Efficient Predicate Encryption Supporting Construction of Fine-Grained Searchable Encryption

Weiyi Cai, Xiaoyuan Yang, Xu An Wang
Department of Electronic Technology, Engineering College of the Armed Police Force, Xi'an, 710086, China
weiyi.wj@gmail.com; xyyangwj@126.com; wangxuaan@gmail.com

Abstract

Predicate Encryption (PE) is a new encryption paradigm which provides more sophisticated and flexible functionality. PE is sufficient for some new applications, such as fine-grained control over access to encrypted data or search on encrypted data. We present an efficient construction of Predicate Encryption which is IND-AH-CPA secure by employing the dual system encryption without random oracle. We research on the relations between PE and Searchable Encryption in detail. The new notion of Public-Key Encryption with Fine-grained Keyword Search (PEFKS) is proposed. We prove that a IND-AH-CPA secure PE scheme implies the existence of a IND-PEFKS-CPA secure PEFKS scheme. We develop the transformation of PE-2-PEFKS and use the transformation to construct an efficient PEFKS scheme from our new PE scheme.
Key Words: Predicate Encryption; Public-Key Encryption with Fine-grained Keyword Search; Composite Order Bilinear Groups; Dual Encryption System

1. Introduction

In traditional Public-Key encryption, most of the systems focus on the point to point secure communication. Data is encrypted to be read by a particular individual who has already established a public key. The ciphertext is decrypted to learn the entire plaintext or nothing about the plaintext. This characteristic is insufficient for new emerging applications, such as cloud computing.

Recently, a new innovative class of encryption system, Predicate Encryption (PE), was proposed by Katz, Sahai and Waters [12]. PE enables one to evaluate more sophisticated and flexible functionality $F: Key_f \times CT_I \to \{0,1\}^*$ given the ciphertext CT_I and secret key Key_f. In a Predicate Encryption system, a key corresponds to a predicate and a ciphertext is associated with an attribute vector. The secret key sk_f corresponding to a predicate f can be used to decrypt a ciphertext using key associated with attribute vector I if and only if $f(I)=1$. PE implies several recent works aimed at constructing different types of fine-grained encryption schemes, such as Identity-Based Encryption(IBE)[18,20,9], Attribute-Based Encryption (ABE)[17,11,7,6], Hidden-Vector Encryption (HVE)[5]. They also introduced attribute-hiding (AH) which is a stronger security notion than payload-hiding. Attribute-hiding requires that a ciphertext conceal the associated attribute as well as the plaintext, while payload-hiding only requires that a ciphertext conceal the plaintext. In some applications which require the privacy of encryption key, payload-hiding is unacceptable. The notion of attribute-hiding addresses the limitation.

The dual system encryption which was introduced by Waters et.al [21,14] is a useful technique to obtain fully secure PE. In a dual encryption system, keys and ciphertexts can take on one of two forms: normal and semi-functional. The semi-functional keys and semi-functional ciphertexts are not used in the real system, only in the proof of security. The proof employs a sequence of security games which are shown to be indistinguishable. The first is the real security game in which both keys and ciphertext are normal. In the second game, the ciphertext is semi-functional and the keys remain normal. In subsequent games, the keys requested by the attacker are changed to be semi-functional one by one. By the final game, none of the keys given out are actually useful for decrypting a semi-functional ciphertext, and proving security becomes relatively easy.

The Public-Key Encryption with Keyword Search (PEKS) scheme was proposed by Boneh et al[3] for some interesting applications. An email user may want the server to deliver his/her emails according to some keywords attached on the emails. The user generates some trapdoors for the keywords and sends them to the server in secretly. The server may test whether there are these keywords in the emails. If the test outputs true, the mail will be sent to the user according to the rule. A practical PEKS must meet two conditions, consistency and security [1]. The consistency is that the decryption will not work unless the trapdoor and the ciphertext are matched. The security is that the ciphertext does not reveal any information about the keywords unless given the trapdoor.

There are many similar properties between the anonymous IBE and PEKS. In [3], Boneh et.al found that PEKS implied IBE. In [1], Abdalla et.al proved that an anonymous IBE could be transformed to a secure and consistent PEKS.

Our Contribution

In this paper, we do further work for PE and PEKS. The results are as follows.

- We present a Predicate Encryption system for the class of inner-product predicates that is fully secure without random oracles. There are several advantages over previous systems. We adopt dual system encryption to prove the security of our construction based on simple assumptions. The cost of our scheme is nearly a half of the existed scheme [12]. There is only one group element for each attribute in the ciphertext and user's key. It only requires one pairing operation for each attribute in the decrytion algorithm.
- In previous searchable encryption, the server only can test whether one keyword was present in the ciphertext. We extern the notion of PEKS to Public-Key Encryption with Fine-grained Keyword Search(PEFKS). PEFKS not only can test whether mutiple kewords were present in the ciphertext, but aslo can evaluate the relations of the keywords, such as equal, disjunction/conjunction. These complicated relations can't be formulated only from single keyword search by adding some relations of keywords, since it leaks unnecessary information to the server [10]. We discuss the consistency via an experiment involving adversary and define the security of PEFKS through the game between the challenge and the adversary.
- We prove that IND-AH-CPA secure PE implies the existence of IND-PEFKS-CPA secure PEFKS and develop a transformation of PE to PEFKS, PE-2-PEFKS. The transformation is efficient. We also use it to construct a PEFKS scheme from our PE.

