PE.Encrypt**, and receives a challenge ciphertext CT'_{PE}. It sets $CT_{PE,h} = CT'_{PE}$.

   (c) For $h+1 \leq k \leq m_b$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((x_b, L_k), M_b, PK_{PE})$.

   (d) For $m_b + 1 \leq k \leq W$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((x_b, \tilde{L}), \tilde{M}, PK_{PE})$.

3. Otherwise (that is, if $m_b + 1 \leq h \leq W$), it proceeds the following steps.

   (a) For $1 \leq k \leq h-1$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((\tilde{x}, \tilde{L}), \tilde{M}, PK_{PE})$.

   (b) For $k = h$, it gives challenge attributes $x'_0 = (x_b, \tilde{L}), x'_1 = (\tilde{x}, \tilde{L})$ and challenge messages $M'_0 = \tilde{M}, M'_1 = \tilde{M}$ to the challenge ciphertext oracle of PE that simulates **PE.Encrypt**, and receives a challenge ciphertext CT'_{PE}. It sets $CT_{PE,h} = CT'_{PE}$.

   (c) For $h+1 \leq k \leq W$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((x_b, \tilde{L}), \tilde{M}, PK_{PE})$.

4. It outputs a ciphertext as $CT = (CT_{PE,\pi(1)}, \dots, CT_{PE,\pi(W)})$ where π is a random permutation.

Phase II: $\mathcal{A}_I$ may continue to request secret keys for predicates and user indexes subject to the same restrictions as before, and $\mathcal{B}$ gives the corresponding secrets keys to $\mathcal{A}_I$.

Guess: Finally $\mathcal{A}_I$ outputs a bit b'. $\mathcal{B}$ sets $c' = b'$ and outputs c'.

To finish the proof, we show that the distribution of the simulation is correct. It is easy to check that the public key and the secret keys are correctly distributed. Let c be the challenger ciphertext oracle's random bit of the PE scheme. If $c = 0$ and $h \leq m_b$, then CT'_{PE} is the encryption of an attribute $x'_0 = (x_b, L_h)$ and a message M_b. If $c = 0$ and $h \geq m_b + 1$, then CT'_{PE} is the encryption of an attribute $x'_0 = (x_b, \tilde{L})$ and a message

$\tilde{M}$. Thus, the challenge ciphertext is the same as $\mathbf{G}_{h-1}^{\gamma}$ if $c = 0$. Similarly, we have that the challenge ciphertext is the same as $\mathbf{G}_{h}^{\gamma}$ if $c = 1$. Therefore, we have

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda) &= \big|\Pr[c=0]\cdot\Pr[c'=c|c=0] + \Pr[c=1]\cdot\Pr[c'=c|c=1] - \frac{1}{2}\big| \\
&= \big|\frac{1}{2}\cdot\Pr[b'=0|c=0] + \frac{1}{2}\cdot(1-\Pr[b'=0|c=1]) - \frac{1}{2}\big| \\
&= \big|\frac{1}{2}\cdot(\Pr[b'=0|c=0]) - \frac{1}{2}\cdot(\Pr[b'=0|c=1])\big| \\
&= \big|\frac{1}{2}\cdot\Pr[S_{A_I}^{G_{h-1}^{\gamma}}] - \frac{1}{2}\cdot\Pr[S_{A_I}^{G_{h}^{\gamma}}]\big|.
\end{aligned}$$

This completes our proof. □

4.3.2 Type-II Adversary

Lemma 4.4. *If there is a Type-II adversary $\mathcal{A}_{II}$ for the RPE scheme, then there exists a PPT algorithm B for the PE scheme such that $\mathbf{Adv}_{\mathcal{A}_{II}}^{RPE,wFH}(\lambda) \leq W\cdot\mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda)$.*

Proof. The main idea of this proof is that a Type-II (i.e. an inside adversary) cannot distinguish the changes of the challenge ciphertext from the encryption of x_0, R, and M to the encryption of x_1, R, and M through hybrid games since the Type-II adversary has the restriction of $R_0 = R_1 = R$ and $M_0 = M_1 = M$. Note that we cannot change the challenge ciphertext to the encryption of a random attribute, a dummy set, and a random message like the proof of the Type-I adversary since the Type-II adversary can easily distinguish this change by decrypting the challenge ciphertext.

