Functional Encryption for Public-Attribute Inner Products: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation

Nuttapong Attrapadung[1] and Benoît Libert[2⋆]

[1] Research Center for Information Security, AIST (Japan)
[2] Université catholique de Louvain, Crypto Group (Belgium)

Abstract. In functional encryption (FE) schemes, ciphertexts and private keys are associated with attributes and decryption is possible whenever key and ciphertext attributes are suitably related. It is known that expressive realizations can be obtained from a simple functional encryption flavor called inner product encryption (IPE), where decryption is allowed whenever ciphertext and key attributes form orthogonal vectors. In this paper, we construct public-attribute inner product encryption (PAIPE) systems, where ciphertext attributes are public (in contrast to attribute-hiding IPE systems). Our PAIPE schemes feature *constant-size* ciphertexts for the zero *and* non-zero evaluations of inner products. These schemes respectively imply an adaptively secure identity-based broadcast encryption scheme and an identity-based revocation mechanism that both feature short ciphertexts and rely on simple assumptions in prime order groups. We also introduce the notion of *negated spatial encryption*, which subsumes non-zero-mode PAIPE and can be seen as the revocation analogue of the spatial encryption primitive of Boneh and Hamburg.

Keywords. Functional encryption, identity-based broadcast encryption, revocation, efficiency.

1 Introduction

Ordinary encryption schemes usually provide coarse-grained access control since, given a ciphertext, only the holder of the private key can obtain the plaintext. In many applications such as distributed file systems, the need for fine-grained and more complex access control policies frequently arises. To address these concerns, several kinds of *functional public key encryption* schemes have been studied.

Functional encryption can be seen as a generalization of identity-based encryption (IBE) [27, 9]. In IBE schemes, the receiver's ability to decrypt is merely contingent on his knowledge of a private key associated with an identity that matches a string chosen by the sender. In contrast, functional encryption (FE) systems make it possible to decrypt using a private key $\mathsf{sk}_{\boldsymbol{x}}$ corresponding to a set $\boldsymbol{x}$ of atomic elements, called *attributes*, that is suitably related – according to some well-defined relation R – to another attribute set $\boldsymbol{y}$ specified by the sender.

The goal of this paper is to describe new (pairing-based) functional encryption constructions providing short ciphertexts (ideally, their length should not depend on the size of attribute sets) while providing security against adaptive adversaries or supporting negation (e.g. decryption should be precisely disallowed to holders of private keys $\mathsf{sk}_{\boldsymbol{x}}$ for which $R(\boldsymbol{x}, \boldsymbol{y}) = 1$).

1.1 Related Work

The first flavor of functional encryption traces back to the work of Sahai and Waters [25] that was subsequently extended in [18, 24]. Their concept, called *attribute-based encryption* (ABE), allows

⋆ This author acknowledges the Belgian National Fund for Scientific Research (F.R.S.-F.N.R.S.) for his support as a "Chargé de Recherches" and the BCRYPT Interuniversity Attraction Pole.

a sender to encrypt data under a set of attributes ω while an authority generates private keys for access control policies $\mathcal{T}$. Decryption rights are granted to anyone holding a private key for a policy $\mathcal{T}$ such that $\mathcal{T}(\omega) = 1$. Identity-based broadcast encryption (IBBE) [2, 26, 15, 11] and revocation (IBR) [22] schemes can also be thought of as functional encryption systems where ciphertexts are encrypted for a set of identities $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_n\}$: in IBBE (resp. IBR) systems, decryption requires to hold a private key $\mathsf{sk}_{\mathsf{ID}}$ for which $\mathsf{ID} \in S$ (resp. $\mathsf{ID} \notin S$).

The above kinds of functional encryption systems are only *payload hiding* in that they keep encrypted messages back from unauthorized parties but ciphertexts do not hide their underlying attribute set. *Predicate encryption* schemes [12, 20, 29, 28] additionally provide *anonymity* as ciphertexts also conceal the attribute set they are associated with, which is known to enable [8, 1] efficient searches over encrypted data. In [20], Katz, Sahai and Waters devised a predicate encryption scheme for inner products: a ciphertext encrypted for the attribute vector $\vec{Y}$ can be opened by any key $\mathsf{sk}_{\vec{X}}$ such that $\vec{X} \cdot \vec{Y} = 0$. As shown in [20], inner product encryption (IPE) suffices to give functional encryption for a number of relations corresponding to the evaluation of polynomials or CNF/DNF formulae.

1.2 Our Contributions

While quite useful, the IPE scheme of [20] strives to anonymize ciphertexts, which makes it difficult to break through the linear complexity barrier (in the vector length n) in terms of ciphertext size. It indeed seems very hard to avoid such a dependency as long as anonymity is required: for instance, anonymous FE constructions [12, 19] suffer from the same overhead. A similar problem appears in the context of broadcast encryption, where the only known scheme [4] that conceals the receiver set also has $O(n)$-size ciphertexts.

This paper focuses on applications of functional encryption schemes, such as identity-based broadcast encryption and revocation systems, where the anonymity property is not fundamental. Assuming public ciphertext attributes rather than anonymity may be useful in other contexts. For instance, suppose that a number of ciphertexts are stored with varying attributes $\boldsymbol{y}$ on a server and we want to decrypt only those for which $R(\boldsymbol{x}, \boldsymbol{y}) = 1$. Anonymous ciphertexts require to decrypt all of them whereas public attributes $\boldsymbol{y}$ make it possible to test whether $R(\boldsymbol{x}, \boldsymbol{y})$ (which is usually faster than decrypting) and only decrypt appropriate ones.

At the expense of sacrificing anonymity, we thus describe public-attribute inner product encryption (which we will call PAIPE to distinguish the primitive from IPE schemes with the attribute hiding property) schemes where the ciphertext overhead reduces to $O(1)$ as long as the description of the ciphertext attribute vector is not considered as being part of the ciphertext, which is a common assumption in the broadcast encryption/revocation applications (*i.e.*, the list of receivers is not seen as a ciphertext component). In addition, the number of pairing evaluations to decrypt is also constant, which significantly improves upon $O(n)$, since pairings calculations still remain costly.

Our first PAIPE system achieves adaptive security, as opposed to the selective model, used in [20], where the adversary has to choose the target ciphertext vector $\vec{Y}$ upfront. To acquire adaptive security, we basically utilize the method used in the Waters' fully secure IBE [31], albeit we also have to introduce a new trick called "n-equation technique" so as to deal with the richer structure of PAIPE. Our system directly yields the first adaptively secure identity-based broadcast encryption scheme with constant-size ciphertexts in the standard model. Previous IBBE with $O(1)$-size ciphertexts were either only selective-ID secure [2, 15, 11, 26] or in the random oracle model [17].

Among IBBE systems featuring compact ciphertexts (including selective-ID secure ones), ours is also the first one relying on simple assumptions in prime order groups: it does not use any "q-type" assumption, where the input includes a sequence of elements $\{g^{(a^i)}\}_{i=0}^{q}$.

It is worth mentioning that techniques developed by Lewko and Waters [23] can be applied to the construction of Boneh and Hamburg [11] (as we show in appendix B) to give fully secure IBBE with short ciphertexts in composite order groups. However, it was not previously known how to obtain such a scheme in prime order groups (at least without relying on the absence of computable isomorphism in asymmetric pairing configurations). Indeed, despite recent progress [16], there is still no black-box way to translate pairing-based cryptosystems from composite to prime order groups. In particular, Freeman's framework [16] does not apply to [23].

Our second contribution is a PAIPE system for non-zero inner products: ciphertexts encrypted for vector $\vec{Y}$ can only be decrypted using $\mathsf{sk}_{\vec{X}}$ if $\vec{X} \cdot \vec{Y} \neq 0$, which – without retaining anonymity – solves a question left open by Katz, Sahai and Waters [20][Section 5.4]. The scheme implies the first identity-based revocation (IBR) mechanism [22] with $O(1)$-size ciphertexts. Like the two schemes of Lewko, Sahai and Waters [22], its security is analyzed in a non-adaptive model where the adversary has to choose which users to corrupt at the outset of the game[3]. In comparison with [22] where ciphertexts grow linearly with the number of revoked users and public/private keys have constant size, our basic IBR construction performs exactly in the dual way since key sizes depend on the maximal number of revoked users. Depending on the application, one may prefer one scheme over the other one. We actually show how to generalize both implementations and obtain a tradeoff between ciphertext and key sizes (and without assuming a maximal number of revoked users): the second scheme of [22] and ours can be seen as lying at opposite extremities of the spectrum. In appendix E, we also describe a somewhat simpler variant of our non-zero PAIPE scheme in groups of composite order.

On a theoretical side, our non-zero PAIPE realization turns out to be a particular case of a more general primitive, that we call *negated spatial encryption*, which we define as a negated mode for the spatial encryption primitive of Boneh and Hamburg [11]. Namely, keys correspond to subspaces and can decrypt ciphertexts encrypted under points that lie *outside* the subspace. This generalized primitive turns out to be non-trivial to implement and we had to use a fully generalized form of our new "n-equation" technique. The proposed scheme is proven secure under a non-standard assumption defined in [22].

1.3 Our Techniques

The core technique of our *non-zero* PAIPE scheme will be used throughout the paper, including in our adaptively secure *zero* PAIPE scheme. This can be viewed analogously to fact that Waters' fully secure IBE [31] uses the revocation technique of [22]. Our non-zero PAIPE also builds on [22]. However, the fact that non-zero PAIPE has much richer structure than revocation scheme and the pursued goal of achieving constant ciphertext size together prevent us from using their techniques directly. To describe the difficulties that arise, we first outline the Lewko-Sahai-Waters revocation scheme in its simplified form where only one user is revoked.

Construction 1. (A SIMPLIFIED REVOCATION SCHEME)

[3] We actually work in a slightly stronger model, called *co-selective-ID*, where the adversary chooses which parties to corrupt at the beginning – before seeing the public key – but is not required to announce the target revoked set until the challenge phase.

▶ Setup: chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p and picks $g \xleftarrow{\$} \mathbb{G}$ as well as $\alpha, \alpha_1, \alpha_2 \xleftarrow{\$} \mathbb{Z}_p$. The public key is $\big(g, g^{\alpha_1}, g^{\alpha_2}, e(g,g)^{\alpha}\big)$. The master key is g^{α}.

▶ KeyGen: chooses $t \xleftarrow{\$} \mathbb{Z}_p$ at random and outputs a private key for an identity $\mathsf{ID} \in \mathbb{Z}_p$ as $(K_0 = g^t,\ K_1 = g^{\alpha+\alpha_1 t},\ K_2 = g^{t(\alpha_1 \mathsf{ID}+\alpha_2)})$.

▶ Encrypt: encrypts M and specifies a revoked identity ID' by choosing $s \xleftarrow{\$} \mathbb{Z}_p$ and computing $(E_0 = \mathsf{M} \cdot e(g,g)^{\alpha s},\ E_1 = g^{s(\alpha_1 \mathsf{ID}'+\alpha_2)},\ E_2 = g^s)$.

▶ Decrypt: decryption computes $e(K_2, E_2)^{\frac{1}{\mathsf{ID}-\mathsf{ID}'}} e(E_1, K_0)^{-\frac{1}{\mathsf{ID}-\mathsf{ID}'}} = e(g,g)^{\alpha_1 ts}$ if $\mathsf{ID} \neq \mathsf{ID}'$. It then computes $e(g,g)^{\alpha s}$ as $e(K_1, E_2)/e(g,g)^{\alpha_1 ts} = e(g,g)^{\alpha s}$.

The scheme can be explained by viewing a key and a ciphertext as forming a linear system of 2 equations in the exponent of $e(g,g)$ with variables $\alpha_1 ts, \alpha_2 ts$.

$$M_{\mathsf{ID},\mathsf{ID}'} \begin{pmatrix} \alpha_1 ts \\ \alpha_2 ts \end{pmatrix} := \begin{pmatrix} \mathsf{ID} & 1 \\ \mathsf{ID}' & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 ts \\ \alpha_2 ts \end{pmatrix} = \begin{pmatrix} \log(e(K_2, E_2)) \\ \log(e(E_1, K_0)) \end{pmatrix}.$$

Computing the blinding factor $e(g,g)^{\alpha_1 ts}$ amounts to solve the system, which is possible when $\det(M_{\mathsf{ID},\mathsf{ID}'}) \neq 0$ (and thus $\mathsf{ID} \neq \mathsf{ID}'$, as required). In particular, decryption computes a linear combination (in the exponent) with coefficients from the first row of $M^{-1}_{\mathsf{ID},\mathsf{ID}'}$ which is $(\frac{1}{\mathsf{ID}-\mathsf{ID}'}, \frac{-1}{\mathsf{ID}-\mathsf{ID}'})$. In [22], this is called "2-equation technique". The scheme is extended to n-dimension, *i.e.,* the revocation of n users $\{\mathsf{ID}'_1, \ldots, \mathsf{ID}'_n\}$, by utilizing n local independent systems of two equations

$$M_{\mathsf{ID},\mathsf{ID}'_j} \Big(\alpha_1 ts_j,\ \alpha_2 ts_j\Big)^{\top} = \Big(\log(e(K_2, E_{2,j})),\ \log(e(E_{1,j}, K_0))\Big)^{\top} \text{ for } j \in [1,n],$$

that yield $2n$ ciphertext components $(E_{1,j}, E_{2,j})$, each one of which corresponds to a share s_j of s such that $s = \sum_1^n s_j$. The decryption at j-th system returns $e(g,g)^{\alpha_1 ts_i}$ if $\mathsf{ID} \neq \mathsf{ID}'_j$. Combining these results finally gives $e(g,g)^{\alpha_1 ts}$.

We aim at *constant-size* ciphertexts for non-zero PAIPE schemes of dimension n. When trying to use the 2-equation technique with n dimensions, the following difficulties arise. First, the "decryptability" condition $\vec{X} \cdot \vec{Y} \neq 0$ cannot be decomposed as simply as the condition of the revocation scheme, which is decomposable as the conjunction of $\mathsf{ID} \neq \mathsf{ID}'_j$ for $j \in [1,n]$. Second, the ciphertext size was $O(n)$ and we want to decrease it to $O(1)$.

Towards solving these problems, we introduce a technique called "n-equation technique". First, we utilize n secret exponents $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)^{\top}$ and let α_1 function as the "master" exponent while $\alpha_2, \ldots, \alpha_n$ serve as the "perturbed" factors. Intuitively, we will set up a system of n linear equations of the form:

$$M_{\vec{X},\vec{Y}}(\alpha_1 ts, \ldots, \alpha_n ts)^{\top} = \big(\log(e(K_{i_1}, E_{j_1})), \ldots, \log(e(K_{i_n}, E_{j_n}))\big)^{\top} \tag{1}$$

where $\{K_{i_k}\}$ and $\{E_{j_k}\}$ are elements of $\mathbb{G}$ defined for a key for $\vec{X}$ and a ciphertext for $\vec{Y}$ respectively. At first, this generalized system seems to require linear-size ciphertexts $(E_{j_1}, \ldots, E_{j_n})$. A trick to resolve this is to reuse ciphertext elements throughout the system: we let $E_{j_k} = E_2 = g^s$ for $k \in [1, n-1]$. This effectively yields a constraint $M_{\vec{X},\vec{Y}} = \big(Q^{\top}_{\vec{X}} \quad R^{\top}\big)^{\top}$, where $Q_{\vec{X}}$ is a $(n-1) \times n$ matrix parameterized only by $\vec{X}$ and R is a $1 \times n$ matrix. The remaining problem is then to choose

$M_{\vec{X},\vec{Y}}$ in such a way that the system has a solution if $\vec{X} \cdot \vec{Y} \neq 0$ (the decryptability condition). To this end, we define

$$M_{\vec{X},\vec{Y}} := \begin{pmatrix} -\frac{x_2}{x_1} & 1 & & \\ -\frac{x_3}{x_1} & & 1 & \\ \vdots & & & \ddots \\ -\frac{x_n}{x_1} & & & 1 \\ y_1 & y_2 & y_3 & \cdots & y_n \end{pmatrix}, \tag{2}$$

where it holds that $\det(M_{\vec{X},\vec{Y}}) = (-1)^{n+1}\vec{X} \cdot \vec{Y}/x_1$. By translating this conceptual view back into algorithms, we obtain a basic non-zero PAIPE scheme. From this, we propose two schemes for non-zero PAIPE: the first one is a special case of negated spatial encryption in section 5.1, while the second one is proven secure under simple assumptions and given in section 5.2.

1.4 Organization

In the forthcoming sections, the syntax and the applications of functional encryption are explained in sections 2 and 3. We describe our zero mode PAIPE system in section 4. Our negated schemes are detailed in section 5.

2 Definitions

2.1 Syntax and Security Definition for Functional Encryption

Let $R : \Sigma_{\mathsf{k}} \times \Sigma_{\mathsf{e}} \to \{0,1\}$ be a boolean function where Σ_{k} and Σ_{e} denote "key attribute" and "ciphertext attribute" spaces. A functional encryption (FE) scheme for R consists of the following algorithms.

- $\mathsf{Setup}(1^\lambda, des) \to (\mathsf{pk}, \mathsf{msk})$: takes as input a security parameter 1^λ and a scheme description des (which usually describes the dimension n), and outputs a master public key pk and a master secret key msk.
- $\mathsf{KeyGen}(\boldsymbol{x}, \mathsf{msk}) \to \mathsf{sk}_{\boldsymbol{x}}$: takes as input a key attribute $\boldsymbol{x} \in \Sigma_{\mathsf{k}}$ and the master key msk. It outputs a private decryption key $\mathsf{sk}_{\boldsymbol{x}}$.
- $\mathsf{Encrypt}(\boldsymbol{y}, \mathsf{M}, \mathsf{pk}) \to C$: takes as input a ciphertext attribute $\boldsymbol{y} \in \Sigma_{\mathsf{e}}$, a message $\mathsf{M} \in \mathcal{M}$, and public key pk. It outputs a ciphertext C.
- $\mathsf{Decrypt}(C, \boldsymbol{y}, \mathsf{sk}_{\boldsymbol{x}}, \boldsymbol{x}) \to \mathsf{M}$: given a ciphertext C with its attribute $\boldsymbol{y}$ and the decryption key $\mathsf{sk}_{\boldsymbol{x}}$ with its attribute $\boldsymbol{x}$, it outputs a message M or $\perp$.

We require the standard correctness of decryption. Namely, for all security parameters $\lambda \in \mathbb{N}$, all key pairs $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\boldsymbol{x} \in \Sigma_{\mathsf{k}}$, all $\mathsf{sk}_{\boldsymbol{x}} \leftarrow \mathsf{KeyGen}(\boldsymbol{x}, \mathsf{msk})$, and all $\boldsymbol{y} \in \Sigma_{\mathsf{e}}$,

- If $R(\boldsymbol{x}, \boldsymbol{y}) = 1$, then $\mathsf{Decrypt}(\mathsf{Encrypt}(\boldsymbol{y}, \mathsf{M}, \mathsf{pk}), \mathsf{sk}_{\boldsymbol{x}}) = \mathsf{M}$.
- If $R(\boldsymbol{x}, \boldsymbol{y}) = 0$, $\mathsf{Decrypt}(\mathsf{Encrypt}(\boldsymbol{y}, \mathsf{M}, \mathsf{pk}), \mathsf{sk}_{\boldsymbol{x}}) = \perp$ with probability nearly 1.

