Large Universe Subset Predicate Encryption Based on Static Assumption (without Random Oracle)

Sanjit Chatterjee and Sayantan Mukherjee

Department of Computer Science and Automation,
Indian Institute of Science, Bangalore
{sanjit,sayantanm}@iisc.ac.in

Abstract. In a recent work, Katz et al. (CANS'17) generalized the notion of Broadcast Encryption to define Subset Predicate Encryption (SPE) that emulates *subset containment* predicate in the encrypted domain. They proposed two selectively secure constructions of SPE in the small universe setting. Their first construction is based on a parameterized assumption while the second one is based on DBDH. Both achieve constant-size secret key while the ciphertext size depends on the size of the privileged set. They also showed some black-box transformation of SPE to well-known primitives like WIBE and ABE to establish the richness of the SPE structure.
This work investigates the question of large universe realization of SPE scheme based on static assumption in the standard model. We propose two constructions both of which achieve constant-size secret key. First construction SPE_1, instantiated in composite order bilinear groups, achieves constant-size ciphertext and is proven secure in a restricted version of selective security model under the subgroup decision assumption (SDP). Our main construction SPE_2 is adaptively secure in the prime order bilinear group under the symmetric external Diffie-Hellman assumption (SXDH). Thus SPE_2 is the first large universe instantiation of SPE to achieve adaptive security without random oracle. Both of our constructions have efficient decryption function suggesting their practical applicability. Thus primitives like WIBE and ABE resulting through the black-box transformation of our constructions become more practical.

1 Introduction

The notion of Identity-Based Encryption (IBE) [8] was generalized by Katz et al. [24] to Predicate Encryption (PE). Predicate encryption emulates a predicate function $R : \mathcal{X} \times \mathcal{Y} \rightarrow \{0,1\}$ in the encrypted domain in the following sense. A key SK associated with key-index x can decrypt a ciphertext CT associated with data-index y if $R(\mathsf{x},\mathsf{y}) = 1$. In such a generalized view, IBE evaluates an equality predicate. Attribute-Based Encryption (ABE) [19] is another example of predicate encryption that emulates boolean function in the encrypted domain. One can view Broadcast Encryption (BE) [9] as a simpler form of ABE where the disjunction predicate is evaluated in the form of membership checking.

Katz et al. [23] recently introduced a new primitive called Subset Predicate Encryption (SPE) that allows checking for *subset containment* in the encrypted domain. Let $\mathcal{ID}$ be the set of identities. Then, in an SPE, a key SK associated with a key-index set $\Omega \subset \mathcal{ID}$ can decrypt a ciphertext CT associated with data-index set $\Theta \subset \mathcal{ID}$ if $\Omega \subseteq \Theta$. There is an obvious connection between BE and SPE in the sense that both encrypt for a privileged set Θ. However, unlike BE, the KeyGen in SPE takes a set of identities Ω as input. In terms of functionality, it is trivial to achieve *subset containment* through multiple *membership testing* as $\Omega \subseteq \Theta \iff (i \in \Omega \Rightarrow i \in \Theta)$.

Thus, one may be tempted to use an efficient BE instantiation [9] to construct a small universe (i.e. $|\mathcal{ID}| = \mathsf{poly}(\lambda)$ for security parameter λ) SPE. In such an instantiation, KeyGen of SPE would simply be a concatenation of output of KeyGen of BE for each $x \in \Omega$ i.e. $\mathsf{SK}_\Omega = (\mathsf{SK}_{x_1}, \ldots, \mathsf{SK}_{x_{\mathfrak{k}}})$ where $\Omega = (x_1, \ldots, x_{\mathfrak{k}})$. However, such a realization of SPE suffers from an obvious security issue. Given a ciphertext CT_Θ, an unprivileged set Ω (having secret key SK_Ω for $\Omega \nsubseteq \Theta$) can easily derive a valid key by stripping SK_Ω as long as $\Omega \cap \Theta \neq \phi$.

In their work, Katz et al. [23] discussed and then ruled out a few generic techniques to construct small universe SPE from Inner-Product Encryption (IPE) [24], Wildcard Identity-Based Encryption (WIBE) [1] and Fuzzy Identity-Based Encryption (FIBE) [26] due to the reason of inefficiency. They proposed two dedicated SPE constructions in the small universe setting. Both of their constructions have constant-size secret key while the ciphertext size depends on the cardinality of the privileged set it is intended to. Informally speaking, their first construction utilized the *exponent inversion* technique [11] and the second one utilized the *commutative blinding* technique [7]. However, both the constructions were proven only *selectively secure*. The security of the first construction is based on a non-static assumption (q-BDHI) whereas the security of the second construction is based on a static assumption (DBDH). In fact, the second construction can be lifted to be selectively secure in the large universe setting, but the proof requires the random oracle assumption.

Given the above results of [23], the main open question in the context of SPE is the following. Can we realize an adaptively secure SPE in the large universe setting under some static assumption without random oracle? In this paper, we answer this question in the affirmative. In addition, we also ask whether one can achieve an SPE with constant-size ciphertext. On this front, this paper reports some partial success through a trade-off in the security model.

We start with a rather obvious observation. Recall the connection between SPE in the small universe and public key broadcast encryption mentioned above. In a similar vein, Identity-Based Broadcast Encryption (IBBE) can be seen as a special case of large universe SPE. In fact, the KeyGen of IBBE can be considered to be a special case of the KeyGen of SPE that always takes a singleton set as input. However, trivially extending the KeyGen of IBBE to that of SPE may still be problematic. The security model of IBBE has a natural restriction that the intersection of challenge identity set and the set of identities compromised in the key extraction phase must be *null*. On the other hand, corresponding natural restriction in the context of SPE would be that none of the set of identities (here we call key-index set) queried in the key extraction phase should be a subset of the challenge identity set (here we call data-index set). In fact, the security model of SPE introduces two *new issues* due to aforementioned involved structure of SPE:

- $\mathscr{P}1$: the challenge data-index set might share a few elements with the queried key-index sets.
- $\mathscr{P}2$: the queried key-index sets might have a non-empty intersection among themselves.

We will discuss in details the relevance of $\mathscr{P}1$ and $\mathscr{P}2$ in the context of the security argument of our first construction.

Our first construction (SPE_1) structurally is a descendant of IBBE by Delerablée [15]. Note that, Delerablée's IBBE was a constant-size ciphertext construction and was proven selectively secure under a q-type parameterized assumption in the random oracle model. Recently, Gong et al. [18] proposed integration of [15] and Déjà Q [30] towards selectively secure IBBE with constant-size ciphertext under static subgroup decision assumptions in the standard model. Recall that, the IBBE KeyGen encodes a single identity whereas the SPE KeyGen encodes a set Ω into a secret key of constant size. However, we show that the KeyGen of [18] can be tweaked appropriately to generate a constant-size secret key corresponding to a set. This observation leads to our first construction SPE_1, a constant-size ciphertext SPE in the large universe setting without the random oracle assumption.

The security reduction of SPE_1 uses Déjà Q framework just as was done in [18]. However, we have to address the above mentioned issues ($\mathscr{P}1$ and $\mathscr{P}2$) that stem from more involved structure of SPE_1. Here we report a complete success against both the issues but in a weaker version of security settings. In some sense, this is the cost we had to bear to construct the first large-universe SPE with constant-size ciphertext. Here we would like to point out that pairing-based construction of an adaptively secure IBBE achieving constant-size secret key and constant-size ciphertext still remains an open problem.