1.1. Related Work

Predicate encryption was presented by Katz, Sahai and Waters in [12] as a generalized notion of IBE. In their predicate encryption scheme, a predicate f was associated with a vector $\vec{v} \in Z_p^n \setminus \{\vec{0}\}$ and the key was associated with attribute $\vec{x} \in Z_p^n \setminus \{\vec{0}\}$, where if $\vec{x} \cdot \vec{v} = 0$ then $f_{\vec{v}}(\vec{x}) = 1$, else $f_{\vec{v}}(\vec{x}) = 0$. Their construction provided attribute-hiding property. However their construction was inefficient and was proved to be selectively secure in the IND-AH-CPA game.

Shi and Waters [19] defined delegation in predicate encryption systems, and proposed a new security definition for delegation. They presented an efficient construction supporting conjunctive queries. Their system also was only proved selective and CPA security.

Okamoto and Takashima [15] presented a hierarchical predicate encryption (HPE) scheme for inner-product predicates that is secure in the standard model based on new assumptions in dual pairing vector spaces (DPVS). But this system was selectively secure. In Crypto2010, they presented a fully secure scheme [16].

Lewko et.al [13] proposed a fully secure (H)PE scheme for inner-product predicates in the standard model by employing the dual system methodology. There scheme was proven to be CPA secure but their scheme was inefficient.

Boneh et al [3] first studied the problem of public-key encryption with keyword search (PEKS). They gave several constructions of PEKS and proved that a PEKS implied a secure IBE. They claimed that it was hard to construct PEKS from IBE.

In [1], Abdalla et.al did further work for PEKS. They made two important contributions. First, they defined computational, statistical and perfect consistency which is formulated via an experiment involving an adversary. Second, they provided a transformation from an anonymous IBE to a secure PEKS that guaranteed consistency and security.

Most of previous work focused on single keyword search. However, in many situations, the search will be on multiple keywords. In [2], Joonsang et.al proposed a PEKS scheme that encrypts multiple keywords that are connected through conjunctive or disjunctive logical connectives.

1.2. Organization

In Section 2, we give the definition for the rest of this paper. We present our construction of PE and prove its security in section 3. In section 4, PEFKS is presented and the PE-2-PEFKS transformation is described. In Section 5 we make our conclusion.

2. Definition

In this section we introduce the notion of Predicate Encryption for the class of inner-product predicates and PEFKS. We also give the necessary background on composite order bilinear groups and our complexity assumptions.

2.1. Predicate Encryption

A Predicate Encryption scheme for the class of inner-product predicates supports functionality $F : Key_{f_{\vec{v}}} \times CT_{\vec{x}} \to \{0,1\}^*$, where $\vec{x} \in Z_p^n \setminus \{\vec{0}\}$ and $\vec{v} \in Z_p^n \setminus \{\vec{0}\}$. If $\vec{x} \cdot \vec{v} = 0$, then $f_{\vec{v}}(\vec{x}) = 1$, else $f_{\vec{v}}(\vec{x}) = 0$. The ciphertext space is $\mathcal{C}$ and the message space is $\mathcal{M}$.

PE scheme consists of four fundamental algorithms: **Setup, KeyGen, Encrypt, and Decrypt**.

Setup given the security parameter 1^{λ}, outputs the public parameters PK and master key MK;

KeyGen given the master key MK and a predicate vector $\vec{v}$, outputs a user key sk_v;

Encrypt given the public parameters PK and an attribute vector $\vec{x}$ and a message $m \in \mathcal{M}$, outputs a ciphertext $c \in \mathcal{C}$;

Decrypt given the user key $sk_{\vec{v}}$ and a ciphertext c, outputs the plaintext m if and only if $f_{\vec{v}}(\vec{x}) = 1$.

In [13], Lewko et.al defines IND-AH-CPA security for PE systems via the following game. In section 4.1, we will see that this security definition is sufficient for the construction of IND-PKES-CPA scheme.

Security Model for PE

Setup The challenger runs the Setup algorithm and gives the public parameters to the adversary;

Phase1 The adversary is allowed to adaptively issue queries for private keys for many predicates vector $\vec{v}$;

Challenge The adversary submits two equal length messages m_0 and m_1 and two attribute vectors $\vec{x}_0, \vec{x}_1$ where $f_{\vec{v}}(\vec{x}_0) \neq 1$ and $f_{\vec{v}}(\vec{x}_1) \neq 1$ for all the key queried in Phase1. The challenger flips a random coin b and encrypts m_b with $\vec{x}_b$. The challenge ciphertext c^* is passed to the adversary;

Phase2 The adversary may continue to issue adaptively queries like Phase1, except the key query for predicate $f_{\vec{v}}(\vec{x}_0) = 1$ and $f_{\vec{v}}(\vec{x}_1) = 1$;

Guess The adversary outputs a guess b' of b.

The advantage of an IND-AH-CPA adversary in this game is defined as $\left| \Pr[b' = b] - \frac{1}{2} \right|$.

Definition 1 A Predicate Encryption scheme is IND-AH-CPA secure if all polynomial time adversaries have at most a negligible advantage in the above security game.

2.2. PEFKS

In practice, one may need to append multiple keywords to one message and describe the relations between them, e.g. "*urgent* and *business*", "*family* or *company*". A Public-key Encryption with Fine-grained Keywords Search (PEFKS) is sufficient for this request. PEFKS allows a user to define the relations of keywords which makes it more appropriate in practice.

PEFKS consists of the following algorithms.