Terminology and Variants. We refer to any encryption primitive A that can be casted as a functional encryption by specifying its corresponding function $R^{\mathsf{A}} : \Sigma_{\mathsf{k}}^{\mathsf{A}} \times \Sigma_{\mathsf{e}}^{\mathsf{A}} \to \{0,1\}$. For a FE primitive A, we can define two variants:

- **Dual Variant**, denoted by $\mathsf{Dual}(\mathsf{A})$, is defined by setting $\Sigma_{\mathsf{k}}^{\mathsf{Dual}(\mathsf{A})} := \Sigma_{\mathsf{e}}^{\mathsf{A}}$ as well as $\Sigma_{\mathsf{e}}^{\mathsf{Dual}(\mathsf{A})} := \Sigma_{\mathsf{k}}^{\mathsf{A}}$ and $R^{\mathsf{A}}(\boldsymbol{x}, \boldsymbol{y}) = R^{\mathsf{Dual}(\mathsf{A})}(\boldsymbol{y}, \boldsymbol{x})$. In a dual variant, the roles of key and ciphertext attributes are swapped from those of its original primitive.
- **Negated Variant**, denoted by $\mathsf{Neg}(\mathsf{A})$, is defined by using the same domains as A and setting $R^{\mathsf{Neg}(\mathsf{A})}(\boldsymbol{x}, \boldsymbol{y}) = 1 \Leftrightarrow R^{\mathsf{A}}(\boldsymbol{x}, \boldsymbol{y}) = 0$. The condition is thus the opposite of the original primitive.

Security Definition. A FE scheme for a function $R : \Sigma_{\mathsf{k}} \times \Sigma_{\mathsf{e}} \to \{0,1\}$ is fully secure if no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ has non-negligible advantage in the following game.

Setup. The challenger runs $\mathsf{Setup}(\mathrm{n})$ and hands the public key pk to $\mathcal{A}$.

Query Phase 1. The challenger answers all private key queries for $\boldsymbol{x} \in \Sigma_{\mathsf{k}}$ by returning $\mathsf{sk}_{\boldsymbol{x}} \leftarrow \mathsf{KeyGen}(\boldsymbol{x}, \mathsf{msk})$.

Challenge. $\mathcal{A}$ submits equal-length messages $\mathsf{M}_0, \mathsf{M}_1$ and a target ciphertext attribute vector $\boldsymbol{y}^\star \in \Sigma_{\mathsf{e}}$ such that $R(\boldsymbol{x}, \boldsymbol{y}^\star) = 0$ for all key attributes $\boldsymbol{x}$ that have been queried so far. The challenger then flips a bit $\beta \xleftarrow{\$} \{0,1\}$ and computes the challenge ciphertext $C^\star \leftarrow \mathsf{Encrypt}(\boldsymbol{y}, \mathsf{M}_\beta, \mathsf{pk})$ which is given to $\mathcal{A}$.

Query Phase 2. The adversary is allowed to make further private key queries $\boldsymbol{x} \in \Sigma_{\mathsf{k}}$ under the same restriction as above, *i.e.,* $R(\boldsymbol{x}, \boldsymbol{y}^\star) = 0$.

Guess. The adversary $\mathcal{A}$ outputs a guess $\beta' \in \{0,1\}$ and wins if $\beta' = \beta$. In the game, $\mathcal{A}$'s advantage is typically defined as $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[\beta = \beta'] - \frac{1}{2}|$.

(Co-)Selective Security. We also consider the notion of selective security [13, 5], where $\mathcal{A}$ has to choose the challenge attribute $\boldsymbol{y}^\star$ before the setup phase, but can adaptively choose the key queries for $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q$. One can consider its "dual" notion where $\mathcal{A}$ must output the q key queries for attribute vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q$ before the setup phase, but can adaptively choose the target challenge attribute $\boldsymbol{y}^\star$. We refer to this scenario as the *co-selective* security model, which is useful in some applications such as revocation. By definition, both notions are incomparable in general and we do not know about their relation yet.

We shall show how one FE primitive can be obtained from another. The following useful lemma from [11] describes a sufficient criterion for implication.

Proposition 1 (Embedding Lemma [11]). *Consider encryption primitives A, B that can be casted as functional encryption for relations $R^{\mathsf{A}}, R^{\mathsf{B}}$, respectively. Suppose there exists efficient injective mappings $f_{\mathsf{k}} : \Sigma_{\mathsf{k}}^{\mathsf{A}} \to \Sigma_{\mathsf{k}}^{\mathsf{B}}$ and $f_{\mathsf{e}} : \Sigma_{\mathsf{e}}^{\mathsf{A}} \to \Sigma_{\mathsf{e}}^{\mathsf{B}}$ such that*

$$R^{\mathsf{B}}(f_{\mathsf{k}}(\boldsymbol{x}), f_{\mathsf{e}}(\boldsymbol{y})) = 1 \Leftrightarrow R^{\mathsf{A}}(\boldsymbol{x}, \boldsymbol{y}) = 1.$$

Let Π_{B} be a construction for primitive B. We construct Π_{A} for primitive A from Π_{B} by applying mappings $f_{\mathsf{k}}, f_{\mathsf{e}}$ to all key attributes and ciphertext attributes, respectively. More precisely, we use the same setup algorithm. As for the key generation and encryption procedures, they can be defined as $\Pi_{\mathsf{A}}.\mathsf{KeyGen}(x, \mathsf{msk}) := \Pi_{\mathsf{B}}.\mathsf{KeyGen}(f_{\mathsf{k}}(x), \mathsf{msk})$ and $\Pi_{\mathsf{A}}.\mathsf{Encrypt}(y, M, \mathsf{pk}) := \Pi_{\mathsf{B}}.\mathsf{Encrypt}(f_{\mathsf{e}}(y), M, \mathsf{pk})$, respectively. Then, if Π_{B} is secure, so is Π_{A}. This holds for adaptive, selective, co-selective security models. We denote this primitive implication by $\mathsf{B} \xrightarrow{f_{\mathsf{k}}, f_{\mathsf{e}}} \mathsf{A}$.

We immediately obtain the next corollary stating that the implication applies to the negated (resp. dual) variant with the same (resp. swapped) mappings.

Corollary 1. $\mathsf{B} \xrightarrow{f_{\mathsf{k}},f_{\mathsf{e}}} \mathsf{A}$ *implies* $\mathsf{Dual(B)} \xrightarrow{f_{\mathsf{e}},f_{\mathsf{k}}} \mathsf{Dual(A)}$ *and* $\mathsf{Neg(B)} \xrightarrow{f_{\mathsf{k}},f_{\mathsf{e}}} \mathsf{Neg(A)}$.

2.2 Complexity Assumptions in Bilinear Groups

We consider groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with an efficiently computable map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$ and $a, b \in \mathbb{Z}$ and $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$. In these groups, we assume the hardness of the (now classical) Decision Bilinear Diffie-Hellman and Decision Linear [6] problems.

Definition 1. *The* **Decision Bilinear Diffie-Hellman Problem** *(DBDH) in bilinear groups* $(\mathbb{G}, \mathbb{G}_T)$ *is, given elements* $(g, g^{\theta_1}, g^{\theta_2}, g^{\theta_3}, \eta) \in \mathbb{G}^4 \times \mathbb{G}_T$ *with* $\theta_1, \theta_2, \theta_3 \xleftarrow{\$} \mathbb{Z}_p$*, to decide whether* $\eta = e(g, g)^{\theta_1\theta_2\theta_3}$ *or* $\eta \in_R \mathbb{G}_T$.

Definition 2. *The* **Decision Linear Problem** *(DLIN) in* $\mathbb{G}$ *consists in, given a tuple of elements* $(g, f, \nu, g^{\theta_1}, f^{\theta_2}, \eta) \in \mathbb{G}^6$ *with* $\theta_1, \theta_2 \xleftarrow{\$} \mathbb{Z}_p$ *and* $f, g, \nu \xleftarrow{\$} \mathbb{G}$*, deciding whether* $\eta = \nu^{\theta_1+\theta_2}$ *or* $\nu \in_R \mathbb{G}$.

2.3 Some Notations

Throughout the paper, we will treat a vector as a column vector, unless specified otherwise. We use the same notations as in [11]. More precisely, for any vector of scalars $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{Z}_p^n$, the notation $g^{\vec{\alpha}}$ stands for the vector of group elements $(g^{\alpha_1}, \ldots, g^{\alpha_n})^\top \in \mathbb{G}^n$. For $\vec{a}, \vec{z} \in \mathbb{Z}_p^n$, we denote their inner product as $\langle \vec{a}, \vec{z} \rangle = \vec{a}^\top \vec{z} = a_1 z_1 + \cdots + a_n z_n$. Given $\vec{A} = g^{\vec{a}}$ and $\vec{z}$, one can easily compute $\vec{A}^{\vec{z}} = (g^{\vec{a}})^{\vec{z}} := g^{\langle \vec{a}, \vec{z} \rangle}$, without knowing $\vec{a}$. We also denote the element-vise product as $\vec{a}\vec{z} = (a_1 z_1, \ldots, a_n z_n)$. We denote by I_n the identity matrix of size n.

3 Functional Encryption Instances and Their Implications

3.1 Public-Attribute Inner Product Encryption and Its Consequences

We underline the power of PAIPE by showing its implications in this section. Each primitive is defined by describing the corresponding boolean function R. We then show how to construct one primitive from another by explicitly describing attribute mappings. In this way, correctness and security are consequences of the embedding lemma. Basically, the approach follows exactly the same way as [20]. A new contribution is that we also consider the negated variant of primitives, which will be useful for non-zero polynomial evaluation and revocation schemes. The implication for negated variants follows from Corollary 1.

Inner Product. A public-attribute inner product encryption (PAIPE) scheme over $\mathbb{Z}_p^n$, for some prime p, is defined as follows. Attribute domains are defined as $\Sigma_{\mathsf{k}}^{\mathsf{PAIPE}_n} = \Sigma_{\mathsf{e}}^{\mathsf{PAIPE}_n} = \mathbb{Z}_p^n$. We consider two distinct PAIPE modes here. The first one is the zero-mode PAIPE where $R^{\mathsf{ZIPE}_n}(\vec{X}, \vec{Y}) = 1$ iff $\vec{X} \cdot \vec{Y} = 0$. The other one is its negated analogue, which we call the non-zero-mode PAIPE, where the relation R^{NIPE_n} is defined in such a way that $R^{\mathsf{NIPE}_n}(\vec{X}, \vec{Y}) = 1$ iff $\vec{X} \cdot \vec{Y} \neq 0$.

Polynomial Evaluation. Functional encryption for the zero evaluation of polynomials of degree $\leq n$ is defined as follows. The ciphertext and key attribute domains are defined as $\Sigma_{\mathsf{e}}^{\mathsf{ZPoly}_{\leq n}} = \mathbb{Z}_p$ and $\Sigma_{\mathsf{k}}^{\mathsf{ZPoly}_{\leq n}} = \{P \in \mathbb{Z}_p[x] \mid \deg(P) \leq n\}$, respectively. The relation is defined by $R^{\mathsf{ZPoly}_{\leq n}}(P, x) = 1$ iff $P(x) = 0$. The non-zero evaluation mode can be defined as its negated primitive $\mathsf{Neg}(\mathsf{ZPoly}_{\leq n})$.

Given a PAIPE scheme over $\mathbb{Z}_p^{n+1}$, one obtain a functional encryption system for polynomial evaluation via the following embedding. For the key attribute, we simply map the polynomial $P[X] = \rho_0 + \rho_1 X + \cdots + \rho_n X^n$ to the vector $\vec{X}_p = (\rho_0, \ldots, \rho_n)$. Regarding ciphertext attributes, each element $w \in \mathbb{Z}_p$ is then mapped onto a vector $\vec{Y}_w = (1, w, w^2, \ldots, w^n)$. Correctness and security hold since $P(w) = 0$ whenever $\vec{X}_p \cdot \vec{Y}_w = 0$. The non-zero evaluation case can be analogously derived from the non-zero-mode PAIPE using the same mappings, due to Corollary 1.

We can also consider other variants such as a scheme that supports multivariate polynomials and a dual variant, where the key attribute corresponds to a fixed point and the ciphertext attribute corresponds to a polynomial, as in [20].

OR, AND, DNF, CNF Formulae. We now consider a FE scheme for some boolean formulae that evaluate disjunctions, conjunctions, and their extensions to disjunctive or conjunctive normal forms. As an example, a functional encryption scheme for boolean formula $R^{\mathsf{OR}_{\leq n}} : \mathbb{Z}_N^{\leq n} \times \mathbb{Z}_N \to \{0,1\}$ can be defined by $R^{\mathsf{OR}_{\leq n}}((I_1, \ldots, I_k), z) \mapsto 1$ (for $k \leq n$) iff $(z = I_1)$ or $\cdots$ or $(z = I_k)$. This can be obtained from a functional encryption for the zero evaluation of a univariate polynomial of degree smaller than n by generating a private key for $f_{\mathsf{OR},I_1,\ldots,I_k}(z) = (z - I_1) \cdots (z - I_k)$, and letting senders encrypting to z.

Other fundamental cases can be considered similarly as in [20] and are shown below. In [20] only non-negated policies (the first three cases below and their extensions) were considered. Schemes supporting negated policies (the latter three cases below and their extensions) are introduced in this paper. The negated case can be implemented by PAIPE for non-zero evaluation. One can combine these cases to obtain DNF, CNF formulae. Below, $r \xleftarrow{\$} \mathbb{Z}_p$ is chosen by KeyGen.[4]

Policy	Implementation
$(z = I_1)$ or $(z = I_2)$	$f_{\mathsf{OR},I_1,I_2}(z) = (z - I_1)(z - I_2) = 0$
$(z_1 = I_1)$ or $(z_2 = I_2)$	$f_{\overline{\mathsf{OR}},I_1,I_2}(z_1, z_2) = (z_1 - I_1)(z_2 - I_2) = 0$
$(z_1 = I_1)$ and $(z_2 = I_2)$	$f_{\mathsf{AND},I_1,I_2}(z_1, z_2) = (z_1 - I_1)r + (z_2 - I_2) = 0$
$(z_1 \neq I_1)$ or $(z_2 \neq I_2)$	$f_{\mathsf{NOR},I_1,I_2}(z_1, z_2) = (z_1 - I_1)r + (z_2 - I_2) \neq 0$
$(z \neq I_1)$ and $(z \neq I_2)$	$f_{\mathsf{NAND},I_1,I_2}(z) = (z - I_1)(z - I_2) \neq 0$
$(z_1 \neq I_1)$ and $(z_2 \neq I_2)$	$f_{\overline{\mathsf{NAND}},I_1,I_2}(z_1, z_2) = (z_1 - I_1)(z_2 - I_2) \neq 0$

ID-based Broadcast Encryption and Revocation. Let $\mathcal{I}$ be an identity space. An ID-based broadcast encryption scheme (IBBE) for maximum n receivers per ciphertext is a functional encryption for $R^{\mathsf{IBBE}_{\leq n}} : \mathcal{I} \times 2^{\mathcal{I}} \to \{0,1\}$ defined by $R^{\mathsf{IBBE}_{\leq n}} : (\mathsf{ID}, S) \mapsto 1$ iff $\mathsf{ID} \in S$. An IBBE system can be constructed by a functional encryption for $R^{\mathsf{Dual}(\mathsf{OR}_{\leq n})}$. To encrypt a message for the receiver set $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_k\}$, one encrypts using the policy $(z = \mathsf{ID}_1)$ or $\cdots$ or $(z = \mathsf{ID}_k)$.

Likewise, identity-based revocation (IBR) [22] for at most n revocations per ciphertext can be casted as a negated IBBE, *i.e.,* $R^{\mathsf{IBR}_{\leq n}} : (\mathsf{ID}, R) \mapsto 1$ iff $\mathsf{ID} \notin R$.

3.2 Spatial Encryption

We now recall the concept of spatial encryption [11]. For a $n \times d$ matrix M of which elements are in $\mathbb{Z}_p$ and a vector $\vec{c} \in \mathbb{Z}_p^n$, we define its corresponding affine space as $\mathrm{Aff}(M, \vec{c}) = \{M\vec{w} + \vec{c} \mid \vec{w} \in \mathbb{Z}_p^d\}$. Let $\mathcal{V}_n \subseteq 2^{(\mathbb{Z}_p^n)}$ be the collection of all affine spaces inside $\mathbb{Z}_p^n$. That is, $\mathcal{V}_n$ is defined as

$$\mathcal{V}_n = \{\mathrm{Aff}(M, \vec{c}) \mid M \in \mathbb{M}_{n \times d}, c \in \mathbb{Z}_p^n, d \leq n\},$$

[4] As noted in [20], the AND (and NOR) case will not work in the adaptive security model since the information on r leaks.

where $\mathbb{M}_{n\times d}$ is the set of all $n \times d$ matrices in $\mathbb{Z}_p$.

A spatial encryption system in $\mathbb{Z}_p^n$ is a functional encryption scheme for a relation $R^{\mathsf{Spatial}} : \mathcal{V}_n \times \mathbb{Z}_p^n \to \{0,1\}$ defined by $R^{\mathsf{Spatial}} : (V, \vec{y}) \mapsto 1$ iff $\vec{y} \in V$.

The notion of spatial encryption was motivated by Boneh and Hamburg [11]. It has many applications as it notably implies broadcast HIBE and multi-authority schemes. Nevertheless, its connection to inner-product encryption has not been investigated so far. In section 4.1, we prove that spatial encryption implies inner product encryption by providing a simple attribute mapping.

As a result of independent interest, we also consider the negated spatial encryption primitive (namely, FE that is defined by $R^{\mathsf{Neg}(\mathsf{Spatial})} : (V, \vec{y}) \mapsto 1$ iff $\vec{y} \notin V$) and provide a construction in section 5.1. From this scheme and Corollary 1 together with our implication result of zero-mode PAIPE from spatial encryption, we then obtain a non-zero-mode PAIPE construction.

4 Functional Encryption for Zero-Mode PAIPE

4.1 Warm-up: Selectively Secure Zero-Mode PAIPE from Spatial Encryption

We first observe that spatial encryption implies zero-mode public-attribute inner product encryption. Indeed, for the key attribute, we map a vector $\vec{X} = (x_1, \ldots, x_n)^\top \in \mathbb{Z}_p^n$ onto a $(n-1)$-dimension affine space $V_{\vec{X}} = \mathrm{Aff}(M_{\vec{X}}, \vec{0}_n) = \{M_{\vec{X}}\vec{w} + \vec{0}_n \mid \vec{w} \in \mathbb{Z}_p^{n-1}\}$ with the matrix $M_{\vec{X}} \in \mathbb{Z}_p^{n\times(n-1)}$

$$M_{\vec{X}} = \begin{pmatrix} -\frac{x_2}{x_1}, -\frac{x_3}{x_1}, \cdots, -\frac{x_n}{x_1} \\ I_{n-1} \end{pmatrix}. \tag{3}$$

For any vector $\vec{Y} = (y_1, \ldots, y_n)^\top \in \mathbb{Z}_p^n$, we then have $\vec{X} \cdot \vec{Y} = 0 \Leftrightarrow \vec{Y} \in V_{\vec{X}}$ since $\vec{X} \cdot \vec{Y} = 0 \Leftrightarrow y_1 = y_2 \cdot (-\frac{x_2}{x_1}) + \cdots + y_n \cdot (-\frac{x_n}{x_1}) \Leftrightarrow \vec{Y} = M_{\vec{X}} \cdot (y_2, \ldots, y_n)^\top \Leftrightarrow \vec{Y} \in V_{\vec{X}}$. By the embedding lemma, we can therefore conclude its implication.