Our main construction (SPE_2) achieves adaptive security in the prime order groups under the SXDH assumption with constant-size secret key. This construction resembles the IBBE structure of [25] which extended JR-IBE [22] to achieve an efficient tag-based IBBE construction. We tweak the KeyGen algorithm of their IBBE_1 [25] to realize adaptive secure SPE in the large universe setting. Again, the non-triviality lies in the security argument. Precisely, in the security model of IBBE_1 [25], for a challenge set $\Theta^* = (y_1, \ldots, y_\ell)$, the set of identities queried for key extraction should be strictly non-overlapping. In the security argument of (SPE_2) however, no such restriction is imposed. In particular, the query (Ω) adversary makes, may contain some elements that also belong to the challenge set Θ^*.

Here we mention that, our second construction handles both the earlier mentioned issues completely that too in the adaptive security model. The KeyGen in SPE_2 introduces enough randomness in the secret key to handle the second issue $\mathscr{P}2$ completely. We are thus able to realize the first large universe adaptively secure SPE without random oracle assumption. Our construction is quite efficient in terms of parameter size, encryption and decryption cost. For example, the encryption does not require any pairing evaluation while the decryption evaluates only 3 pairings. Only limitation on this construction is obvious: the ciphertext size depends on the size of the privileged set it is intended to.

We conclude with a brief discussion on the effect of the black-box transformations due to Katz et al. [23] on our SPE_2 constructions. We achieve the first adaptively secure CP-DNF (CP-ABE with DNF policy) evaluation with a constant-size secret key.

Organization of the Paper. In Section 2 we recall a few definitions and present the notations that will be followed in this paper. In Section 3 we define Subset Predicate Encryption (SPE) and its security model. In Sections 4 and 5, we present two SPE constructions along with their proofs. Section 6 concludes this paper.

2 Preliminaries

Notations. Here we denote $[a, b] = \{i \in \mathbb{N} : a \leq i \leq b\}$ and for any $n \in \mathbb{N}$, $[n] = [1, n]$. The security parameter is denoted by 1^λ where $\lambda \in \mathbb{N}$. By $s \hookleftarrow S$ we denote a uniformly random choice s from S. By $\mathfrak{P}(S)$ we denote the power set of set S. We use $A \approx_\epsilon B$ to denote that A and B are computationally indistinguishable such that for any PPT adversary $\mathcal{A}$, $|\mathsf{Pr}[\mathcal{A}(A) \to 1] - \mathsf{Pr}[\mathcal{A}(B) \to 1]| \leq \epsilon$ where $\epsilon \leq \mathsf{neg}(\lambda)$ for $\mathsf{neg}(\lambda)$ denoting negligible function. We use $\mathsf{Adv}^{\mathit{\Pi}}_{\mathcal{A},M}(\lambda)$ to denote the advantage adversary $\mathcal{A}$ has against protocol $\mathit{\Pi}$ in security model M and $\mathsf{Adv}^{\mathsf{HP}}_{\mathcal{A}}(\lambda)$ is used to denote the advantage of $\mathcal{A}$ to solve the hard problem HP.

2.1 Bilinear Groups

This paper presents two subset predicate encryption schemes. The first construction is instantiated in the composite order symmetric bilinear groups whereas the second one is instantiated in the prime order asymmetric bilinear groups.

Composite Order Bilinear Pairings. A composite order symmetric bilinear group generator $\mathcal{G}_{\mathsf{sbg}}$, apart from security parameter 1^λ takes an additional parameter n and returns $(n+3)$-tuple $(p_1, \cdots, p_n, \mathsf{G}, \mathsf{G}_{\mathrm{T}}, e)$ where both $\mathsf{G}, \mathsf{G}_{\mathrm{T}}$ are cyclic groups of order $N = \prod_{i \in [n]} p_i$ where all p_i are large primes and $e : \mathsf{G} \times \mathsf{G} \to \mathsf{G}_{\mathrm{T}}$ is an admissible, non-degenerate bilinear pairing [8, 10]. Here, G_{p_i} denotes a subgroup of G of order p_i. This notation is naturally extended to $\mathsf{G}_{p_i \cdots p_j}$ denoting a subgroup of G of order $p_i \times \cdots \times p_j$. By convention $g_{i \cdots j}$ is an element of subgroup $\mathsf{G}_{p_i \cdots p_j}$. It is evident that $e(h_i, h_j) = 1$ if $i \neq j$ for $h_i \in \mathsf{G}_{p_i}$ and $h_j \in \mathsf{G}_{p_j}$. This is called the *orthogonality* property of the composite order bilinear pairing.

Prime Order Bilinear Pairings. The prime order asymmetric bilinear group generator $\mathcal{G}_{\mathsf{abg}}$, takes security parameter 1^λ and returns a 5 tuple $(p, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_{\mathrm{T}}, e)$ where all of $\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_{\mathrm{T}}$ are cyclic groups of order large prime p and $e : \mathsf{G}_1 \times \mathsf{G}_2 \to \mathsf{G}_{\mathrm{T}}$ is an admissible, non-degenerate bilinear pairing [17].

2.2 Hardness Assumptions

Composite Order Setting. Let $(p_1, p_2, p_3, \mathsf{G}, \mathsf{G}_{\mathrm{T}}, e) \leftarrow \mathcal{G}_{\mathsf{sbg}}(1^\lambda, 3)$ be the output of symmetric bilinear group generator where both $\mathsf{G}, \mathsf{G}_{\mathrm{T}}$ are cyclic groups of order $N = p_1p_2p_3$ where p_1, p_2, p_3 are large primes. We define two variants of subgroup decision problems [30] as follows:

$\mathsf{SD1}$: The $\mathsf{SD1}$ problem in group G is defined as following.

Given $g_1 \hookleftarrow \mathsf{G}_{p_1}, g_3 \hookleftarrow \mathsf{G}_{p_3}, g_{12} \hookleftarrow \mathsf{G}_{p_1p_2}$, $Z \hookleftarrow \mathsf{G}_{p_1p_2}$; decide if $Z \in \mathsf{G}_{p_1}$ or $Z \in \mathsf{G}_{p_1p_2}$.

The advantage of a probabilistic polynomial time algorithm $\mathcal{A}$ to solve $\mathsf{SD1}$ is $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SD1}}(\lambda) = |\mathsf{Pr}[\mathcal{A}(g_1, g_3, g_{12}, Z) = 1 : Z \in \mathsf{G}_{p_1}] - \mathsf{Pr}[\mathcal{A}(g_1, g_3, g_{12}, Z) = 1 : Z \in \mathsf{G}_{p_1p_2}]|$ where the probability is calculated over the random choice of $g_1 \in \mathsf{G}_{p_1}$, $g_3 \in \mathsf{G}_{p_3}$, $g_{12} \in \mathsf{G}_{p_1p_2}$, $Z \in \mathsf{G}_{p_1p_2}$ as well as the random bits used by $\mathcal{A}$. The $\mathsf{SD1}$ assumption holds if for any probabilistic polynomial time algorithm $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SD1}}(\lambda) \leq \mathsf{neg}(\lambda)$.

$\mathsf{SD2}$: The $\mathsf{SD2}$ problem in group G is defined as following.

Given $g_1 \hookleftarrow \mathsf{G}_{p_1}, g_3 \hookleftarrow \mathsf{G}_{p_3}, g_{12} \hookleftarrow \mathsf{G}_{p_1p_2}, g_{23} \hookleftarrow \mathsf{G}_{p_2p_3}$, $Z \hookleftarrow \mathsf{G}$; decide if $Z \in \mathsf{G}_{p_1p_3}$ or $Z \in \mathsf{G}$.