KG $(1^{\lambda}) \to (pk, sk)$, the key generation algorithm, which takes in security parameter λ and outputs a secret key sk and a public key pk;

Td $(sk, \vec{w}) \to t_{\vec{w}}$, the trapdoor generation algorithm, which outputs $t_{\vec{w}}$ for keywords vector $\vec{w}$;

PEFKS $(pk, \vec{x}) \to \sigma$, the encryption algorithm, which outputs ciphertext σ for keywords vector $\vec{x}$;

Test $(t_{\vec{w}}, \sigma) \to \{0,1\}$, the verification algorithm, which outputs 1 if $\vec{w} \cdot \vec{x} = 0$, otherwise outputs 0.

We will discuss the consistency and security of the PEFKS.

Consistency. By analogy with the definition of [1], we define the consistency notion via an experiment involving an adversary. The experiment is as follows:

$$Exp^{PEFKS-CONSISTENCY}_{\Sigma_{PEFKS},\mathcal{L}}(\lambda):$$
$$(pk,sk)\leftarrow KG(1^{\lambda});$$
$$(\vec{w},\vec{x})\leftarrow \mathcal{L}(pk), \vec{w}\cdot\vec{x}\neq 0;$$
$$t_{\vec{w}}\leftarrow Td(sk,\vec{w});$$
$$\sigma\leftarrow PEFKS(pk,\vec{x});$$
$$\text{if } \vec{w}\cdot\vec{x}\neq 0 \text{ and } Test(t_{\vec{w}},\sigma)=1 \text{ then return 1 else return 0}$$

The advantage of $\mathcal{L}$ is defined as $Adv^{PEFKS-CONSISTENCY}_{\Sigma_{PEFKS},\mathcal{L}}(\lambda)\equiv \Pr[Exp^{PEFKS-CONSISTENCY}_{\Sigma_{PEFKS},\mathcal{L}}(\lambda)=1]$.

The scheme is said to be perfectly consistent if this advantage is 0 for all adversaries $\mathcal{L}$ (computationally unrestricted), statistically consistent if it is negligible for all adversaries $\mathcal{L}$ (computationally unrestricted), computational consistent if it is negligible for all polynomial time adversaries $\mathcal{L}$. Computational consistency is weaker than statistical consistency and perfect consistency but still adequate in practice.

Security. We define the semantic security notion (IND-PKFES-CPA) for the PEFKS.

Security Model for PEFKS

Setup The challenger runs the KG algorithm to get (pk,sk) and gives pk to the adversary;

Phase1 The adversary is allowed to adaptively issue queries for trapdoors $t_{\vec{w}}$ for many keywords vector $\vec{w}$;

Challenge The adversary submits two keywords vectors $\vec{x}_0,\vec{x}_1$ where $\vec{x}_0\cdot\vec{w}\neq 0$ and $\vec{x}_1\cdot\vec{w}\neq 0$ for all keys queried in Phase1. The challenger flips a random coin b and gives the adversary $\sigma^*\leftarrow PEFKS(pk,\vec{x}_b)$;

Phase2 The adversary may continue to issue adaptively queries like Phase1, except that $\vec{x}_0\cdot\vec{w}=0$ and $\vec{x}_1\cdot\vec{w}=0$;

Guess The adversary outputs a guess b' of b.

The advantage of an IND-PEFKS-CPA adversary in this game is defined as $\left|\Pr[b'=b]-\frac{1}{2}\right|$.

Definition 2 A PEFKS scheme is IND-PEFKS-CPA secure if all polynomial time adversaries have at most a negligible advantage in the above security game.

2.3. Assumption

Our systems are constructed in composite order bilinear groups. Composite order bilinear groups was first introduced by Boneh, Goh, and Nissim [4]. The only known instantiations of composite order bilinear groups use elliptic curves over finite fields. Since the elliptic curve group order N must be infeasible to factor, it must be at least (say) 1024 bits [8].

We define a group generator $\mathcal{G}$, an algorithm which takes in a security parameter 1^{λ} and output $(N=p_1p_2p_3,G,G_T,e)$, where p_1,p_2,p_3 are distinct primes, G and G_T are cyclic groups of order N, and $e:G\times G\rightarrow G_T$ is a map that:

1. Bilinear: for all $u,v\in G$ and $a,b\in Z_p$, we have $e(u^a,v^b)=e(u,v)^{ab}$.
2. Non-degeneracy: their exist g, s.t. $e(g,g)$ has order N in G_T.

We say that G is a bilinear group if the group operation in G and the bilinear map $e:G\times G\rightarrow G_T$ are both efficiently computable. Let G_{p1}, G_{p2}, G_{p3} denote the subgroups of G. We can see that they have the orthogonality property[14], namely, when $h_i\in G_{pi}$, $h_j\in G_{pj}$, $i\neq j$, $e(h_i,h_j)$ is the identity element in G_T. We will implement this property in our construction.

We now state the complexity assumptions that we will rely on to prove security of our systems. These assumptions are an extension of [14]. Introducing the additional term $h\in G_{p_1}$ still does not help to add advantage to $\mathcal{A}$, since h is independent of the challenge.

Assumption 1: Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G}=(N=p_1p_2p_3,G,G_T,e)\leftarrow\mathcal{G},$$
$$g,h\leftarrow G_{p_1}, X_3\leftarrow G_{p_3},$$
$$D=(\mathbb{G},g,h,X_3),$$

$$T_0 \leftarrow G_{p_1}, T_1 \leftarrow G_{p_1 p_2}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be: $Adv1_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D,T_0)=0] - \Pr[\mathcal{A}(D,T_1)=0]|$

Definition 3: We say that $\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of 1^λ for any polynomial time algorithm $\mathcal{A}$.