In [11], Boneh and Hamburg described a selectively secure construction of spatial encryption that achieves constant-size ciphertexts (by generalizing the Boneh-Boyen-Goh HIBE [7]). From their scheme, we thus immediately obtain a selectively secure zero PAIPE scheme with constant-size ciphertext as shown below.

We first give some notations here. For a vector $\vec{a} = (a_1, \ldots, a_n)^\top \in \mathbb{Z}_p^n$, we write $g^{\vec{a}}$ to denote $(g^{a_1}, \ldots, g^{a_n})^\top$. Given $g^{\vec{a}}, \vec{z}$, one can easily compute $(g^{\vec{a}})^{\vec{z}} := g^{\langle \vec{a}, \vec{z} \rangle}$, where $\langle \vec{a}, \vec{z} \rangle$ denotes the inner product $\vec{a} \cdot \vec{z} = \vec{a}^\top \vec{z}$.

Construction 2. (Selectively secure zero-mode PAIPE)

▶ $\mathsf{Setup}(1^\lambda, n)$: chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with a generator $g \xleftarrow{\$} \mathbb{G}$. It chooses $\alpha, \alpha_0, \ldots, \alpha_n \xleftarrow{\$} \mathbb{Z}_p$ at random and defines $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$. The public key is defined to be $\mathsf{pk} = \left(g, g^{\alpha_0}, \vec{H} = g^{\vec{\alpha}}, Z = e(g,g)^\alpha\right)$ and the master secret key is $\mathsf{msk} = g^\alpha$.

▶ $\mathsf{KeyGen}(\vec{X}, \mathsf{msk}, \mathsf{pk})$: chooses $t \xleftarrow{\$} \mathbb{Z}_p$ and parses $\vec{X}$ as $(x_1, \ldots, x_n)$ and returns $\perp$ if $x_1 = 0$. It outputs the private key as $\mathsf{sk}_{\vec{X}} = (D_0, D_1, K_2, \ldots, K_n)$ where

$$D_0 = g^t, \qquad D_1 = g^{\alpha+\alpha_0 t}, \qquad \{K_i = (g^{-\alpha_1 \frac{x_i}{x_1}} g^{\alpha_i})^t\}_{i=2,\ldots,n}.$$

▶ $\mathsf{Encrypt}(\vec{Y}, \mathsf{pk})$: the encryption algorithm first picks $s \xleftarrow{\$} \mathbb{Z}_p$. It parses $\vec{Y}$ as $(y_1, \ldots, y_n)$ and computes the ciphertext as

$$E_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, \qquad E_1 = (g^{\alpha_0} g^{\langle \vec{\alpha}, \vec{Y} \rangle})^s, \qquad E_2 = g^s.$$

▶ $\mathsf{Decrypt}(C, \vec{Y}, \mathsf{sk}_{\vec{X}}, \vec{X}, \mathsf{pk})$: to decrypt, the algorithm computes the blinding factor $e(g,g)^{\alpha \cdot s}$ as $\frac{e(D_1 K_2^{y_2} \cdots K_n^{y_n}, E_2)}{e(E_1, D_0)} = e(g,g)^{\alpha \cdot s}$.

The selective security of this scheme is a consequence of a result given in [11].

Theorem 1. *Construction 2 is selectively secure under the n-Decisional Bilinear Diffie-Hellman Exponent assumption* (see [10, 11] for a description of the latter).

4.2 Adaptively Secure Zero-Mode PAIPE under Simple Assumptions

We extend the above selectively secure zero-mode PAIPE to acquire adaptive security by applying the Waters' dual system method [31]. However, we have to use our "n-equation technique" as opposed to 2-equation technique used for IBE in [31]. The reason is that we have to deal with the difficulties arising from the richer structure of PAIPE and the aggregation of ciphertexts into a constant number of elements, analogously to what we described in section 1.

The scheme basically goes as follows. A ciphertext contains a random tag tagc in the element E_1 while each private key contains $n-1$ tags (tagk_i for each K_i element) that are aggregated into $\mathsf{tagk} = \sum_{i=2}^{n} \mathsf{tagk}_i y_i$ upon decryption of a ciphertext intended for $\vec{Y}$. The receiver is able to decrypt whenever $\mathsf{tagk} \neq \mathsf{tagc}$ (and $\vec{X} \cdot \vec{Y} = 0$), which occurs with overwhelming probability.

Construction 3. (Adaptively secure zero-mode PAIPE)

▶ $\mathsf{Setup}(1^\lambda, n)$: chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. It then picks generators $g, v, v_1, v_2 \xleftarrow{\$} \mathbb{G}$ and chooses $\alpha, \alpha_0, \alpha_1, \ldots, \alpha_n, a_1, a_2, b \xleftarrow{\$} \mathbb{Z}_p$. Let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and $\vec{H} = (h_1, \ldots, h_n) = g^{\vec{\alpha}}$. The public key consists of

$$\mathsf{pk} = \left(\begin{array}{c} g,\ w = g^{\alpha_0},\ Z = e(g,g)^{\alpha \cdot a_1 \cdot b},\ \vec{H} = g^{\vec{\alpha}},\ A_1 = g^{a_1},\ A_2 = g^{a_2}, B = g^b, \\ B_1 = g^{b \cdot a_1},\ B_2 = g^{b \cdot a_2},\ \tau_1 = v \cdot v_1^{a_1},\ \tau_2 = v \cdot v_2^{a_2},\ T_1 = \tau_1^b,\ T_2 = \tau_2^b \end{array} \right)$$

The master key is defined to be $\mathsf{msk} = (g^\alpha, g^{\alpha a_1}, v, v_1, v_2)$.

▶ $\mathsf{Keygen}(\vec{X}, \mathsf{msk}, \mathsf{pk})$: parses $\vec{X}$ as $(x_1, \ldots, x_n)$ and returns $\perp$ if $x_1 = 0$. Otherwise, it randomly picks $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$, $z_1, z_2 \xleftarrow{\$} \mathbb{Z}_p$, $\mathsf{tagk}_2, \ldots, \mathsf{tagk}_n \xleftarrow{\$} \mathbb{Z}_p$, sets $r = r_1 + r_2$ and generates the private key $\mathsf{sk}_{\vec{X}} = (D_1, \ldots, D_7, K_2, \ldots, K_n, \mathsf{tagk}_2, \ldots, \mathsf{tagk}_n)$ by computing

$$\mathsf{sk}_{\mathsf{core}} = \left\{ K_i = \left(g^{-\alpha_1 \frac{x_i}{x_1}} \cdot g^{\alpha_i} \cdot g^{\alpha_0 \cdot \mathsf{tagk}_i} \right)^{r_1} \right\}_{i=2,\ldots,n},$$

$$\mathsf{sk}_{\mathsf{adapt}} = \begin{pmatrix} D_1 = g^{\alpha a_1} \cdot v^r, & D_2 = g^{-\alpha} \cdot v_1^r \cdot g^{z_1}, & D_3 = B^{-z_1}, & D_4 = v_2^r \cdot g^{z_2}, \\ D_5 = B^{-z_2}, & D_6 = B^{r_2}, & D_7 = g^{r_1} & \end{pmatrix}.$$

▶ $\mathsf{Encrypt}(\vec{Y}, \mathsf{M}, \mathsf{pk})$: to encrypt $\mathsf{M} \in \mathbb{G}_T$ under $\vec{Y} = (y_1, \ldots, y_n) \in (\mathbb{Z}_p)^n$, pick $s_1, s_2, t, \mathsf{tagc} \xleftarrow{\$} \mathbb{Z}_p$ and compute $C = (C_1, \ldots, C_7, E_0, E_1, E_2, \mathsf{tagc})$ where

$$C_{\mathsf{core}} = \left(E_0 = \mathsf{M} \cdot Z^{s_2}, \quad E_1 = (g^{\alpha_0 \cdot \mathsf{tagc}} \cdot g^{\langle \vec{\alpha}, \vec{Y} \rangle})^t, \quad E_2 = g^t \right),$$

$$C_{\mathsf{adapt}} = \begin{pmatrix} C_1 = B^{s_1 + s_2}, & C_2 = B_1^{s_1}, & C_3 = A_1^{s_1}, & C_4 = B_2^{s_2}, \\ C_5 = A_2^{s_2}, & C_6 = \tau_1^{s_1} \cdot \tau_2^{s_2}, & C_7 = T_1^{s_1} \cdot T_2^{s_2} \cdot w^{-t} & \end{pmatrix}.$$

▶ $\mathsf{Decrypt}(C, \vec{Y}, \mathsf{sk}_{\vec{X}}, \vec{X}, \mathsf{pk})$: computes $\mathsf{tagk} = \mathsf{tagk}_2 y_2 + \cdots + \mathsf{tagk}_n y_n$ and then

$$W_1 = \prod_{j=1}^{5} e(C_j, D_j) \cdot (\prod_{j=6}^{7} e(C_j, D_j))^{-1} = e(g,g)^{\alpha \cdot a_1 \cdot b \cdot s_2} \cdot e(g,w)^{r_1 t},$$

as well as $W_2 = \left(\frac{e(K_2^{y_2} \cdots K_n^{y_n}, E_2)}{e(E_1, D_7)}\right)^{\frac{1}{\mathsf{tagk}-\mathsf{tagc}}} = e(g,w)^{r_1 t}$. It finally recovers the plaintext as

$$\mathsf{M} = E_0 / Z^{s_2} = E_0 / e(g,g)^{\alpha \cdot a_1 \cdot b \cdot s_2} \leftarrow E_0 \cdot W_2 \cdot W_1^{-1}.$$

The correctness of W_2 at decryption is shown in appendix A.1, while the rest follows from [31]. As we can see, ciphertexts have the same size as in the IBE scheme of [31], no matter how large the vector $\vec{Y}$ is. Also, decryption entails a constant number of pairing evaluations (whereas ciphertexts cost $O(n)$ pairings to decrypt in [20]).

Theorem 2. *Construction 3 is adaptively secure under the DLIN and DBDH assumptions.*

Proof. The proof uses the dual system methodology similar to [31], which involves ciphertexts and private keys that can be normal or semi-functional.

- Semi-functional ciphertexts are generated as in [31] by first computing a normal ciphertext $(C_1', \ldots, C_7', E_0', E_1', E_7', \mathsf{tagc}')$ and then choosing $\chi \xleftarrow{\$} \mathbb{Z}_p$ before replacing (C_4', C_5', C_6', C_7'), respectively, by

$$C_4 = C_4' \cdot g^{b a_2 \chi}, \qquad C_5 = C_5' \cdot g^{a_2 \chi}, \qquad C_6 = C_6' \cdot v_2^{a_2 \chi}, \qquad C_7 = C_7' \cdot v_2^{a_2 b \chi}. \tag{4}$$

- From a normal key $(D_1', \ldots, D_7', K_2', \ldots, K_n', \mathsf{tagk}_2', \ldots, \mathsf{tagk}_n')$, semi-functional keys are obtained by choosing $\gamma \xleftarrow{\$} \mathbb{Z}_p$ and replacing (D_1', D_2', D_4') by

$$D_1 = D_1' \cdot g^{-a_1 a_2 \gamma}, \qquad D_2 = D_2' \cdot g^{a_2 \gamma}, \qquad D_4 = D_4' \cdot g^{a_1 \gamma}. \tag{5}$$

The proof proceeds with a game sequence starting from Game_{Real}, which is the actual attack game. The following games are defined below.

Game_0 is the real attack game but the challenge ciphertext is semi-functional.

Game_k (for $1 \leq k \leq q$) is identical to Game_0 except that the first i private key generation queries are answered by returning a semi-functional key.

Game_{q+1} is as Game q but the challenge ciphertext is a semi-functional encryption of a random element of $\mathbb{G}_T$ instead of the actual plaintext.

We prove the indistinguishability between two consecutive games under some assumptions. The sequence ends in Game_{q+1}, where the challenge ciphertext is independent of the challenger's bit β, hence any adversary has no advantage.

The indistinguishability of Game_{Real} and Game_0 as well as that of Game_q and Game_{q+1} can be proved exactly in the same way as in [31] and the details are given in appendix C for completeness.

Lemma 1. *If DLIN is hard,* Game_0 *is indistinguishable from* Game_{Real}.

Lemma 2. *For any* $1 \leq k \leq q$, *if an adversary* $\mathcal{A}$ *can distinguish* Game_k *from* Game_{k-1}, *we can build a distinguisher for the DLIN problem.*

This lemma is the most non-trivial part in the theorem. The main issue is that, in order to enable adaptive security, the reduction must be done in such a way that the simulator $\mathcal{B}$ can create semi-functional keys for any vector $\vec{X}$, including those for which $\vec{X} \cdot \vec{Y}^{\star} = 0$. However, the crucial point is that we must prevent $\mathcal{B}$ from directly deciding whether the k^{th} queried private key is normal or semi-functional by generating a semi-functional ciphertext for itself. Indeed, if this were possible, the reduction from $\mathcal{A}$ would not be established.

To resolve this, we use a secret exponent vector $\vec{\zeta} \in \mathbb{Z}_p^n$ and embed the DLIN instance in such a way that $\mathcal{B}$ can only answer the k^{th} private key query for $\vec{X}$ using a vector of tags $(\mathsf{tagk}_2, \ldots, \mathsf{tagk}_n)$ and the challenge ciphertext for $\vec{Y}^{\star}$ using a tag $\mathsf{tagc}^{\star}$ that satisfy the relation $(\mathsf{tagk}_2, \ldots, \mathsf{tagk}_n, \mathsf{tagc}^{\star})^{\top} = -M_{\vec{X},\vec{Y}^{\star}}\vec{\zeta}$, where $M_{\vec{X},\vec{Y}}$ is the $n \times n$ matrix defined in Eq.(2). We thereby conceptually use the n-equation technique here. A particular consequence is that, if we have $\vec{X} \cdot \vec{Y}^{\star} = 0$, then it holds that

$$\mathsf{tagk} = \sum_{i=2}^{n} \mathsf{tagk}_i y_i^{\star} = \zeta_1 \sum_{i=2}^{n} \frac{x_i}{x_1} y_i^{\star} - \sum_{i=2}^{n} \zeta_i y_i^{\star} = \zeta_1 \cdot (-y_1^{\star}) - \sum_{i=2}^{n} \zeta_i y_i^{\star} = \mathsf{tagc}^{\star},$$

which is the exact condition under which decryption is hampered. In this situation, $\mathcal{B}$ cannot distinguish the k^{th} private key by itself, as desired. We are now ready to describe the proof of Lemma 2.

Proof. The distinguisher $\mathcal{B}$ receives $(g, f, \nu, g^{\theta_1}, f^{\theta_2}, \eta)$ and decides if $\eta = \nu^{\theta_1+\theta_2}$.

Setup. Algorithm $\mathcal{B}$ picks $\alpha, a_1, a_2, \delta_{v_1}, \delta_{v_2} \xleftarrow{\$} \mathbb{Z}_p$ and sets $g = g$, $Z = e(f, g)^{\alpha a_1}$,

$$\begin{aligned}
A_1 &= g^{a_1}, & A_2 &= g^{a_2}, & B &= g^b = f, & v_1 &= \nu^{a_2} \cdot g^{\delta_{v_1}} \\
B_1 &= g^{ba_1} = f^{a_1}, & B_2 &= g^{ba_2} = f^{a_2}, & v &= \nu^{-a_1 a_2}, & v_2 &= \nu^{a_1} \cdot g^{\delta_{v_2}}, \\
\tau_1 &= v v_1^{a_1} = g^{\delta_{v_1} a_1}, & \tau_2 &= v v_2^{a_2} = g^{\delta_{v_2} a_2}, & \tau_1^b &= f^{\delta_{v_1} a_1}, & \tau_2^b &= f^{\delta_{v_2} a_2}.
\end{aligned}$$

Next, algorithm $\mathcal{B}$ chooses $\delta_w \xleftarrow{\$} \mathbb{Z}_p, \vec{\zeta} = (\zeta_1, \ldots, \zeta_n) \xleftarrow{\$} \mathbb{Z}_p^n$, $\vec{\delta} = (\delta_1, \ldots, \delta_n) \xleftarrow{\$} \mathbb{Z}_p^n$, then defines $w = g^{\alpha_0} = f \cdot g^{\delta_w}$, and $h_i = g^{\alpha_i} = f^{\zeta_i} \cdot g^{\delta_i}$ for $i = 1, \ldots, n$. Note that, as in the proof of lemma 2 in [31] , $\mathcal{B}$ knows $\mathsf{msk} = (g^{\alpha}, g^{\alpha a_1}, v, v_1, v_2)$.

Key Queries. When $\mathcal{A}$ makes the j^{th} private key query, $\mathcal{B}$ does as follows.

[**Case** $j > k$] It generates a normal key, using the master secret key msk.

[**Case** $j < k$] It creates a semi-functional key, which it can do using $g^{a_1 a_2}$.

[**Case** $j = k$] It defines $\mathsf{tagk}_2, \ldots, \mathsf{tagk}_n$ as $\mathsf{tagk}_i = \zeta_1 \cdot \frac{x_i}{x_1} - \zeta_i$ for $i = 2, \ldots, n$, which implies that $(h_1^{-x_i/x_1} \cdot h_i \cdot w^{\mathsf{tagk}_i}) = g^{-\delta_1(x_i/x_1)+\delta_i+\delta_w \mathsf{tagk}_i}$, for $i = 2, \ldots, n$. Using these tags, it generates a normal private key $(D_1', \ldots, D_7', K_2', \ldots, K_n')$ using random exponents $r_1', r_2', z_1', z_2' \xleftarrow{\$} \mathbb{Z}_p$. Then, it sets

$$\begin{aligned}
D_1 &= D_1' \cdot \eta^{-a_1 a_2}, & D_2 &= D_2' \cdot \eta^{a_2} \cdot (g^{\theta_1})^{\delta_{v_1}}, & D_3 &= D_3' \cdot (f^{\theta_2})^{\delta_{v_1}}, \\
D_4 &= D_4' \cdot \eta^{a_1} \cdot (g^{\theta_1})^{\delta_{v_2}}, & D_5 &= D_5' \cdot (f^{\theta_2})^{\delta_{v_2}}, & D_6 &= D_6' \cdot f^{\theta_2},
\end{aligned}$$

as well as $D_7 = D_7' \cdot (g^{\theta_1})$ and $K_i = K_i' \cdot (g^{\theta_1})^{-\delta_1(x_i/x_1)+\delta_i+\delta_w \mathsf{tagk}_i}$ for $i = 2, \ldots, n$.

If $\eta = \nu^{\theta_1+\theta_2}$, $\mathsf{sk}_{\vec{X}} = (D_1, \ldots, D_7, K_2, \ldots, K_n, \mathsf{tagk}_2, \ldots, \mathsf{tagk}_n)$ is easily seen to form a normal key where $r_1 = r_1' + \theta_1$, $r_2 = r_2' + \theta_2$, $z_1 = z_1' - \delta_{v_1}\theta_2$, $z_2 = z_2' - \delta_{v_2}\theta_2$ are the underlying random exponents. If $\eta \in_R \mathbb{G}$, it can be written $\eta = \nu^{\theta_1+\theta_2} \cdot g^{\gamma}$ for some $\gamma \in_R \mathbb{Z}_p$, so that $\mathsf{sk}_{\vec{X}}$ is distributed

as a semi-functional key. We note that $\mathsf{tagk}_2, \ldots, \mathsf{tagk}_n$ are independent and uniformly distributed since $\zeta_1, \ldots, \zeta_n$ (which are the solutions of a system of $n-1$ equations with n unknowns) are uniformly random and perfectly hidden from $\mathcal{A}$'s view.