The advantage of a probabilistic polynomial time algorithm $\mathcal{A}$ to solve $\mathsf{SD2}$ is $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SD2}}(\lambda) = |\mathsf{Pr}[\mathcal{A}(g_1, g_3, g_{12}, g_{23}, Z) = 1 : Z \in \mathsf{G}_{p_1p_3}] - \mathsf{Pr}[\mathcal{A}(g_1, g_3, g_{12}, g_{23}, Z) = 1 : Z \in \mathsf{G}]|$ where the probability is calculated over the random choice of $g_1 \in \mathsf{G}_{p_1}$, $g_3 \in \mathsf{G}_{p_3}$, $g_{12} \in \mathsf{G}_{p_1p_2}$, $g_{23} \in \mathsf{G}_{p_2p_3}$, $Z \in \mathsf{G}$ as well as the random bits used by $\mathcal{A}$. The $\mathsf{SD2}$ assumption holds if for any probabilistic polynomial time algorithm $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SD2}}(\lambda) \leq \mathsf{neg}(\lambda)$.

Prime Order Setting. Let $(p, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_{\mathrm{T}}, e) \leftarrow \mathcal{G}_{\mathsf{abg}}(1^\lambda)$ be the output of asymmetric bilinear group generator where $\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_{\mathrm{T}}$ are cyclic groups of order p that is a large prime.

Symmetric External Diffie-Hellman Assumption (SXDH). The Symmetric External Diffie-Hellman assumption was introduced by Ateniese et al. in [3]. This assumption is defined based on decisional Diffie-Hellman assumption (DDH) on both the source groups G_1 and G_2.

$\mathsf{DDH}_{\mathsf{G}_1}$: The decisional Diffie-Hellman problem (DDH) in group G_1 is defined as following.

Given $g_1, g_2, g_1^b, g_1^{bs}, Z = g_1^{s+\hat{s}} \in \mathsf{G}_1$ where g_1 is a generator of G_1, g_2 is a generator of G_2 and $b, s \hookleftarrow \mathbb{Z}_p$; decide if $\hat{s} = 0$ or $\hat{s} \in \mathbb{Z}_p^*$.

The advantage of a probabilistic polynomial time algorithm $\mathcal{A}$ to solve $\mathsf{DDH}_{\mathsf{G}_1}$ is $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}_{\mathsf{G}_1}}(\lambda) = |\mathsf{Pr}[\mathcal{A}(g_1, g_2, g_1^b, g_1^{bs}, Z) = 1 : Z = g_1^s] - \mathsf{Pr}[\mathcal{A}(g_1, g_2, g_1^b, g_1^{bs}, Z) = 1 : Z = g_1^{s+\hat{s}}, \hat{s} \hookleftarrow \mathbb{Z}_p^*]|$ where the probability is calculated over the random choice of $g_1 \in \mathsf{G}_1$, $g_2 \in \mathsf{G}_2$, $b, s \in \mathbb{Z}_p$ as well as the random bits used by $\mathcal{A}$. The $\mathsf{DDH}_{\mathsf{G}_1}$ assumption holds if for any probabilistic polynomial time algorithm $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}_{\mathsf{G}_1}}(\lambda) \leq \mathsf{neg}(\lambda)$.

$\mathsf{DDH}_{\mathsf{G}_2}$: The decisional Diffie-Hellman problem (DDH) in group G_2 is defined as following.

Given $g_1, g_2, g_2^c, g_2^r, Z = g_2^{cr+\hat{r}} \in \mathsf{G}_2$ where g_1 is a generator of G_1, g_2 is a generator of G_2 and $c, r \hookleftarrow \mathbb{Z}_p$; decide if $\hat{r} = 0$ or $\hat{r} \in \mathbb{Z}_p^*$.

The advantage of a probabilistic polynomial time algorithm $\mathcal{A}$ to solve $\mathsf{DDH}_{\mathsf{G}_2}$ is $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}_{\mathsf{G}_2}}(\lambda) = \mathsf{Pr}[\mathcal{A}(g_1, g_2, g_2^c, g_2^r, Z) = 1 : Z = g_2^{cr}] - \mathsf{Pr}[\mathcal{A}(g_1, g_2, g_2^c, g_2^r, Z) = 1 : Z = g_2^{cr+\hat{r}}, \hat{r} \hookleftarrow \mathbb{Z}_p^*]$ where the probability is calculated over the random choice of $g_1 \in \mathsf{G}_1$, $g_2 \in \mathsf{G}_2$, $c, r \in \mathbb{Z}_p$ as well as the random bits used by $\mathcal{A}$. The $\mathsf{DDH}_{\mathsf{G}_2}$ assumption holds if for any probabilistic polynomial time algorithm $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}_{\mathsf{G}_2}}(\lambda) \leq \mathsf{neg}(\lambda)$.

Note that, in the above description, $\mathsf{DDH}_{\mathsf{G}_1}$ and $\mathsf{DDH}_{\mathsf{G}_2}$ are presented in different manner. The representation of $\mathsf{DDH}_{\mathsf{G}_1}$ given here is called the 1-Lin problem which in reality is equivalent to the standard form of DDH in G_1.

3 Subset Predicate Encryption

We next give the definition of Subset Predicate Encryption (SPE) in terms of a predicate encryption [24] and formally model its security requirement.

3.1 Definition

Let $\mathcal{ID}$ be the identity space. For a key-index set $\Omega \in \mathcal{X} \subset \mathfrak{P}(\mathcal{ID})$ and a data-index set $\Theta \in \mathcal{Y} \subset \mathfrak{P}(\mathcal{ID})$, the predicate function for SPE is

$$\mathsf{R}_s(\Omega, \Theta) = \begin{cases} 1 & \text{if } \Omega \subseteq \Theta \\ 0 & \text{otherwise} \end{cases}.$$

The following description of an SPE scheme is presented here as a Key-Encapsulation Mechanism (KEM) where $\mathcal{C}$, $\mathcal{SK}$ and $\mathcal{K}$ denote ciphertext space, secret key space, and encapsulation key space respectively.

- Setup: It takes $m \in \mathbb{N}$ along with security parameter 1^λ. It outputs a master secret key msk and corresponding public key mpk.
- KeyGen: It takes mpk, msk and a key-index set $\Omega \in \mathcal{X}$ of size $k \leq m$ as input. It generates a secret key $\mathsf{SK} \in \mathcal{SK}$ corresponding to key-index set Ω.
- $\mathsf{Encrypt}$: It takes mpk, a data-index set $\Theta \in \mathcal{Y}$ of size $\ell \leq m$ as input. It generates the encapsulation key $\kappa \in \mathcal{K}$ and the ciphertext $\mathsf{CT} \in \mathcal{C}$.
- Test: It takes (SK, Ω) and (CT, Θ) as input. Outputs κ or $\perp$.

Correctness. For all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all key-index set $\Omega \in \mathcal{X}$, all $\mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \Omega)$, all data-index set $\Theta \in \mathcal{Y}$, all $(\kappa, \mathsf{CT}) \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \Theta)$,

$$\mathsf{Decrypt}(\mathsf{mpk}, (\mathsf{SK}, \Omega), (\mathsf{CT}, \Theta)) = \begin{cases} \kappa & \text{if } \mathsf{R}_s(\Omega, \Theta) = 1 \\ \perp & \text{otherwise} \end{cases}.$$

Remark 1. The Setup algorithms takes an additional parameter m along with the security parameter λ. This is because, both our constructions are large universe constructions. The cardinality of the sets processed in ciphertext generation and key generation in both of our constructions will be upper bounded by m like any other available standard model large universe construction [5, 25].

