The Relation and Transformation between Hierarchical Inner Product Encryption and Spatial Encryption

Jie Chen, Hoon Wei Lim, San Ling, and Huaxiong Wang

Nanyang Technological University, Singapore
s080001@e.ntu.edu.sg
{hoonwei,lingsan,hxwang}@ntu.edu.sg

Abstract. Hierarchical inner product encryption (HIPE) and spatial encryption (SE) are two important classes of functional encryption (FE) that have numerous applications. Although HIPE and SE both involve some notion of linear algebra, the former works in vectors while the latter is based on (affine) spaces. Moreover, they currently possess different properties in terms of security, anonymity (payload/attribute-hiding) and ciphertext sizes, for example. In this paper, we formally study the relation between HIPE and SE. In our work, we discover some interesting and novel property-preserving transformation techniques that enable generic construction of an SE scheme from an HIPE scheme, and vice versa.

1 Introduction

The concept of *identity-based encryption* (IBE) [33, 5, 14, 20, 10, 36, 17, 37, 3] has been extensively studied, particularly in the past decade. It allows a public (encryption) key to be constructed based on some public identification information (or identifier). To encrypt a message, one needs only the intended recipient's identifier and a set of public system parameters. The secret (decryption) key corresponding to a public key is generated by and obtained from a trusted authority.

Over the years, the concept of IBE has been generalized. Since identifiers in the context of IBE can be any arbitrary strings, public keys and their matching private keys can also be associated with attributes, policies, predicates and so on. In recent endeavors toward a more generalized notion of encryption, a new class of cryptographic primitive called *functional encryption* (FE) [9, 32, 31, 18, 8, 21, 22, 26, 13] emerged. Generally, in an FE system for functionality $\mathsf{F}(\cdot,\cdot)$ defined over some key space $\mathcal{K}$ and some plaintext space $\mathcal{X}$, an authority holding a master key can generate a key sk_v for $v \in \mathcal{K}$ that enables the computation of the function $\mathsf{F}(v,\cdot)$ on encrypted data. More precisely, the decryptor can compute $\mathsf{F}(v,x)$ from an encryption of $x \in \mathcal{X}$ using sk_v [13]. In many applications, a plaintext $x \in \mathcal{X}$ is itself a pair $(ind, m) \in \mathcal{I} \times \mathcal{M}$ where ind is an element of an index space $\mathcal{I}$ and m is an element of a payload message space $\mathcal{M}$. There are two types of secrecy in these applications: payload-hiding (with public index or non-anonymous) and attribute-hiding (with hidden index or anonymous) [21]. Roughly speaking, the former requires that only the message m be concealed from an adversary, while the latter requires that both the index ind and the message m be concealed from the adversary. We note that when an FE scheme has v and ind as strings and $\mathsf{F}(v,(ind,m))$ outputs the corresponding payload message m iff $ind = v$, it is essentially an IBE scheme. Given such flexibility and generalization, FE has numerous applications, particularly in the domains of access control, content distribution, mail filtering, data searching, broadcasting, tracing, and biometrics [31, 18, 30, 4, 35, 9, 8, 7, 12, 32].

Most recent work on FE was centered around specific instantiations of FE, such as *hierarchical inner product encryption* (HIPE) [25, 22, 26], *spatial encryption* (SE) [6, 38], and *attribute-based* *encryption* (ABE) [18, 4, 22, 26]. Given the rapid proliferation of various instantiations of FE, it seems to be a useful exercise to examine their relations. In this paper, we focus on investigating the relation between HIPE and SE.

1.1 Motivation

Inner product encryption (IPE) was first proposed by Katz, Sahai and Waters [21].[1] In IPE, $v \in \mathcal{K}$ and $ind \in \mathcal{I}$ are vectors in $\mathbb{Z}_q^n$ for some integer n and prime q. A ciphertext for $\mathbf{x} \in \mathcal{I}$ can be decrypted by a secret key $\mathsf{sk}_{\mathbf{v}}$ for $\mathbf{v} \in \mathcal{K}$, that is $\mathsf{F}(\mathbf{v}, (\mathbf{x}, m)) = m$, iff the inner product $\mathbf{x} \cdot \mathbf{v} = 0$. Subsequently, Okamoto and Takashima [25] proposed HIPE (or "delegatable" IPE), where $\mathsf{F}((\mathbf{v}_1, \ldots, \mathbf{v}_r), ((\mathbf{x}_1, \ldots, \mathbf{x}_h), m)) = m$ for hierarchical vectors $(\mathbf{v}_1, \ldots, \mathbf{v}_r) \in \mathcal{K}$ and $(\mathbf{x}_1, \ldots, \mathbf{x}_h) \in \mathcal{I}$ iff $r \leq h$ and $\mathbf{x}_i \cdot \mathbf{v}_i = 0$ for all $i \in [r]$.[2] Moreover, using the secret key $\mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)}$ for $(\mathbf{v}_1, \ldots, \mathbf{v}_r)$, one can delegate a secret key for a deeper hierarchical vector $(\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{v}_{r+1}) \in \mathcal{K}$, i.e. key delegation.

From an IPE scheme, one can obtain an IBE scheme and a hidden-vector encryption (HVE) scheme. Moreover, we get other variants of public-key encryption schemes supporting polynomial evaluation, equality queries (which is also called keyword search [9]), subset queries, comparison queries, conjunctive normal form, and disjunctive normal form [8, 21]. By making use of the delegation mechanism in HIPE, the applications of IPE can be broaden and generalized to cover other cryptographic primitives such as hierarchical IBE (HIBE), generalized keyword search [1, 25], and delegatable HVE [34]. Indeed, (H)IPE is seen as a very powerful tool or building block for constructing a broad range of public-key encryption schemes.

On the other hand, the notion of SE was proposed by Boneh and Hamburg [6]. Here, $\mathcal{K}$ corresponds to affine spaces in $\mathbb{Z}_q^n$, while $\mathcal{I}$ corresponds to vectors in $\mathbb{Z}_q^n$. For a space $\mathcal{S} \in \mathcal{K}$ and a vector $\mathbf{x} \in \mathcal{I}$, we have $\mathsf{F}(\mathcal{S}, (\mathbf{x}, m)) = m$ iff $\mathbf{x} \in \mathcal{S}$. Furthermore, using the secret key $\mathsf{sk}_{\mathcal{S}}$ for $\mathcal{S}$, one can create a secret key for a subspace $\mathcal{S}'$ of $\mathcal{S}$.

As illustrated in [6], SE is a very generalized encryption system in the sense that many other types of IBE, such as HIBE, inclusive IBE, co-inclusive IBE, broadcast HIBE and forward-secure IBE, are embedded in it. Moreover, we can construct a "product" scheme by embedding multiple instances of SE to obtain additional system properties. For example, an SE scheme can be extended to support multiple authorities by taking the product of the SE scheme and a broadcast scheme whose identities are the names of authorities. See [6, 19] for more embeddings of other cryptosystems into SE.

Although HIPE and SE are related in the sense that they both are instantiations of FE, the former works in vectors while the latter is based on spaces. We notice that existing HIPE schemes [22, 27, 28] possess some properties that are complementary to the SE scheme [6], and vice versa. For example, the HIPE schemes are fully secure and attribute-hiding (anonymous) but do not have short ciphertexts, while the SE scheme has short ciphertexts but is only payload-hiding (non-anonymous) and proven secured in the selective security model. Note that the main tool for constructing an HIPE scheme is dual pairing vector spaces (DPVS) [24, 25], which has very natural orthogonal structure by randomly generating a pair of orthonormal bases. However, one limitation of randomly generated a pair of orthonormal bases is that the size of each element of DPVS is

[1] In [21], IPE is called predicate encryption supporting inner products. We note that the term predicate encryption typically refers to schemes with the attribute-hiding property. But, generally, the notion of (H)IPE also includes schemes with public index.

[2] We use $[r]$ to denote the set $\{1, \ldots, r\}$ and $[r, r']$ to denote the set $\{r, r+1, \ldots, r'\}$.

$\mathcal{O}(n)$, where n is the dimension of the relevant vector space. Hence, the resulting ciphertexts, such as those in previous works [22, 27, 28], are at least $\mathcal{O}(n)$. Okamoto and Takashima [29] recently showed how to reduce the size of ciphertexts by randomly generating a pair of orthonormal bases with special form, such that the resulting HIPE scheme has constant-size ciphertexts. However, the size of ciphertexts is still relatively large in comparison with that of the Boneh-Hamburg SE scheme. Meanwhile, the security of the Boneh-Hamburg SE scheme is analyzed based on the conventional partitioning techniques [10, 11]. It seems difficult to construct an attribute-hiding SE scheme that enjoys full security proven using the partitioning strategy [36].