We first define a sequence of hybrid games $\mathbf{H}_0,\ldots,\mathbf{H}_L$. Let $R_0 = R_1 = R$ and $M_0 = M_1 = M$ since it is a Type-II adversary. We formally define the games as follows:

Game $\mathbf{H}_0$. This game is the original security game except that the challenger bit b is fixed to 0.

Game $\mathbf{H}_h$. We define a new game $\mathbf{H}_h$ for $1 \leq h \leq W-1$. This game is almost identical to $\mathbf{H}_{h-1}$ except that the challenge ciphertext component $CT_{PE,h}$ is changed from the encryption of x_0, R, and M to the encryption of x_1, R, and M. The challenge ciphertext is constructed as follows:

1. It first obtains a covering set $CV_R = \{S_{i_1},\ldots,S_{i_m}\}$ by running $\mathbf{CS.Cover}(\mathcal{BT}, R)$. Let $L_k = ID(S_{i_k})$ for $1 \leq k \leq m$. For $m+1 \leq k \leq W$, it sets $L_k = \tilde{L}$.
2. For $1 \leq k \leq h$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((x_1, L_k), M, PK_{PE})$.
3. For $h+1 \leq k \leq W$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((x_0, L_k), M, PK_{PE})$.
4. It outputs a ciphertext as $CT = \big(CT_{PE,\pi(1)},\ldots,CT_{PE,\pi(W)}\big)$ where π is a random permutation.

Game $\mathbf{H}_W$. In this game $\mathbf{H}_W$, the challenge ciphertext is constructed as follows:

1. It first obtains a covering set $CV_R = \{S_{i_1},\ldots,S_{i_m}\}$ by running $\mathbf{CS.Cover}(\mathcal{BT}, R)$. Let $L_k = ID(S_{i_k})$ for $1 \leq k \leq m$. For $m+1 \leq k \leq W$, it sets $L_k = \tilde{L}$.
2. For $1 \leq k \leq W$, it computes $CT_{PE,k}$ by running $\mathbf{PE.Encrypt}((x_1, L_k), M, PK_{PE})$.
3. It outputs a ciphertext as $CT = \big(CT_{PE,\pi(1)},\ldots,CT_{PE,\pi(W)}\big)$ where π is a random permutation.

This game is the same as the original security game except that the challenge bit b is fixed to 1.

Let $S_{\mathcal{A}_{II}}^{H_h}$ be an event that $\mathcal{A}_{II}$ outputs 0 in the game $\mathbf{H}_h$. From Claim 4.5, we have that it is hard for the Type-II adversary to distinguish $\mathbf{H}_{h-1}$ from $\mathbf{H}_h$ if the PE scheme is secure. Thus, we have that

$$\begin{aligned}\big|\Pr[S_{\mathcal{A}_{II}}^{H_0}] - \Pr[S_{\mathcal{A}_{II}}^{H_W}]\big| &= \big|\Pr[S_{\mathcal{A}_{II}}^{H_0}] + \sum_{h=1}^{W-1}\big(\Pr[S_{\mathcal{A}_{II}}^{H_h}] - \Pr[S_{\mathcal{A}_{II}}^{H_h}]\big) - \Pr[S_{\mathcal{A}_{II}}^{H_W}]\big| \\ &\leq \sum_{h=1}^{W}\big|\Pr[S_{\mathcal{A}_{II}}^{H_{h-1}}] - \Pr[S_{\mathcal{A}_{II}}^{H_h}]\big| = 2W \cdot \mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda).\end{aligned}$$

Finally, we can obtain the following equation as

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}_{II}}^{RPE,wFH}(\lambda) &= \big|\Pr[b=0]\cdot\Pr[b'=b|b=0] + \Pr[b=1]\cdot\Pr[b'=b|b=1] - \frac{1}{2}\big| \\ &= \big|\frac{1}{2}\cdot\Pr[b'=0|b=0] + \frac{1}{2}\cdot(1-\Pr[b'=0|b=1]) - \frac{1}{2}\big| \\ &= \frac{1}{2}\cdot\big|\Pr[b'=0|b=0] - \Pr[b'=0|b=1]\big| \\ &= \frac{1}{2}\cdot\big|\Pr[S_{\mathcal{A}_{II}}^{H_0}] - \Pr[S_{\mathcal{A}_{II}}^{H_W}]\big| \leq W\cdot\mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda).\end{aligned}$$