3.2 Security Definition

Adaptive CPA-Security of SPE. The $\mathsf{IND\text{-}ID\text{-}CPA}$ security game of SPE is defined between challenger $\mathcal{C}$ and adversary $\mathcal{A}$ as following:

- **Setup**: The challenger $\mathcal{C}$ gives mpk to the adversary $\mathcal{A}$ and keeps msk as secret.
- **Query Phase-I**: Given a key-index set Ω, $\mathcal{C}$ returns $\mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \Omega)$.
- **Challenge**: $\mathcal{A}$ provides challenge data-index Θ^* (such that $\mathsf{R}_s(\Omega, \Theta^*) = 0$ for all previous key queries). $\mathcal{C}$ then generates $(\kappa_0, \mathsf{CT}) \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \Theta^*)$ and chooses $\kappa_1 \hookleftarrow \mathcal{K}$. It returns $(\mathsf{CT}, \kappa_{\mathfrak{b}})$ to adversary for $\mathfrak{b} \hookleftarrow \{0, 1\}$.
- **Query Phase-II**: Given a key-index Ω such that $\mathsf{R}_s(\Omega, \Theta^*) = 0$, $\mathcal{C}$ returns $\mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \Omega)$.
- **Guess**: Finally $\mathcal{A}$ outputs its guess $\mathfrak{b}' \in \{0, 1\}$ and wins if $\mathfrak{b} = \mathfrak{b}'$.

For any adversary $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\mathcal{A},\mathsf{SPE}}(\lambda) = |\mathsf{Pr}[\mathfrak{b} = \mathfrak{b}'] - 1/2|.$$

We say SPE is $\mathsf{IND\text{-}ID\text{-}CPA}$ secure if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\mathcal{A},\mathsf{SPE}}(\lambda) \leq \mathsf{neg}(\lambda)$. If there is an **Init** phase before the **Setup** where the adversary $\mathcal{A}$ commits to the challenge data-index set Θ^*, we call such model selective $\mathsf{IND\text{-}ID\text{-}CPA}$ or $\mathsf{IND\text{-}sID\text{-}CPA}$ security model.

4 SPE_1: Realizing Constant Size Ciphertext

We present the first SPE construction having constant-size secret key and constant-size ciphertext in the composite order pairing setting.

4.1 Construction

SPE_1 is defined by the following four algorithms.

- $\mathsf{Setup}(1^\lambda, m)$: The symmetric bilinear group generator outputs $(p_1, p_2, p_3, \mathsf{G}, \mathsf{G}_\mathrm{T}, e) \leftarrow \mathcal{G}_{\mathsf{sbg}}(1^\lambda, 3)$ where both $\mathsf{G}, \mathsf{G}_\mathrm{T}$ are cyclic groups of order $N = p_1 p_2 p_3$. Then pick $\alpha, \beta \hookleftarrow N$, generators $g_1, u \hookleftarrow \mathsf{G}_{p_1}$ and $g_3 \hookleftarrow \mathsf{G}_{p_3}$. Choose $R_{3,i} \hookleftarrow \mathsf{G}_{p_3}$ for all $i \in [m]$. Define the $\mathsf{msk} = (\alpha, \beta, u, g_3)$ and the public parameter is
$$\mathsf{mpk} = (g_1, g_1^\beta, \left(G_i = g_1^{\alpha^i}, U_i = u^{\alpha^i} \cdot R_{3,i}\right)_{i \in [m]}, e(g_1, u)^\beta, \mathsf{H})$$
where $\mathsf{H} : \mathsf{G}_\mathrm{T} \to \mathcal{K}$ is a universal hash function for encapsulation key space $\mathcal{K} = \{0,1\}^n$ where $n = \mathsf{poly}(\lambda)$.
- $\mathsf{KeyGen}(\mathsf{msk}, \Omega)$: Given a key-index set Ω, such that $|\Omega| = \Bbbk \leq m$; define the polynomial $P_\Omega(z) = \prod_{x \in \Omega}(z + x) = d_0 + d_1 z + d_2 z^2 + \ldots + d_\Bbbk z^\Bbbk$, pick $Y_3 \hookleftarrow \mathsf{G}_{p_3}$ and define secret key as
$$\mathsf{SK}_\Omega = u^{\frac{\beta}{P_\Omega(\alpha)}} \cdot Y_3 = u^{\frac{\beta}{\prod_{x \in \Omega}(\alpha + x)}} \cdot Y_3.$$
- $\mathsf{Encrypt}(\mathsf{mpk}, \Theta)$: Given a data-index set Θ, such that $|\Theta| = \ell \leq m$; the polynomial $P_\Theta(z) = \prod_{y \in \Theta}(z + y) = c_0 + c_1 z + c_2 z^2 + \ldots + c_\ell z^\ell$. Choose $s \hookleftarrow \mathbb{Z}_p$ and compute κ and $\mathsf{CT}_\Theta = (\mathsf{C}_0, \mathsf{C}_1)$ such that
$$\kappa = \mathsf{H}(e(g_1, u)^{s\beta}), \mathsf{C}_0 = g_1^{s\beta}, \mathsf{C}_1 = g_1^{sP_\Theta(\alpha)} = \left(g_1^{c_0} \prod_{i \in [\ell]} G_i^{c_i}\right)^s.$$
- $\mathsf{Decrypt}((\mathsf{SK}_\Omega, \Omega), (\mathsf{CT}_\Theta, \Theta))$: As $\Omega \subseteq \Theta$, compute $P_{\Theta \setminus \Omega}(\alpha) = \prod_{w \in \Theta \setminus \Omega}(\alpha + w)$ $= a_0 + a_1 \alpha + a_2 \alpha^2 + \ldots + a_t \alpha^t$ where $t = |\Theta \setminus \Omega|$. Then compute $\kappa = \mathsf{H}((B/A)^{1/a_0})$ where
$$A = e(\mathsf{C}_0, \prod_{i \in [t]} U_i^{a_i}), B = e(\mathsf{C}_1, \mathsf{SK}_\Omega).$$

Correctness. Notice that, due to orthogonality property,
$$A = e(\mathsf{C}_0, \prod_{i \in [t]} U_i^{a_i}) = e(g_1^{s\beta}, u^{P_{\Theta \setminus \Omega}(\alpha) - a_0} \prod_{i \in [t]} R_{3,i}^{a_i}) = e(g_1, u)^{s\beta\left(P_{\Theta \setminus \Omega}(\alpha) - a_0\right)},$$
$$B = e(\mathsf{C}_1, \mathsf{SK}_\Omega) = e(g_1^{sP_\Theta(\alpha)}, u^{\frac{\beta}{P_\Omega(\alpha)}} \cdot Y_3) = e(g_1, u)^{s\beta P_{\Theta \setminus \Omega}(\alpha)}.$$

Then, $B/A = e(g_1, u)^{s\beta a_0}$, $\mathsf{H}((B/A)^{1/a_0}) = \mathsf{H}(e(g_1, u)^{s\beta}) = \kappa$.

4.2 Security

Recall that SPE_1 has its root in the Gong et al. [18] modification of the IBBE construction of Delerablée [15]. Following [18], we have applied Déjà Q [12, 30] to prove the security of SPE_1 in the standard model. However, the Déjà Q based argument requires further restriction on the adversary in terms of the key extraction queries and we elaborate on this point first.