Assumption 2: Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \leftarrow \mathcal{G},$$

$$g, h, X_1 \leftarrow G_{p_1}, X_2, Y_2 \leftarrow G_{p_2}, X_3, Y_3 \leftarrow G_{p_3},$$

$$D = (\mathbb{G}, g, h, X_3, X_1 X_2, Y_2 Y_3),$$

$$T_0 \leftarrow G_{p_1 p_3}, T_1 \leftarrow G.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be: $Adv2_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D,T_0)=0] - \Pr[\mathcal{A}(D,T_1)=0]|$

Definition 4: We say that $\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of 1^λ for any polynomial time algorithm $\mathcal{A}$.

Assumption 3: Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \leftarrow \mathcal{G}, \quad r \in Z_N,$$

$$g, h \leftarrow G_{p_1}, X_2, Y_2, Z_2 \leftarrow G_{p2}, X_3 \leftarrow G_{p_3},$$

$$D = (\mathbb{G}, g, X_3, Z_2, g^r X_2, hY_2),$$

$$T_0 = e(g,h)^r, T_1 \leftarrow G_T.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be: $Adv3_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D,T_0)=0] - \Pr[\mathcal{A}(D,T_1)=0]|$

Definition 5: We say that $\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of 1^λ for any polynomial time algorithm $\mathcal{A}$.

3. Efficient PE

In this section, we present an IND-AH-CPA secure Predicate Encryption system that support inner-product predicates. The class of predicates is $\mathcal{F} = \{f_{\vec{v}} \mid \vec{v} \in Z_p^n \setminus \{\vec{0}\}\}$, with $f_{\vec{v}}(\vec{x}) = 1$ if $\vec{x} \cdot \vec{v} = 0 \bmod N$. In our construction, subgroup G_{p_1} will be used for encryption and decryption; G_{p_3} will be used for key randomizing; G_{p_3} will be used for semi-functional keys and semi-functional ciphertext, which is not used in real encryption system. The attributes of the ciphertext and predicate of the user are expressed as a vector. We define that each element of the vector must not be 0，and $\sum_{i=1,\cdots,n} x_i \neq 0$.

Setup(1^λ) The KGC first runs $\mathcal{G}(1^\lambda)$ to get $\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e)$. It then choose random generators $g, h \in G_{p_1}$, $X_3 \in G_{p_3}$, and random $a \in Z_N$, $t_i \in Z_N$, $i = 1, \cdots, n$, where $a \neq t_i$. The public parameters and master key are given as

$$PK = \{g, h, g_1 = g^a, \{T_i = g^{t_i}\}_{i=1,\cdots,n}\}$$

$$MK = \{a, \{t_i\}_{i=1,\cdots,n}\}$$

KeyGen(MK, $\vec{v}$) The KGC runs this algorithm to generate a user key for user who is qualified with predicate vector $\vec{v}$. First, it choose a random value $s \in Z_N$, and $W_i \in G_{p_3}$, $i = 1, \cdots, n$. Let $\vec{v} = \{v_1, \cdots, v_n\}$, It creates the private key as

$$sk_v = \{\{d_i = (hg^{sv_i} W_i)^{1/(a-t_i)}\}_{i=1,\cdots,n}\}$$

Encrypt(PK, $\vec{x}$, m) To encrypt $m \in \mathcal{M}$ with attribute vector $\vec{x}$ the sum of which must not be 0, the sender chooses random $r \in Z_N$ then it sets

$$c = \{c_0 = m \cdot e(g,h)^{-r\sum_{i=1,\cdots,n} x_i}, \{c_i = (g_1 T_i^{-1})^{rx_i}\}_{i=1,\cdots,n}\}$$

Decrypt($sk_{\vec{v}}$, c **)** The receiver downloads the ciphertext. It computes

$$c_0 \cdot \prod_{i=1,\cdots,n} e(c_i, d_i)$$

Correctness To see that correctness holds, we assume the ciphertext is well-formed:

$$c_0 \cdot \prod_{i=1,\cdots,n} e(c_i, d_i)$$

$$= m \cdot e(g,h)^{-r\sum_{i=1,\cdots,n} x_i} \prod_{i=1,\cdots,n} e((g_1^r T_i^{-r})^{x_i}, (hg^{s\cdot v_i} W_i)^{1/(a-t_i)})$$

$$= m \cdot e(g,h)^{-r\sum_{i=1,\cdots,n} x_i} \cdot e(g,h)^{r\sum_{i=1,\cdots,n} x_i} e(g,g)^{sr\sum_{i=1,\cdots,n} x_i \cdot v_i}$$

$$= m \cdot e(g,g)^{s\cdot r\sum_{i=1,\cdots,n} x_i \cdot v_i}$$

If $\vec{x} \cdot \vec{v} \neq 0 \bmod p_1$, then the decryption algorithm evaluates to a random element in the group of G_T . If $\vec{x} \cdot \vec{v} = 0 \bmod p_1$, namely $f_{\vec{v}}(\vec{x}) = 1$, the receiver can get the message.

3.1. Efficiency

We now consider the efficiency of the scheme in terms of ciphertext size, private key size, and computation time for decryption and encryption as compared with[12,13].

Table 1 Comparisons of Efficiency

Schemes	Key Size	Ciphertext Size	Encryption time	Decryption time	Security
Katz[12]	(2n+1) g	(2n+1) g +1 g_T	(4n+2) p	(2n+1) e	selective
Lewko[13]	N g	N g +1 g_T	N p	N e	adaptive
Ours	n g	n g +1 g_T	(1+n) p	n e	adaptive

Note: N>2n+3. g denotes one group element in G and g_T denotes one group element in G_T. p and e are the power operation and pairing operation.