This completes our proof. □

Claim 4.5. *If there is a Type-II adversary that distinguishes between $\boldsymbol{H}_{h-1}$ and $\boldsymbol{H}_h$ with non-negligible advantage, then there is a PPT algorithm $\mathcal{B}$ for the PE scheme such that $\big|\Pr[S_{\mathcal{A}_{II}}^{H_{h-1}}] - \Pr[S_{\mathcal{A}_{II}}^{H_h}]\big| = 2\cdot\boldsymbol{Adv}_{\mathcal{B}}^{PE,AH}(\lambda)$.*

Proof. The basic idea of this proof is that a simulator can use the challenge ciphertext oracle of PE to construct the challenge ciphertext component $CT_{PE,h}$ of RPE since the only difference between $\mathbf{H}_{h-1}$ and $\mathbf{H}_h$ is the challenge ciphertext component $CT_{PE,h}$. Suppose there exists a Type-II adversary $\mathcal{A}_{II}$ that distinguishes between $\mathbf{H}_{h-1}$ and $\mathbf{H}_h$ with non-negligible advantage. A simulator $\mathcal{B}$ that attacks the PE scheme is first given: a challenge public key PK_{PE}. Then $\mathcal{B}$ that interacts with $\mathcal{A}_{II}$ is described as follows:

Setup: $\mathcal{B}$ first sets $PK = (PK_{PE}, \mathcal{BT})$ and gives PK to $\mathcal{A}_{II}$.

Phase I: $\mathcal{A}_{II}$ adaptively requests a secret key for a predicate f and a user index u. $\mathcal{B}$ proceeds the secret key query as follows:

1. It first obtains a private set $PV_u = \{S_{j_0}, \dots, S_{j_d}\}$ by running **CS.Assign**$(\mathcal{BT}, u)$.
2. For $0 \leq k \leq d$, it sets a new predicate P_{f,L_k} and requests a secret key $SK_{PE,k}$ for a new predicate P_{f,L_k} to the key generation oracle that simulates **PE.GenKey**.
3. It sets the secret key $SK_{f,u} = (SK_{PE,0}, \dots, SK_{PE,d})$ and gives this to $\mathcal{A}_{II}$.

Challenge: $\mathcal{A}_{II}$ outputs challenge attributes $x_0, x_1 \in \Sigma$, challenge revoked sets $R_0, R_1 \subseteq \mathcal{N}$ such that $R_0 = R_1 = R$, and challenge messages $M_0, M_1 \in \mathcal{M}$ such that $M_0 = M_1 = M$ subject to the restrictions. It sets $b = 0$ and proceeds as follows:

1. It first obtains a covering set $CV_R = \{S_{i_1}, \dots, S_{i_m}\}$ by running **CS.Cover**$(\mathcal{BT}, R)$. Let $L_k = ID(S_{i_k})$ for $1 \leq k \leq m$. For $m+1 \leq k \leq W$, it sets $L_k = \tilde{L}$.
2. For $1 \leq k \leq h-1$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((x_1, L_k), M, PK_{PE})$.

3. For $k = h$, it gives challenge attributes $x'_0 = (x_0, L_k), x'_1 = (x_1, L_k)$ and challenge messages $M'_0 = M, M'_1 = M$ to the challenge ciphertext oracle that simulates **PE.Encrypt**, and receives CT'_{PE}. It sets $CT_{PE,h} = CT'_{PE}$.
4. For $h+1 \leq k \leq W$, it computes $CT_{PE,k}$ by running **PE.Encrypt**$((x_0, L_k), M, PK_{PE})$.
5. It outputs a ciphertext as $CT = (CT_{PE,\pi(1)}, \ldots, CT_{PE,\pi(W)})$ where π is a random permutation.

Phase II: $\mathcal{A}_{II}$ may continue to request secret keys for additional predicates and user indexes subject to the same restrictions as before, and $\mathcal{B}$ gives the corresponding secrets keys to $\mathcal{A}_{II}$.
Guess: Finally $\mathcal{A}_{II}$ outputs a bit b'. $\mathcal{B}$ sets $c' = b'$ and outputs c'.