Challenge. Once the first phase is over, $\mathcal{A}$ outputs $M_0, M_1 \in \mathbb{G}_T$ along with a target vector $\vec{Y}^\star = (y_1^\star, \ldots, y_n^\star)$. Then, $\mathcal{B}$ flips a coin $\beta \xleftarrow{\$} \{0,1\}$ and computes the tag $\mathsf{tagc}^\star = -\langle \vec{Y}^\star, \vec{\zeta} \rangle$ for which $\mathcal{B}$ will be able to prepare the semi-functional ciphertext. To this end, $\mathcal{B}$ first computes a normal encryption $(C_1', \ldots, C_7', E_0', E_1', E_2', \mathsf{tagc}^\star)$ of M_β using exponents s_1', s_2', t'. It then chooses $\chi \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$C_4 = C_4' \cdot f^{a_2 \cdot \chi}, \qquad C_5 = C_5' \cdot g^{a_2 \cdot \chi}, \qquad C_7 = C_7' \cdot \nu^{-\delta_w \cdot a_1 \cdot a_2 \cdot \chi} \cdot f^{\delta_{v_2} \cdot a_2 \cdot \chi},$$
$$C_6 = C_6' \cdot v_2^{a_2 \cdot \chi}, \qquad E_2 = E_2' \cdot \nu^{a_1 \cdot a_2 \cdot \chi}, \qquad E_1 = E_1' \cdot (\nu^{\delta_w \cdot \mathsf{tagc}^\star + \langle \vec{Y}^\star, \vec{\delta} \rangle})^{a_1 \cdot a_2 \cdot \chi}.$$

We claim that $(C_1', C_2', C_3', C_4, C_5, C_6, C_7, E_0, E_1, E_2, \mathsf{tagc}^\star)$ is a semi-functional ciphertext with underlying exponents $\chi, s_1 = s_1', s_2 = s_2'$ and $t = t' + \log_g(\nu) a_1 a_2 \chi$. To prove this, we observe that

$$\begin{aligned} C_7 &= T_1^{s_1} \cdot T_2^{s_2} \cdot w^{-t} \cdot v_2^{a_2 b \chi} = T_1^{s_1} \cdot T_2^{s_2} \cdot w^{-t' - \log_g(\nu) a_1 a_2 \chi} \cdot (\nu^{a_1} \cdot g^{\delta_{v_2}})^{a_2 b \chi} \\ &= T_1^{s_1} \cdot T_2^{s_2} \cdot w^{-t'} \cdot (f \cdot g^{\delta_w})^{-\log_g(\nu) a_1 a_2 \chi} \cdot (\nu^{a_1} \cdot g^{\delta_{v_2}})^{a_2 b \chi} \\ &= C_7' \cdot \nu^{-\delta_w a_1 a_2 \chi} \cdot f^{\delta_{v_2} a_2 \chi}, \end{aligned}$$

where the unknown term in $v_2^{a_2 b \chi}$ is canceled out by w^{-t}. Also,

$$\begin{aligned} E_1 &= E_1' \cdot \left(h_1^{y_1^\star} \cdots h_n^{y_n^\star} \cdot w^{\mathsf{tagc}^\star} \right)^{\log_g(\nu) a_1 a_2 \chi} \\ &= E_1' \cdot \left((f^{\zeta_1} g^{\delta_1})^{y_1^\star} \cdots (f^{\zeta_n} g^{\delta_n})^{y_n^\star} \cdot (f g^{\delta_w})^{-\langle \vec{Y}^\star, \vec{\zeta} \rangle} \right)^{\log_g(\nu) a_1 a_2 \chi} \\ &= E_1' \cdot (\nu^{\langle \vec{Y}^\star, \vec{\delta} \rangle + \delta_w \cdot \mathsf{tagc}^\star})^{a_1 a_2 \chi}, \end{aligned}$$

where the unknown $f^{\log_g(\nu)}$ vanishes due to our definition of $\mathsf{tagc}^\star$. It then remains to show that $\mathsf{tagc}^\star, \mathsf{tagk}_2, \ldots, \mathsf{tagk}_n$ are still n-wise independent. But this holds since their relations form a system

$$M \cdot \vec{\zeta} := \begin{pmatrix} -\frac{x_2}{x_1} & 1 & & \\ -\frac{x_3}{x_1} & & 1 & \\ \vdots & & & \ddots \\ -\frac{x_n}{x_1} & & & 1 \\ y_1^\star & y_2^\star & y_3^\star & \ldots \; y_n^\star \end{pmatrix} \begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \\ \zeta_n \end{pmatrix} = - \begin{pmatrix} \mathsf{tagk}_2 \\ \mathsf{tagk}_3 \\ \vdots \\ \mathsf{tagk}_n \\ \mathsf{tagc}^\star \end{pmatrix},$$

which has a solution in $\vec{\zeta}$ whenever $\det(M) = (-1)^{n+1} \vec{X} \cdot \vec{Y}^\star / x_1 \neq 0$.

Eventually, $\mathcal{A}$ outputs a bit β' and $\mathcal{B}$ outputs 0 if $\beta = \beta'$. As in [31], we see that $\mathcal{A}$ is playing Game_{k-1} if $\eta = \nu^{\theta_1 + \theta_2}$ and Game_k otherwise.

Lemma 3. *If DBDH is hard,* Game_q *and* Game_{q+1} *are indistinguishable.*

4.3 Comparisons

In this section, we give a detailed comparison among the various IPE and PAIPE that can be found in the literature. This is shown in Table 1. Their efficiency is measured in terms of ciphertext and key sizes as well as the number of exponentiations and pairing evaluations to decrypt.

Table 1. Performances of IPE and PAIPE Schemes

IPE/PAIPE schemes	Ciphertext overhead	Private key size	Decryption cost	Attribute hiding?	Security
KSW [20], TO [30]	$O(n) \times \lvert\mathbb{G}\rvert$	$O(n) \times \lvert\mathbb{G}\rvert$	$O(n)$ p.	Yes	Selective
LOSTW [21]	$O(n) \times \lvert\mathbb{G}\rvert$	$O(n) \times \lvert\mathbb{G}\rvert$	$O(n)$ p.	Yes	Adaptive
This work	$O(1) \times \lvert\mathbb{G}\rvert$	$O(n) \times \lvert\mathbb{G}\rvert$	$O(1)$ p. + $O(n)$ exp.	No	Adaptive

n:dimension, p.:pairing applications, exp.:group exponential applications

Table 2 also summarizes the features of all identity-based broadcast encryption schemes that have been described in prime order groups. Comparisons are given in terms of performances and security guarantees. As for the latter criterion, three dimensions are considered depending on the strength of underlying assumptions (*i.e.*, simple assumptions vs q-type assumptions), on whether security holds in the standard model or the random oracle model, as well as in the adaptive or selective sense.

Table 2. Performances of IBBE systems

IBBE schemes	Ciphertext overhead	Private key size	Decryption cost	Security	RO?	Assumptions
AKN [2]	$O(1) \times \lvert\mathbb{G}\rvert$	$O(n^2) \times \lvert\mathbb{G}\rvert$	$O(1)$ p. + $O(n)$ exp.	Selective	No	q-type
Del. [15]	$O(1) \times \lvert\mathbb{G}\rvert$	$O(1) \times \lvert\mathbb{G}\rvert$	$O(1)$ p. + $O(n)$ exp.	Selective	No	q-type
BH [11]	$O(1) \times \lvert\mathbb{G}\rvert$	$O(n) \times \lvert\mathbb{G}\rvert$	$O(1)$ p. + $O(n)$ exp.	Selective	No	q-type
GW [17]	$O(1) \times \lvert\mathbb{G}\rvert$ + $O(1) \times \lvert\mathbb{G}_T\rvert$	$O(1) \times \lvert\mathbb{G}\rvert$ + $O(1) \times \lvert\mathbb{Z}_p\rvert$	$O(1)$ p. + $O(n)$ exp.	Adaptive	Yes	q-type
This work	$O(1) \times \lvert\mathbb{G}\rvert$	$O(n) \times \lvert\mathbb{G}\rvert$	$O(1)$ p. + $O(n)$ exp.	Adaptive	No	Simple

n:the number of users, p.:pairing applications, exp.:group exponential applications

Our system is asymptotically on par with the Boneh-Hamburg realization [11] in all efficiency metrics, with the advantage of providing full security under simple assumptions (which appears to be a unique property in existing IBBE schemes).

In appendix B, we describe a less efficient but conceptually simpler (notably in its similarity with construction 2) variant of our PAIPE scheme in groups of composite order. It is derived from an adaptively secure spatial encryption system obtained by applying the Lewko-Waters techniques [23] to the construction of Boneh and Hamburg.

5 Functional Encryption for Non-Zero-Mode PAIPE

5.1 Negated Spatial Encryption

We begin this section by providing a co-selectively-secure construction of negated spatial encryption, which is motivated by its implication of non-zero-mode PAIPE. At a high-level, our scheme can be viewed as a "negative" analogue of the Boneh-Hamburg spatial encryption [11], in very much the same way as the Lewko-Sahai-Waters revocation scheme [22] is a negative analogue of the Boneh-Boyen IBE [5]. The intuition follows exactly from section 1, where we have to use "n-equation technique". In spatial encryption, we have to deal with, in general, how we can set up a system of n equations similarly to Eq.(1). To this end, we confine the vector subspaces that we can use as follows. Our construction is a functional encryption for the relation $R^{\mathsf{Neg(Spatial)}} : \mathcal{W}_n \times \mathbb{Z}_p^n \to \{0,1\}$, for a collection $\mathcal{W}_n \subseteq \mathcal{V}_n$ of vector subspaces in $\mathbb{Z}_p^n$ defined as

$$\mathcal{W}_n = \{\mathrm{Aff}(M, \vec{0}) \in \mathcal{V}_n \mid \mathrm{rank}(M_{(-1)}) = n-1\},$$

where we denote $M_{(-1)}$ as the matrix obtained by deleting the first row $M_1 \in \mathbb{Z}_p^{1\times d}$ of M.

Construction 4. (Co-selectively secure negated spatial encryption)

▶ $\mathsf{Setup}(1^\lambda, n)$: chooses a bilinear group $\mathbb{G}$ of prime order $p > 2^\lambda$ with a random generator $g \xleftarrow{\$} \mathbb{G}$. It randomly chooses $\alpha, \alpha_1, \ldots, \alpha_n \xleftarrow{\$} \mathbb{Z}_p$. Let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$. The public key consists of $\mathsf{pk} = \big(g, g^{\vec{\alpha}}, g^{\alpha_1\vec{\alpha}}, e(g,g)^\alpha\big)$. The corresponding master secret key is $\mathsf{msk} = (\alpha, \vec{\alpha})$.

▶ $\mathsf{KeyGen}(V, \mathsf{msk}, \mathsf{pk})$: suppose that $V = \mathrm{Aff}(M, \vec{0})$, from a matrix $M \in (\mathbb{Z}_p)^{n\times d}$. The algorithm picks $t \xleftarrow{\$} \mathbb{Z}_p$ and outputs $\mathsf{sk}_V = (D_0, D_1, \vec{K}) \in \mathbb{G}^{d+2}$ where

$$D_0 = g^t, \qquad D_1 = g^{\alpha + t\alpha_1^2}, \qquad \vec{K} = g^{tM^\top\vec{\alpha}}.$$

▶ $\mathsf{Encrypt}(\vec{y}, \mathsf{M}, \mathsf{pk})$: picks $s \xleftarrow{\$} \mathbb{Z}_p$ and computes (C_0, C_1, C_2, C_3) as

$$C_0 = \mathsf{M}\cdot e(g,g)^{\alpha s}, \qquad C_1 = g^{s\alpha_1\langle\vec{y},\vec{\alpha}\rangle}, \qquad C_2 = g^s, \qquad C_3 = g^{\alpha_1 s}.$$

▶ $\mathsf{Decrypt}(C, \vec{y}, \mathsf{sk}_V, V, \mathsf{pk})$: the algorithm first obtains M from V. We also recall the notation of M_1, which is the vector of the first row of M. It first solves the system of equations in $\vec{w}$ from $M_{(-1)}\vec{w} = (y_2, \ldots, y_n)^\top$, which it can do since $V \in \mathcal{W}_n$. It computes the message blinding factor as

$$e(g,g)^{\alpha s} = e(D_1, C_2)\cdot\left(\frac{e(C_1, D_0)}{e(\vec{K}\vec{w}, C_3)}\right)^{\frac{1}{M_1\vec{w}-y_1}} = e(g^{\alpha+t\alpha_1^2}, g^s)\cdot\left(\frac{e(g^{s\alpha_1\langle\vec{y},\vec{\alpha}\rangle}, g^t)}{g^{t\vec{w}^\top M^\top\vec{\alpha}}, g^{\alpha_1 s})}\right)^{\frac{1}{M_1\vec{w}-y_1}}.$$

Computability. We claim that the decryption can be computed if $y \notin V$. Indeed, we prove that if $y \notin V$ then $M_1\vec{w} - y_1 \neq 0$ (and the above equation is well-defined). To prove the contrapositive, suppose that $M_1\vec{w} - y_1 = 0$. Then, we must have $\vec{y} \in V$ since $M\vec{w} = \begin{bmatrix} M_1 \\ M_{(-1)}\end{bmatrix}\vec{w} = \begin{bmatrix} M_1\vec{w} \\ M_{(-1)}\vec{w}\end{bmatrix} = \vec{y}$.

Correctness. We verify that decryption is correct as follows. First, we note that due to our definition of $\vec{w}$, we have $\langle M\vec{w} - \vec{y}, \vec{\alpha}\rangle = (M_1\vec{w} - y_1)\alpha_1$. Therefore, the correctness follows from the fact that

$$\left(\frac{e(g^{s\alpha_1\langle\vec{y},\vec{\alpha}\rangle}, g^t)}{e(g^{t\vec{w}^\top M^\top\vec{\alpha}}, g^{\alpha_1 s})}\right)^{\frac{1}{M_1\vec{w}-y_1}} = \left(\frac{1}{e(g,g)^{ts\alpha_1\langle M\vec{w}-\vec{y},\vec{\alpha}\rangle}}\right)^{\frac{1}{M_1\vec{w}-y_1}} = e(g,g)^{-st\alpha_1^2}.$$

Security. The co-selective security of the scheme relies on a q-type assumption defined in [22].

Definition 3 ([22]). *The* q-**Decision Multi-Exponent Bilinear Diffie-Hellman Problem** *(or q-MEBDH) in* $(\mathbb{G}, \mathbb{G}_T)$ *is, given* $Z \in \mathbb{G}_T$ *and a set of elements*

$$P = \left\{ \begin{array}{cc} & g,\ g^s,\ e(g,g)^\alpha \\ \forall_{1\le i,j\le q} & g^{a_i},\ g^{a_i s},\ g^{a_i a_j},\ g^{\alpha/a_i^2} \\ \forall_{1\le i,j,k\le q, i\neq j} & g^{a_i a_j s},\ g^{\alpha a_j/a_i^2},\ g^{\alpha a_i a_j/a_k^2},\ g^{\alpha a_i^2/a_j^2} \end{array} \right\},$$

where $g \xleftarrow{\$} \mathbb{G}$, $s, \alpha, a_1, \ldots, a_q \xleftarrow{\$} \mathbb{Z}_p$, *to decide whether* $Z = e(g,g)^{\alpha s}$ *or* $Z \in_R \mathbb{G}_T$.

Theorem 3. *Construction 4 is co-selectively secure under the q-Decisional Multi-Exponent Bilinear Diffie-Hellman assumption, where q is the number of key queries.* (The proof is given in appendix D).

Implications. For a vector $\vec{X} \in \mathbb{Z}_p^n$, the embedding $V_{\vec{X}} = \mathrm{Aff}(M_{\vec{X}}, \vec{0_n})$ defined in Eq.(3) is easily seen to be in the limited domain $\mathcal{W}_n$ since $(M_{\vec{X}})_{(-1)}$ is an identity matrix of size $n-1$ and hence $\mathrm{rank}((M_{\vec{X}})_{(-1)}) = n-1$. From Corollary 1, the above scheme thus implies non-zero-mode PAIPE.

5.2 Non-Zero-Mode PAIPE under Simple Assumptions

We proved the co-selective security of our negated spatial encryption scheme under a non-standard q-type assumption introduced in [22]. In this section, we show that the dual system technique [31] makes it possible to rest on simple assumptions such as DBDH and DLIN.

The scheme is very similar to the zero PAIPE scheme of section 4.2 and we only state the differences. Again, the intuition follows exactly from section 1 and the security proof uses similar techniques as in [22]. In appendix E, we describe an intuitively simpler variant of the scheme in composite order groups.

Construction 5. (Co-selectively secure non-zero-mode PAIPE)

▶ $\mathsf{Setup}(1^\lambda, n)$: outputs pk exactly as in the construction 3 except that we define $w = g^{\alpha_1} (= h_1)$ in this scheme, instead of g^{α_0}.

▶ $\mathsf{Keygen}(\vec{X}, \mathsf{msk}, \mathsf{pk})$: outputs $\mathsf{sk}_{\vec{X}} = (\mathsf{sk}_{\mathsf{adapt}}, \mathsf{sk}_{\mathsf{core}})$ where $\mathsf{sk}_{\mathsf{adapt}}$ is the same as in the construction 3 (with $w = g^{\alpha_1}$) and $\mathsf{sk}_{\mathsf{core}} = \{K_i = \left(g^{-\alpha_1 \frac{x_i}{x_1}} \cdot g^{\alpha_i}\right)^{r_1}\}_{i=2,\ldots,n}$.. Namely, it parses $\vec{X}$ as $(x_1, \ldots, x_n)$ and returns $\perp$ if $x_1 = 0$. Otherwise, it chooses $r_1, r_2, z_1, z_2 \xleftarrow{\$} \mathbb{Z}_p$. It sets $r = r_1 + r_2$ and outputs $\mathsf{sk}_{\vec{X}} = (D_1, \ldots, D_7, K_2, \ldots, K_n)$ where

$$
\begin{array}{llll}
D_1 = g^{\alpha a_1} \cdot v^r, & D_2 = g^{-\alpha} \cdot v_1^r \cdot g^{z_1}, & D_3 = B^{-z_1}, & D_4 = v_2^r \cdot g^{z_2}, \\
D_5 = B^{-z_2}, & D_6 = B^{r_2}, & D_7 = g^{r_1}, & \{K_i = \left(w^{-x_i/x_1} \cdot h_i\right)^{r_1}\}_{i=2,\ldots,n}
\end{array}
$$

▶ $\mathsf{Encrypt}(\vec{Y}, \mathsf{M}, \mathsf{pk})$: outputs $C = (C_{\mathsf{adapt}}, C_{\mathsf{core}})$ where C_{adapt} is as in the construction 3 (with $w = g^{\alpha_1}$) and $C_{\mathsf{core}} = \left(E_0 = \mathsf{M} \cdot Z^{s_2},\ E_1 = (g^{\langle \vec{\alpha}, \vec{Y} \rangle})^t,\ E_2 = g^t\right)$. In more details, to encrypt $\mathsf{M} \in \mathbb{G}_T$ under the vector $\vec{Y} = (y_1, \ldots, y_n) \in (\mathbb{Z}_p)^n$, pick $s_1, s_2, t \xleftarrow{\$} \mathbb{Z}_p$ and compute the ciphertext $C = (C_1, \ldots, C_7, E_0, E_1, E_2)$ where

$$
\begin{array}{llll}
E_0 = \mathsf{M} \cdot Z^{s_2}, & C_1 = B^{s_1 + s_2}, & C_2 = B_1^{s_1}, & C_3 = A_1^{s_1}, \\
C_4 = B_2^{s_2}, & C_5 = A_2^{s_2}, & C_6 = \tau_1^{s_1} \cdot \tau_2^{s_2}, & C_7 = T_1^{s_1} \cdot T_2^{s_2} \cdot w^{-t}.
\end{array} \tag{6}
$$

$$
E_1 = \left(w^{y_1} \cdot h_2^{y_2} \cdots h_n^{y_n}\right)^t = \left(w^{y_1} \cdot g^{\alpha_2 y_2 + \cdots + \alpha_n y_n}\right)^t, \qquad E_2 = g^t. \tag{7}
$$

▶ $\mathsf{Decrypt}(C, \vec{Y}, \mathsf{sk}_{\vec{X}}, \vec{X}, \mathsf{pk})$: computes W_1 as in construction 3 and W_2 as

$$
W_2 = \left(\frac{e(K_2^{y_2} \cdots K_n^{y_n}, E_2)}{e(E_1, D_7)}\right)^{-\frac{x_1}{\vec{X} \cdot \vec{Y}}} = e(g, w)^{r_1 t}.
$$

The correctness of the scheme is showed in appendix A.2. Its security proof relies on the DLIN and DBDH assumptions.