As we can see, the cost of our construction is only a half of [12,13]. The ciphertext size will be approximately one group element in G for each attribute, while two in [12,13]. User's private keys will consist of one group elements in G for each attribute, while two in [12,13]. In the encryption procedure, $e(g,h)$ and $(g_1 T_i^{-1})$ can be pre-computed, so it doesn't need any pairing operation. It only requires one power operation for each attribute, while four in [12] and more than two in [13]. The decryption procedure needs one pairing operation for every each attribute, while two in[12] and more than two in [13].

We can conclude that, comparing with other typical PE scheme, our construction is also more efficient.

3.2. Security

To prove the security, we will adopt the dual system encryption methodology which was used in[21,14]. We define two additional structures: semi-functional ciphertexts and keys. These will not be used in the real system, but will be needed in our proof.

Semi-functional Ciphertext Let g_2 denote a generator of G_{p_2}. $c \in Z_N$, and $\{z_i \in Z_N\}_{i=1,\cdots,n}$ are random values. A semi-functional ciphertext is formed as follows.

$$\{c_i = (g^{r_1 x_i} g_2^{cz_i})^{a-t_i}\}_{i=1,\cdots,n}$$

Semi-functional Key Let g_2 denote a generator of G_{p_2}. $d \in Z_N$ and $\{y_i \in Z_N\}_{i=1,\cdots,n}$ are random values. A semi-functional key is formed as follows.

$$\{d_i = (hg^{sv_i} W_i g_2^{dy_i})^{1/(a-t_i)}\}_{i=1,\cdots,n}$$

A normal key can decrypt both normal and semi-functional ciphertexts, while a normal ciphertexts can be decrypted by both normal and semi-functional keys. When we use a semi-functional key to decrypt a semi-functional ciphertext, we are left with additional term $e(g_2, g_2)^{cd\sum_{i=1,\cdots,n} y_i \cdot z_i}$. Notice that if a semi-functional key which is satisfy that $\sum_{i=1,\cdots,n} y_i \cdot z_i = 0$ is used to decrypt a semi-functional ciphertext, decryption will still work.

Based on the assumptions, we will prove the security of our system using a sequence of games. Game_{real} is the real security game in which both keys and ciphertext are normal. In the second game, Game_0, the ciphertext is semi-functional and all keys are normal. In game Game_k, the first k key queries are semi-functional and the rest are normal. By the final game, Game_{final}, all of the key queries are semi-functional and the challenge ciphertext is a semi-functional encryption of a random message. We will prove these games are indistinguishable in the following lemmas.

Lemma 1 Assume there is an a polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{\mathrm{Game}_{real}} - Adv_{\mathcal{A}}^{\mathrm{Game}_0} = \varepsilon$. Then we can construct a polynomial time simulator $\mathcal{B}$ with advantage ε in breaking Assumption 1.

Proof. $\mathcal{B}$ is given a challenge sample of Assumption 1, $(\mathbb{G}, g, h, X_3, T)$, which is used as an input of the Setup algorithm. $\mathcal{B}$ chooses random value $a \in Z_N$, $t_i \in Z_N$, $i = 1, \cdots, n$. The public parameters are set the same as Setup algorithm. $\mathcal{B}$ will simulate Game_{real} and Game_0 with $\mathcal{A}$.

As to the key queries $\vec{v}$, $\mathcal{B}$ can generate normal key by using the KeyGen algorithm, since it knows the MK.

As to the challenge (m_0, m_1) and ($\vec{x}_0, \vec{x}_1$), $\mathcal{B}$ will imbeds the Assumption 1 into the challenge ciphertext. It first flips a random coin b, and sets:

$$c^* = \{c_0 = m_b \cdot e(T, h)^{-\sum_{i=1,\cdots,n} x_i^b}, \{c_i = T^{x_i^b (a - t_i)}\}_{i=1,\cdots,n}\}$$

If $T \in G_{p_1}$, namely $T = g^r$, it is clearly that this is a properly distributed normal ciphertext. If $T \in G_{p_1 p_2}$, namely $T = g^r g_2{}^c$, we implicitly set $z_i = x_i$. However, the values of $x_i \bmod p_1$ are uncorrelated from $z_i \bmod p_2$ by the Chinese Remainder Theorem. This is a properly distributed semi-functional ciphertext.

$\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage ε in breaking Assumption 1 after all.

Lemma 2 Assume there is an a polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{\mathrm{Game}_{k\text{-}1}} - Adv_{\mathcal{A}}^{\mathrm{Game}_k} = \varepsilon$. Then we can construct a polynomial time simulator $\mathcal{B}$ with advantage ε in breaking Assumption 2.

Proof. $\mathcal{B}$ is given a challenge sample of Assumption 2, $(\mathbb{G}, g, h, X_3, X_1X_2, Y_2Y_3, T)$. The public parameters are generated just like that in the proof of Lemma 1. $\mathcal{B}$ will simulate Game_{k-1} and Game_k with $\mathcal{A}$.

As to the key queries $\vec{v}$, $\mathcal{B}$ forms normal keys for queries>k, semi-functional keys queries<k, and either normal or semi-functional for k_{th} query.

To the queries>k, $\mathcal{B}$ can generate normal key by using the KeyGen algorithm by using its knowledge of MK. To the queries<k, $\mathcal{B}$ chooses random value $s, d \in Z_N$, then the semi-functional key can then be defined as:

$$\{d_i = (hg^{sv_i} W_i (Y_2Y_3)^{dy_i})^{1/(a-t_i)}\}_{i=1,\cdots,n}$$

To the k_{th} key, $\mathcal{B}$ uses the value of T in the challenge, and choose random value. The key will be set as:

$$\{d_i = (hT^{v_i} W_i)^{1/(a-t_i)}\}_{i=1,\cdots,n}$$

If $T \in G_{p_1} G_{p_3}$, it is clearly that this is a properly distributed normal key. If $T \in G_{p_1 p_2 p_3}$, namely $T = g^s g_2{}^d g_3^f$, we implicitly set $y_i = v_i$. According to the Chinese Remainder Theorem, this is a properly distributed semi-functional key.