To finish the proof, we show that the distribution of the simulation is correct. It is easy to check that the public key and the secret keys are correctly distributed. Let c be the challenger ciphertext oracle's random bit of the PE scheme. If $c = 0$, then CT'_{PE} is the encryption of an attribute $x'_0 = (x_0, L_k)$ and a message M. If $c = 1$, then CT'_{PE} is the encryption of an attribute $x'_1 = (x_1, L_k)$ and a message M. Thus, the challenge ciphertext is the same as $\mathbf{H}_{h-1}$ if $c = 0$. Similarly, we obtain that the challenge ciphertext is the same as $\mathbf{H}_h$ if $c = 1$. Therefore, we have

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{B}}^{PE,AH}(\lambda) &= \big|\Pr[c=0]\cdot\Pr[c'=c|c=0] + \Pr[c=1]\cdot\Pr[c'=c|c=1] - \frac{1}{2}\big| \\
&= \big|\frac{1}{2}\cdot\Pr[b'=0|c=0] + \frac{1}{2}\cdot(1-\Pr[b'=0|c=1]) - \frac{1}{2}\big| \\
&= \frac{1}{2}\cdot\big|\Pr[b'=0|b=0,c=0] - \Pr[b'=0|b=0,c=1]\big| \\
&= \frac{1}{2}\cdot\big|\Pr[S_{A_{II}}^{H_{h-1}}] - \Pr[S_{A_{II}}^{H_h}]\big|.
\end{aligned}$$

This completes our proof. □

4.4 Instantiations

Our weakly full-hiding RPE scheme uses a PE scheme for the predicate class $\mathcal{P}$ corresponding to conjunctions of some predicate query and an equality query as a building block. In this section, we show that a PE scheme for $\mathcal{P}$ can be constructed from HVE, IPE, or FE schemes.

4.4.1 Instantiation from HVE

HVE is a specific type of PE that supports conjunctions of equality queries [11,18,25]. In HVE, a ciphertext is associated with a vector $\vec{x} = (x_1, \ldots, x_l)$ of attribute elements and a secret key is also associated with a vector $\vec{y} = (y_1, \ldots, y_l)$ of attribute elements or a wildcard $*$. If all attribute elements of $\vec{x}$ are equal to the elements of $\vec{y}$ in their position except the wildcard position, then the ciphertext with $\vec{x}$ can be decrypted by the secret key with $\vec{y}$. Boneh and Waters [11] showed that an HVE scheme can be used to build a PKE scheme that supports equality, comparison, subset, and conjunctive queries on encrypted data. Let $\mathcal{I}$ be a set and $\mathcal{I}_* = \mathcal{I} \cup \{*\}$ where $*$ plays the role of a wildcard. HVE supports the class of predicates $\mathcal{F}_{HVE} = \{f_{\vec{y}} \mid \vec{y} = (y_1, \ldots, y_l) \in \mathcal{I}_*^l\}$ such that

$$f_{\vec{y}}(\vec{x}) = \begin{cases} 1 & \text{if } ((y_i = x_i) \vee (y_i = *)) \text{ for all } 1 \leq i \leq l, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that there is an HVE scheme that supports the class of predicates $\mathcal{F}_{HVE}$ over $\mathcal{I}_*^{l+1}$ which consists of algorithms **HVE.Setup**, **HVE.GenKey**, **HVE.Encrypt**, and **HVE.Decrypt**. We construct a PE scheme for the class of predicates $\mathcal{P} = \{P_{f,u} | f \in \mathcal{F}_{HVE}, u \in \mathcal{I}\}$ by using this HVE scheme as follows:

PE.Setup(1^λ): This algorithm outputs PK and MK by running **HVE.Setup**(1^λ).

PE.GenKey($P_{\vec{y},u}, MK, PK$): This algorithm takes as input a predicate $P_{\vec{y},u} = (\vec{y}, u)$ where $\vec{y} = (y_1, \ldots, y_l) \in \mathcal{I}_*^l$ and $u \in \mathcal{I}$, and the master secret key MK. It sets a new vector $\vec{w} = (y_1, \ldots, y_l, u) \in \mathcal{I}_*^l \times \mathcal{I}$ and outputs a secret key $SK_{P_{\vec{y},u}}$ by running **HVE.GenKey**$(\vec{w}, MK, PK)$.

PE.Encrypt(x, M, PK): This algorithm takes as input an attribute $x = (\vec{x}, v)$ where $\vec{x} = (x_1, \ldots, x_l) \in \mathcal{I}^l$ and $v \in \mathcal{I}$, a message M, and the public key PK. It sets a new vector $\vec{v} = (x_1, \ldots, x_l, v) \in \mathcal{I}^{l+1}$ and outputs a ciphertext CT by running **HVE.Encrypt**$(\vec{v}, M, PK)$.