Theorem 4. *Construction 5 is co-selectively secure under the DLIN and DBDH assumptions.*

Proof. The proof uses exactly the same sequence of games as in Theorem 2. Semi-functional ciphertexts and keys are also defined identically to those of the previous scheme, *i.e.,* as those stated in Eq.(4,5), except that no tag components are used.

The proofs of indistinguishability between Game_{Real} and Game_0 as well as that of Game_q and Game_{q+1} proceed almost as those of Lemma 1 and 2.

Lemma 4. Game_0 *is indistinguishable from* Game_{Real} *under the DLIN assumption.*

Proof. The proof is identical to the one of lemma 1 since the switch from normal to semi-functional ciphertexts only affect elements (C_4, C_5, C_6, C_7) and, in their normal or semi-functional form, these ciphertexts components are identical to those of the PAIPE scheme of section 4.

Lemma 5. *For any* $1 \leq k \leq q$*, if an adversary* $\mathcal{A}$ *can distinguish* Game_k *from* Game_{k-1}*, we can build a distinguisher for the DLIN problem.*

Proof. The distinguisher $\mathcal{B}$ takes as input a DLIN instance $(g, f, \nu, g^{\theta_1}, f^{\theta_2}, \eta)$ and has to decide whether $\eta = \nu^{\theta_1+\theta_2}$ or not.

Init. The adversary $\mathcal{A}$ first outputs the vectors to be queried $\vec{X}_1, \ldots, \vec{X}_q$. We parse the k^{th} vector $\vec{X}_k$ as $(x_1, \ldots, x_n)$.

Setup. The algorithm $\mathcal{B}$ first randomly chooses $\alpha, a_1, a_2, \delta_{v_1}, \delta_{v_2} \xleftarrow{\$} \mathbb{Z}_p$ and sets $g = g$,

$$A_1 = g^{a_1}, \qquad A_2 = g^{a_2}, \qquad B = g^b = f, \qquad v_1 = \nu^{a_2} \cdot g^{\delta_{v_1}}$$
$$B_1 = g^{ba_1} = f^{a_1}, \qquad B_2 = g^{ba_2} = f^{a_2}, \qquad v = \nu^{-a_1 a_2}, \qquad v_2 = \nu^{a_1} \cdot g^{\delta_{v_2}},$$

as well as $e(g,g)^{\alpha a_1 b} = e(f,g)^{\alpha a_1}$, which allows defining

$$\tau_1 = v v_1^{a_1} = g^{\delta_{v_1} a_1}, \qquad \tau_2 = v v_2^{a_2} = g^{\delta_{v_2} a_2}, \qquad \tau_1^b = f^{\delta_{v_1} a_1}, \qquad \tau_2^b = f^{\delta_{v_2} a_2}.$$

Next, $\mathcal{B}$ chooses $\delta_w, \delta_1, \ldots, \delta_n \xleftarrow{\$} \mathbb{Z}_p$, and defines $w = f \cdot g^{\delta_w}$,

$$h_i = w^{x_i/x_1} \cdot g^{\delta_i} \qquad \text{for } i = 2, \ldots, n.$$

As in the proof of lemma 2 and the one of lemma 2 in [31], the simulator $\mathcal{B}$ knows the master secret key $\mathsf{msk} = (g^\alpha, g^{\alpha a_1}, v, v_1, v_2)$ of the system.

Key Queries. When $\mathcal{A}$ makes the j^{th} private key query, $\mathcal{B}$ does as follows.
[**Case** $j > k$] It generates a normal key, using the master secret key msk.
[**Case** $j < k$] It creates a semi-functional key using $g^{a_1 a_2}$.
[**Case** $j = k$] In this case, it generates a key by computing

$$D_1 = D_1' \cdot \eta^{-a_1 a_2}, \qquad D_2 = D_2' \cdot \eta^{a_2} \cdot (g^{\theta_1})^{\delta_{v_1}}, \quad D_3 = D_3' \cdot (f^{\theta_2})^{\delta_{v_1}},$$
$$D_4 = D_4' \cdot \eta^{a_1} \cdot (g^{\theta_1})^{\delta_{v_2}}, \quad D_5 = D_5' \cdot (f^{\theta_2})^{\delta_{v_2}}, \qquad D_6 = D_6' \cdot f^{\theta_2},$$

and $D_7 = D_7' \cdot (g^{\theta_1})$, as well as elements

$$K_i = K_i' \cdot (g^{\theta_1})^{\delta_i} \qquad \text{for } i = 2, \ldots, n,$$

which are all computable since we have $w^{x_i/x_1} \cdot h_i = g^{\delta_i}$ and $K_i = K_i' \cdot (g^{\theta_1})^{\delta_1} = (w^{x_i/x_1} h_i)^{r_1' + \theta_1}$, with $r_1' = \log_g(D_7')$. As in the proof of lemma 2, if $\eta = \nu^{\theta_1+\theta_2}$, $\mathsf{sk}_{\vec{X}} = (D_1, \ldots, D_7, K_2, \ldots, K_n)$ forms a normal key where $r_1 = r_1' + \theta_1$, $r_2 = r_2' + \theta_2$, $z_1 = z_1' - \delta_{v_1}\theta_2$, $z_2 = z_2' - \delta_{v_2}\theta_2$ are the implicitly defined underlying exponents. On the other hand, if ν is random, it can be expressed as $\eta = \nu^{\theta_1+\theta_2} \cdot g^\gamma$ for some $\gamma \in_R \mathbb{Z}_p$, so that $\mathsf{sk}_{\vec{X}}$ is distributed as a semi-functional key.

Challenge. At the challenge phase, $\mathcal{A}$ outputs two messages $\mathsf{M}_0, \mathsf{M}_1 \in \mathbb{G}_T$ along with a vector $\vec{Y}^\star = (y_1^\star, \ldots, y_n^\star)$ such that $\vec{X}_i \cdot \vec{Y}^\star = 0$ for each $i \in \{1, \ldots, q\}$. At this stage, $\mathcal{B}$ flips a coin $\beta \xleftarrow{\$} \{0,1\}$ and generates a normal encryption $(C_1', \ldots, C_7', E_0', E_1', E_2')$ of M_β. It then chooses $\chi \xleftarrow{\$} \mathbb{Z}_p$ and computes a perturbed ciphertext as

$$C_4 = C_4' \cdot f^{a_2 \cdot \chi}, \qquad C_5 = C_5' \cdot g^{a_2 \cdot \chi}, \qquad C_6 = C_6' \cdot v_2^{a_2 \cdot \chi},$$
$$C_7 = C_7' \cdot \nu^{-\delta_w \cdot a_1 \cdot a_2 \cdot \chi} \cdot f^{\delta_{v_2} \cdot a_2 \cdot \chi} = T_1^{s_1'} \cdot T_2^{s_2'} \cdot w^{-t'} \cdot \nu^{-\delta_w \cdot a_1 \cdot a_2 \cdot \chi} \cdot f^{\delta_{v_2} \cdot a_2 \cdot \chi},$$
$$E_1 = E_1' \cdot (\nu^{\sum_{i=2}^n y_i^\star \delta_i})^{a_1 \cdot a_2 \cdot \chi}, \qquad E_2 = E_2' \cdot \nu^{a_1 \cdot a_2 \cdot \chi}.$$

As in the proof of Lemma 2, the semi-functional component C_7 is created by implicitly setting $t = \log_g(E_2)$ as $t = t' + \log_g(\nu) a_1 a_2 \chi$. The term E_1 is computed as above since

$$\begin{aligned} E_1 &= \left(w^{y_1^\star} \cdot h_2^{y_2^\star} \cdots h_n^{y_n^\star}\right)^t = \left(w^{y_1^\star} \cdot (w^{x_2/x_1} g^{\delta_2})^{y_2^\star} \cdots (w^{x_n/x_1} g^{\delta_n})^{y_n^\star}\right)^t \\ &= \left(w^{\vec{X} \cdot \vec{Y}^\star / x_1} \cdot (g^{\delta_2})^{y_2^\star} \cdots (g^{\delta_n})^{y_n^\star}\right)^t = \left(g^{\sum_{j=2}^n \delta_j y_j^\star}\right)^t = E_1' \cdot (\nu^{\sum_{i=2}^n y_i^\star \delta_i})^{a_1 \cdot a_2 \cdot \chi}, \end{aligned}$$

where the unknown term w^t disappears due to the requirement $\vec{X} \cdot \vec{Y}^\star = 0$ on the challenge vector $\vec{Y}^\star$. We then can conclude that $(C_1', C_2', C_3', C_4, C_5, C_6, C_7, E_0, E_1, E_2)$ is properly distributed as a semi-functional ciphertext.

Eventually, $\mathcal{A}$ outputs a bit β' and $\mathcal{B}$ outputs 0 if $\beta = \beta'$. As in [31], we see that $\mathcal{A}$ is playing Game_{k-1} if $\eta = \nu^{\theta_1 + \theta_2}$ and Game_k otherwise.

Lemma 6. *If the DBDH assumption holds, no PPT distinguisher can tell apart* Game_q *and* Game_{q+1}.

Proof. Mutatis mutandis, the proof is identical to the one of lemma 3. As in the case of lemma 4, variable assignments are unchanged since all values that are implicitly defined (*i.e.*, not explicitly known to the simulator) appear in key or ciphertext components that are identical to those of the zero-mode PAIPE scheme. The only difference is that no tags are introduced in keys or ciphertexts.

5.3 A Generalization of the Scheme and Its Application

Extended Ciphertext Attribute Domain. The above scheme is a functional encryption for the relation $R^{\mathsf{NIPE}_n} : \mathbb{Z}_p^n \times \mathbb{Z}_p^n \to \{0,1\}$. It can be extended so as to support relations of the form $R^{\mathsf{NIPE}_n^*} : \mathbb{Z}_p^n \times (\mathbb{Z}_p^n)^d \to \{0,1\}$, for some $d \in \mathsf{poly}(\lambda)$, and defined as $R^{\mathsf{NIPE}_n^*}(\vec{X}, (\vec{Y}_1, \ldots, \vec{Y}_d)) = 1$ if and only if for all $i = 1, \ldots, d$: $\vec{X} \cdot \vec{Y}_i \neq 0$.

We construct this extended system by setting up exactly the same public and private keys (for $\vec{X}$) as in the original scheme. To encrypt to $(\vec{Y}_1, \ldots, \vec{Y}_d)$, the scheme generates C_{adapt} and E_0 as usual with the underlying exponents s_1, s_2, t. Then, it chooses $t_1, \ldots, t_d \in \mathbb{Z}_p$ so that $t = t_1 + \cdots + t_d$ and for $i = 1, \ldots, d$, parses $\vec{Y}_i = (y_{i,1}, \ldots, y_{i,n})$ and computes $E_{1,i} = (g^{\langle \vec{\alpha}, \vec{Y}_i \rangle})^{t_i} = (h_1^{y_{i,1}} \cdots h_n^{y_{i,n}})^{t_i}$ and $E_{2,i} = g^{t_i}$, in such a way that the ciphertext is $(C_1, \ldots, C_7, E_0, \{E_{1,i}, E_{2,i}\}_{i=1,\ldots,d})$. Decryption requires to first compute

$$W_{2,i} = \left(\frac{e(K_2^{y_{i,2}} \cdots K_n^{y_{i,n}}, E_{2,i})}{e(E_{1,i}, D_7)}\right)^{-\frac{x_1}{\vec{X} \cdot \vec{Y}_i}} = e(g, w)^{r_1 t_i},$$

for $i = 1, \ldots, d$, from which the receiver obtains[5] $W_2 = W_{2,1} \cdots W_{2,d} = e(g,w)^{r_1 t}$. The remaining calculations are carried out as in the basic scheme and we now explain how the security proof must be adapted.

The proof of co-selective security is almost identical to the proof of Lemma 5, except that simulating the challenge ciphertext is instead done as follows. First, recall that the challenge vector set $(\vec{Y}_1, \ldots, \vec{Y}_d)$ is legal iff, for each private key query $\vec{X}_\ell$ ($\ell \in [1,q]$), there exists $j \in [1,d]$ such that $\vec{X}_\ell \cdot \vec{Y}_j = 0$. Let us consider the k^{th} query as in the proof of Lemma 5 and let $j \in [1,d]$ be such that $\vec{X}_k \cdot \vec{Y}_j = 0$. For all $i \in [1,d]$ such that $i \neq j$, the simulator $\mathcal{B}$ picks $t_i \xleftarrow{\$} \mathbb{Z}_p$ and computes $E_{1,i}, E_{2,i}$ as specified by the scheme. We let $t'' = \sum_{i \neq j} t_i$ be the sum of these values. The simulator then implicitly defines the exponent $t_j = t' - t'' + \log_g(\nu) a_1 a_2 \chi$. Analogously to the proof of Lemma 5, it can compute $E_{1,j} = E'_1 \cdot (g^{-t''} \cdot \nu^{a_1 a_2 \chi})^{\sum_{i=2}^n y_i^\star \delta_i}$ and $E_{2,j} = E'_2 \cdot g^{-t''} \cdot \nu^{a_1 a_2 \chi}$ (where E'_1, E'_2 are part of a normal ciphertexts obtained as in the basic scheme) in such a way that the term w^{t_j} is canceled out due to $\vec{X} \cdot \vec{Y}_j = 0$. The rest of the proof then follows similarly.

Applications. We can obtain an identity-based revocation scheme with parameter tradeoff from the aforementioned extension. The instantiation of ID-based revocation scheme ($\mathsf{IBR}_{\leq n}$) from our non-zero inner-product system NIPE_{n+1} yields a construction with $O(1)$-size ciphertexts and $O(n)$-size private keys, where n denotes the maximal number of revoked users.

From our extended scheme NIPE^*_{n+1}, we can obtain an ID-based revocation scheme $\mathsf{IBR}_{\mathsf{poly}(\lambda)}$, without a fixed maximal number of revoked users. To revoke the set R where $|R| = r$, we divide it into a disjointed union $R = R_1 \cup \cdots \cup R_{r/n}$, where $|R_i| = n$ for all i (we assume that n divides r). We then simply construct the vector $\vec{Y}_i$ from the revocation subset R_i for each $i \in [1, r/n]$, in the same way as we use NIPE_{n+1} to instantiate $\mathsf{IBR}_{\leq n}$. We then finally encrypt using the set of vectors $(\vec{Y}_1, \ldots, \vec{Y}_{r/n})$. The correctness and security properties are easily seen to hold since we have

$$R^{\mathsf{IBR}_{\leq n}}(\mathsf{ID}, R) = 1 \;\Leftrightarrow\; \mathsf{ID} \notin R \;\Leftrightarrow\; \forall i \in [1, r/n] : \mathsf{ID} \notin R_i \Leftrightarrow\; R^{\mathsf{IBR}_{\mathsf{poly}(\lambda)}}(\mathsf{ID}, (R_1, \ldots, R_{r/n})) = 1$$

As far as efficiency goes, the construction has $O(r/n)$-size ciphertexts and $O(n)$-size private keys. Interestingly, we note that the second scheme described by Lewko, Sahai and Waters [22] (which indeed inspires ours) can be viewed as a special case of our scheme where $n = 1$.

Comparison Between Revocation Schemes. Table 3 gives a comparative efficiency between the two revocation schemes described by Lewko, Sahai and Waters [22] and our constructions.

From a security point of view, all schemes are proven secure in the standard model and in a non-adaptive sense. Comparisons are thus given w.r.t. the same metrics as in section 4.3 as well as according to whether the number of revocations must be fixed in advance or not.

6 Conclusion

This paper proposed new constructions of functional encryption with short ciphertexts in the public attribute setting, where ciphertexts do not have to be anonymous. In prime order groups, these new schemes gave rise to the first adaptively secure identity-based broadcast encryption scheme with constant-size ciphertexts in the standard model. In their negated counterpart, they also imply the

[5] If $n < r^{1/2}$, the receiver can more efficiently compute $W_2 = \frac{\prod_{j=2}^n e\left(K_j, \prod_{i=1}^d E_{2,i}^{-y_{i,j} \cdot x_1/(\vec{X} \cdot \vec{Y}_i)}\right)}{e\left(\prod_{i=1}^d E_{1,i}^{-x_1/(\vec{X} \cdot \vec{Y}_i)}, D_7\right)}$, so that only $O(n)$ pairing evaluations are required (instead of $O(r/n)$).

Table 3. Performances of revocation systems

Revocation schemes	Ciphertext overhead	Private key size	Decryption cost	Assumption	Bounded number of revocations?
LSW1 [22]	$O(r) \times \|\mathbb{G}\|$	$O(1) \times \|\mathbb{G}\|$	O(1) p. + O(r) exp.	q-MEBDH	No
LSW2 [22]	$O(r) \times \|\mathbb{G}\|$	$O(1) \times \|\mathbb{G}\|$	O(1) p. + O(r) exp.	DLIN + DBDH	No
Basic scheme	$O(1) \times \|\mathbb{G}\|$	$O(r_{\max}) \times \|\mathbb{G}\|$	O(1) p. + O(r) exp.	DLIN + DBDH	Yes
Tradeoff scheme	$O(\frac{r}{n}) \times \|\mathbb{G}\|$	$O(n) \times \|\mathbb{G}\|$	$O(\min(\frac{r}{n}, n))$ p. + O(r) exp.	DLIN + DBDH	No

n:any parameter, r:the number of revoked users, $r_{\max}$: the maximum size of r (if required),
p.:pairing applications, exp.:group exponential applications

first (identity-based) revocation schemes featuring constant-size ciphertexts, no matter how many users are revoked. We also introduced a revocation analogue of the spatial encryption primitive of Boneh and Hamburg, which we showed to imply revocation.

In composite order groups, we also described conceptually simpler variants of the above constructions using the Lewko-Waters techniques. Our fully secure zero-mode PAIPE scheme (construction 3) was notably generalized into an adaptively secure spatial encryption system, which provides a simpler answer to a question left open by Boneh and Hamburg.

These results leave a few open problems. First, it would be nice to completely (namely, with a proper delegation mechanism) implement fully secure spatial encryption in groups of prime order with any (*i.e.*, not only asymmetric) pairing configuration. This would require new ideas to eliminate tags or get the delegation technique of [11] to suitably interact with them. Another open problem is to move beyond the co-selective model when it comes to prove the security of our negated schemes. In particular, it would be interesting to have adaptively secure negated spatial encryption realizations (ideally, under simple assumptions).