The main motivation for our work is to understand the relation between HIPE and SE. Particularly, we are looking for an efficient "black-box" or generic construction of SE from HIPE and vice versa. Here, we first observe that there is a simple connection between SE and (non-hierarchical) IPE. It is indeed not difficult to transform an SE scheme to an IPE scheme. We note that the condition $\mathbf{x} \cdot \mathbf{v} = 0$ in IPE holds iff $\mathbf{x} \in \mathcal{S}^{\perp}(\mathbf{v})$, where $\mathcal{S}^{\perp}(\mathbf{v})$ is the orthogonal space of $\mathbf{v}$ (i.e., the Euclidean space spanned by $\mathbf{x}$ such that $\mathbf{x} \cdot \mathbf{v} = 0$). This can be viewed as $\mathsf{F}(\mathcal{S}^{\perp}(\mathbf{v}), (\mathbf{x}, m)) = m$ in the SE setting. In other words, to generate a secret key associated with a vector $\mathbf{v} \in \mathcal{K}$ in an IPE scheme, we can run the key generation algorithm of an SE scheme for the space $\mathcal{S}^{\perp}(\mathbf{v})$. Attrapadung and Libert [2] showed that an IPE scheme can be constructed from an SE scheme in a similar way. They obtained a selective secure, payload-hiding IPE scheme with short ciphertexts from the SE scheme of [6]. This, in turn, implies that given an SE scheme, we can derive an IPE scheme and a broad range of public-key encryption schemes with properties similar to those of the SE scheme.

1.2 Our Results

From studying the relation between HIPE and SE, we discover some interesting and novel techniques for constructing an SE scheme from an HIPE scheme, and vice versa, in such a way that transformation between an SE scheme and an HIPE scheme is property-preserving. The "bridging" between HIPE and SE using our techniques converts the problem of constructing an HIPE scheme with certain properties into a problem of constructing an SE scheme with the required (similar) properties, and vice versa. Our results can be summarized as follows:

(1) Although HIPE and SE are conceptually similar, namely they both involve some notion of linear algebra, their relation from a technical view point is less obvious. It is not at all trivial and clear how one could construct an HIPE scheme from an SE scheme, and vice versa. This is so because of the following reason. In HIPE, $\mathbf{x}_i \cdot \mathbf{v}_i = 0 \Leftrightarrow \mathbf{x}_i \in \mathcal{S}^{\perp}(\mathbf{v}_i)$ for all $i \in [r]$. We require that these orthogonal (Euclidean) spaces be independent from each other. However, in the SE setting, we make use of relations between (affine) spaces and their (affine) subspaces to allow key delegation. This seems to be a contradicting requirement from that for HIPE. Hence, there is no obvious way to directly construct an SE scheme from an HIPE scheme. Nevertheless, it seems more feasible to construct the latter from the former. One possible and natural way to do that is by making use of multiple SE schemes to construct an HIPE scheme such that each SE scheme generates a secret key for an orthogonal space associated with a specific level of a hierarchy. However, such transformation is complicated, inefficient and the security of key delegation is not guaranteed. Furthermore, it is likely that such transformation will not lead to generic construction of HIPE.
In our work, we discover some tricks from linear algebra on which our transformation techniques are based to overcome the aforementioned challenges.

(2) By applying our techniques to the fully secure and attribute-hiding HIPE scheme under a simple assumption of [27], we immediately obtain an SE scheme possessing the same properties. Similarly, using our techniques, one can obtain a fully secure SE scheme with constant-size ciphertexts under a simple assumption and a selective secure and payload-hiding HIPE scheme with short ciphertexts from the HIPE scheme of [29] and the SE scheme of [6] respectively.

(3) As a further contribution, we show how to derive a fully secure SE scheme supporting negation (dealing with the absence of specific vectors) under a simple assumption from the scheme of [26]. That is, given a space $\mathcal{S} \in \mathcal{K}$ and a vector $\mathbf{x} \in \mathcal{I}$, $\mathsf{F}(\mathcal{S}, (\mathbf{x}, m)) = m$ iff $\mathbf{x} \notin \mathcal{S}$. Previous work has achieved only co-selective security [2].

1.3 Organization

We organize the rest of the paper as follows. In Section 2, we provide the definitions and security models of HIPE and SE. In Section 3, we introduce some concepts from linear algebra. Sections 4 & 5 present our property-preserving transformation techniques. We show how to construct SE from HIPE, and vice versa. In Section 6, we further discuss how our work can be extended, particularly to obtain a fully secure SE scheme supporting negation under a simple assumption. In Section 7, we make comparisons between existing and our derived schemes.

2 Definitions

In what follows, we borrow the definition and the game-based security model for FE from [13] which are adequate to define all encryption systems in this paper.

2.1 Functional Encryption

As in [13], we first describe a functionality F of the syntactic definition of FE. The functionality F describes the functions of a plaintext that can be learned from the ciphertext:

Definition 1. *A functionality* F *defined over* $(\mathcal{K}, \mathcal{X})$ *is a function* $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \{0,1\}^*$ *described as a (deterministic) Turing Machine. The set* $\mathcal{K}$ *is called the key space and the set* $\mathcal{X}$ *is called the plaintext space. We require that the key space* $\mathcal{K}$ *contain a special key called the empty key denoted* ϵ.

An FE scheme for the functionality F enables one to evaluate $\mathsf{F}(v, x)$ given the encryption of x and a secret key sk_v for v. The algorithm for evaluation $\mathsf{F}(v, x)$ using sk_v is called *decrypt*. More precisely, an FE scheme is defined as follows:

Definition 2. *A functional encryption scheme (FE) for a functionality* F *defined over* $(\mathcal{K}, \mathcal{X})$ *is a tuple of four probabilistic polynomial-time (PPT) algorithms (*Setup, KeyGen, Enc, Dec*) and an additional, but optional PPT algorithm* KeyDel *satisfying the following correctness condition for all*

$v \in \mathcal{K}$ and $x \in \mathcal{X}$:

$(\mathsf{pp}, \mathsf{mk}) \leftarrow \mathsf{Setup}(1^\lambda)$ (*generate a public and master secret key pair*)
$\mathsf{sk}_v \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{mk}, v)$ (*generate a secret key for v*)
$\mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{pp}, x)$ (*encrypt a plaintext x*)
$y \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_v, \mathsf{ct}_x)$ (*use sk_v to compute $\mathsf{F}(v, x)$ from ct_x*)
$\mathsf{sk}_{v'} \leftarrow \mathsf{KeyDel}(\mathsf{pp}, v, \mathsf{sk}_v, v')$ (*generate a secret key for v' that satisfies a partial order relation denoted as $v' \preceq v$*)

then we require that $y = \mathsf{F}(v, x)$ with probability 1.

The empty key ϵ: The special key ϵ in $\mathcal{K}$ captures all the information about the plaintext that intentionally leaks from the ciphertext. The secret key for ϵ is empty and also denoted by ϵ. Thus, anyone can run $\mathsf{Dec}(\mathsf{pp}, \epsilon, \mathsf{ct}_x)$ on a ciphertext $\mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{pp}, x)$ and obtain all the information about x that intentionally leaks from ct_x. Take IBE for example, $\mathsf{F}(\epsilon, (id, m))$ outputs only $|m|$ (the length of message m) in the attribute-hiding setting while it outputs $|m|$ and the identity id in the payload-hiding setting. Henceforth, we assume that every FE scheme contains the empty key ϵ in the key space $\mathcal{K}$ and we will not explicitly mention it.

We now define the security model for FE. For the plaintext pair $(x_{(0)}, x_{(1)})$ of an adversary's choice, we need the following requirement to make the experiment non-trivial:

$$\mathsf{F}(v, x_{(0)}) = \mathsf{F}(v, x_{(1)}) \text{ for all } v \text{ for which the adversary has } \mathsf{sk}_v. \tag{1}$$

Then we define a security game for an FE scheme as follows:

Definition 3. *For $\beta = 0, 1$ define an experiment β for an adversary $\mathcal{A}$ as follows:*

- $\mathsf{Sim.Setup}$*: It runs* $(\mathsf{pp}, \mathsf{mk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and gives* pp *to $\mathcal{A}$.*
- $\mathsf{Sim.Query}$*: $\mathcal{A}$ adaptively makes repeated key queries of one of three types:*
  - $\mathsf{Sim.KeyGen}$ *$\mathcal{A}$ submits a key generation query $v \in \mathcal{K}$. The challenger generates a secret key* sk_v *for v, but does not give it to $\mathcal{A}$. It instead adds the key to the set $\mathcal{T}$ and gives the adversary a reference to it.*
  - $\mathsf{Sim.KeyDel}$ *$\mathcal{A}$ specifies a key* sk_v *in the set $\mathcal{T}$ for $v \in \mathcal{K}$, then it submits a key delegation query for $v' \in \mathcal{K}$, where $v' \preceq v$. The challenger runs the* $\mathsf{KeyDel}(\mathsf{pp}, v, \mathsf{sk}_v, v')$ *algorithm to get a new secret key* $\mathsf{sk}_{v'}$ *and adds this to the set $\mathcal{T}$.*
  - $\mathsf{Sim.KeyRev}$ *$\mathcal{A}$ specifies an element of the set $\mathcal{T}$ for a secret key* sk_v*. The challenger removes the item from the set $\mathcal{T}$ and gives $\mathcal{A}$ the secret key. We note at this point there is no need for the challenger to allow more delegate queries on the key since $\mathcal{A}$ can run them itself.*
- $\mathsf{Sim.Challenge}$*: $\mathcal{A}$ submits two plaintexts $x_{(0)}$, $x_{(1)} \in \mathcal{X}$ satisfying requirement (1) and in return, it receives* $\mathsf{Enc}(\mathsf{pp}, x_{(\beta)})$.
- $\mathsf{Sim.Guess}$*: $\mathcal{A}$ continues to issue key queries as before subject to requirement (1) and eventually outputs a bit in* $\{0, 1\}$.