PE.Decrypt($CT, SK_{f_{\vec{y},u}}, PK$): This algorithm takes as input a ciphertext CT and a secret key $SK_{f_{\vec{y},u}}$, and outputs a message M by running **HVE.Decrypt**$(CT, SK_{f_{\vec{y},u}}, PK)$.

We first show the one-to-one correspondence such that $P_{\vec{y},u}(\vec{x}, v) = 1$ if and only if $((w_i = v_i) \vee (w_i = *))$ for all $1 \le i \le l+1$ where $\vec{v} = (v_1, \ldots, v_l, v_{l+1}) = (x_1, \ldots, x_l, v)$ and $\vec{w} = (w_1, \ldots, w_l, w_{l+1}) = (y_1, \ldots, y_l, u)$. If $P_{\vec{y},u}(\vec{x}, v) = 1$, then we easily have $((y_i = x_i) \vee (y_i = *))$ for all $1 \le i \le l$ and $u = v$. If $((w_i = v_i) \vee (w_i = *))$ for all $1 \le i \le l+1$, then we also have $((y_i = x_i) \vee (y_i = *))$ for all $1 \le i \le l$ and $u = v$ since $w_{l+1} = u \neq *$.

The correctness and security of PE easily follow from those of HVE because of the above one-to-one correspondence. That is, if there exists an adversary $\mathcal{A}$ that attacks the PE scheme, then it is directly converted to another adversary $\mathcal{B}$ that attacks the underlying HVE scheme.

4.4.2 Instantiation from IPE

IPE is a specific type of PE that supports inner-product queries [3, 16, 17, 23, 24]. In IPE, a ciphertext is associated with a vector $\vec{x} = (x_1, \ldots, x_l)$ of attribute elements and a secret key is also associated with a vector $\vec{y} = (y_1, \ldots, y_l)$ of attribute elements. If the inner-product operation between two vectors is zero, then the ciphertext with $\vec{x}$ can be decrypted by the secret key with $\vec{y}$. Katz, Sahai, and Waters [16] showed that an IPE scheme can be used to build an anonymous IBE scheme, an HVE scheme, and a PE scheme supporting polynomial evaluation. Let $\mathcal{I}$ be a set. IPE supports the class of predicates $\mathcal{F}_{IPE} = \{f_{\vec{y}} \mid \vec{y} = (y_1, \ldots, y_l) \in \mathcal{I}^l\}$ such that

$$f_{\vec{y}}(\vec{x}) = \begin{cases} 1 & \text{if } \langle \vec{y}, \vec{x} \rangle = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that there is an IPE scheme that supports the class of predicates $\mathcal{F}_{IPE}$ over $\mathcal{I}^{l+2}$ which consists of algorithms **IPE.Setup**, **IPE.GenKey**, **IPE.Encrypt**, and **IPE.Decrypt**. We construct a PE scheme for the class of predicates $\mathcal{P} = \{P_{f,u} | f \in \mathcal{F}_{IPE}, u \in \mathcal{I}\}$ by using this IPE scheme as follows:

PE.Setup(1^λ): This algorithm outputs PK and MK by running **IPE.Setup**(1^λ).

PE.GenKey($P_{\vec{y},u}, MK, PK$): This algorithm takes as input a predicate $P_{\vec{y},u}$ where $\vec{y} = (y_1, \ldots, y_l) \in \mathcal{I}^l$ and a value $u \in \mathcal{I}$, and the master secret key MK. It sets a new vector $\vec{w} = (y_1, \ldots, y_l, u, 1) \in \mathcal{I}^{l+2}$ and outputs a secret key $SK_{f_{\vec{y},u}}$ by running **IPE.GenKey**$(\vec{w}, MK, PK)$.

PE.Encrypt(x, M, PK): This algorithm takes as input an attribute $x = (\vec{x}, v)$ where $\vec{x} = (x_1, \ldots, x_l) \in \mathcal{I}^l$ and $v \in \mathcal{I}$, a message M, and the public key PK. It sets a new vector $\vec{v} = (x_1, \ldots, x_l, s, -sv) \in \mathcal{I}^{l+2}$ by selecting a random value $s \in \mathcal{I}$ and outputs a ciphertext CT by running **IPE.Encrypt**$(\vec{v}, M, PK)$.