Now, we will discuss whether $\mathcal{B}$ can distinguish the simulated k_{th} query which is semi-functional in Game_k and normal in $\mathrm{Game}_{k\text{-}1}$ by himself. Assuming that $\mathcal{B}$ has constructed a valid semi-functional ciphertext by itself, namely $\vec{x} \cdot \vec{v} = 0$, the simulated ciphertext must contain the element X_1X_2, since it is the only one can be used for semi-functional ciphertext and $\mathcal{B}$ doesn't know the factor of N. It implies that $z_i = x_i$. Then we have $\vec{z} \cdot \vec{y} = \vec{x} \cdot \vec{v} = 0$. Decryption still work. Therefore, $\mathcal{B}$ can't distinguish the simulated k_{th} key by itself and it only can rely on the output of $\mathcal{A}$ to solve the Assumption.

As to the challenge (m_0, m_1) and ($\vec{x}_0, \vec{x}_1$), $\mathcal{B}$ flips a random coin b, and sets:

$$c^* = \{c_0 = m_b \cdot e(X_1X_2, h)^{-\sum_{i=1,\cdots,n} x_i^b},\ \{c_i = (X_1X_2)^{x_i^b (a-t_i)}\}_{i=1,\cdots,n}\}$$

Let $X_1 X_2 = g^r g_2^c$, we implicitly set $z_i = x_i$, but these values are also actually uncorrelated in the subgroups p_1, p_2 according to the Chinese Remainder Theorem. This is a properly distributed semi-functional ciphertext.

$\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage ε in breaking Assumption 2.

Lemma 3 Assume there is a polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{\text{Game}_q} - Adv_{\mathcal{A}}^{\text{Game}_{final}} = \varepsilon$. Then we can construct a polynomial time simulator $\mathcal{B}$ with advantage ε in breaking Assumption 3.

Proof. $\mathcal{B}$ is given a challenge sample of Assumption 3, $(\mathbb{G}, g, X_3, Z_2, g^r X_2, hY_2, T)$. $\mathcal{B}$ chooses random values $a \in Z_N$, $t_i \in Z_N$, $i = 1, \cdots, n$. The public parameters are set as: $PK = \{g, hY_2, g_1 = g^a, \{T_i = g^{t_i}\}_{i=1,\cdots,n}\}$. hY_2 will seem undistinguishable from h to $\mathcal{A}$, since it is hard to find a non-trivial factor of *N*. $\mathcal{B}$ will simulate Game_q and Game_{final} with $\mathcal{A}$.

As to the key queries $\vec{v}$, $\mathcal{B}$ chooses random $s, y_i' \in Z_N$, and sets the semi-functional key as:

$$\{d_i = (hY_2 g^{sv_i} Z_2^{y_i'} W_i)^{1/(a-t_i)}\}_{i=1,\cdots,n}$$

Let $Y_2 = g_2^f$, $Z_2 = g_2^d$, we implicitly set $y_i = y_i' + f/d$. Thus, this is a properly distributed semi-functional ciphertext.

As to the challenge (m_0, m_1) and ($\vec{x}_0, \vec{x}_1$), $\mathcal{B}$ will imbeds the Assumption 3 into the challenge ciphertext. It flips a random coin *b*, and sets:

$$c* = \{c_0 = m_b \cdot T^{-\sum_{i=1,\cdots,n} x_i^b}, \{c_i = (g^r X_2)^{x_i^b (a-t_i)}\}_{i=1,\cdots,n}\}$$

If $T = e(g,h)^r$, it is a valid semi-functional ciphertext. If $T \in G_T$, this will be a semi-functional encryption of a random message and it is a perfect simulation of Game_{final}.

$\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage ε in breaking Assumption 3.

Theorem 1 If Assumptions 1, 2, and 3 hold, then our PE system is IND-AH-CPA secure.

Proof. If Assumptions 1, 2, and 3 hold, the real security game is indistinguishable from Game_{final} according to the previous lemmas. In Game_{final}, the challenge ciphertext will give no information about b. Therefore, $\mathcal{A}$ only can attain negligible advantage in breaking our construction. This is clear that the PE system is IND-AH-CPA secure.

4. PE-2-PEFKS Transformation

In[3], Boneh et.al proved that an IND-ID-CCA secure IBE could rise from a secure PEKS, but they claimed that it was hard to construct a PEKS from a secure IBE. In[1], Abdalla et.al found that IND-ANO-CPA secure IBE implied the existence of IND-PEKS-CPA secure PEKS. They also proposed a general way to transform any IND-ANO-CPA secure IBE into an IND-PEKS-CPA secure and computationally consistent PEKS. But this kind of PEKS only can test whether the keyword in the ciphertext is match to that in the trapdoor. According to the definition of PEFKS in section 2.2, we will propose a general way to transform IND-AH-CPA secure PE into a PEFKS.

The PE-PEFKS transformation consists of the following steps:

a) Setup(1^λ) can be used as KG (1^λ) to generate (pk, sk);

b) KeyGen algorithm can be used as Td $(sk, \vec{w})$ to get $t_{\vec{w}}$ which will be delivered to server;

c) Choosing a random element R, Encrypt(PK, $\vec{x}$, R) $\to c$ can be used as PEFKS $(pk, \vec{x})$ to encrypt keywords $\vec{x}$, and set $\sigma = (R, c)$;

d) If Decrypt($sk_{\vec{v}}$, c) $\to R$, Test $(t_{\vec{w}}, \sigma) \to 1$. Otherwise Test $(t_{\vec{w}}, \sigma) \to 0$.