Recall that, in case of SPE, the adversary $\mathcal{A}$ makes key extraction queries on sets. Thus, unlike IBBE, the same identity can be present in more than one distinct key extraction queries. Suppose $\mathcal{A}$ makes key extraction queries on $\Omega' = \{\mathsf{x}_1, \mathsf{x}_2\}$, $\Omega'' = \{\mathsf{x}_2, \mathsf{x}_3\}$ and $\Omega''' = \{\mathsf{x}_1, \mathsf{x}_3\}$ where $\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3 \in \mathcal{ID}$. Note that, the corresponding secret keys in SPE_1 are $\mathsf{SK}_{\Omega'} = u^{\frac{1}{(\alpha+\mathsf{x}_1)(\alpha+\mathsf{x}_2)}} \cdot X_{3,1}$, $\mathsf{SK}_{\Omega''} = u^{\frac{1}{(\alpha+\mathsf{x}_2)(\alpha+\mathsf{x}_3)}} \cdot X_{3,2}$ and $\mathsf{SK}_{\Omega'''} = u^{\frac{1}{(\alpha+\mathsf{x}_1)(\alpha+\mathsf{x}_3)}} \cdot X_{3,3}$. Then, for $\Omega = \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3\}$,

$$\mathsf{SK}_\Omega = u^{\frac{1}{(\alpha+\mathsf{x}_1)(\alpha+\mathsf{x}_2)(\alpha+\mathsf{x}_3)}} = \left(\frac{\mathsf{SK}_{\Omega'}}{\mathsf{SK}_{\Omega''}}\right)^{(\mathsf{x}_3-\mathsf{x}_1)^{-1}} = \left(\frac{\mathsf{SK}_{\Omega'}}{\mathsf{SK}_{\Omega'''}}\right)^{(\mathsf{x}_3-\mathsf{x}_2)^{-1}} \quad (1)$$

Precisely, given any two elements from the set $\{\mathsf{SK}_{\Omega'}, \mathsf{SK}_{\Omega''}, \mathsf{SK}_{\Omega'''}\}$, an adversary can easily compute SK_Ω. Collusion of this sort has already been studied in the literature and goes by the name Aggregate [16].

If the above-mentioned query sequence $\{\Omega', \Omega'', \Omega'''\}$ (which results in corresponding secret key sequence $\{\mathsf{SK}_{\Omega'}, \mathsf{SK}_{\Omega''}, \mathsf{SK}_{\Omega'''}\}$) is allowed in the security game, both the sets $\{\mathsf{SK}_{\Omega'}, \mathsf{SK}_{\Omega''}\}$ and $\{\mathsf{SK}_{\Omega'}, \mathsf{SK}_{\Omega'''}\}$ can be combined to make the same SK_Ω. Such a dependency relation also sinks into corresponding semi-functional components making the Déjà Q-based proof fail since it required the semi-functional secret key components to be independent. We call such a query sequence as *claw*. Allowing claws in the key extraction queries thus lead to a technical problem for the Déjà Q-based security argument. This will become clear in the context of our proof of SPE_1 and is discussed further in Footnote 1.

The easiest workaround to avoid such *claw* would be to ensure that no two queries have any element in common i.e. $\Omega_i \cap \Omega_j = \phi$ for all distinct $i, j \in [q]$. In IND-s*ID-CPA security model, we put a much weaker restriction on the adversary who is allowed to make key queries only on the *cover-free* sets. Like IND-sID-CPA security model, the challenge data-index set Θ^* here is committed before the Setup phase. But the security model restricts the adversary in terms of the queries it can make for key extraction to avoid *claws*. Precisely, the adversary in this security model, is allowed to make key extraction queries on the key-index sets $(\Omega_1, \Omega_2, \ldots, \Omega_q)$ adaptively with the restriction $\Omega_i \setminus (\bigcup_{j\in[q]\setminus\{i\}} \Omega_j) \neq \phi$. We say SPE is selective* CPA-secure (IND-s*ID-CPA) if, for any PPT adversary $\mathcal{A}$ that gives out the challenge data-index set Θ^* during **Init** and the queries it make following above restriction, $\mathsf{Adv}^{\mathsf{ind\text{-}s^*id\text{-}cpa}}_{\mathcal{A},\mathsf{SPE}}(\lambda) \leq \mathsf{neg}(\lambda)$.

Here we mention that we do not see any ready vulnerability in our construction due to Aggregate (or any other way for that matter). This is because, given the secret keys corresponding to Ω_i and Ω_j, the Aggregate computes secret key for *bigger* set Ω (precisely $\Omega = \Omega_i \cup \Omega_j$ for distinct Ω_i, Ω_j). Now, for a challenge data-index set Θ^*, the natural restriction ensures $\Omega_i, \Omega_j \not\subset \Theta^*$ and therefore $\Omega \not\subset \Theta^*$. Although the existence of Aggregate function allows $\mathcal{A}$ to compute the corresponding secret key SK_Ω, our security argument establishes that it does not help the adversary to win in the security game with non-negligible advantage.

We reiterate that we do not put any restriction on the relation between the challenge data-index set Θ^* and the queried key-index sets Ω apart from the natural restriction $\Omega \not\subseteq \Theta^*$. The $\mathsf{IND\text{-}s^*ID\text{-}CPA}$ model in this respect behaves exactly same as the $\mathsf{IND\text{-}sID\text{-}CPA}$ model. The restriction imposed in selective*-model is sort of a proof artifact that is inherited in the context of SPE_1 from earlier Déjà Q based proofs [18, 30]. The question of arguing security of SPE_1 or similar scheme without the above restriction remains an interesting open problem.

4.2.1 Formal Proof of Security

Theorem 1. *For any adversary $\mathcal{A}$ of SPE construction* SPE_1 *in the* $\mathsf{IND\text{-}s^*ID\text{-}CPA}$ *model that makes at most q many secret key queries, there exist adversary $\mathcal{B}_1$, $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{ind\text{-}s^*id\text{-}cpa}}_{\mathcal{A},\mathsf{SPE}_1}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}_1}(\lambda) + (m+q+2) \cdot \mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}_2}(\lambda) + \frac{((m+q)(m+q+1)+1)}{p_2} + 2^{-\lambda}.$$

Proof. The proof is established via a hybrid argument. The idea is to modify each game only a small amount that allows the solver $\mathcal{B}$ to model the intermediate games properly. The hybrid argument is based on Wee's [29] porting of Déjà Q framework introduced by Chase and Meiklejohn [12]. In the first game Game_0, both the challenge ciphertext and secret keys are normal. Game_1 differs from Game_0 as we introduce some simplifying natural restrictions. Game_2 reformulates the challenge ciphertext to a different representation that will be useful in later games. In Game_3, we replace the challenge ciphertext component C_0 with a random G_{p_1} element. Then, in Game_4, we introduce semi-functional component in the challenge ciphertext. We next define a sub-sequence of games ($\mathsf{Game}_{5,1,0}, \mathsf{Game}_{5,1,1}, \mathsf{Game}_{5,2,0}, \mathsf{Game}_{5,2,1}, \ldots, \mathsf{Game}_{5,m+q+1,0}, \mathsf{Game}_{5,m+q+1,1}$) to introduce entropy into the semi-functional components of secret key and few related public parameters. Note that till this point, we mostly have followed the proof of [18]. Now, the above sub-sequence of games effectively introduces enough entropy in the semi-functional component such that we can replace it with pure random choice in Game_6. This game, in particular, uses the property that key-queries are done on *cover-free* sets only. Finally, in Game_7, we show that semi-functional components as a whole supply enough entropy to hide the encapsulation key κ.