PE.Decrypt$(CT, SK_{f_{\vec{y},u}}, PK)$**:** This algorithm takes as input a ciphertext CT and a secret key $SK_{f_{\vec{y},u}}$, and outputs a message M by running **IPE.Decrypt**$(CT, SK_{f_{\vec{y},u}}, PK)$.

We first show the one-to-one correspondence such that $P_{\vec{y},u}(\vec{x}, v) = ((\langle \vec{x}, \vec{y} \rangle = 0) \wedge (u = v)) = 1$ if and only if $\langle \vec{v}, \vec{w} \rangle = \langle \vec{x}, \vec{y} \rangle + s \cdot (u - v) = 0$ where $\vec{v}, \vec{w}$ are vectors of the IPE scheme. If $f_{\vec{y},u}(\vec{x}, v) = 1$, then $\langle \vec{v}, \vec{w} \rangle = 0$ since $\langle \vec{x}, \vec{y} \rangle = 0$ and $u = v$. If $\langle \vec{v}, \vec{w} \rangle = 0$, then we consider two cases such as $\langle \vec{x}, \vec{y} \rangle = 0$ or $\langle \vec{x}, \vec{y} \rangle \neq 0$. In case of $\langle \vec{v}, \vec{w} \rangle = 0$ and $\langle \vec{x}, \vec{y} \rangle = 0$, we have $f_{\vec{y},u}(\vec{x}, v) = 1$ since $u - v = 0$. In case of $\langle \vec{v}, \vec{w} \rangle = 0$ and $\langle \vec{x}, \vec{y} \rangle \neq 0$, the probability of $f_{\vec{y},u}(\vec{x}, v) = 1$ is negligible since s is randomly chosen.

The correctness and security of PE easily follow from those of IPE because of the above one-to-one correspondence. That is, if there exists an adversary $\mathcal{A}$ that attacks the PE scheme, then it is directly converted to another adversary $\mathcal{B}$ that attacks the underlying IPE scheme.

4.4.3 Instantiation from FE for General Circuits

Recently, Garg et al. [13] proposed a function encryption (FE) scheme for all polynomial-size circuits by using indistinguishability obfuscation for circuits, public-key encryption, and non-interactive zero knowledge. From the FE scheme for all circuits, we can build a PE scheme **PE** for all polynomial-size predicates $\mathcal{P}$ since the predicate class $\mathcal{P}$ is the subclass of the function circuit $\mathcal{F}$ class. Note that the output of a predicate is just $\{0, 1\}$ whereas the output of a function circuit is any value. If we instantiate our weakly full-hiding RPE scheme by using this PE scheme for all predicates, then our RPE scheme can supports all polynomial-size predicates.

4.5 Discussions

Efficiency. Suppose that we use an IPE scheme for our RPE scheme. Let $N_{max} = 2^d$ be the maximum number of users and l be the size of vectors for inner product operations in the RPE scheme. In this case, the underlying IPE scheme should support vectors of size $l + 2$ for inner product operations. The secret key of the RPE scheme consists of $\log(N_{max})$ secret keys of the IPE scheme, the ciphertext of the RPE scheme consists of $r \log(N_{max}/r)$ ciphertexts of the IPE scheme, and the decryption algorithm of the RPE scheme requires $r \log^2(N_{max})$ decryption operations of the IPE scheme. For instance, if we use the IPE scheme of Okamoto and Takashima [23] that has constant number of secret key elements, then the public key, the secret key, and the ciphertext of the RPE scheme consist of $O(l)$ group elements, $11 \log(N_{max})$ group elements, and $(5l + 11) r \log(N_{max})$ group elements respectively. Additionally, the decryption algorithm of the RPE scheme requires $(l + 2) r \log^2(N_{max})$ exponentiations and $11 r \log^2(N_{max})$ pairing operations.

Efficient Decryption. The decryption algorithm of our RPE scheme requires $r \log^2(N_{max})$ decryption operations of the underlying PE scheme. We can improve the efficiency of the decryption algorithm by using the technique of Barth et al. [5], redefined as anonymous hint systems by Libert et al. [19]. If we use the anonymous hint system of Barth et al. [5], then we can reduce the decryption overhead from $r \log^2(N_{max})$ decryption operations of the PE scheme to $r \log^2(N_{max})$ exponentiations and one decryption operation of the PE scheme. For instance, if we use the IPE scheme of Okamoto and Takashima [23] that has constant number of secret key elements, then the decryption algorithm just requires $r \log^2(N_{max})$ exponentiations and 11 pairing operations.