For $\beta = 0, 1$ let W_β be the event that the adversary outputs 1 *in Experiment β and define*

$$\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda) := |\Pr[W_0] - \Pr[W_1]|.$$

Definition 4. *An FE scheme is secure if for all PPT adversaries $\mathcal{A}$ the function* $\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}}(\lambda)$ *is negligible.*

2.2 Hierarchical Inner Product Encryption and Spatial Encryption

In HIPE and SE, a plaintext $x \in \mathcal{X}$ is itself a pair $(ind, m) \in \mathcal{I} \times \mathcal{M}$ where ind is called an index and m is called the payload message.

In the HIPE setting, a functionality F is defined over a key space and an index space using sets of hierarchical vectors. Let a hierarchy of depth d vector spaces have the form of $\boldsymbol{\mu} := (n, d; \mu_1, \ldots, \mu_d)$ where $\mu_0 = 0 < \mu_1 < \mu_2 < \ldots < \mu_d = n$. Let $\mathit{\Phi}_i := \mathbb{Z}_q^{\mu_i - \mu_{i-1}} \backslash \{\mathbf{0}\}$ for $i \in [d]$ be the sets of vectors. Let $\mathit{\Phi} := \bigcup_{i=1}^{d} (\mathit{\Phi}_1 \times \ldots \times \mathit{\Phi}_d)$, where the union is a disjoint union. The key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) for HIPE then corresponds to all hierarchical vectors of the form $(\mathbf{v}_1, \ldots, \mathbf{v}_r)$ (resp. $(\mathbf{x}_1, \ldots, \mathbf{x}_h)$) of depth at most d in $\mathit{\Phi}$. Here

$$\mathsf{F}((\mathbf{v}_1, \ldots, \mathbf{v}_r), ((\mathbf{x}_1, \ldots, \mathbf{x}_h), m)) := \begin{cases} m \text{ if } r \leq h \text{ and } \mathbf{x}_i \cdot \mathbf{v}_i = 0 \text{ for all } i \in [r] \\ \perp \text{ otherwise.} \end{cases}$$

Moreover, $(\mathbf{v}'_1, \ldots, \mathbf{v}'_{r'}) \preceq (\mathbf{v}_1, \ldots, \mathbf{v}_r)$ iff $r \leq r'$ and $\mathbf{v}'_i = \mathbf{v}_i$ for all $i \in [r]$. Namely $(\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{v}_{r+1}, \ldots, \mathbf{v}_{r'}) \preceq (\mathbf{v}_1, \ldots, \mathbf{v}_r)$.

In the SE setting, a functionality F is defined over a key space and an index space using sets of spaces and vectors respectively. In an n-dimensional SE scheme, the key space $\mathcal{K}$ corresponds to all affine spaces in $\mathbb{Z}_q^n$ and the index space $\mathcal{I}$ corresponds to all vectors in $\mathbb{Z}_q^n$. Here

$$\mathsf{F}(\mathcal{S}, (\mathbf{x}, m)) := \begin{cases} m \text{ if } \mathbf{x} \in \mathcal{S} \\ \perp \text{ otherwise.} \end{cases}$$

Moreover, $\mathcal{S}' \preceq \mathcal{S}$ iff $\mathcal{S}'$ is a subspace of $\mathcal{S}$.

Let $x_{(0)} = (ind_{(0)}, m_{(0)}), x_{(1)} = (ind_{(1)}, m_{(1)}) \in \mathcal{X}$ be the adversary's choice of plaintext pair. The security game for both HIPE and SE can then be defined using Definition 3 with the following variations:

- If the adversary outputs challenge indices $ind_{(0)}, ind_{(1)}$ before the Setup phase, the security game is then under the *selective security* model. Otherwise it is under the *full security* model.
- If the adversary outputs challenge indices such that $ind_{(0)} = ind_{(1)}$, the security game is then under the *payload-hiding* security model, that is $\mathsf{F}(\epsilon, (ind, m)) = (ind, |m|)$. Otherwise it is under the *attribute-hiding* security model, that is $\mathsf{F}(\epsilon, (ind, m)) = |m|$.

We note that requirement (1) also satisfies the attribute-hiding security model of [21] that allows key queries v_i in which $\mathsf{F}(v_i, (ind_{(0)}, m_{(0)})) = m_{(0)} = m_{(1)} = \mathsf{F}(v_i, (ind_{(1)}, m_{(1)}))$. However, some existing HIPE schemes, for example [25, 22, 27], consider a weaker model in which $\mathsf{F}(v_i, (ind_{(0)}, m_{(0)}))$ (resp. $\mathsf{F}(v_i, (ind_{(1)}, m_{(1)}))$) does not reveal $m_{(0)}$ (resp. $m_{(1)}$) for all key queries v_i.

3 Preliminaries

3.1 Notation

In the remainder of the paper, if not explicitly specified, we assume that all vectors are row vectors in $\mathbb{Z}_q^n$ for some integer n and prime q and spaces are Euclidean spaces spanned by row vectors. Table 1 summarizes some notation used in the remainder of the paper.

Table 1. Notation.

$t \xleftarrow{\$} \mathcal{T}$	t is chosen uniformly at random from the set $\mathcal{T}$
$\mathcal{S}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$	the space spanned by $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$
$\mathcal{S}^\perp(\mathbf{v}_1, \ldots, \mathbf{v}_r)$	the orthogonal space of $\mathcal{S}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$, i.e., the space spanned by all $\mathbf{x}$ where $\mathbf{x} \cdot \mathbf{v}_i = 0$ for $i \in [r]$
$\mathcal{B}(\mathcal{S})$	a basis of the space $\mathcal{S}$
$\mathcal{B}^\perp(\mathcal{S})$	a basis of the space $\mathcal{S}^\perp$
$\mathsf{dim}(\mathcal{S})$	the dimension of the space $\mathcal{S}$
$\mathbf{M}_i$	the i-th row of the matrix $\mathbf{M}$
$\mathcal{S}(\mathbf{M})$	the space spanned by the row vectors of the matrix $\mathbf{M}$
$\mathcal{S}(\mathbf{M}, \mathbf{z})$	the affine space $\{\mathbf{aM} + \mathbf{z} : \mathbf{a} \in \mathbb{Z}_q^r\}$, where $\mathbf{M} \in \mathbb{Z}_q^{r \times n}$

Moreover, we use subscript to indicate the set type of a scheme. For example, we use $\mathcal{K}_{\mathsf{HIPE}}$ (resp. $\mathcal{I}_{\mathsf{HIPE}}$) to denote the key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) with regards to HIPE.

3.2 Concepts from Linear Algebra

We now derive some lemmas associated with vectors, matrices and linear spaces, and which are essential for our property-preserving transformation between HIPE and SE.

Lemma 1. *Given a space $\mathcal{S}$, then $\mathsf{dim}(\mathcal{S}) + \mathsf{dim}(\mathcal{S}^\perp) = n$ and $(\mathcal{S}^\perp)^\perp = \mathcal{S}$.*

The proof for the above lemma is obvious and is omitted from here.

Lemma 2. *Given a space $\mathcal{S}$, there exists a polynomial-time algorithm taking as input $\mathcal{S}$ and outputting a basis $\mathcal{B}^\perp$ of $\mathcal{S}^\perp$.*

Proof. Let the rows of a matrix $\mathbf{M}$ be formed by the row vectors which span $\mathcal{S}$. Then we can use an algorithm (such as Algorithm 2.3.1 of [15]) that takes as input $\mathbf{M}$ and outputs a basis of the kernel of $\mathbf{M}$. This basis is also a basis of $\mathcal{S}^\perp$. □

Henceforth, we refer to the algorithm described in Lemma 2 as the $\mathsf{BasisGen}$ algorithm. We will later use it to transform an element in $\mathcal{K}_{\mathsf{HIPE}}$ to an element in $\mathcal{K}_{\mathsf{SE}}$, and vice versa.