The consistency and security of our scheme may be reduced to the security of PE. If an adversary can ruin the consistency and security of PEFKS, we can construct an algorithm to break the PE scheme. In theorem 2, we give the formal result and proof.

Theorem 2 If PE is IND-AH-CPA secure, then PEFKS is computational consistency and IND-PEFKS-CPA secure.

Proof. Assuming there is a polynomial time adversary $\mathcal{L}_1$ that can break the computational consistency of PEFKS. Let $\mathcal{A}$ be a polynomial time adversary of PE. In the key queries phase, $\mathcal{A}$ runs $\mathcal{L}_1(pk)$ to get predicate vector $\vec{v}'$ and attribute vector $\vec{x}$ such that $\vec{x} \cdot \vec{v}' \neq 0$ but Test still output 1. $\mathcal{A}$ also get (R_0, R_1) that is used to break the computational consistency by $\mathcal{L}_1$. $\mathcal{A}$ then issue the challenge query, (R_0, R_1) and $\vec{x}$, and is given the challenge ciphertext $c*$ encrypting R_b

under $\vec{x}$. $\mathcal{A}$ makes key query for $\vec{v}'$, and runs Decrypt($sk_{\vec{v}'}, c^*$) to find *b*. It is easy to see that we can construct an algorithm to break the data privacy property of PE scheme, namely

$$Adv_{\Sigma_{PEFKS},\mathcal{L}_1}^{PEFKS-CONSISTENCY}(\lambda) \le Adv_{\Sigma_{PE},\mathcal{A}}^{PE-IND-CPA}(\lambda).$$

Assuming there is a polynomial time adversary $\mathcal{L}_2$ that can break the IND-PEFKS-CPA security of PEFKS. Let $\mathcal{A}$ be a polynomial time adversary of PE. In the key queries phase, $\mathcal{A}$ runs $\mathcal{L}_2$ (*pk*) to get challenge attribute vectors $\vec{x}_0, \vec{x}_1$ with *R*. Given the challenge ciphertext c^* encrypting *R* under $\vec{x}_b$, $\mathcal{A}$ runs $\mathcal{L}_2$ to find *b*. During this phase, $\mathcal{A}$ answers any trapdoor query of $\mathcal{L}_2$ via its key queries. It is clear that we can construct an algorithm to break the attribute hiding property of PE scheme, namely

$$Adv_{\Sigma_{PEFKS},\mathcal{L}_2}^{PEFKS-IND-CPA}(\lambda) \le Adv_{\Sigma_{PE},\mathcal{A}}^{PE-IAH-CPA}(\lambda).$$

4.1. Our PEFKS

Based on the efficient PE and theorem 2, we can construct an IND-PEFKS-CPA secure PEFKS. The PEFKS works as follows:

KG(1^λ **)** KG**(**1^λ **)=**Setup(1^λ) $\rightarrow$

$$pk = PK = \{g, h, g_1 = g^a, \{T_i = g^{t_i}\}_{i=1,\cdots,n}\},$$

$$sk = MK = \{a, \{t_i\}_{i=1,\cdots,n}\};$$

Td $(sk, \vec{w})$ Td $(sk, \vec{w})$ = KeyGen(MK , $\vec{w}$ **)** $\rightarrow$

$$t_w = \{\{d_i = (hg^{sw_i}W_i)^{1/(a-t_i)}\}_{i=1,\cdots,n}\};$$

PEFKS $(pk, \vec{x})$ To encrypt keyword vector $\vec{x}$, the sender first Chooses a random element R, then runs the Encrypt(PK , $\vec{x}$, R **)** to get c

$$c = \{c_0 = R \cdot e(g,h)^{-r\sum_{i=1,\cdots,n} x_i}, \{c_i = (g_1 T_i^{-1})^{rx_i}\}_{i=1,\cdots,n}\}$$

The ciphertext is set as $\sigma = (R, c)$;

Test $(t_{\vec{w}}, \sigma)$ The server computes $c_0 \cdot \prod_{i=1,\cdots,n} e(c_i, d_i)$. If it values to R, namely $\vec{w} \cdot \vec{x} = 0$, the server sets Test $(t_{\vec{w}}, \sigma) = 1$. Otherwise it sets Test $(t_{\vec{w}}, \sigma) = 0$.

4.2. Application of PEFKS

Previous PEKS scheme only support for equal relation. The new notion of PEFKS opens up a much larger world for searchable encryption. It can provide more sophisticated and flexible relations between the encryption-keyword and trapdoor-keyword.

Previous PEKS could be seem as a subclass of PEFKS. It means that PEFKS support equal relation. E.g. For the keyword w in previous PEKS, the keyword vector is set as $\vec{w} = (1, w)$ and the encrypted keyword vector is set as $\vec{x} = (w', -1)$. If $w = w'$, namely $\vec{w} \cdot \vec{x} = 0$, correctness and security follow.

PEFKS artfully provides multiple keywords search that are connected through conjunctive or disjunctive logical connectives.