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, H. Shi. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In *Crypto'05*, *LNCS* 3621, pp. 205–222, 2005.
2. M. Abdalla, E. Kiltz, G. Neven. Generalized Key Delegation for Hierarchical Identity-Based Encryption. In *ESORICS'07*, *LNCS* 4734, pp. 139–154. Springer, 2007.
3. N. Attrapadung, B. Libert. Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In *PKC'10*, *LNCS* 6056, pp. 384–402. Springer, 2010.
4. A. Barth, D. Boneh, B. Waters. Privacy in Encrypted Content Distribution Using Private Broadcast Encryption. In *Financial Cryptography 2006*, *LNCS* 4107, pp. 52–64, 2006.
5. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.
6. D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55, 2004.
7. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical Identity-Based encryption with Constant Size Ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.
8. D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano. Public Key Encryption with Keyword Search. In *Eurocrypt'04*, *LNCS* 3027, pp. 506–522, 2004.
9. D. Boneh, M. Franklin. Identity-Based Encryption from the Weil Pairing. In *SIAM* Journal of Computing 32(3), pp. 586–615, 2003, earlier version in *Crypto'01*, *LNCS* 2139, pp. 213–229, 2001.
10. D. Boneh, C. Gentry, B. Waters. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *Crypto'05*, *LNCS* 3621, pp. 258–275, 2005.
11. D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt'08*, *LNCS* 5350, pp. 455–470, 2008

12. D. Boneh, B. Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. In *4th Theory of Cryptography Conference* (TCC 2007), *LNCS* 4392, pp. 535–554, 2007.
13. R. Canetti, S. Halevi, J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Eurocrypt'03*, *LNCS* 2656, pp. 254–271, 2003.
14. R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt'04*, *LNCS* 3027, pp. 207–222, 2004.
15. C. Delerablée. Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In *Asiacrypt'07*, *LNCS* 4833, pp. 200–215, 2007.
16. D. Freeman. Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups. In *Eurocrypt'10*, *LNCS* series, to appear, 2010.
17. C. Gentry, B. Waters. Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In *Eurocrypt'09*, *LNCS* 5479, pp. 171–188, 2009.
18. V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS'06*, pp. 89–98, 2006.
19. V. Iovino, G. Persiano. Hidden-Vector Encryption with Groups of Prime Order. In *Pairing'08*, *LNCS* 5209, pp. 75–88, 2008.
20. J. Katz, A. Sahai, B. Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *Eurocrypt'08*, *LNCS* 4965, pp. 146-162, 2008.
21. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt 2010*, *LNCS* 6110, pp. 62–91, 2010.
22. A. Lewko, A. Sahai, B. Waters. Revocation Systems with Very Small Private Keys. In IEEE Symposium on Security and Privacy (S&P) 2010, to appear.
23. A. Lewko, B. Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010*, LNCS 5978, pp. 455-479, Springer, 2010.
24. R. Ostrovsky, A. Sahai, B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM CCS'07*, pp. 195–203, 2007.
25. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt'05*, *LNCS* 3494, pp. 457–473, 2005.
26. R. Sakai, J. Furukawa. Identity-Based Broadcast Encryption. In Cryptology ePrint Archive: Report 2007/217, `http://eprint.iacr.org/2007/217`, 2007.
27. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Crypto'84*, *LNCS* 196, pp. 47–53, 1984.
28. E. Shen, E. Shi, B. Waters. Predicate Privacy in Encryption Systems. In *TCC'09*, *LNCS* 5444, pp. 457–473, 2009.
29. E. Shi, B. Waters. Delegating Capabilities in Predicate Encryption Systems. In *ICALP'08*, *LNCS* 5126, pp. 560–578, 2008.
30. K. Takashima, T. Okamoto. Hierarchical Predicate Encryption for Inner-Products. In *Asiacrypt'09*, *LNCS* 5912, pp. 214–231, 2009.
31. B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Crypto'09*, *LNCS* series, 2009.

A Verifying Correctness in Decryption

A.1 For the Zero-Mode PAIPE Scheme of Section 4.2

$$W_2 = \left(\frac{e(\prod_{i=2}^n K_i^{y_i}, E_2)}{e(E_1, D_7)}\right)^{\frac{1}{\mathsf{tagk}-\mathsf{tagc}}} = \left(\frac{e\Big(\prod_{i=2}^n (g^{-\alpha_1 \frac{x_i}{x_1}} g^{\alpha_i} w^{\mathsf{tagk}_i})^{r_1 y_i}, g^t\Big)}{e\Big(\big(g^{\langle \vec{\alpha}, \vec{Y}\rangle} \cdot w^{\mathsf{tagc}}\big)^t, g^{r_1}\Big)}\right)^{\frac{1}{\mathsf{tagk}-\mathsf{tagc}}}$$

$$= \left(\frac{e\Big((g^{-\alpha_1 \frac{x_2 y_2 + \cdots + x_n y_n}{x_1}} g^{\alpha_2 y_2 + \cdots + \alpha_n y_n} w^{\mathsf{tagk}_2 y_2 + \cdots + \mathsf{tagk}_n y_n})^{r_1}, g^t\Big)}{e\Big(\big(g^{\alpha_1 y_1 + \alpha_2 y_2 + \cdots + \alpha_n y_n} \cdot w^{\mathsf{tagc}}\big)^t, g^{r_1}\Big)}\right)^{\frac{1}{\mathsf{tagk}-\mathsf{tagc}}}$$

$$= e\Big(g^{-\alpha_1 (\frac{x_2 y_2 + \cdots x_n y_n}{x_1} + y_1)} w^{(\mathsf{tagk}-\mathsf{tagc})}, g\Big)^{\frac{r_1 t}{\mathsf{tagk}-\mathsf{tagc}}}$$

$$= e\Big(g^{-\alpha_1 \frac{\vec{X}\cdot\vec{Y}}{x_1}} w^{(\mathsf{tagk}-\mathsf{tagc})}, g\Big)^{\frac{r_1 t}{\mathsf{tagk}-\mathsf{tagc}}} = e(g, w)^{r_1 t}.$$

A.2 For the Non-Zero-Mode PAIPE Scheme of Section 5.2

$$W_2 = \left(\frac{e(\prod_{i=2}^n K_i^{y_i}, E_2)}{e(E_1, D_7)}\right)^{-\frac{x_1}{\vec{X}\cdot\vec{Y}}} = \left(\frac{e\Big(\prod_{i=2}^n (g^{-\alpha_1 \frac{x_i}{x_1}} g^{\alpha_i})^{r_1 y_i}, g^t\Big)}{e\Big(\big(g^{\alpha_1 y_1 + \cdots + \alpha_n y_n}\big)^t, g^{r_1}\Big)}\right)^{-\frac{x_1}{\vec{X}\cdot\vec{Y}}}$$

$$= \left(\frac{e\Big((w^{-\frac{x_2 y_2 + \cdots + x_n y_n}{x_1}} g^{\alpha_2 y_2 + \cdots + \alpha_n y_n})^{r_1}, g^t\Big)}{e\Big(\big(w^{y_1} \cdot g^{\alpha_2 y_2 + \cdots + \alpha_n y_n}\big)^t, g^{r_1}\Big)}\right)^{-\frac{x_1}{\vec{X}\cdot\vec{Y}}}$$

$$= e\Big(w^{\frac{\vec{X}\cdot\vec{Y}}{x_1}}, g\Big)^{r_1 t \cdot \frac{x_1}{\vec{X}\cdot\vec{Y}}} = e(g, w)^{r_1 t}.$$

B Adaptively Secure Spatial Encryption in Composite Order Groups

The Lewko-Waters techniques [23] apply to provide a simpler realization (that avoids the use of tags and achieves perfect correctness) of our PAIPE scheme in groups whose order is a product $N = p_1 p_2 p_3$ of three distinct primes.

However, as showed in Section 4.1, spatial encryption is a more general primitive than PAIPE as it includes zero-mode PAIPE as a special case. For this reason, we only describe a fully secure spatial encryption construction in composite order groups, a special case of which is a fully secure zero-mode PAIPE with a similar efficiency and based on the same assumptions.

In a nutshell, the idea is to use groups of order $N = p_1 p_2 p_3$ to turn the selectively secure scheme of Boneh and Hamburg [11] into an adaptively secure scheme.

B.1 Construction

We first briefly recall the concept of spatial encryption [11]. For a matrix $M \in \mathbb{Z}_N^{n\times d}$ and a vector $\vec{c} \in \mathbb{Z}_N^n$, one considers the affine space $\mathrm{Aff}(M,\vec{c}) = \{M\vec{w} + \vec{c} \mid \vec{w} \in \mathbb{Z}_N^d\}$. Let $\mathcal{V}_n \subseteq 2^{(\mathbb{Z}_p^n)}$ be the collection of all affine spaces inside $\mathbb{Z}_N^n$. That is, $\mathcal{V}_n$ is defined as

$$\mathcal{V}_n = \{\mathrm{Aff}(M,\vec{c}) \mid M \in \mathbb{M}_{n\times d}, c \in \mathbb{Z}_N^n, d \leq n\},$$

where $\mathbb{M}_{n\times d}$ is the set of all $n \times d$ matrices in $\mathbb{Z}_N$.

In a spatial encryption scheme, private keys correspond to affine subspaces and ciphertexts are associated with a vector and can be decrypted by any private key associated with a subspace containing that vector. In addition, a private key corresponding to an affine subspace V_1 allows deriving (using algorithm Delegate below) a private key for any subspace V_2 such that $V_2 \subset V_1$.

In [11], Boneh and Hamburg gave a construction of spatial encryption with short ciphertexts. It is inspired by the Boneh-Boyen-Goh hierarchical identity-based encryption scheme [7]. We show that the Lewko-Waters techniques [23] indeed apply to tweak the Boneh-Hamburg construction and render it adaptively secure.

The description hereafter thus uses groups $(\mathbb{G}, \mathbb{G}_T)$ of order $N = p_1p_2p_3$. In this kind of group, for each $i, j \in \{1, 2, 3\}$, we denote by $\mathbb{G}_{p_i}$ the subgroup of $\mathbb{G}$ of order p_i while $\mathbb{G}_{p_ip_j}$ stands for the subgroup of order p_ip_j.

Ciphertexts are generated exactly in the same way as in [11] but they live in the subgroup of order p_1. Private keys are also generated as in the underlying basic schemes and are then multiplied by a random element of order p_3. These randomizers of order p_3 vanish upon decryption since pairing two elements of order p_i and p_j, with $i \neq j$, always gives the identity element $1_{\mathbb{G}_T}$.

The security proof relies on the fact that, at the first transition in the sequence of games, normal ciphertexts are indistinguishable from semi-functional ones, where ciphertexts components live in the subgroup $\mathbb{G}_{p_1p_2}$ according to a certain distribution. In addition, at some step of the sequence of games, normal private keys are computatinally indistinguishable from semi-functional keys, the elements of which have a non-trivial component of order p_2.

Construction 6. (Fully Secure Spatial Encryption)

▶ $\mathsf{Setup}(n)$: chooses bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of order $N = p_1p_2p_3$. It then picks a random generator $g \xleftarrow{\$} \mathbb{G}_{p_1}$ and $X_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. It randomly chooses $\alpha, \alpha_0, \ldots, \alpha_n \xleftarrow{\$} \mathbb{Z}_N$. Let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_N^n$. The public key is $\mathsf{pk} = \big(g, g^{\vec{\alpha}}, e(g,g)^{\alpha}, X_3\big)$. The master secret key is $\mathsf{msk} = g^{\alpha}$.

▶ $\mathsf{KeyGen}(V, \mathsf{msk}, \mathsf{pk})$: suppose that $V = \mathrm{Aff}(M,\vec{c})$, from a matrix $M \in (\mathbb{Z}_p)^{n\times d}$ and a vector $\vec{c} \in \mathbb{Z}_N^n$. The algorithm chooses $t \xleftarrow{\$} \mathbb{Z}_N$, $R_0', R_1' \xleftarrow{\$} \mathbb{G}_{p_3}$, and $\vec{R}' \xleftarrow{\$} \mathbb{G}_{p_3}^d$. It outputs the private key as $\mathsf{sk}_V = (D_0, D_1, \vec{K}) \in \mathbb{G}_{p_1p_3}^{d+2}$ where

$$D_0 = g^t R_0', \qquad D_1 = g^{\alpha + t(\alpha_0 + \langle \vec{c}, \vec{\alpha}\rangle)} R_1', \qquad \vec{K} = g^{tM^\top \vec{\alpha}} \vec{R}'.$$

▶ $\mathsf{Delegate}(\mathsf{msk}, \mathsf{pk}, V_1, D_{V_1}, V_2)$: takes as input two subspaces $V_1 = \mathrm{Aff}(M_1, \vec{c_1})$, $V_2 = \mathrm{Aff}(M_2, \vec{c_2})$ for some matrices $M_1 \in \mathbb{Z}_N^{n\times d_1}$, $M_2 \in \mathbb{Z}_N^{n\times d_2}$. It outputs $\perp$ if it turns out that $V_2 \not\subset V_1$. Otherwise, we must have $M_2 = M_1T$ and $\vec{c_2} = \vec{c_1} + M_1\vec{x}$ for some efficiently computable matrix $T \in \mathbb{Z}_N^{d_1\times d_2}$ and some vector $\vec{x} \in \mathbb{Z}_N^{d_1}$. Given $D_{V_1} = (D_0, D_1, \vec{K}) \in \mathbb{G}_{p_1p_3}^{d_1+2}$, these allow computing a delegated

key $D_{V_2} = (D'_0, D'_1, \vec{K}') \in \mathbb{G}_{p_1 p_3}^{d_2+2}$ as

$$\begin{aligned} D_{V_2} &= \big(D_0 \cdot g^{t_1} \cdot R''_0,\ D_1 \cdot \vec{K}^{x^\top} \cdot g^{\alpha_0 t_1} \cdot g^{t_1 \langle \vec{c_2}, \vec{\alpha} \rangle} \cdot R''_1,\ \vec{K}^{T^\top} \cdot g^{t_1 M_2^\top \vec{\alpha}} \cdot \vec{R}'' \big) \\ &= \big(g^{t'} \cdot R'''_0,\ g^{\alpha} \cdot g^{\alpha_0 t'} \cdot g^{t' \langle \vec{c_2}, \vec{\alpha} \rangle} \cdot R'''_1,\ g^{t' M_2^\top \vec{\alpha}} \cdot \vec{R}''' \big), \end{aligned}$$

where $t' = t + t_1$, for some randomly drawn $t_1 \xleftarrow{\$} \mathbb{Z}_p$, $R''_0, R''_1 \xleftarrow{\$} \mathbb{G}_{p_3}$, $\vec{R}'' \xleftarrow{\$} \mathbb{G}_{p_3}^{d_2}$.

▶ $\mathsf{Encrypt}(\vec{Y}, \mathsf{M}, \mathsf{pk})$: picks $s \xleftarrow{\$} \mathbb{Z}_N$ and computes the ciphertext as

$$C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, \qquad C_1 = g^{s(\alpha_0 + \langle \vec{Y}, \vec{\alpha} \rangle)}, \qquad C_2 = g^s.$$

▶ $\mathsf{Decrypt}(C, \vec{Y}, \mathsf{sk}_V, V, \mathsf{pk})$: the decryption algorithm first obtain $M \in \mathbb{Z}_N^{n \times d}, \vec{c} \in \mathbb{Z}_N^n$ from V. Suppose that $\vec{Y} \in V$ (so that decryption is possible). Therefore, there must exist $\vec{w} \in \mathbb{Z}_N^d$ such that $M\vec{w} + \vec{c} = \vec{Y}$. It then solves this system of linear equations to obtain $\vec{w}$. It then computes the message blinding factor as

$$\frac{e(D_1 \vec{K}^{\vec{w}^\top}, C_2)}{e(C_1, D_0)} = e(g,g)^{\alpha s}.$$

The correctness of the scheme can be verified by observing that

$$\frac{e(D_1 \cdot \vec{K}^{\vec{w}^\top}, C_2)}{e(C_1, D_0)} = \frac{e(g^{\alpha + t(\alpha_0 + \langle \vec{c}, \vec{\alpha} \rangle)} g^{t \vec{w}^\top M^\top \vec{\alpha}}, g^s)}{e(g^{s(\alpha_0 + \langle \vec{Y}, \vec{\alpha} \rangle)}, g^t)} = \frac{e(g^{\alpha + t(\alpha_0 + \langle \vec{c} + M\vec{w}, \vec{\alpha} \rangle)}, g^s)}{e(g^{s(\alpha_0 + \langle \vec{Y}, \vec{\alpha} \rangle)}, g^t)} = e(g,g)^{\alpha s}.$$

B.2 Security Proof

The above spatial encryption is adaptively secure under Assumptions 1, 2 and 3 stated by Lewko and Waters in [23], which are described as follows.

1. Given $g \xleftarrow{\$} \mathbb{G}_{p_1}, X_3 \xleftarrow{\$} \mathbb{G}_{p_3}$, and $T \in \mathbb{G}$, decide if $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$ or $\mathbb{G}_{p_1}$.
2. Let $g, X_1 \xleftarrow{\$} \mathbb{G}_{p_1}, X_2, Y_2 \xleftarrow{\$} \mathbb{G}_{p_2}, X_3, Y_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Given $(g, X_1 X_2, X_3, Y_2 Y_3)$, and $T \in \mathbb{G}$, decide if $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$ or $\mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$.
3. Let $g \xleftarrow{\$} \mathbb{G}_{p_1}$, $X_2, Y_2, Z_2 \xleftarrow{\$} \mathbb{G}_{p_2}$, $X_3, Z_3 \xleftarrow{\$} \mathbb{G}_{p_3}$ and $\alpha, s \xleftarrow{\$} \mathbb{Z}_N$. Given a tuple of elements $(g, g^\alpha X_2, X_3, g^s Y_2, Z_2 Z_3)$ and $T \in \mathbb{G}_T$, decide if $T = e(g,g)^{\alpha s}$ or not.

The proof follows exactly the same strategy as that of [23]. It uses the same sequence of games as in Theorem 2, except only that we insert one more game namely $\mathsf{Game}_{Restricted}$ between Game_{Real} and Game_0. This game, $\mathsf{Game}_{Restricted}$, will be the same as Game_{Real}, except that the adversary is not allowed to ask for keys of $\vec{X}$ such that $M\vec{w} + \vec{c} = \vec{Y}^\star \bmod p_2$ for some $\vec{w} \in \mathbb{Z}_{p_2}^d$. The semi-functional ciphertexts and keys are defined as follows. Let g_2 denote a generator of $\mathbb{G}_{p_2}$.

- Semi-functional ciphertexts are generated from a normal ciphertext (C'_0, C'_1, C'_2) by choosing random $x, z_{\mathsf{c}} \xleftarrow{\$} \mathbb{Z}_N$ and setting

$$C_0 = C'_0, \qquad C_1 = C'_1 \cdot g_2^{x z_{\mathsf{c}}}, \qquad C_2 = C'_2 \cdot g_2^x$$

- Semi-functional keys are obtained from a normal key $(D'_0, D'_1, D'_2, \vec{K}')$ by choosing random $\gamma, z_{\mathsf{k}} \xleftarrow{\$} \mathbb{Z}_N$, $\vec{z} \xleftarrow{\$} \mathbb{Z}_N^d$ and setting

$$D_0 = D'_0 \cdot g_2^{\gamma}, \qquad D_1 = D'_1 \cdot g_2^{\gamma z_{\mathsf{k}}}, \qquad \vec{K} = \vec{K}' \cdot g_2^{\gamma \vec{z}}$$

We note that if one attempts to decrypt a semi-functional ciphertext with a semi-functional key, the output from decryption will be the correct mask $e(g,g)^{\alpha s}$ multiplied by the perturbation factor $e(g_2,g_2)^{x\gamma(z_{\mathsf{k}}+\langle\vec{z},\vec{w}\rangle-z_{\mathsf{c}})}$.