Lemma 3. *Given a space $\mathcal{S}$, a subspace $\mathcal{S}'$ and a basis $\mathcal{B}^\perp$ of $\mathcal{S}^\perp$, there exists a polynomial-time algorithm taking as input $\mathcal{S}, \mathcal{S}', \mathcal{B}^\perp$ and outputting a basis $\mathcal{B}'^\perp$ of $\mathcal{S}'^\perp$, which contains all the vectors of $\mathcal{B}^\perp$.*

Proof. We first run the $\mathsf{BasisGen}(\mathcal{S}')$ algorithm to output a basis $\mathcal{B}''^\perp$ of $\mathcal{S}'^\perp$. We then want to transform this basis to a basis of $\mathcal{S}'^\perp$ which contains $\mathcal{B}^\perp$ (we note that if $\mathcal{S}'$ is a subspace of $\mathcal{S}$, then $\mathcal{S}^\perp$ is a subspace of $\mathcal{S}'^\perp$). Let the rows of matrices $\mathbf{T}$ and $\mathbf{T}'$ be formed by $\mathcal{B}^\perp$ and $\mathcal{B}''^\perp$, respectively. We can then use an algorithm (such as Algorithm 2.3.7 of [15]) that takes as input $\mathbf{T}^\top$, $\mathbf{T}'^\top$ (the transpose of matrices $\mathbf{T}$ and $\mathbf{T}'$, respectively) and outputs a basis for a supplement of $\mathcal{S}^\perp$ in $\mathcal{S}'^\perp$. We add the output basis to $\mathcal{B}^\perp$, which, in turn, is a basis of $\mathcal{S}'^\perp$. □

Henceforth, we refer to the algorithm described in Lemma 3 as the $\mathsf{BasisDel}$ algorithm. It is used during key delegation to transform a partial order associated with $\mathcal{K}_{\mathsf{SE}}$, such as $\mathcal{S}' \preceq \mathcal{S}$, to a partial order associated with $\mathcal{K}_{\mathsf{HIPE}}$, such as $(\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{v}_{r+1}, \ldots, \mathbf{v}_{r'}) \preceq (\mathbf{v}_1, \ldots, \mathbf{v}_r)$.

4 Property-Preserving Transformation from HIPE to SE

Our main idea of transformation from HIPE to SE is transforming the vector inclusion by space relation of SE into the hierarchical inner product relation of HIPE, meanwhile, it preserves the partial order relation in the key space of SE. Our technique involves two key steps: we first describe the transformation from HIPE to SE that works in a subset of affine spaces—Euclidean spaces; we then sketch in Section 4.3 how to derive an n-dimensional SE scheme in affine spaces from an $(n+1)$-dimensional SE scheme in Euclidean spaces.

4.1 Idea for Transforming HIPE to SE

The idea of transformation from HIPE to n-dimensional SE in Euclidean spaces is as follows.

We consider an HIPE scheme with a hierarchy such that each level is n dimensional and which produces ciphertexts that are associated with only hierarchical vectors of the form $(\mathbf{x}, \dots, \mathbf{x})$ (that is all levels with the same vector $\mathbf{x}$). In such a scheme, a secret key associated with a hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathcal{K}_{\mathsf{HIPE}}$ can also be viewed as being associated with the space $\mathcal{S}^{\perp}(\mathbf{v}_1, \dots, \mathbf{v}_r)$. In other words, we can interpret the relation between a ciphertext and a secret key as $\mathsf{F}(\mathcal{S}^{\perp}(\mathbf{v}_1, \dots, \mathbf{v}_r), (\mathbf{x}, m)) = m$ in the SE setting.

With the above observation and the property $(\mathcal{S}^{\perp})^{\perp} = \mathcal{S}$ from Lemma 1, instead of generating a secret key for an $(n-r)$-dimensional space $\mathcal{S} \in \mathcal{K}_{\mathsf{SE}}$, we generate a secret key for a basis $\mathcal{B}^{\perp}(\mathcal{S}) = \{\mathbf{v}_1, \dots, \mathbf{v}_r\} \in \mathcal{K}_{\mathsf{HIPE}}$ of $\mathcal{S}^{\perp}$ by using the KeyGen algorithm of the HIPE scheme. Here, $\mathcal{B}^{\perp}(\mathcal{S})$ can be generated by running the $\mathsf{BasisGen}$ algorithm described in Lemma 2 and we can arrange the basis as a hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_r)$ in the HIPE setting. The initial order of the arranged basis is arbitrary, but the order should be fixed after key generation or delegation.

Furthermore, we note that when more linearly independent vectors are added in the hierarchy, the dimension of the orthogonal space gets smaller. Namely $\mathcal{S}^{\perp}(\mathbf{v}_1, \dots, \mathbf{v}_r, \mathbf{v}_{r+1}, \dots, \mathbf{v}_{r'})$ is a subspace of $\mathcal{S}^{\perp}(\mathbf{v}_1, \dots, \mathbf{v}_r)$. This property is crucial for key delegation. Given a space $\mathcal{S}$, a subspace $\mathcal{S}'$ of $\mathcal{S}$, and a "fixed" basis $\mathcal{B}^{\perp} = \{\mathbf{v}_1, \dots, \mathbf{v}_r\}$ of $\mathcal{S}^{\perp}$, we can derive a basis in the form $\mathcal{B}'^{\perp} = \{\mathbf{v}_1, \dots, \mathbf{v}_r, \mathbf{v}_{r+1}, \dots, \mathbf{v}_{r'}\}$ of $\mathcal{S}'^{\perp}$ by running the $\mathsf{BasisDel}$ algorithm described in Lemma 3. Then we can arrange the basis $\mathcal{B}^{\perp}$, $\mathcal{B}'^{\perp}$ such that $\mathcal{B}'^{\perp} \preceq \mathcal{B}^{\perp}$ in the HIPE setting for key delegation.

4.2 Construction of SE from HIPE

We now describe the construction of an SE scheme from an HIPE scheme using the idea described before. From the property $\mathsf{dim}(\mathcal{S}) + \mathsf{dim}(\mathcal{S}^{\perp}) = n$ of Lemma 1, to construct an n-dimensional SE scheme, we require an HIPE scheme with hierarchy $\boldsymbol{\mu} := ((n-1)n, n-1; n, 2n, \dots, (n-1)n)$. Given an HIPE scheme with five algorithms: $\mathsf{HIPE.Setup}$, $\mathsf{HIPE.KeyGen}$, $\mathsf{HIPE.Enc}$, $\mathsf{HIPE.Dec}$, and $\mathsf{HIPE.KeyDel}$, we construct an SE scheme with the corresponding five algorithms: $\mathsf{SE.Setup}$, $\mathsf{SE.KeyGen}$, $\mathsf{SE.Enc}$, $\mathsf{SE.Dec}$, and $\mathsf{SE.KeyDel}$, as follows:

- $\mathsf{SE.Setup}(1^{\lambda}, n)$ runs
$$(\mathsf{pp}, \mathsf{mk}) \leftarrow \mathsf{HIPE.Setup}(1^{\lambda}, \boldsymbol{\mu}).$$
It outputs the public parameters pp and master key mk.
- $\mathsf{SE.KeyGen}(\mathsf{pp}, \mathsf{mk}, \mathcal{S})$ generates a secret key for an $(n-r)$-dimensional space $\mathcal{S}$. It first runs
$$\mathcal{B}^{\perp}(\mathcal{S}) := \{\mathbf{v}_1, \dots, \mathbf{v}_r\} \leftarrow \mathsf{BasisGen}(\mathcal{S}),$$

and then runs

$$\mathsf{sk}_{\mathcal{S}} \leftarrow \mathsf{HIPE.KeyGen}(\mathsf{pp}, \mathsf{mk}, \mathcal{B}^{\perp}(\mathcal{S})).$$

It outputs the secret key $\mathsf{sk}_{\mathcal{S}}$ with $\mathcal{B}^{\perp}(\mathcal{S})$.

- $\mathsf{SE.Enc}(\mathsf{pp}, \mathbf{x}, m)$ runs

  $$\mathsf{ct}_{\mathbf{x}} \leftarrow \mathsf{HIPE.Enc}(\mathsf{pp}, (\mathbf{x}, \ldots, \mathbf{x}), m).$$

  It outputs the ciphertext $\mathsf{ct}_{\mathbf{x}}$.
- $\mathsf{SE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathcal{S}}, \mathsf{ct}_{\mathbf{x}})$ runs

  $$m \leftarrow \mathsf{HIPE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathcal{S}}, \mathsf{ct}_{\mathbf{x}}).$$

  It outputs the message m.
- $\mathsf{SE.KeyDel}(\mathsf{pp}, \mathcal{S}, \mathsf{sk}_{\mathcal{S}}, \mathcal{S}')$ delegates a secret key to an $(n - r')$-dimensional subspace $\mathcal{S}'$ of $\mathcal{S}$, where $\mathcal{B}^{\perp}(\mathcal{S}) = \{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$. It first runs

  $$\mathcal{B}^{\perp}(\mathcal{S}') := \{\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{v}_{r+1}, \ldots, \mathbf{v}_{r'}\} \leftarrow \mathsf{BasisDel}(\mathcal{S}, \mathcal{S}', \mathcal{B}^{\perp}(\mathcal{S}))$$

  and then runs

  $$\mathsf{sk}_{\mathcal{S}'} \leftarrow \mathsf{HIPE.KeyDel}(\mathsf{pp}, \mathcal{B}^{\perp}(\mathcal{S}), \mathsf{sk}_{\mathcal{S}}, \mathcal{B}^{\perp}(\mathcal{S}')).$$

  It outputs the secret key $\mathsf{sk}_{\mathcal{S}'}$ with $\mathcal{B}^{\perp}(\mathcal{S}')$.