- For a disjunctive logical connective, " w_1 and w_2 " which corresponds to the polynomial evaluation $p = r(w_1 - x_1) + (w_2 - x_2)$, the keyword vector is set as $\vec{w} = (rw_1, -r, w_2, -1)$. If $\vec{x} = (1, x_1, 1, x_2)$ and $p = 0$, the Test will be evaluated to 1.
- For a conjunctive logical connective, " w_1 or w_2 " which corresponds to the polynomial evaluation $p = (w_1 - x_1)(w_2 - x_2)$, the keyword vector is set as $\vec{w} = (w_2 w_1, -w_1, -w_2, -1)$. If $\vec{x} = (1, x_1, x_2, x_1 x_2)$ and $p = 0$, the Test will be evaluated to 1.

Conjunctive or disjunctive logical connectives can extend to more complex combinations for boolean formulas. The above polynomial evaluation also can extend to more general polynomial evaluation $p = w_0 + w_1 x + \cdots + w_d x^d$.

We will give a simple application of PEFKS. An email user wants the server to deliver his\her email immediately if the email is appended with keywords "*urgent* and *business*". "*urgent*" and "*business*" may be denote as some value defined by the system. The user set the trapdoor $t_{\vec{w}}$ as $\vec{w} = (w_2 w_1, -w_1, -w_2, -1)$, where w_1 denotes "*urgent*" and w_2 denotes "*business*". If the email is "*urgent* and *business*", a sender send ciphertext of the email and append with the ciphertext σ of keywords vector $\vec{x} = (1, w_1, w_2, w_1 w_2)$. The server then can test wether this email is "*urgent* and *business*" by running **Test** $(t_{\vec{w}}, \sigma)$.

5. Conclusion

We present an Inner-product Predicate Encryption system that is practical based on composite order bilinear groups. The security of our construction is proven IND-AH-CPA secure by adopting the dual system encryption, which is sufficient for PE-2-PEFKS transformation. PEFKS is proposed in this paper. The new notion will be more useful for applications.

There are still some interesting directions. One is to design more sophisticated and flexible functionality $F : Key \times CT \to \{0,1\}^*$ which will be more expressive than inner-product. Another is the possibility of transformation of PEFKS to PE.

References

[1] Abdalla M., Bellare M., Catalano D., Kiltz E., Kohno T., Lange T., Malone-Lee J., Neven G., Paillier P. and Shi H.. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In Advances in Cryptology-Crypto 2005, LNCS 3621, Springer-Verlag, 2005, pp. 205-222.

[2] Baek J., Safiavi-Naini R., Susilo W.. Public Key Encryption with Keyword Search Revisited. Cryptology ePrint Archive, Report 2005/119.

[3] Boneh D., Crescenzo G. D., Ostrovsky R., and Persiano G.. Public Key Encryption with Keyword Search. In Eurocrypt 2004, LNCS 3027, Springer-Verlag, 2004, pp. 506-522.

[4] Boneh D., Goh E., Nissim K.. Evaluating 2-dnf formulas on ciphertexts. In TCC, 2005, pp. 325-342.

[5] Boneh D., Waters B.. Conjunctive, Subset, and Range Queries on Encrypted Data. In Theory of Cryptography Conference, LNCS 4392, Springer-Verlag, 2007, pp. 535-554.

[6] Chase M.. Multi-Authority Attribute Based Encryption. TCC 2007, pp. 515–534.

[7] Cheung L., Newport C.. Provably Secure Ciphertext Policy ABE. In ACM Conference on Computer and Communications Security 2007, pp. 456-465.

[8] Freeman D. M.. Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups. In EUROCRYPT 2010, LNCS 6110, pp. 44–61.

[9] Gentry C.. Practical Identity-Based Encryption Without Random Oracles. In EUROCRYPT 2006, LNCS 4004, pp. 445–464.

[10] Golle P., Staddon J., and Waters B.. Secure Conjunctive Search over Encrypted Data. In ACNS 2004, LNCS 3089, Springer-Verlag, 2004, pp. 31-45.

[11] Goyal V., Pandey O., Sahai A., Waters B.. Attribute-based encryption for finegrained access control of encrypted data. In ACM Conference on Computer and Communications Security, 2006, pp.89-98.

[12] Katz J., Sahai A., Waters B.. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In EUROCRYPT 2008. LNCS 4965, Springer-Verlag, 2008, pp. 146–162.

[13] Lewko A., Okamoto T., Sahai A., Takashima K., Waters B.. Fully Secure Functional Encryption: Attribute-based Encryption and (Hierarchical) Inner Product Encryption. In EUROCRYPT 2010, LNCS 6110, Springer-Verlag, 2010, pp. 62–91.

[14] Lewko A., Waters B.. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In TCC 2010, LNCS 5978, Springer-Verlag, 2010, pp. 455-479.

[15] Okamoto T., Takashima K.. Hierarchical Predicate Encryption for Inner-Products. In ASIACRYPT 2009, LNCS 5912, Springer-Verlag, 2009: 214-231.

[16] Okamoto T., Takashima K.. Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In CRYPTO 2010, LNCS 6223, Springer-Verlag, 2010, pp. 191–208.

[17] Sahai A., Waters B.. Fuzzy Identity Based Encryption. In EUROCRYPT 2005, Springer-Verlag, 2005, pp. 457-473.

[18] Shamir A.. Identity-Based Cryptosystems and Signature Schemes. In CRYPTO 1984, LNCS 196, Springer-Verlag, 2001, pp. 47-53.

[19] Shi E., Waters B.. Delegating Capabilities in Predicate Encryption Systems. In ICALP 2008, Part II, LNCS 5126, Springer-Verlag, 2008, pp. 560–578.

[20] Waters B.. Efficient Identity-Based Encryption without Random Oracles. In EUROCRYPT 2005,

LNCS 3494, Springer-Verlag, 2005, pp.114-127.
[21] Waters B.. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In CRYPTO 2009, LNCS 5677, Springer-Verlag, 2009, pp. 619-636.