Let the adversary $\mathcal{A}$ make challenge query on Θ^* and q many key extraction queries on the sets $(\Omega_1, \Omega_2, \ldots, \Omega_q)$ where $\Omega_i \not\subseteq \Theta^*$ for all $i \in [q]$. Let us denote $\Theta^* = \{y_1, y_2, \ldots, y_\ell\}$ and $\Omega_i = \{x_{i,1}, \ldots, x_{i,\Bbbk_i}\}$ for all $i \in [q]$. Then we define sets $C'_i = \Theta^* \setminus \Omega_i$ and $C_i = \Omega_i \setminus \Theta^*$ for all $i \in [q]$ and denote their cardinality by ℓ'_i and ℓ_i respectively. The set $M_{i,j} = \Omega_i \setminus \Omega_j$ is the set of identities that is queried in i^{th} query but not in j^{th} query for all $i, j \in [q]$ and $i \neq j$. Let us denote X_i be the event that $\mathcal{A}$ has won the game Game_i.

Game_0. This is same as the real game.
Game_1. The following natural assumptions are made on the game.

– For all $z \in (\Theta^* \cup \bigcup_{i\in[q]} \Omega_i)$, $(\alpha + z)$ is not divisible by p_1. Otherwise, $\mathcal{B}$ can easily solve the subgroup decision problem SD1 by computing $\mathsf{gcd}((\alpha + z), N)$.
– For all $i, j \in [q]$ and $i \neq j$, for all $x, x' \in M_{i,j}$, if $x \neq x' \mod N$ then $x \neq x' \mod p_2$. Otherwise, $\mathcal{B}$ can easily solve the subgroup decision problem SD2 by computing $\mathsf{gcd}((x - x'), N)$.

Therefore, $|\mathsf{Pr}[X_1] - \mathsf{Pr}[X_0]| \leq \mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}}(\lambda)$.

Game_2. We perform a conceptual change to Game_1 here. Given the challenge data-index set $\Theta^* = \{y_1, \ldots, y_\ell\}$, pick $\alpha, \tilde{\beta}, u \hookleftarrow \mathbb{Z}_N^2 \times \mathsf{G}_{p_1}$. Define polynomial $P_{\Theta^*}(z) = \prod_{y\in\Theta^*} (z + y)$ and set $\beta = \tilde{\beta} \cdot P_{\Theta^*}(\alpha) \mod N$. In mpk, this affects only g_1^{β}. Rest of the public parameters in mpk are defined the same as in Game_1. The secret keys corresponding to Ω_i is $\mathsf{SK}_{\Omega_i} = u^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot Y_3$ for $i \in [q]$. The encapsulation key and challenge ciphertext are (κ, CT) where $\mathsf{CT} = (\mathsf{C}_0, \mathsf{C}_1, \mathsf{C}_2)$ such that

$$\kappa = \mathsf{H}(e(\mathsf{C}_0, U_0)), \mathsf{C}_0 = g_1^{s\tilde{\beta}P_{\Theta^*}(\alpha)}, \mathsf{C}_1 = g_1^{sP_{\Theta^*}(\alpha)} = \mathsf{C}_0^{1/\tilde{\beta}},$$

where $U_0 = u \cdot \mathsf{R}_3$ for $\mathsf{R}_3 \hookleftarrow \mathsf{G}_{p_3}$. Note that, the replacement $\beta = \tilde{\beta} \cdot P_{\Theta^*}(\alpha) \mod N$ doesn't change the ciphertext distribution as $\tilde{\beta}$ is uniformly random and $P_{\Theta^*}(\alpha) \neq 0 \mod p_1$. Therefore, $\mathsf{Pr}[X_2] = \mathsf{Pr}[X_1]$.

Game_3. Another conceptual change to Game_2 is performed here. Choose $\mathsf{C}_0 \hookleftarrow \mathsf{G}_{p_1}$. The rest of the ciphertext is defined the same as in Game_2. As both κ and C_1 are functions of C_0, namely $\kappa = \mathsf{H}(e(\mathsf{C}_0, U_0))$ and $\mathsf{C}_1 = \mathsf{C}_0^{1/\tilde{\beta}}$, such a replacement doesn't change the distribution of the challenge ciphertext or the challenge encapsulation key. Therefore, $\mathsf{Pr}[X_3] = \mathsf{Pr}[X_2]$.

Game_4. Here the subgroup decision assumption SD1 is used to choose C_0 from the group $\mathsf{G}_{p_1p_2}$ uniformly at random. Other ciphertext components and secret keys are generated similar to Game_3. Therefore, $|\mathsf{Pr}[X_4] - \mathsf{Pr}[X_3]| \leq \mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}}(\lambda)$. We provide an informal argument here. Given the problem instance SD1, $\mathcal{B}$ chooses $\alpha, \tilde{\beta} \hookleftarrow \mathbb{Z}_N$. This allows $\mathcal{B}$ to compute all of mpk similar to Game_3. As $\mathcal{B}$ holds both α and $\tilde{\beta}$, it can answer any key extraction query. In the challenge phase, it uses the target T of SD1 problem instance to simulate C_0. If T was from G_{p_1}, C_0 is normal whereas if T was from $\mathsf{G}_{p_1p_2}$, then C_0 is semi-functional. Since C_0 determines the challenge ciphertext completely, the distribution from which T was chosen determines if the challenge ciphertext is normal or semi-functional.

Game_5. At this point, we change all the secret keys $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$ and the public parameters $\{U_i\}_{i\in[m]}$ to semi-functional. At the same time U_0 is also converted into semi-functional. This is done via intermediate games $\{\mathsf{Game}_{5,k,0}, \mathsf{Game}_{5,k,1}\}_{k\in[m+q+1]}$. Informally speaking, this sub-sequence of games inject semi-functional component to all of secret keys $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$ and parameters $\{U_i\}_{i\in[0,m]}$ *multiple times*. This is done by repeating the following steps exactly $(m + q + 1)$ times. On every step (i.e. k^{th} step where $k \in [m + q + 1]$),

(i) first add $g_2^{\alpha \bmod p_2}$ in all SK_{Ω_i} and all U_i for $\alpha \in \mathbb{Z}_N$ fixed during Setup, (ii) then replace $g_2^{\alpha \bmod p_2}$ with $g_2^{\alpha_k \bmod p_2}$ where α_k is a freshly chosen randomness. Here we follow the approach of Wee [30] which lets us argue that all semi-functional components in $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$ and $\{U_i\}_{i\in[0,m]}$ are random to adversary (in Game_6).

– In $\mathsf{Game}_{5,k,0}$, the parameters $\{U_i\}_{i\in[0,m]}$ and the secret keys $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$ are defined as following:

$$U_i = u^{\alpha^i} \cdot \boxed{g_2^{r\alpha^i}} \cdot g_2^{\sum_{j\in[k-1]} r_j \alpha_j^i} R'_{3,i} \tag{2}$$

$$\mathsf{SK}_{\Omega_i} = u^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot \boxed{g_2^{\frac{r\cdot\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}}} \cdot g_2^{\sum_{j\in[k-1]} \frac{r_j\cdot\tilde{\beta}\cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} Y'_3 \tag{3}$$

Informally speaking, this game "adds" the boxed parts to $\{U_i\}_{i\in[0,m]}$ and $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$ of $\mathsf{Game}_{5,k-1,1}$ where r is a uniformly random choice from $\mathbb{Z}_N$. Note that, $\mathcal{B}$ does not have explicit access to g_2 and simulates this using the "target" element of $\mathsf{SD2}$ problem instance. We then claim in Lemma 1 that such a modification is invisible to any probabilistic polynomial time adversary $\mathcal{A}$ under the hardness assumption $\mathsf{SD2}$.