We now show that the resulting SE scheme works correctly and is indeed secure. Since our generic construction of SE scheme requires only a single instance of an HIPE scheme (along with some additional efficient algorithms), we use techniques similar to those for the Embedding Lemma of [6, 19].

Theorem 1. *The SE scheme transformed from the HIPE scheme works correctly.*

Proof. From our construction, the only algorithm that needs to be considered is delegation. Given a partial order pair $\mathcal{S}' \preceq \mathcal{S}$ of $\mathcal{K}_{\mathsf{SE}}$, we transform it into a pair $(\mathcal{B}^{\perp}(\mathcal{S}'), \mathcal{B}^{\perp}(\mathcal{S}))$ such that $\mathcal{B}^{\perp}(\mathcal{S}') \preceq \mathcal{B}^{\perp}(\mathcal{S})$ in the HIPE setting. Thus, the key delegation algorithm also works.

Given a plaintext $(\mathbf{x}, m) \in \mathcal{X}_{\mathsf{SE}}$ and an $(n - r)$-dimensional space $\mathcal{S} \in \mathcal{K}_{\mathsf{SE}}$, we transform them into $((\mathbf{x}, \ldots, \mathbf{x}), m) \in \mathcal{X}_{\mathsf{HIPE}}$ and $(\mathbf{v}_1, \ldots, \mathbf{v}_r) \in \mathcal{K}_{\mathsf{HIPE}}$ respectively, where $\mathcal{B}^{\perp}(\mathcal{S}) = \{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ is a basis of $\mathcal{S}^{\perp}$. Then we have

$$\begin{aligned} \mathsf{F}(\mathcal{S}, (\mathbf{x}, m)) = m &\Leftrightarrow \mathbf{x} \in \mathcal{S} \\ &\Leftrightarrow \mathbf{x} \cdot \mathbf{v}_i = 0 \text{ for all } \mathbf{v}_i \in \mathcal{B}^{\perp}(\mathcal{S}) \\ &\Leftrightarrow \mathsf{F}((\mathbf{v}_1, \ldots, \mathbf{v}_r), ((\mathbf{x}, \ldots, \mathbf{x}), m) = m. \end{aligned}$$

This implies that the resulting SE scheme inherits the decryptability ($\mathsf{F}(\mathcal{S}, (\mathbf{x}, m)) = m$ iff $\mathbf{x} \in \mathcal{S}$ as defined in Section 2.2) from the original HIPE scheme. □

Theorem 2. *For any adversary $\mathcal{A}$ against the SE scheme in the same security model for the original HIPE scheme, there is an adversary $\mathcal{D}$ against the HIPE scheme, running in about the same time as $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{SE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{HIPE}}_{\mathcal{D}}(\lambda).$$

Moreover, the SE scheme preserves properties from the original HIPE scheme.

Proof. Given any adversary $\mathcal{A}$ against the SE scheme in the same security model for the original HIPE scheme (fully/selective secure, attribute/payload-hiding), we simulate an adversary $\mathcal{D}$ with advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SE}}(\lambda)$ against the HIPE scheme as follows:

- $\mathsf{Sim.Setup}$: It runs a real game $\mathcal{RG}$ of HIPE and forwards pp to adversary $\mathcal{A}$.
- $\mathsf{Sim.Query}$: It answers $\mathcal{A}$'s queries by querying $\mathcal{RG}$'s key generation and delegation oracles.
- $\mathsf{Sim.Challenge}$: It forwards $\mathcal{A}$'s challenge to $\mathcal{RG}$ and then returns $\mathcal{RG}$'s output to $\mathcal{A}$.
- $\mathsf{Sim.Guess}$: It answers $\mathcal{A}$'s queries as $\mathsf{Sim.Query}$ phase and forwards $\mathcal{A}$'s guess to $\mathcal{RG}$.

In the above security game, we can efficiently transform the elements in $\mathcal{K}_{\mathsf{SE}}$ and $\mathcal{X}_{\mathsf{SE}}$, as required by the HIPE setting. From Theorem 1 and its proof, all the oracles of the SE scheme can be simulated correctly. Moreover, the plaintexts $x_{(0)} = (\mathbf{x}_{(0)}, m_{(0)})$, $x_{(1)} = (\mathbf{x}_{(1)}, m_{(1)}) \in \mathcal{X}_{\mathsf{SE}}$, any space $\mathcal{S} \in \mathcal{K}_{\mathsf{SE}}$ of $\mathcal{A}$'s choices and those in the HIPE setting satisfy requirement (1) (and other restrictions of the security model) simultaneously. Thus, the simulation is perfect and we conclude that $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{HIPE}}(\lambda)$ is at least $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SE}}(\lambda)$.

It is clear that properties such as full/selective security (under simple assumptions) and attribute/payload-hiding are preserved in the transformation since the model of security game we simulate for the SE scheme is identical to that for the original HIPE scheme. Moreover, should the original HIPE scheme work in prime order bilinear groups and have short ciphertexts (and other properties), the derived SE scheme would also inherit such properties since the SE scheme can be viewed as a "restricted" form or an embedding of the HIPE scheme. □

Remark: We sometimes may require that the key space $\mathcal{K}_{\mathsf{SE}}$ to be containing all vectors in $\mathbb{Z}_q^n$. To achieve this, we can use n extra levels each with two dimensions for the hierarchy $\boldsymbol{\mu}$ in the HIPE scheme to "fix" a vector in the transformation. More precisely, we require a hierarchy $\boldsymbol{\mu} := ((n+1)n, 2n-1; n, 2n, \ldots, (n-1)n, (n-1)n+2, \ldots, (n+1)n)$ and encode $\mathbf{z} = (z_1, \ldots, z_n) \in \mathcal{K}_{\mathsf{SE}}$ as $((1, z_1), \ldots, (1, z_n))$, $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{I}_{\mathsf{SE}}$ as $((x_1, -1), \ldots, (x_n, -1))$ in the last n levels. Nevertheless, we will shortly explain why this is unnecessary for the construction of SE in affine spaces.

4.3 Construction of SE in Affine Spaces

Now, we briefly show how to construct an n-dimensional SE scheme in affine spaces from an $(n+1)$-dimensional SE scheme in Euclidean spaces.

Given an affine space $\mathcal{S}(\mathbf{M}, \mathbf{z}) = \{\mathbf{aM} + \mathbf{z} : \mathbf{a} \in \mathbb{Z}_q^r\}$, where $\mathbf{M} \in \mathbb{Z}_q^{r \times n}$, we embed it in $\mathcal{S}((\mathbf{M}_1, 0), \ldots, (\mathbf{M}_r, 0), (\mathbf{z}, 1)) \in \mathcal{X}_{\mathsf{SE}}$. Given a vector $\mathbf{x} \in \mathbb{Z}_q^n$, we embed it in $(\mathbf{x}, 1) \in \mathcal{I}_{\mathsf{SE}}$. Then it is not difficult to check that $\mathbf{x} \in \mathcal{S}(\mathbf{M}, \mathbf{z})$ iff $(\mathbf{x}, 1) \in \mathcal{S}((\mathbf{M}_1, 0), \ldots, (\mathbf{M}_r, 0), (\mathbf{z}, 1))$. Moreover, if $\mathcal{S}(\mathbf{M}', \mathbf{z}')$ is a subspace of $\mathcal{S}(\mathbf{M}, \mathbf{z})$, namely there is some matrix $\mathbf{T} \in \mathbb{Z}_q^{r' \times r}$ and vector $\mathbf{b} \in \mathbb{Z}_q^r$ such that $\mathbf{M}' = \mathbf{TM}$ and $\mathbf{z}' = \mathbf{z} + \mathbf{bM}$, then $\mathcal{S}((\mathbf{M}'_1, 0), \ldots, (\mathbf{M}'_{r'}, 0), (\mathbf{z}', 1))$ is a subspace of $\mathcal{S}((\mathbf{M}_1, 0), \ldots, (\mathbf{M}_r, 0), (\mathbf{z}, 1))$, and vice versa.

Remark: We note that a vector $\mathbf{z}$ in the key space can be viewed as a special affine space $\mathcal{S}(\mathbf{O}, \mathbf{z})$ (where $\mathbf{O}$ is the zero matrix) and $\mathbf{x} = \mathbf{z}$ is equivalent to $\mathbf{x} \in \mathcal{S}(\mathbf{O}, \mathbf{z})$. Therefore, no additional structure is required to add vectors in the key space $\mathcal{K}_{\mathsf{SE}}$ in our transformation.

5 Property-Preserving Transformation from SE to HIPE

We now turn our attention to the transformation from SE to HIPE. Here, we only require an SE scheme in Euclidean spaces.

5.1 Idea for Transforming SE to HIPE

We first give the idea of constructing an HIPE scheme with hierarchy $\boldsymbol{\mu} := (n, d; \mu_1, \ldots, \mu_d)$ from an SE scheme. Given a vector $\mathbf{v} \in \Phi_i$ (resp. $\mathbf{x} \in \Phi_i$), we use

$$\mathbf{w}^{(i)} := (0, \ldots, 0, \mathbf{v}, 0, \ldots, 0) \quad (\text{resp } \mathbf{y}^{(i)} := (0, \ldots, 0, \mathbf{x}, 0, \ldots, 0))$$

to denote an n-dimensional vector, where $\mathbf{v}$ (resp. $\mathbf{x}$) is embedded in the $(\mu_{i-1}+1)$-th to the μ_i-th scalars.