The indistinguishability between games $\mathsf{Game}_{Real}/\mathsf{Game}_{Restricted}$, between $\mathsf{Game}_{Restricted}/\mathsf{Game}_0$ as well as $\mathsf{Game}_q/\mathsf{Game}_{q+1}$ can be proved almost exactly in the same way as in [23]. These proofs rely on Assumption 1 and 2, Assumption 1, Assumption 3, respectively. We thus omit them here and focus on the following lemma.

Lemma 7. *For any $1 \le k \le q$, if an adversary $\mathcal{A}$ can distinguish Game_k from Game_{k-1}, we can build an algorithm $\mathcal{B}$ that breaks Assumption 2 given in [23].*

Proof. The distinguisher $\mathcal{B}$ takes in a problem instance $(g, X_1X_2, X_3, Y_2Y_3, T)$ for Assumption 2. Its task is to decide whether $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$ or $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$.

Setup. Algorithm $\mathcal{B}$ first picks $\alpha, \alpha_0, \ldots, \alpha_n \xleftarrow{\$} \mathbb{Z}_N$ and prepares $\mathsf{pk} = \left(g, g^{\vec{\alpha}}, e(g,g)^{\alpha}, X_3\right)$ as usual. It sends pk to $\mathcal{A}$.

Key Queries. When $\mathcal{A}$ makes the j^{th} private key query, $\mathcal{B}$ does as follows.

[**Case** $j > k$] $\mathcal{B}$ generates a normal key as in the construction. This can be done since it knows the master key $\mathsf{msk} = g^{\alpha}$.

[**Case** $j < k$] In this case, $\mathcal{B}$ creates a semi-functional key. To do so, it first computes a normal private key $(D'_0, D'_1, \vec{K}')$. It then chooses $\tilde{\gamma}, z_{\mathsf{k}} \xleftarrow{\$} \mathbb{Z}_N$, $\vec{z} \xleftarrow{\$} \mathbb{Z}_N^d$ and sets

$$D_0 = D'_0 \cdot (Y_2Y_3)^{\tilde{\gamma}}, \qquad D_1 = D'_1 \cdot (Y_2Y_3)^{\tilde{\gamma} z_{\mathsf{k}}}, \qquad \vec{K} = \vec{K}' \cdot (Y_2Y_3)^{\tilde{\gamma}\vec{z}}.$$

This is a properly distributed semi-functional key with $g_2^{\gamma} = Y_2^{\tilde{\gamma}}$.

[**Case** $j = k$] The distinguisher $\mathcal{B}$ picks $u \xleftarrow{\$} \mathbb{Z}_N$, $\vec{u} \xleftarrow{\$} \mathbb{Z}_N^d$ and constructs the private key $(D_0, D_1, D_2, K_2, \ldots, K_n)$ as

$$D_0 = T, \qquad D_1 = g^{\alpha} \cdot T^{\alpha_0 + \langle \vec{c}, \vec{\alpha} \rangle} \cdot X_3^u, \qquad \vec{K} = T^{M^{\top}\vec{\alpha}} \cdot X_3^{\vec{u}}.$$

If $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then this is a normal key with g^t being equal to the $\mathbb{G}_{p_1}$ component of T. If $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then this is a semi-functional key with g_2^{γ} being equal to the $\mathbb{G}_{p_2}$ component of T and $z_{\mathsf{k}} = \alpha_0 + \langle \vec{c}, \vec{\alpha} \rangle \bmod p_2$, $\vec{z} = M^{\top}\vec{\alpha} \bmod p_2$. We note that $\vec{\alpha} \bmod p_1$ is not correlated with $\vec{\alpha} \bmod p_2$ and these values are properly distributed.

Challenge. In the challenge phase, $\mathcal{A}$ outputs messages $\mathsf{M}_0, \mathsf{M}_1 \in \mathbb{G}_T$ along with her target $\vec{Y}^{\star}$. Then, $\mathcal{B}$ flips a coin $\beta \xleftarrow{\$} \{0,1\}$ and forms the challenge ciphertext as

$$C_0 = \mathsf{M}_{\beta} \cdot e(X_1X_2, g)^{\alpha}, \qquad C_1 = (X_1X_2)^{\alpha_0 + \langle \vec{Y}^{\star}, \vec{\alpha} \rangle}, \qquad C_2 = X_1X_2.$$

We claim that this is a properly distributed semi-functional ciphertext with $g^s = X_1$, $g_2^x = X_2$ and $z_{\mathsf{c}} = \alpha_0 + \langle \vec{Y}^{\star}, \vec{\alpha} \rangle$. To prove this, we must show that $z_{\mathsf{k}}, \vec{z}$ (from each queried private key) and z_{c} are all independently distributed from $\mathcal{A}$'s view. Hence, it suffices to show that

$$\begin{pmatrix} 1 & \vec{c}^{\top} \\ \vec{0} & M^{\top} \\ 1 & (\vec{Y}^{\star})^{\top} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} z_{\mathsf{k}} \\ \vec{z} \\ z_{\mathsf{c}} \end{pmatrix} \bmod p_2$$

has always a solution in $(\alpha_0, \vec{\alpha})$ modulo p_2. To this end, it suffices to show that, in the matrix of the left-hand-side member, the last row (which relates to the information revealed by the challenge ciphertext) is independent of all the other rows. This independence is guaranteed by the inequality $M\vec{w} + \vec{c} \neq \vec{Y}^{\star} \bmod p_2$ for any $\vec{w} \in \mathbb{Z}_{p_2}^d$, which is exactly our requirement in the game.

Eventually, the adversary $\mathcal{A}$ outputs a bit β' and $\mathcal{B}$ outputs 0 if $\beta = \beta'$. As in [23], we see that $\mathcal{A}$ is playing Game_{k-1} in the event that $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$ and Game_k otherwise (*i.e.,* $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$).

B.3 Zero-Mode PAIPE in Composite Order Groups

As a special case of the fully secure spatial encryption system, we outline a simple construction of PAIPE scheme which is very similar to the selectively-secure scheme of section 4.1.

Construction 7. (Simpler Fully Secure Zero-Mode PAIPE)

▶ $\mathsf{Setup}(n)$: proceeds almost as in appendix B.1. Namely, it chooses bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of order $N = p_1p_2p_3$ and picks random $g \xleftarrow{\$} \mathbb{G}_{p_1}$ and $X_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. It chooses $\alpha, \alpha_0, \ldots, \alpha_n \xleftarrow{\$} \mathbb{Z}_N$. Let $\vec{\alpha} = (\alpha_0, \ldots, \alpha_n)$. The public key is $\mathsf{pk} = \big(g, g^{\vec{\alpha}}, e(g,g)^{\alpha}, X_3\big)$. The master key consists of $\mathsf{msk} = g^{\alpha}$.

▶ $\mathsf{KeyGen}(\vec{X}, \mathsf{msk}, \mathsf{pk})$: chooses $t \xleftarrow{\$} \mathbb{Z}_N$ and $R'_0, R'_1, R_2, \ldots, R_n \xleftarrow{\$} \mathbb{G}_{p_3}$. The algorithm parses $\vec{X}$ as a vector $(x_1, \ldots, x_n) \in (\mathbb{Z}_N)^n$. Then, it outputs the private key as $\mathsf{sk}_{\vec{X}} = (D_0, D_1, K_2, \ldots, K_n)$ where

$$D_0 = g^t R'_0, \qquad D_1 = g^{\alpha+\alpha_0 t} R'_1, \qquad \{K_i = (g^{-\alpha_1 \frac{x_i}{x_1}} g^{\alpha_i})^t R_i\}_{i=2,\ldots,n}.$$

▶ $\mathsf{Encrypt}(\vec{Y}, \mathsf{M}, \mathsf{pk})$: the encryption algorithm first picks $s \xleftarrow{\$} \mathbb{Z}_N$ at random and parses $\vec{Y}$ as $\vec{Y} = (y_1, \ldots, y_n)$. It then computes the ciphertext as

$$C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, \qquad C_1 = (g^{\alpha_0} g^{\alpha_1 y_1} g^{\alpha_2 y_2} \cdots g^{\alpha_n y_n})^s, \qquad C_2 = g^s.$$

▶ $\mathsf{Decrypt}(C, \vec{Y}, \mathsf{sk}_{\vec{X}}, \vec{X}, \mathsf{pk})$: computes the message blinding factor as

$$\frac{e(D_1 K_2^{y_2} \cdots K_n^{y_n}, C_2)}{e(C_1, D_0)} = e(g,g)^{\alpha s}.$$

If $\vec{X} \cdot \vec{Y} = 0$, correctness can be verified almost identically to appendix A. Indeed, since $C_2, D_0 \in \mathbb{G}_{p_1}$, we have that the elements in $\mathbb{G}_{p_3}$ from the keys will be canceled out when computing pairing, and the computation is thus exactly the same as that of the selectively secure scheme.

C Proofs of Lemmas 1 and 3

C.1 Proof of Lemma 1

The proof proceeds identically to that of lemma 1 in [31]. The simulator $\mathcal{B}$ is given a DLIN instance $(g, f, \nu, g^{\theta_1}, f^{\theta_2}, \eta \stackrel{?}{=} \nu^{\theta_1+\theta_2})$. It first sets $g = g$, $A_1 = g^{a_1} = f$, $A_2 = g^{a_2} = \nu$ and chooses random exponents $\alpha, b, \delta_v, \delta_{v_1}, \delta_{v_2} \xleftarrow{\$} \mathbb{Z}_p$ to define

$$B = g^b, \quad B_1 = g^{ba_1} = f^b, \quad B_2 = g^{ba_2} = \nu^b, \quad v = g^{\delta_v}, \quad v_1 = g^{\delta_{v_1}}, \quad v_2 = g^{\delta_{v_2}},$$

which allow calculating $\tau_1, \tau_2, \tau_1^b, \tau_2^b$ and $Z = e(g,g)^{\alpha \cdot a_1 \cdot b} = e(g,f)^{\alpha \cdot b}$. Finally, $\mathcal{B}$ picks random group elements $w, h_1, \ldots, h_n \stackrel{\$}{\leftarrow} \mathbb{G}$, which completes the generation of mpk. Since $\mathcal{B}$ knows the entire master secret key $\mathsf{msk} = (g^\alpha, g^{\alpha \cdot a_1}, v, v_1, v_2)$, it is able to generate normal private keys for any vector $\vec{X}$ throughout the game.

To generate the challenge ciphertext for the adversarially-chosen vector $\vec{Y}^\star$, $\mathcal{B}$ first computes a normal ciphertext $(C'_1, \ldots, C'_7, E'_0, E'_1, E'_2, \mathsf{tagc})$ using random exponents $s'_1, s'_2, t' \stackrel{\$}{\leftarrow} \mathbb{Z}_p$. The adversary $\mathcal{A}$ is given

$$C = (C_1, C_2, C_3, C_4, C_5, C_6, C_7, E_0, E_1, E_2, \mathsf{tagc})$$

where

$$E_0 = E'_0 \cdot \big(e(g^{\theta_1}, f) \cdot e(g, f^{\theta_2})\big)^{b \cdot \alpha}, \quad C_1 = C'_1 \cdot (g^{\theta_1})^b, \quad C_2 = C'_2 \cdot (f^{\theta_2})^{-b}$$

$$C_3 = C'_3 \cdot (f^{\theta_2})^{-1}, \quad C_4 = C'_4 \cdot \eta^b, \quad C_5 = C'_5 \cdot \eta,$$

$$C_6 = C'_6 \cdot (g^{\theta_1})^{\delta_v} \cdot (f^{\theta_2})^{-\delta_{v_1}} \cdot \eta^{\delta_{v_2}}, \qquad C_7 = C'_7 \cdot \big((g^{\theta_1})^{\delta_v} \cdot (f^{\theta_2})^{-\delta_{v_1}} \cdot \eta^{\delta_{v_2}}\big)^b$$

and $(E_1, E_2) = (E'_1, E'_2)$. As in [31][lemma 1], the challenge ciphertext C has the distribution of a normal ciphertext (where $s_1 = -\theta_2 + s'_1$, $s_2 = s'_2 + \theta_1 + \theta_2$ and $s = \theta_1 + s'_1 + s'_2$ are the implicitly defined encryption exponents) if $\eta = \nu^{\theta_1 + \theta_2}$. On the other hand, if $\eta \in_R \mathbb{G}$, C has the shape of a semi-functional ciphertext. □

C.2 Proof of Lemma 3

The simulator $\mathcal{B}$ takes in a Decision Bilinear Diffie-Hellman instance $(g, g^{\theta_1}, g^{\theta_2}, g^{\theta_3}, \eta)$ and has to decide whether $\eta = e(g,g)^{\theta_1 \theta_2 \theta_3}$.

To prepare the master public key mpk, $\mathcal{B}$ chooses $a_1, b, \delta_v, \delta_{v_1}, \delta_{v_2}, \delta_w \stackrel{\$}{\leftarrow} \mathbb{Z}_p$. It sets $g = g$, $Z = e(g,g)^{\alpha \cdot a_1 \cdot b} = e(g^{\theta_1}, g^{\theta_2})^{a_1 \cdot b}$, $w = g^{\delta_w}$ and

$$\begin{array}{llll} B = g^b, & A_1 = g^{a_1}, & A_2 = (g^{\theta_2}), & B_1 = g^{ba_1} = f^{a_1}, \\ B_2 = (g^{\theta_2})^b, & v = g^{\delta_v}, & v_1 = g^{\delta_{v_1}}, & v_2 = g^{\delta_{v_2}}, \end{array}$$

which implicitly define $\alpha = \theta_1 \theta_2$, $a_2 = \theta_2$, and chooses $\vec{H} = (h_1, \ldots, h_n) \in \mathbb{G}^n$ at random. In the reduction, $\mathcal{B}$ does not entirely know the master secret key since parts $g^\alpha = g^{\theta_1 \cdot \theta_2}$ and $g^{\alpha \cdot a_1} = g^{\theta_1 \cdot \theta_2 \cdot a_1}$ are not available. Nevertheless, it will be able to compute semi-functional keys (recall that generated ciphertexts and private keys are all semi-functional in Game q and Game $q+1$).

To generate a private key for some vector $\vec{X} = (x_1, \ldots, x_n)$, $\mathcal{B}$ randomly draws $r_1, r_2, z_1, z_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, $\mathsf{tagk}_2, \ldots, \mathsf{tagk}_n \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and $\gamma' \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, sets $r = r_1 + r_2$ and calculates

$$D_1 = (g^{\theta_2})^{-\gamma' \cdot a_1} \cdot v^r, \qquad D_2 = (g^{\theta_2})^{\gamma'} \cdot v_1^r \cdot g^{z_1}, \qquad D_3 = (g^b)^{-z_1}$$

$$D_4 = (g^{\theta_1})^{a_1} \cdot g^{a_1 \cdot \gamma'} \cdot v_2^r \cdot g^{z_2}, \qquad D_5 = (g^b)^{-z_2}, \qquad D_6 = g^{r_2 \cdot b}, \qquad D_7 = g^{r_1},$$

$$K_2 = (h_1^{x_2/x_1} \cdot h_2 \cdot w^{\mathsf{tagk}_2})^{r_1}, \quad \ldots, \quad K_n = (h_1^{x_n/x_1} \cdot h_n \cdot w^{\mathsf{tagk}_n})^{r_1}.$$

The above forms a valid key $\mathsf{sk}_{\vec{X}} = (D_1, \ldots, D_7, K_2, \ldots, K_n, \mathsf{tagk}_2, \ldots, \mathsf{tagk}_n)$ where the variable (which acts as a randomizer making the key semi-functional) is implicitly set to $\gamma = \theta_1 + \gamma'$.

At the challenge phase, $\mathcal{A}$ outputs a vector $\vec{Y}^\star = (y_1^\star, \ldots, y_n^\star)$ and a pair of messages $\mathsf{M}_0, \mathsf{M}_1 \in \mathbb{G}_T$. Then, $\mathcal{B}$ picks $\beta \xleftarrow{\$} \{0,1\}$ and sets $E_0 = \mathsf{M}_\beta \cdot \eta^{a_1 \cdot b}$. Next, it chooses a tag $\mathsf{tagc} \xleftarrow{\$} \mathbb{Z}_p$, encryption exponents $s_1, t \xleftarrow{\$} \mathbb{Z}_p$ and another exponent $\chi' \xleftarrow{\$} \mathbb{Z}_p$ that will be used to implicitly define $\chi = -\theta_3 + \chi'$. It computes

$$C_1 = g^{s_1 \cdot b} \cdot (g^{\theta_3})^b, \quad C_2 = g^{b \cdot a_1 \cdot s_1}, \quad C_3 = g^{a_1 \cdot s_1}, \quad C_4 = (g^{\theta_2})^{\chi' \cdot b}, \quad C_5 = (g^{\theta_2})^{\chi'},$$
$$C_6 = \tau_1^{s_1} \cdot (g^{\theta_3})^{\delta_v} \cdot (g^{\theta_2})^{\delta_{v_2} \cdot \chi'}, \qquad C_7 = (\tau_1^b)^{s_1} \cdot (g^{\theta_3})^{\delta_v \cdot b} \cdot (g^{\theta_2})^{\delta_{v_2} \cdot \chi' \cdot b} \cdot w^{-t}$$

$$E_1 = (h_1^{y_1^\star} \cdots h_n^{y_n^\star} \cdot w^{\mathsf{tagc}})^t, \qquad E_2 = g^t.$$

As in the proof of lemma 3 in [31], if $\eta = e(g,g)^{\theta_1 \theta_2 \theta_3}$, the game mirrors Game q where the encryption exponent s_2 is set to be θ_3. In contrast, if $\eta \in_R \mathbb{G}_T$, the game corresponds to Game $q+1$. □

D Proof of Theorem 3

Proof. Towards a contradiction, we assume there is a co-selective adversary $\mathcal{A}$ with non-negligible advantage and show that it implies an algorithm $\mathcal{B}$ that solves the Decision q-MEBDH problem in $\mathbb{G}$. Algorithm $\mathcal{B}$ is given a q-MEBDH challenge (Z, P). Let $\vec{a} = (a_1, \ldots, a_q)^\top$. It proceeds as follows.

Init. The co-selective security game begins with $\mathcal{A}$ first choosing $V_1, \ldots, V_q$, where $V_k = \mathrm{Aff}(M^{(k)}, \vec{0}_n)$ is an affine subspace corresponds to the k^{th} query.

Setup. First, for each $k = 1, \ldots, q$, it solves a linear equation system with variables $(b_{2,k}, \ldots, b_{n,k})$:

$$(M^{(k)})^\top \vec{b}_k := (M^{(k)})^\top (1, b_{2,k}, \cdots, b_{n,k})^\top = \vec{0}.$$

This can be done since $\mathrm{rank}(M^{(k)}_{(-1)}) = n-1$. It defines a $n \times q$ matrix $B = [\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_q]$, which comprises $\vec{b}_k$ as the k^{th} *column*. It chooses $\vec{\delta} = (\delta_1, \ldots, \delta_n)^\top \xleftarrow{\$} \mathbb{Z}_p^n$ and implicitly defines the vector $\vec{\alpha} = B\vec{a} + B\vec{\delta}$, where $\vec{a} = (a_1, \ldots, a_q)^\top$ is the unknown vector of exponents from the problem instance, by defining public key components as

$$g^{\alpha_1} = g^{a_1 + \cdots + a_q + \delta_1 + \cdots + \delta_q}, \qquad g^{\alpha_1 \vec{\alpha}} = g^{\alpha_1 B \vec{a}} g^{\alpha_1 B \vec{\delta}} = g^{B(a_1 + \cdots + a_q + \delta_1 + \cdots + \delta_q)\vec{a}} g^{\alpha_1 B \vec{\delta}},$$

which are computable from elements $g^{a_i}, g^{a_i a_j}$ that are available from the problem instance, just like $e(g,g)^\alpha$ that completes the public key.