– In $\mathsf{Game}_{5,k,1}$, as mentioned earlier, we replace $g_2^{\alpha \bmod p_2}$ with $g_2^{\alpha_k \bmod p_2}$ for a fresh randomness $\alpha_k \in \mathbb{Z}_N$. The change on the parameters $\{U_i\}_{i\in[0,m]}$ and the secret key $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$ distributions are shown in the boxed part below. It is easy to see that we can rewrite U_i and SK_{Ω_i} in a simpler manner as presented in Equations (4) and (5).

$$\begin{aligned} U_i &= u^{\alpha^i} \cdot \boxed{g_2^{r_k\alpha_k^i}} \cdot g_2^{\sum_{j\in[k-1]} r_j \alpha_j^i} R'_{3,i} \\ &= u^{\alpha^i} \cdot g_2^{\sum_{j\in[k]} r_j \alpha_j^i} R'_{3,i}. \end{aligned} \tag{4}$$

$$\begin{aligned} \mathsf{SK}_{\Omega_i} &= u^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot \boxed{g_2^{\frac{r_k\cdot\tilde{\beta}\cdot P_{\Theta^*}(\alpha_k)}{P_{\Omega_i}(\alpha_k)}}} \cdot g_2^{\sum_{j\in[k-1]} \frac{r_j\cdot\tilde{\beta}\cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} Y'_3 \\ &= u^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_2^{\sum_{j\in[k]} \frac{r_j\cdot\tilde{\beta}\cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} Y'_3 \end{aligned} \tag{5}$$

Here we claim that $|\mathsf{Pr}[X_{5,k,0}] - \mathsf{Pr}[X_{5,k,1}]| = 0$. First note that, at the end of Game_4, CT^* and κ are independent of $\alpha \bmod p_2$. Moreover, due to *Chinese Remainder Theorem*, the public parameters $g_1^{\beta}, (G_i)_{i\in[m]}$ are also independent of $\alpha \bmod p_2$. Thus, $\alpha \bmod p_2$ is only present in the parameters $\{U_i\}_{i\in[0,m]}$ and in the secret keys $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$. As a result, the replacement of $\alpha \bmod p_2$ using $\alpha_k \bmod p_2$ is information-theoretically hidden to the adversary for freshly chosen uniformly random $\alpha_k \in \mathbb{Z}_N$. Moreover, $r \bmod p_2$ is involved only in $\{U_i\}_{i\in[0,m]}$ and in $\{\mathsf{SK}_{\Omega_i}\}_{i\in[q]}$ and can be replaced with uniformly random choice $r_k \in \mathbb{Z}_N$ trivially. Thus these changes are invisible to any adversary $\mathcal{A}$ and its winning probability doesn't change.

Here, we denote Game_4 by $\mathsf{Game}_{5,0,1}$ and Game_5 by $\mathsf{Game}_{5,m+q+1,1}$ to get a sub-sequence of games that injected requied amount of entropy to all the secret keys and related parameters. The changes that happened between Game_4 and Game_5 are summed up in Equations (6) and (7) where $r_1, \ldots, r_{m+q+1}, \alpha_1, \ldots, \alpha_{m+q+1} \hookleftarrow \mathbb{Z}_N$.

$$u^{\alpha^i} \cdot R_{3,i} \to u^{\alpha^i} \cdot g_2^{\sum_{j\in[m+q+1]} r_j \alpha_j^i} \cdot R'_{3,i} \tag{6}$$

$$u^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot Y_3 \to u^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_2^{\sum_{j\in[m+q+1]} \frac{r_j\cdot\tilde{\beta}\cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} \cdot Y'_3 \tag{7}$$

Thus, we have $|\Pr[X_5] - \Pr[X_4]| \leq \sum_{k\in[m+q+1]} |\Pr[X_{5,k-1,1}] - \Pr[X_{5,k,0}]|$
$\leq (m+q+1)\cdot \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD2}}(\lambda)$.

Game_6**.** Our aim in this game is to show that semi-functional component of U_0 is independent of semi-functional components of $(U_i)_{i\in[m]}$ and $(\mathsf{SK}_{\Omega_i})_{i\in[q]}$. To this end, we first replace all semi-functional components of $(U_i)_{i\in[0,m]}$ and $(\mathsf{SK}_{\Omega_i})_{i\in[q]}$ by freshly chosen uniformly random values and then show that such a replacement is possible without adversary noticing the change.

Thus, we replace the G_{p_2} components of $(U_i)_{i\in[0,m]}$ and $(\mathsf{SK}_{\Omega_i})_{i\in[q]}$ of Equations (6) and (7) with $g_2^{z_0}, g_2^{z_1}, \ldots, g_2^{z_{m+q}}$ respectively for randomly chosen elements $z_0, z_1, \ldots, z_{m+q} \hookleftarrow \mathbb{Z}_N$. Precisely, for all $i \in [0,m]$ and for all $j \in [q]$ we rewrite the Equations (6) and (7) as follows.

$$U_i = u^{\alpha^i} \cdot g_2^{z_i} \cdot \hat{R}_{3,i}, \qquad \mathsf{SK}_{\Omega_j} = u^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_2^{z_{m+j}} \hat{X}_{3,j}.$$

This change between Game_5 and Game_6 can be represented as a linear system $\mathbf{z} = \mathbf{Ar}$ in Equation (8). In Lemma 2, we show that except with negligible probability, $\mathbf{A}$ is a non-singular matrix.[1] This ensures $\mathbf{z}$ to be a random vector in the span of $\mathbf{A}$ and thus the above modification is invisible to the adversary.

[1] This is where the restriction imposed on $\mathcal{A}$ in the $\mathsf{IND\text{-}s^*ID\text{-}CPA}$ security model is useful. As otherwise any clawed query sequence would trivially make $\mathbf{A}$ singular. Let us revisit the toy query sequence $\{\mathsf{SK}_{\Omega'}, \mathsf{SK}_{\Omega''}, \mathsf{SK}_{\Omega'''}\}$ that resulted in claw. Let $\mathbf{A}[m+1+a]$ encodes Ω', $\mathbf{A}[m+1+b]$ encodes Ω'' and $\mathbf{A}[m+1+c]$ encodes Ω''' for some $a, b, c \in [q]$. Then, the following linear combinations $\frac{\mathbf{A}[m+1+b]-\mathbf{A}[m+1+a]}{(\mathsf{x}_1-\mathsf{x}_3)^{-1}}$ and $\frac{\mathbf{A}[m+1+c]-\mathbf{A}[m+1+a]}{(\mathsf{x}_2-\mathsf{x}_3)^{-1}}$ makes the rows $\mathbf{A}[m+1+b]$ and $\mathbf{A}[m+1+c]$ identical thereby making $\mathbf{A}$ singular. This is where the $\mathscr{P}2$ issue affects our Déjà Q based proof argument. The issue $\mathscr{P}1$, on the other hands, does not create any problem in our proof as $\frac{P_{\Theta^*}(z)}{P_{\Omega_i}(z)} = \frac{\prod_{y_j\in\Theta^*}(z+y_j)}{\prod_{x_j\in\Omega_i}(z+x_j)}$ cancels out $P_{\Omega_i\cap\Theta^*}(z)$ for all $i \in [q]$.

$$\begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_m \\ z_{m+1} \\ \vdots \\ z_{m+q} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{m+q+1} \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_{m+q+1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^m & \alpha_2^m & \cdots & \alpha_{m+q+1}^m \\ \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_1)}{P_{\Omega_1}(\alpha_1)} & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_2)}{P_{\Omega_1}(\alpha_2)} & \cdots & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_{m+q+1})}{P_{\Omega_1}(\alpha_{m+q+1})} \\ \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_1)}{P_{\Omega_2}(\alpha_1)} & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_2)}{P_{\Omega_2}(\alpha_2)} & \cdots & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_{m+q+1})}{P_{\Omega_2}(\alpha_{m+q+1})} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_1)}{P_{\Omega_q}(\alpha_1)} & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_2)}{P_{\Omega_q}(\alpha_2)} & \cdots & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_{m+q+1})}{P_{\Omega_q}(\alpha_{m+q+1})} \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{m+q+1} \end{pmatrix}. \quad (8)$$

As a result, we get that $|\mathsf{Pr}[X_6] - \mathsf{Pr}[X_5]| \leq (m+q)(m+q+1)/p_2$.