The key idea of our transformation is adopting some "space partition" technique. We use an n-dimensional SE scheme to embed the i-th level of $\boldsymbol{\mu}$ into the $(\mu_{i-1}+1)$-th up to the μ_i-th scalars. Namely, given a hierarchical vector $(\mathbf{v}_1, \ldots, \mathbf{v}_r) \in \mathcal{K}_{\mathsf{HIPE}}$ (resp. $(\mathbf{x}_1, \ldots, \mathbf{x}_h) \in \mathcal{I}_{\mathsf{HIPE}}$), we embed $\mathbf{v}_i$ (resp. $\mathbf{x}_i$) in $\mathbf{w}_i^{(i)}$ (resp. $\mathbf{y}_i^{(i)}$), which makes each level "independent". More precisely, we generate secret keys and ciphertexts as follows:

(1) A secret key generated using the KeyGen algorithm of the SE scheme for the space $\mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_r^{(r)}) \in \mathcal{K}_{\mathsf{SE}}$ is set to be a secret key for a hierarchical vector $(\mathbf{v}_1, \ldots, \mathbf{v}_r) \in \mathcal{K}_{\mathsf{HIPE}}$;

(2) A message to be encrypted is now associated with the vector $\mathbf{y} = \sum_{i=1}^{h} \mathbf{y}_i^{(i)} \in \mathcal{I}_{\mathsf{SE}}$ instead of a hierarchical vector $(\mathbf{x}_1, \ldots, \mathbf{x}_h) \in \mathcal{I}_{\mathsf{HIPE}}$.

If $r \leq h$, then we have

$$\begin{aligned} \mathbf{x}_i \cdot \mathbf{v}_i = 0 \text{ for } i \in [r] &\Leftrightarrow \mathbf{y} \cdot \mathbf{w}_i^{(i)} = 0 \text{ for } i \in [r] \\ &\Leftrightarrow \mathbf{y} \in \mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_r^{(r)}). \end{aligned}$$

This indicates that the functionality F of the resulting HIPE scheme correctly outputs the payload message (as defined in Section 2.2) if the original SE scheme does.

For a partial order pair $(\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{v}_{r+1}, \ldots, \mathbf{v}_{r'}) \preceq (\mathbf{v}_1, \ldots, \mathbf{v}_r)$ of HIPE, we have $\mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_{r'}^{(r')})$ being a subspace of $\mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_r^{(r)})$. We can then run the delegation algorithm of the SE scheme for $\mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_{r'}^{(r')})$ and the resulting key is set to be the secret key associated with $(\mathbf{v}_1, \ldots, \mathbf{v}_{r'})$.

Up to now, we have considered the case $r \leq h$. However, there is an issue when $h < r$. If we encrypt a message associated with a vector $\mathbf{y} = \sum_{i=1}^{h} \mathbf{y}_i^{(i)} \in \mathcal{I}_{\mathsf{SE}}$ and generate a secret key associated with a space $\mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_r^{(r)}) \in \mathcal{K}_{\mathsf{SE}}$ where $\mathbf{y} \cdot \mathbf{w}_i^{(i)} = 0$ for $i \in [h]$, then we still have $\mathbf{y} \in \mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_r^{(r)})$. This does not meet the security required for HIPE. To address this problem, we use one additional dimension for each level of hierarchy $\boldsymbol{\mu}$ of HIPE in SE. Namely, we require an $(n+d)$-dimensional SE scheme. It will always make $\mathbf{y} \notin \mathcal{S}^{\perp}(\mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_r^{(r)})$ if $h < r$.

5.2 Construction of HIPE From SE

To construct an HIPE scheme with hierarchy $\boldsymbol{\mu} := (n, d; \mu_1, \ldots, \mu_d)$, as explained before, we require an SE scheme with dimension $n' = n + d$. Given a vector $\mathbf{v} \in \Phi_i$ (resp. $\mathbf{x} \in \Phi_i$) and $\hat{b} \in \{0, 1\}$, we use

$$\mathbf{w}^{(i,\hat{b})} := (0, \ldots, 0, (\mathbf{v}, \hat{b}), 0, \ldots, 0) \quad (\text{resp } \mathbf{y}^{(i,\hat{b})} := (0, \ldots, 0, (\mathbf{x}, \hat{b}), 0, \ldots, 0))$$

to denote an n'-dimensional vector, where $(\mathbf{v}, \hat{b})$ (resp. $(\mathbf{x}, \hat{b})$) is embedded in the $(\mu_{i-1} + i)$-th up to the $(\mu_i + i)$-th scalars. Let

$$\mathbf{1}^{(i)} := (0, \ldots, 0, 1, 0, \ldots, 0)$$

be an n'-dimensional vector, where the $(\mu_i + i)$-th scalar is 1.

Given an SE scheme with five algorithms: SE.Setup, SE.KeyGen, SE.Enc, SE.Dec, and SE.KeyDel, we construct an HIPE scheme with the corresponding five algorithms: HIPE.Setup, HIPE.KeyGen, HIPE.Enc, HIPE.Dec, and HIPE.KeyDel, as follows:

- HIPE.Setup$(1^\lambda, \boldsymbol{\mu})$ runs
$$(\mathsf{pp}, \mathsf{mk}) \leftarrow \mathsf{SE.Setup}(1^\lambda, n').$$
It outputs the public parameters pp and master key mk.
- HIPE.KeyGen$(\mathsf{pp}, \mathsf{mk}, (\mathbf{v}_1, \ldots, \boldsymbol{v}_r))$ first runs
$$\mathcal{B}^\perp := \mathcal{B}^\perp(\mathcal{S}(\mathbf{w}_1^{(1,1)}, \ldots, \mathbf{w}_r^{(r,1)})) \leftarrow \mathsf{BasisGen}(\mathcal{S}(\mathbf{w}_1^{(1,1)}, \ldots, \mathbf{w}_r^{(r,1)}))$$
and then runs
$$\mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)} \leftarrow \mathsf{SE.KeyGen}(\mathsf{pp}, \mathsf{mk}, \mathcal{S}(\mathcal{B}^\perp))$$
It outputs the secret key $\mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)}$ with $\mathcal{S}(\mathcal{B}^\perp)$.
- HIPE.Enc$(\mathsf{pp}, (\mathbf{x}_1, \ldots, \mathbf{x}_h), m)$ runs
$$\mathsf{ct}_{(\mathbf{x}_1,\ldots,\mathbf{x}_h)} \leftarrow \mathsf{SE.Enc}(\mathsf{pp}, \sum_{i=1}^{d} \mathbf{y}_i^{(i,0)}, m),$$
where $\mathbf{y}_i^{(i,0)} = \mathbf{1}^{(i)}$ for $i \in [h+1, d]$. It outputs the ciphertext $\mathsf{ct}_{(\mathbf{x}_1,\ldots,\mathbf{x}_h)}$.
- HIPE.Dec$(\mathsf{pp}, \mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)}, \mathsf{ct}_{(\mathbf{x}_1,\ldots,\mathbf{x}_h)})$ runs
$$m \leftarrow \mathsf{SE.Dec}(\mathsf{pp}, \mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)}, \mathsf{ct}_{(\mathbf{x}_1,\ldots,\mathbf{x}_h)}).$$
It outputs the message m.
- HIPE.KeyDel$(\mathsf{pp}, (\mathbf{v}_1, \ldots, \mathbf{v}_r), \mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)}, (\mathbf{v}_1, \ldots, \mathbf{v}_{r'}))$ where the secret key $\mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)}$ is associated with a space $\mathcal{S}(\mathcal{B}^\perp)$. It first runs
$$\mathcal{B}'^\perp := \mathcal{B}^\perp(\mathcal{S}(\mathbf{w}_1^{(1,1)}, \ldots, \mathbf{w}_{r'}^{(r',1)})) \leftarrow \mathsf{BasisGen}(\mathcal{S}(\mathbf{w}_1^{(1,1)}, \ldots, \mathbf{w}_{r'}^{(r',1)}))$$
and then runs
$$\mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_{r'})} \leftarrow \mathsf{SE.KeyDel}(\mathsf{pp}, \mathcal{S}(\mathcal{B}^\perp), \mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_r)}, \mathcal{S}(\mathcal{B}'^\perp)).$$
It outputs the secret key $\mathsf{sk}_{(\mathbf{v}_1,\ldots,\mathbf{v}_{r'})}$ with $\mathcal{S}(\mathcal{B}'^\perp)$.

We show the resulting HIPE scheme works correctly and is indeed secure. The proof techniques are similar to those for Theorems 1 and 2.