Supporting an Exponential Number of Users. The setup algorithm of the RPE scheme takes the maximum (polynomial) number of users N_{max} as an input. To support an unbounded (i.e. exponential) number of users, we can set $N_{max} = 2^{\lambda}$ since the security loss is just related with the logarithm of N_{max}. Note that the maximum number of revoked users r should be polynomial since the size of ciphertexts and the security loss depend on the number of revoked users. In this case, the public key does not need to keep the binary tree $\mathcal{BT}$ since the information for $\mathcal{BT}$ can be deterministically generated when it is needed.

Chosen-Ciphertext Security. To construct a weakly full-hiding RPE scheme with chosen-ciphertext security (CCA), we can combine a PE scheme that provides chosen-ciphertext security and an one-time signature (OTS) scheme that provides the strong unforgeability. That is, the encryption algorithm **RPE.Encrypt** of RPE generates the verification key OVK and the signing key OSK of OTS, and then it builds ciphertext components by running **PE.Encrypt** on a input $OVK||M$ instead of M. Additionally, it attaches an one-time signature σ_{OTS} to the ciphertext by running the signing algorithm of OTS on the ciphertext as an input. The security proof of this modified CCA secure RPE scheme also similarly follows from that of CCA secure PE.

5 Conclusion

In this paper, we revisited the notion of privacy preserving RPE and its general construction. We first introduced two security notions of RPE, namely the strongly full-hiding security and the weakly full-hiding security. Next, we proposed a general general RPE scheme that provides the weakly full-hiding security. Main advantage of our RPE scheme is that it can use any PE scheme since it uses the PE scheme in a black-box way. That is, if we instantiate our RPE scheme from a PE scheme that supports polynomial-size function circuits, our RPE scheme also can supports polynomial-size predicate circuits. Another advantage of our RPE constructions is that it can take efficiency and security advantage of the underlying PE scheme. Additionally, we can improve the efficiency of the decryption algorithm of our RPE scheme by using anonymous hint systems. One interesting problem is to propose a general RPE scheme from any PE scheme that has sublinear size of ciphertexts and provides the strongly full-hiding security.

References

[1] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222. Springer, 2005.

[2] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 480–497. Springer, 2010.

[3] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2011.

[4] Nuttapong Attrapadung and Hideki Imai. Conjunctive broadcast and attribute-based encryption. In Hovav Shacham and Brent Waters, editors, *Pairing*, volume 5671 of *Lecture Notes in Computer Science*, pages 248–265. Springer, 2009.

[5] Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In Giovanni Di Crescenzo and Aviel D. Rubin, editors, *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, pages 52–64. Springer, 2006.

[6] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM Conference on Computer and Communications Security*, pages 417–426. ACM, 2008.

[7] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.

[8] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[9] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[10] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011.

[11] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007.

[12] Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 225–242. Springer, 2012.

[13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. Cryptology ePrint Archive, Report 2013/451, 2013. `http://eprint.iacr.org/2013/451`.

[14] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

[15] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.

[16] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.

[17] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. *J. Cryptology*, 26(2):191–224, 2013.

[18] Kwangsu Lee and Dong Hoon Lee. Improved hidden vector encryption with short ciphertexts and tokens. *Designs Codes Cryptogr.*, 58(3):297–319, 2011.

[19] Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 206–224. Springer, 2012.

[20] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.

[21] Juan Manuel González Nieto, Mark Manulis, and Dongdong Sun. Fully private revocable predicate encryption. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP*, volume 7372 of *Lecture Notes in Computer Science*, pages 350–363. Springer, 2012.

[22] Tatsuaki Okamoto and Katsuyuki Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *CANS*, volume 7092 of *Lecture Notes in Computer Science*, pages 138–159. Springer, 2011.

[23] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 591–608. Springer, 2012.

[24] Jong Hwan Park. Inner-product encryption under standard assumptions. *Designs Codes Cryptogr.*, 58(3):235–257, 2011.

[25] Jong Hwan Park, Kwangsu Lee, Willy Susilo, and Dong Hoon Lee. Fully secure hidden vector encryption under standard assumptions. *Inf. Sci.*, 232:188–207, 2013.

[26] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 199–217. Springer, 2012.

[27] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[28] Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 560–578. Springer, 2008.