Key Queries. To compute a private key for k^{th} query, $\mathcal{B}$ does as follows. It chooses $t'_k \xleftarrow{\$} \mathbb{Z}_p$ and implicitly defines $t_k = t'_k - \alpha/a_k^2$ by setting

$$D_0 = g^{t'_k} g^{-\alpha/a_k^2}, \qquad D_1 = g^{\alpha + t_k \alpha_1^2} = g^{\alpha + (t'_k - \alpha/a_k^2)\alpha_1^2} = (g^{\alpha_1^2})^{t'_k} \cdot g^{\alpha(1-\alpha_1^2)/a_k^2}$$

as well as

$$\vec{K} = g^{t_k (M^{(k)})^\top \vec{\alpha}} = g^{t_k (M^{(k)})^\top (B\vec{a} + B\vec{\delta})} = g^{t'_k (M^{(k)})^\top B\vec{a}} \cdot g^{-\frac{\alpha (M^{(k)})^\top B\vec{a}}{a_k^2}} \cdot g^{t_k (M^{(k)})^\top B\vec{\delta}}.$$

The term D_1 can be computed from $g^{\alpha/a_k^2}, g^{\alpha a_j^2/a_k^2}$ for $1 \leq j,k \leq n$ from the instance. We then claim that the term a_k does not appear in $(M^{(k)})^\top B\vec{a}$. From this claim, one can see that the middle term $g^{-\alpha(M^{(k)})^\top B\vec{a}/a_k^2}$ (which is the only non-trivial one here) can be computed from the term $g^{\alpha a_j/a_k^2}$ for $j \neq k$ from the instance. Indeed, more importantly, the unknown term g^{α/a_k} is canceled out here. The claim is justified by the fact that the coefficient column vector of a_k is exactly the k^{th} column of $(M^{(k)})^\top B$, which is indeed $(M^{(k)})^\top \vec{b}_k = \vec{0}$.

Challenge. Eventually, $\mathcal{A}$ outputs $\mathsf{M}_0, \mathsf{M}_1 \in \mathbb{G}_T$ along with a vector $\vec{y}^\star$. Recall that $\mathcal{A}$ is required to choose the latter in such a way that $\vec{y}^\star \in V_k$ for all $k \in [1,q]$. Hence, for all $k \in [1,q]$, there must exist $\vec{w}_k \in \mathbb{Z}_p^{n-1}$ such that $\vec{y}^\star = M^{(k)}\vec{w}_k$. To construct the challenge ciphertext, $\mathcal{B}$ flips a coin $\beta \xleftarrow{\$} \{0,1\}$ and chooses $s' \xleftarrow{\$} \mathbb{Z}_p$ to compute

$$C_0 = \mathsf{M}_\beta \cdot Z \cdot e(g,g)^{\alpha s'}, \qquad C_1 = g^{s\alpha_1\langle \vec{y}^\star, \vec{\alpha}\rangle} g^{s'\alpha_1\langle \vec{y}^\star, \vec{\alpha}\rangle} = g^{s\alpha_1 (\vec{y}^\star)^\top B\vec{a}} g^{s\alpha_1(\vec{y}^\star)^\top B\vec{\delta}} g^{s'\alpha_1\langle \vec{y}^\star, \vec{\alpha}\rangle},$$

$$C_2 = g^s g^{s'}, \qquad C_3 = g^{sa_1}\cdots g^{sa_q} \cdot (g^s)^{\delta_1+\cdots+\delta_q} \cdot (g^{\alpha_1})^{s'}.$$

We claim that $g^{s\alpha_1(\vec{y}^\star)^\top B\vec{a}}$ (which is the only non-trivial term in C_1 since elements of the form $g^{sa_i^2}$ are not given) equals 1. The claim follows from the fact that, for each $k \in [1,q]$, the coefficient of a_k in $(\vec{y}^\star)^\top B\vec{a}$ is 0. Indeed, this coefficient of a_k is exactly the k^{th} element of the vector $(\vec{y}^\star)^\top B$, which equals $(\vec{y}^\star)^\top \vec{b}_k$. Since we have $(\vec{y}^\star)^\top \vec{b}_k = (M^{(k)}\vec{w}_k)^\top \vec{b}_k = (\vec{w}_k)^\top (M^{(k)})^\top \vec{b}_k = (\vec{w}_k)^\top \vec{0} = 0$, by our definition of $\vec{b}_k$, the claim is established.

Finally, $\mathcal{A}$ outputs $\beta' \in \{0,1\}$. If $\beta = \beta'$ then $\mathcal{B}$ outputs 1 (meaning $Z = e(g,g)^{\alpha s}$). Otherwise, it outputs 0 (meaning Z is random in $\mathbb{G}_T$).

We easily see that, if $Z \in_R \mathbb{G}_T$, then $\Pr[\mathcal{B}(Z,P) = 0] = \frac{1}{2}$. In contrast, if we have $Z = e(g,g)^{\alpha s}$, then $|\Pr[\mathcal{B}(Z,P) = 0] - \frac{1}{2}| \geq \epsilon$. It follows that $\mathcal{B}$ has advantage at least ϵ in solving q-MEBDH problem.

E Co-Selectively Secure Non-Zero PAIPE in Composite Order Groups

As in appendix B, we consider groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1p_2p_3$. In this setting, we rely on the following three assumptions. The first one and the last one were already used in appendix B whereas the second one is analogous to the one use in [31][Appendix E].

Assumption 1: Given $g \xleftarrow{\$} \mathbb{G}_{p_1}, X_3 \xleftarrow{\$} \mathbb{G}_{p_3}$, and $T \in \mathbb{G}$, decide if $T \in \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$ or $T \in \mathbb{G}_{p_1}$.

Assumption 2: Let $g, w, g^t, X_1 \xleftarrow{\$} \mathbb{G}_{p_1}$ with $t \xleftarrow{\$} \mathbb{Z}_N$, $X_2, Y_2, Z_2 \xleftarrow{\$} \mathbb{G}_{p_2}, X_3, Y_3, Z_3 \xleftarrow{\$} \mathbb{G}_{p_3}$. Given elements $(g, w, g^t, X_1X_2, X_3, Y_2Y_3)$, and $T \in \mathbb{G}$, decide if $T = w^t Z_3$ or $T = w^t Z_2 Z_3$.

Assumption 3: Let $g \xleftarrow{\$} \mathbb{G}_{p_1}, X_2, Y_2, Z_2 \xleftarrow{\$} \mathbb{G}_{p_2}, X_3 \xleftarrow{\$} \mathbb{G}_{p_3}, \alpha, s \xleftarrow{\$} \mathbb{Z}_N$. Given

$$(g, g^\alpha X_2, X_3, g^s Y_2, Z_2),$$

and $T \in \mathbb{G}_T$, decide if $T = e(g,g)^{\alpha s}$ or not.

Using the above assumptions, a relatively simple non-zero PAIPE can then be obtained as follows in the co-selective model.

Construction 8. (Co-selectively secure Non-Zero PAIPE in Composite Order Groups)

▶ $\mathsf{Setup}(n)$: given a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of order $N = p_1p_2p_3$, where $p_i > 2^\lambda$ for each $i \in \{1,2,3\}$. Choose $\alpha \xleftarrow{\$} \mathbb{Z}_N$, $g \xleftarrow{\$} \mathbb{G}_{p_1}$, $X_{p_3} \xleftarrow{\$} \mathbb{G}_{p_3}$, $h_i \xleftarrow{\$} \mathbb{G}_{p_1}$ for $i = 0, \ldots, n$. The master public key is defined to be $\mathsf{pk} := \big(g,\ e(g,g)^\alpha,\ \{h_i\}_{i=0,\ldots,n}\big)$ while the master secret key consists of $\mathsf{msk} := \big(g^\alpha, X_{p_3}\big)$.

▶ $\mathsf{KeyGen}(\vec{X}, \mathsf{msk}, \mathsf{pk})$: on input of a vector $\vec{X} = (x_1, \ldots, x_n) \in \mathbb{Z}_N^n$ and the master private key $\mathsf{msk} = \big(g^\alpha, X_{p_3}\big)$, choose $r \xleftarrow{\$} \mathbb{Z}_N$, $R_3, R_3' \xleftarrow{\$} \mathbb{G}_{p_3}$ and $R_{3,i} \xleftarrow{\$} \mathbb{G}_{p_3}$, for $i = 1$ to n, and compute

$$D_1 = g^\alpha \cdot h_0^r \cdot R_3, \qquad D_2 = g^r \cdot R_3', \qquad \{K_i = \big(h_0^{x_i} \cdot h_i\big)^r \cdot R_{3,i}\}_{i=1}^n$$

before returning $\mathsf{sk}_{\vec{X}} = \big(D_1, D_2, \{K_i\}_{i=1}^n\big)$.

▶ $\mathsf{Encrypt}(\vec{Y}, \mathsf{M}, \mathsf{pk})$: to encrypt $\mathsf{M} \in \mathbb{G}_T$ under $\vec{Y} = (y_1, \ldots, y_n) \in (\mathbb{Z}_N)^n$, choose $s \xleftarrow{\$} \mathbb{Z}_N$ and compute

$$C_0 = \mathsf{M} \cdot e(g,g)^{\alpha \cdot s}, \qquad C_1 = g^s, \qquad C_2 = \big(h_1^{y_1} \cdots h_n^{y_n}\big)^s.$$

The ciphertext is $C = \big(C_0, C_1, C_2\big)$.

▶ $\mathsf{Decrypt}(C, \vec{Y}, \mathsf{sk}_{\vec{X}}, \vec{X}, \mathsf{pk})$: parse $\mathsf{sk}_{\vec{X}}$ as (D_1, D_2, D_3), compute

$$K' = \prod_{i=1}^n K_i^{y_i} = \big(h_0^{\vec{X}\cdot\vec{Y}} \cdot h_1^{y_1} \cdots h_n^{y_n}\big)^r,$$

and compute $e(g,g)^{\alpha \cdot s} = e(D_1, C_1) \cdot \left(\frac{e(K', C_1)}{e(C_2, D_2)}\right)^{-\frac{1}{\vec{X}\cdot\vec{Y}}}$ as well as $\mathsf{M} = C_0 / e(g,g)^{\alpha s}$.

The correctness of the scheme is showed by observing that

$$e(K', C_1) = e(g, h_0)^{r \cdot s \cdot \vec{X}\cdot\vec{Y}} \cdot e(\prod_{i=1}^n h_i^{x_i}, g^{rs}) = e(g, h_0)^{r \cdot s \cdot \vec{X}\cdot\vec{Y}} \cdot e(C_2, D_2)$$

and $e(D_1, C_1) = e(g,g)^{\alpha \cdot s} \cdot e(g, h_0)^{r \cdot s}$.

Lemma 8. *Any adversary distinguishing Game_{Real} from Game_0 with non-negligible advantage contradicts Assumption 1.*

Proof. Let $\mathcal{B}$ be an algorithm that receives (g, X_3, T) and aims at deciding whether $T \in_R \mathbb{G}_{p_1p_2}$ or $T \in_R \mathbb{G}_{p_1}$.

To prepare the public key mpk, $\mathcal{B}$ chooses $\alpha \xleftarrow{\$} \mathbb{Z}_N$ as well as $a_i \xleftarrow{\$} \mathbb{Z}_N$ for $i = 0$ to n. Then, it computes $e(g,g)^\alpha$ and sets $X_{p_3} = X_3$ as well as $h_i = g^{a_i}$ for $i = 0$ to n. Since $\mathcal{B}$ knows $\mathsf{msk} = \big(g^\alpha, X_3\big)$, it is able to answer all key generation queries.

At the challenge step, $\mathcal{A}$ outputs a pair of equal-length messages $\mathsf{M}_0, \mathsf{M}_1 \in \mathbb{G}$ as well as a vector $\vec{Y} = (y_1, \ldots, y_n) \in (\mathbb{Z}_N)^n$. To generate the challenge ciphertext, $\mathcal{B}$ flips a random coin $\beta \xleftarrow{\$} \{0,1\}$ and computes $C_1 = T$, $C_2 = T^{\sum_{i=1}^n a_i y_i}$ and $C_0 = \mathsf{M}_\beta \cdot e(T, g)^\alpha$.

If $T \in_R \mathbb{G}_{p_1p_2}$, the challenge ciphertext has the same distribution as in Game_0 whereas $\mathcal{B}$ is playing Game_{Real} if $T \in_R \mathbb{G}_{p_1}$.

Lemma 9. *For each $k \in \{0, \ldots, q\}$, any PPT distinguisher $\mathcal{A}$ between $Game_k$ from $Game_{k+1}$ can be used to break Assumption 2.*

Proof. We show an algorithm $\mathcal{B}$ that takes as input $\big(g, w, g^t, X_1X_2, X_3, Y_2Y_3, T\big)$, and uses $\mathcal{A}$ to decide whether $T = w^t Z_2 Z_3 \in_R \mathbb{G}_{p_1p_2p_3}$ or $T = w^t Z_3 \in_R \mathbb{G}_{p_1p_3}$.

Init. As in the proof of lemma 5, the co-selective security game begins with the adversary $\mathcal{A}$ announcing the set of private key queries $\vec{X}_1, \ldots, \vec{X}_q$ that she intends to make and we parse as $\vec{X}_k = (x_1, \ldots, x_n)$ the k^{th} of these private key queries.

Setup. To prepare the public key pk, $\mathcal{B}$ uses $\vec{X}_k = (x_1, \ldots, x_n)$. It begins by choosing $\alpha \xleftarrow{\$} \mathbb{Z}_N$ as well as $a_i \xleftarrow{\$} \mathbb{Z}_N$ for $i = 1$ to n. It sets $X_{p_3} = X_3$, $h_0 = w$ and $h_i = h_0^{-x_i} g^{a_i}$ for $i = 1$ to n.

Challenge. At some point, $\mathcal{A}$ chooses messages $\mathsf{M}_0, \mathsf{M}_1$ and a vector $\vec{y} = (y_1, \ldots, y_n)$. Then, $\mathcal{B}$ flips a random coin $\beta \xleftarrow{\$} \{0, 1\}$ and constructs the challenge ciphertext as

$$C_0 = \mathsf{M}_\beta \cdot e(X_1X_2, g)^\alpha, \qquad C_1 = X_1X_2, \qquad C_2 = (X_1X_2)^{\sum_{i=1}^n a_i y_i},$$

which is easily seen to form a semi-functional ciphertext for which $z_c = \sum_{i=1}^n a_i y_i$ (note that this value is taken modp_2 and is thus uncorrelated to the values of $a_i \bmod p_1$) since we must have $\vec{X} \cdot \vec{Y} = 0$ for each private key query $\vec{X}$.

Key Queries. To respond private key queries for vectors $\vec{X} = (x_1, \ldots, x_n)$, $\mathcal{B}$ considers three cases depending on the index j of the query:

[**Case** $i < k$] $\mathcal{B}$ generates a semi-functional key by choosing $r, z_1 \xleftarrow{\$} \mathbb{Z}_N$ as well as $R_3 \xleftarrow{\$} \mathbb{G}_{p_3}$ and $R_{3,i} \xleftarrow{\$} \mathbb{G}_{p_3}$ for $i = 1$ to n and computing

$$D_1 = g^\alpha h_0^r \cdot (Y_2Y_3)^{z_1} \qquad D_2 = g^r \cdot R_3 \qquad \{K_i = (h_0^{x_i} \cdot h_i)^r \cdot R_{3,i}\}_{i=1}^n.$$

[**Case** $i > k$] $\mathcal{B}$ computes a normal key using $\mathsf{msk} = \big(g^\alpha, X_3\big)$.

[**Case** $i = k$] if $i = k$, $\mathcal{B}$ uses the input element T. Namely, it chooses $R_3 \xleftarrow{\$} \mathbb{G}_{p_3}$ and $R_{3,i} \xleftarrow{\$} \mathbb{G}_{p_3}$ for $i = 1, \ldots, n$. Then, it computes

$$D_1 = g^\alpha \cdot T, \qquad D_2 = g^t \cdot R_3' \qquad \{K_i = (g^t)^{a_i} \cdot R_{3,i}\}_{i=1}^n.$$

We easily observe that the challenger is playing Game_k if $T = w^t Z_3$. If it turns out that $T = w^t Z_2 Z_3$, $\mathcal{B}$ is rather playing Game_{k+1} since T.

Lemma 10. *Any PPT algorithm $\mathcal{A}$ distinguishing $Game_q$ from $Game_{q+1}$ implies a distinguisher $\mathcal{B}$ for Assumption 3.*

Proof. We outline an algorithm $\mathcal{B}$ that takes as input $\big(g, g^\alpha X_2, X_3, g^s Y_2, Z_2, T\big)$ with the aim of deciding whether $T = e(g, g)^{\alpha s}$ or $T \in_R \mathbb{G}_T$ using its interaction with $\mathcal{A}$. To this end, $\mathcal{B}$ generates the public key $\mathsf{mpk} = \big(g, e(g, g)^\alpha, \{h_i\}_{i=0,\ldots,n}\big)$ by choosing $a_0, \ldots, a_n \xleftarrow{\$} \mathbb{Z}_N$ and setting $X_{p_3} = X_3$, $e(g, g)^\alpha = e(g^\alpha X_2, g)$ as well as $h_i = g^{a_i}$ for $i = 0$ to n.

When the adversary $\mathcal{A}$ makes a private key query $\vec{X} = (x_1, \ldots, x_n)$, $\mathcal{B}$ chooses $r, w \xleftarrow{\$} \mathbb{Z}_N$, $R_3, R_3' \xleftarrow{\$} \mathbb{G}_{p_3}$, $R_{3,i} \xleftarrow{\$} \mathbb{G}_{p_3}$, for $i = 1$ to n, and computes

$$D_1 = (g^\alpha X_2) \cdot h_0^r \cdot R_3, \qquad D_2 = g^r \cdot R_3' \qquad \{K_i = (h_0^{x_i} \cdot h_i)^r \cdot R_{3,i}\}_{i=1}^n$$

which has the distribution of a semi-functional key.

At the challenge phase, $\mathcal{A}$ outputs $\mathsf{M}_0, \mathsf{M}_1 \in \mathbb{G}_T$ and $\vec{Y} = (y_1, \ldots, y_n)$. To construct the challenge ciphertext, $\mathcal{B}$ chooses $\beta \stackrel{\$}{\leftarrow} \{0,1\}$ and computes

$$C_0 = \mathsf{M}_\beta \cdot T, \qquad C_1 = g^s Y_2, \qquad C_2 = (g^s Y_2)^{\sum_{i=1}^n a_i y_i}.$$

Similarly to the proof of lemma 8 in [23], the game is easily seen to correspond to Game_q if $T = e(g,g)^{\alpha s}$ and to Game_{q+1} if $T \in_R \mathbb{G}_T$.

In Game_{q+1}, the adversary's advantage is easily seen to be zero since the challenge ciphertext carries no information on $\beta \in \{0,1\}$.