Theorem 3. *The HIPE scheme transformed from the SE scheme works correctly.*

Proof. From our construction, the only algorithm that needs to be considered is key delegation. Given a partial order pair $(\mathbf{v}_1, \dots, \mathbf{v}_{r'}) \preceq (\mathbf{v}_1, \dots, \mathbf{v}_r)$ of $\mathcal{K}_{\mathsf{HIPE}}$, we transform it into a pair $(\mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{r'}^{(r',1)}),\ \mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_r^{(r,1)}))$ such that $\mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{r'}^{(r',1)}) \preceq \mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_r^{(r,1)})$ in the SE setting. Thus, the key delegation algorithm also works.

Given a plaintext $((\mathbf{x}_1, \dots, \mathbf{x}_h), m) \in \mathcal{X}_{\mathsf{HIPE}}$ and a hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathcal{K}_{\mathsf{HIPE}}$, we transform them into $\mathbf{y} = \sum_{i=1}^{d} \mathbf{y}_i^{(i,0)} \in \mathcal{X}_{\mathsf{SE}}$ and $\mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_r^{(r,1)}) \in \mathcal{K}_{\mathsf{SE}}$ respectively. Then we have

- if $r > h$, we always let $\mathbf{y}_i^{(i,0)} = \mathbf{1}^{(i)}$ for $i \in [h+1, d]$ in our transformation, then $\mathbf{y} \cdot \mathbf{w}_i^{(i,1)} = 1$ for $i \in [h+1, r]$, implying that $\mathbf{y} \notin \mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_r^{(r,1)})$;
- if $r \leq h$, then

$$\begin{aligned}
\mathsf{F}((\mathbf{v}_1, \dots, \mathbf{v}_r), ((\mathbf{x}_1, \dots, \mathbf{x}_h), m)) = m &\Leftrightarrow \mathbf{x}_i \cdot \mathbf{v}_i = 0 \text{ for } i \in [r] \\
&\Leftrightarrow \mathbf{y} \cdot \mathbf{w}_i^{(i,1)} = 0 \text{ for } i \in [r] \\
&\Leftrightarrow \mathbf{y} \in \mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_r^{(r,1)}) \\
&\Leftrightarrow \mathsf{F}(\mathcal{S}^{\perp}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_r^{(r,1)}), (\mathbf{y}, m)) = m.
\end{aligned}$$

This shows that the resulting HIPE scheme inherits the decryptability from the original SE scheme, i.e., we have $\mathsf{F}((\mathbf{v}_1, \dots, \mathbf{v}_r), ((\mathbf{x}_1, \dots, \mathbf{x}_h), m)) = m$ iff $r \leq h$ and $\mathbf{x}_i \cdot \mathbf{v}_i = 0$ for all $i \in [r]$ as defined in Section 2.2. □

Theorem 4. *For any adversary $\mathcal{A}$ against the HIPE scheme in the same security model for the original SE scheme, there is an adversary $\mathcal{D}$ against the SE scheme, running in about the same time as $\mathcal{A}$, such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HIPE}}(\lambda) \leq \mathsf{Adv}_{\mathcal{D}}^{\mathsf{SE}}(\lambda).$$

Moreover, the HIPE scheme preserves properties from the original SE scheme.

Proof. Given any adversary $\mathcal{A}$ against the HIPE scheme in the same security model for the original SE scheme (fully/selective secure, attribute/payload-hiding), we simulate an adversary $\mathcal{D}$ with advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HIPE}}(\lambda)$ against the SE scheme as with the proof for Theorem 2. From Theorem 3 and its proof, we can conclude that all the oracles of the HIPE scheme can be simulated correctly. Furthermore, the plaintexts $x_{(0)} = ((\mathbf{x}_{1,0}, \dots, \mathbf{x}_{h_0,0}), m_{(0)}), x_{(1)} = ((\mathbf{x}_{1,1}, \dots, \mathbf{x}_{h_1,1}), m_{(1)}) \in \mathcal{X}_{\mathsf{HIPE}}$, any hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathcal{K}_{\mathsf{HIPE}}$ of $\mathcal{A}$'s choices and those in the SE setting satisfy requirement (1) (and other restrictions of the security model) simultaneously.

Using the same argument, the simulation is perfect and we conclude that $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{SE}}(\lambda)$ is at least $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HIPE}}(\lambda)$. Preservation of properties could also be considered in a similar way. □

6 Extensions

6.1 Working in $\mathbb{Z}_N$

Up to this point, we have worked in $\mathbb{Z}_q$, where q is prime (e.g. prime order bilinear groups). Here, we explain how an SE or HIPE scheme is defined and why our transformation techniques also work in $\mathbb{Z}_N$, where N is composite and hard to factor (e.g. composite order bilinear groups).

Given a hard-factoring composite integer N and let $\mathsf{Z}_N = \mathbb{Z}_N^* \cup \{0\}$, operations in Z_N are then closed (i.e., no element in $\mathbb{Z}_N \backslash \mathsf{Z}_N$) except for a negligible probability. In what follows, we assume that all operations are in Z_N without mentioning the negligible probability, since if some elements in $\mathbb{Z}_N \backslash \mathsf{Z}_N$ are involved in any operation in Z_N with this negligible probability, then factoring N becomes trivial.

We first explain how to define HIPE and SE in $\mathbb{Z}_N$. We note that HIPE can be defined in $\mathbb{Z}_N$ in a similar way as before (Section 2.2), since it does not involve any notion of field. On the other hand, to define SE, we first define an "affine space" by considering the combination of a set of vectors that is in the form $\{\mathbf{aM} + \mathbf{z} : \mathbf{a} \in \mathbb{Z}_N^r\}$, where $\mathbf{M} \in \mathbb{Z}_N^{r \times n}$. We still use $\mathcal{S}(\mathbf{M}, \mathbf{z})$ to denote such a "space". A "space" $\mathcal{S}(\mathbf{M}', \mathbf{z}')$ is called a "subspace" of $\mathcal{S}(\mathbf{M}, \mathbf{z})$ if there is some (efficiently computable) matrix $\mathbf{T} \in \mathbb{Z}_N^{r' \times r}$ and vector $\mathbf{b} \in \mathbb{Z}_N^r$ such that $\mathbf{M}' = \mathbf{TM}$ and $\mathbf{z}' = \mathbf{z} + \mathbf{bM}$. With these in mind, we can then define SE in $\mathbb{Z}_N$ as before.

Next, we consider transformation between HIPE and SE in $\mathbb{Z}_N$. Here, orthogonal spaces are essential elements required by the transformation and they can be defined in $\mathbb{Z}_N$ in a similar way.

Definition 5. *Given a set of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ in $\mathbb{Z}_N^n$, the **"orthogonal space"** $\mathcal{S}^{\perp}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$ is formed by a set of vectors $\mathbf{x}$ that satisfy the condition that $\mathbf{x} \cdot \mathbf{v}_i = 0$ for all $i \in [r]$.*

Let $\mathbf{M}$ be a matrix whose rows are formed by a set of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$. We can use row transformation of matrix to obtain a reduced echelon form of $\mathbf{M}$, represented by $\mathbf{E_M}$. It is not difficult to see that $\mathcal{S}(\mathbf{M}, \mathbf{0})$ and $\mathcal{S}(\mathbf{E_M}, \mathbf{0})$ are identical. Here, we still use $\mathcal{S}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$ to denote the "space" $\mathcal{S}(\mathbf{M}, \mathbf{0})$ and let $\mathsf{dim}(\mathcal{S}(\mathbf{v}_1, \ldots, \mathbf{v}_r))$ ("dimension" of $\mathcal{S}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$) denote the number of non-zero rows in $\mathbf{E_M}$. From $\mathbf{E_M}$, i.e. the reduced echelon form of $\mathbf{M}$, we can compute a "basis" of $\mathcal{S}^{\perp}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$, which can also form a matrix in a reduced echelon form if we only switch some columns. Conversely if we consider the "basis" of $\mathcal{S}^{\perp}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$, then $\mathbf{E_M}$ (or $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$) forms the "orthogonal space" of $\mathcal{S}^{\perp}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$. Moreover, $\mathsf{dim}(\mathcal{S}(\mathbf{v}_1, \ldots, \mathbf{v}_r)) + \mathsf{dim}(\mathcal{S}^{\perp}(\mathbf{v}_1, \ldots, \mathbf{v}_r)) = n$.

From the above, it is easy to see that we obtain the similar properties as in Lemma 1. Moreover, it is also not difficult to check that algorithms 2.3.1, 2.3.7 of [15] still work in $\mathbb{Z}_N$ and the proofs for Lemmata 2 and 3 still hold. Therefore, our transformation techniques also work in $\mathbb{Z}_N$.

6.2 A Fully Secure SE Scheme Supporting Negation

Okamoto and Takashima [27] recently proposed FE schemes for a wide class of relations that are specified by non-monotone access structures in combination with inner product relations. As with ABE, they proposed two types of FE schemes, the ciphertext-policy FE (CP-FE) and key-policy FE (FE-FE) schemes. Here, we focus on CP-FE and give a sketch of how to obtain an SE scheme supporting negation (dealing with the absence of specific vectors, i.e., given a space $\mathcal{S} \in \mathcal{K}$ and a vector $\mathbf{x} \in \mathcal{I}$, $\mathsf{F}(\mathcal{S}, (\mathbf{x}, m)) = m$ iff $\mathbf{x} \notin \mathcal{S}$) from the CP-FE scheme of [27] by using our transformation techniques described earlier.

Similarly to HIPE, there is a structure (but non-hierarchical) in CP-FE. Let a structure of size d vector spaces have the form of $\boldsymbol{n} := (d; n_1, \ldots, n_d)$. In CP-FE, $v \in \mathcal{K}$ is a set of labeled vectors $\mathcal{T} := \{(i, \mathbf{v}_i \in \mathbb{Z}_q^{n_i}) | i \in [d]\}$ and $ind \in \mathcal{I}$ is a non-monotone access structure $\mathbb{A}$, namely, a span program $\hat{\mathbf{M}} := (\mathbf{M} \in \mathbb{Z}_q^{h \times \ell}, \rho)$ along with a set of labeled vectors $\{(i_1, \mathbf{x}_1 \in \mathbb{Z}_q^{n_{i_1}}), \ldots, (i_h, \mathbf{x}_h \in \mathbb{Z}_q^{n_{i_h}})\}$ such that $\rho : [h] \to \{(i_1, \mathbf{x}_1), \ldots, (i_h, \mathbf{x}_h), \neg(i_1, \mathbf{x}_1), \ldots, \neg(i_h, \mathbf{x}_h)\}$. Here, $\mathsf{F}(\mathcal{T}, (\mathbb{A}, m)) = m$ holds iff $\{\mathbf{x}_j \cdot \mathbf{v}_{i_j} = 0 \text{ or not}\}_{j \in [h]}$ is input to span program $\hat{\mathbf{M}}$ and the truth-value vector of $(\varrho(\mathbf{x}_1 \cdot \mathbf{v}_{i_1} = 0), \ldots, \varrho(\mathbf{x}_h \cdot \mathbf{v}_{i_h} = 0))$ is accepted by span program $\hat{\mathbf{M}}$.

To construct an n-dimensional SE scheme supporting negation, we require a CP-FE scheme of [26] with structure $\boldsymbol{n} := (n-1; n, \ldots, n)$. The Setup, KeyGen, and Dec algorithms are essentially in the same way as those in the transformation from HIPE to SE shown in Section 4 and the Enc algorithm is run by setting the non-monotone access structure to be 1-out-of-$(n-1)$ secret sharing for the non-zero inner product (i.e. $\mathbf{x} \cdot \mathbf{v}_i \neq 0$). With this and the following property:

$$\mathbf{x} \notin \mathcal{S} \Leftrightarrow \mathbf{x} \cdot \mathbf{v}_i \neq 0 \text{ for some } \mathbf{v}_i \in \mathcal{B}^{\perp}(\mathcal{S}),$$

we derive a fully secure n-dimensional SE scheme supporting negation in Euclidean spaces under a simple assumption. To construct the scheme in affine spaces, we use the same techniques described in Section 4.3 and require that a structure be $\boldsymbol{n} := (n; n+1, \ldots, (n+1))$. Note that the previous work has achieved only co-selective security [2].

7 Discussion and Open Problem

We have seen in the previous sections how the property-preserving transformation is performed between an SE and an HIPE scheme. Namely, we have shown the following relations:

$$\text{SE (in affine spaces)} \Longleftarrow \text{SE (in Euclidean spaces)} \Longleftrightarrow \text{HIPE}.$$

Since the inverse direction of the first arrow is trivial, we obtain:

$$\text{SE} \Longleftrightarrow \text{HIPE}.$$

Let us now compare some selected existing HIPE and SE schemes with schemes derived using our property-preserving transformation techniques. Let $\text{HIPE}_{\text{OT10}}$ denote the HIPE scheme of [27], $\text{HIPE}_{\text{OT11}}$ denote the HIPE scheme of [29], SE_{BH08} denote the SE scheme of [6]. We also let $\text{SE}^*_{\text{OT10}}$, $\text{SE}^*_{\text{OT11}}$, and $\text{HIPE}^*_{\text{BH08}}$ be the corresponding transformed schemes, for example $\text{SE}^*_{\text{OT10}}$ is derived from $\text{HIPE}_{\text{OT10}}$. We compare them in terms of the size of public parameters pp, the size of secret keys sk, the size of ciphertexts ct, the number # of pairing computation for decryption, and their properties. Here, the sizes are in the number of group elements. Moreover, all the SE schemes are n-dimensional and the HIPE schemes are with hierarchy $\boldsymbol{\mu} := (n, d; \mu_1, \ldots, \mu_d)$. A summary of the comparisons is presented in Table 2, where n_μ denotes the maximal value of $\{\mu_i - \mu_{i-1} : i \in [d]\}$.[3]

[3] We have not listed all the existing schemes such as the selective secure HIPE scheme of [25] and the SE scheme in composite order bilinear groups of [23].

Table 2. Comparisons between existing and derived schemes.

	$\mathrm{HIPE}_{\mathrm{OT10}}$	$\mathrm{SE}^*_{\mathrm{OT10}}$	$\mathrm{HIPE}_{\mathrm{OT11}}$	$\mathrm{SE}^*_{\mathrm{OT11}}$	$\mathrm{SE}_{\mathrm{BH08}}$	$\mathrm{HIPE}^*_{\mathrm{BH08}}$
size of pp	$\mathcal{O}(n_\mu^2 d)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^4)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
size of sk	$\mathcal{O}(n_\mu^2 d^2)$	$\mathcal{O}(n^4)$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
size of ct	$\mathcal{O}(n_\mu d)$	$\mathcal{O}(n^2)$	14	14	3	3
# pairings	$\mathcal{O}(n_\mu d)$	$\mathcal{O}(n^2)$	13	13	2	2
fully secure	Yes	Yes	Yes	Yes	No	No
anonymous	Yes	Yes	No	No	No	No
short ct	No	No	Yes	Yes	Yes	Yes
prime order	Yes	Yes	Yes	Yes	Yes	Yes
simple assumptions	Yes	Yes	Yes	Yes	No	No

In comparison with $\mathrm{SE}_{\mathrm{BH}}$, as shown in Table 2, $\mathrm{SE}^*_{\mathrm{OT10}}$, $\mathrm{SE}^*_{\mathrm{OT11}}$ achieve full security at the expense of increase in the sizes of the public parameters, secret keys and ciphertexts, and the number of pairing evaluations, approximately in the factor of between n^2 and n^3. Moreover, $\mathrm{SE}^*_{\mathrm{OT10}}$ achieves attribute-hiding and $\mathrm{SE}^*_{\mathrm{OT11}}$ has constant-size ciphertexts. On the other hand, $\mathrm{HIPE}^*_{\mathrm{BH}}$ requires roughly about a factor of n elements less than $\mathrm{HIPE}_{\mathrm{OT10}}$ and $\mathrm{HIPE}_{\mathrm{OT11}}$ in terms of the sizes of the public parameters and secret keys. It achieves significant improvement in terms of the size of ciphertexts and the number of pairing computations during decryption, with only 3 group elements and 2 pairings, respectively.

We have obtained some new instances of HIPE and SE schemes with different properties. However, constructing these schemes that are fully secure, attribute-hiding, and with short ciphertexts in prime order bilinear groups (which are more efficient [16]) still remains an open problem, particularly those that work under simple assumptions.

Acknowledgements The author thanks the reviewers for their helpful comments and suggestions for this paper. This work is supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03.

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *J. Cryptology*, 21(3):350–391, 2008.

[2] N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *Public Key Cryptography*, pages 384–402, 2010.

[3] M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In *TCC*, pages 235–252, 2011.

[4] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[5] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[6] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.

[7] D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *ACM Conference on Computer and Communications Security*, pages 211–220, 2006.

[8] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007.

[9] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[10] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

[11] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.

[12] D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT*, pages 573–592, 2006.

[13] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.

[14] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[15] H. Cohen, editor. *A Course in Computational Algebraic Number Theory.* Springer, 1996. ISBN 0387556400.

[16] D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT*, pages 44–61, 2010.

[17] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[19] M. Hamburg. Spatial encryption. *IACR Cryptology ePrint Archive*, 2011:389, 2011.

[20] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT*, pages 466–481, 2002.

[21] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

[22] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.

[23] D. Moriyama and H. Doi. A fully secure spatial encryption scheme. *IEICE Transactions*, 94-A (1):28–35, 2011.

[24] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.

[25] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.

[26] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.

[27] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. *IACR Cryptology ePrint Archive*, 2010:563, 2010.

[28] T. Okamoto and K. Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. *IACR Cryptology ePrint Archive*, 2011:543, 2011.

[29] T. Okamoto and K. Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In *CANS*, pages 138–159, 2011.

[30] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pages 195–203, 2007.

[31] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In *ACM Conference on Computer and Communications Security*, pages 99–112, 2006.

[32] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[33] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[34] E. Shi and B. Waters. Delegating capabilities in predicate encryption systems. In *ICALP (2)*, pages 560–578, 2008.

[35] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

[36] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[37] B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

[38] M. Zhou and Z. Cao. Spatial encryption under simpler assumption. In *ProvSec*, pages 19–31, 2009.