Game_7. Now we replace $\kappa_0 = \mathsf{H}(e(\mathsf{C}_0, U_0))$ by a uniform random choice from $\mathcal{K}$. The reason behind this is U_0 now is $u \cdot g_2^{z_0} \cdot R_3$. As we saw in the last game, z_0 is a uniformly random quantity independent of all $(z_i)_{i\in[q+m]}$. Thus $e(\mathsf{C}_0, U_0) = e(\mathsf{C}_0, u) \cdot e(\mathsf{C}_0, g_2^{z_0})$ has $\log p_2$ bits of min-entropy due to z_0 mod p_2. Due to left-over hash lemma [20], $\kappa_0 = \mathsf{H}(e(\mathsf{C}_0, U_0))$ is at most $2^{-\lambda}$ distance from the uniform distribution on $\mathcal{K}$ provided G_{p_2} component in C_0 is not 1. The probability that the G_{p_2} component of C_0 is 1 is $1/p_2$. Therefore $|\mathsf{Pr}[X_7] - \mathsf{Pr}[X_6]| \leq 1/p_2 + 2^{-\lambda}$. κ_0 now is a random choice and it hides $\mathfrak{b}$ completely i.e. $\mathsf{Pr}[X_7] = 1/2$.

This completes the proof of Theorem 1 assuming Lemmas 1 to 3 holds. □

Lemma 1. *There exists PPT adversary $\mathcal{B}$ such that,* $|\mathsf{Pr}[X_{5,k-1,1}] - \mathsf{Pr}[X_{5,k,0}]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SD2}}(\lambda)$.

Proof. The solver $\mathcal{B}$ is given the problem instance $D = (g_1, g_3, g_{12}, g_{23})$ and the target T.

Setup. The adversary $\mathcal{A}$ sends the challenger target set Θ^*. $\mathcal{B}$ chooses $\alpha, \tilde{\beta} \hookleftarrow \mathbb{Z}_N^2$ to generate the public parameters $g_1^{\beta}, (G_i)_{i\in[m]}$ efficiently where $\beta = \tilde{\beta} \cdot P_{\Theta^*}(\alpha)$ mod N, $G_i = g_1^{\alpha^i}$. It then chooses $\{\hat{r}_j, \alpha_j\}_{j\in[k-1]} \hookleftarrow \mathbb{Z}_N$. The public parameters $(U_i)_{i\in[m]}$ are generated as follows along with U_0 which is used to compute $e(g_1, u)^{\beta} = e(g_1, U_0)^{\beta}$. For $R'_{3,i} \hookleftarrow \mathsf{G}_{p_3}$,

$$U_i = T^{\alpha^i} g_{23}^{\sum_{j\in[k-1]} \hat{r}_j \alpha_j^i} R'_{3,i}.$$

$\mathcal{B}$ then outputs public parameter

$$\mathsf{mpk} = (g_1, g_1^{\beta}, (G_i, U_i)_{i\in[m]}, e(g_1, U_0)^{\beta}, \mathsf{H}),$$

where H is randomly chosen universal hash function.

Phase-I Queries. On a secret key query on Ω_i, $\mathcal{B}$ chooses $Y_3' \hookleftarrow \mathsf{G}_{p_3}$ and sets

$$\mathsf{SK}_{\Omega_i} = T^{\frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_{23}^{\sum_{j\in[k-1]} \frac{\hat{r}_j\cdot\tilde{\beta}\cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} Y_3'.$$

Challenge. $\mathcal{B}$ here computes κ_0 and $\mathsf{CT}_{\Theta^*} = (\mathsf{C}_0, \mathsf{C}_1)$ where $\mathsf{C}_0 = g_{12}$, $\mathsf{C}_1 = \mathsf{C}_0^{1/\tilde{\beta}}$ and $\kappa_0 = \mathsf{H}(e(\mathsf{C}_0, U_0))$. $\mathcal{B}$ chooses $\kappa_1 \hookleftarrow \mathcal{K}$ and outputs $(\kappa_{\mathfrak{b}}, \mathsf{C}_0, \mathsf{C}_1)$ for $\mathfrak{b} \hookleftarrow \{0,1\}$.

Phase-II Queries. Same as Phase-I queries.

Guess. $\mathcal{B}$ outputs 1 if $\mathcal{A}$'s guess $\mathfrak{b}'$ is same as $\mathcal{B}$'s choice $\mathfrak{b}$.

If $T \in \mathsf{G}_{p_1p_3}$, then the game distribution is same as $\mathsf{Game}_{5,k-1,1}$. On the other hand, if $T \in \mathsf{G}$, then the game distribution is same as $\mathsf{Game}_{5,k,0}$ (see Equations (4) and (5)). $\square$

Lemma 2. *The matrix* $\mathbf{A}$ *in Equation* (8) *is non-singular except with probability* $(m+q)(m+q+1)/p_2$.

Proof. We rephrase the matrix $\mathbf{A}$ from Equation (8) as $\mathbf{A} = \begin{pmatrix}\mathbf{B}\\ \mathbf{P}\end{pmatrix}$ where $\mathbf{B} \in \mathbb{Z}_{p_2}^{(m+1)\times(m+q+1)}$ denotes the first $(m+1)$ rows of $\mathbf{A}$ and $\mathbf{P} \in \mathbb{Z}_{p_2}^{q\times(m+q+1)}$ denotes the last q rows of $\mathbf{A}$. Each entry of $\mathbf{B}$ and $\mathbf{P}$ are respectively evaluation of following polynomials with the indeterminant z taking values $(\alpha_1, \alpha_2, \cdots, \alpha_{m+q+1})$. Therefore for any $\ell \in [m+q+1]$, each $[i,\ell]^{th}$ entry of $\mathbf{B}$ and $\mathbf{P}$ are respectively:

$$\begin{aligned} \mathbf{B}[i,\ell] &= z^i && \text{for } i \in [0,m], \\ \mathbf{P}[i,\ell] &= \frac{\tilde{\beta}\cdot P_{\Theta^*}(z)}{P_{\Omega_i}(z)} && \text{for } i \in [q]. \end{aligned} \tag{9}$$

We simplify the $\mathbf{P}[i,\ell]$ entry next for arbitrary $i \in S_A$, $\ell \in [m+q+1]$. By natural restriction, for all queries, $\Omega_i \not\subset \Theta^*$. Therefore, the polynomial $P_{\Omega_i}(z) \nmid P_{\Theta^*}(z)$ for all $i \in S_A$ where z is the indeterminant. Notice that, $P_{\Theta^*}(z) = \prod\limits_{j\in[\ell]} (z+y_j)$ and $P_{\Omega_i}(z) = \prod\limits_{j\in[\Bbbk_i]} (z+x_j)$ are splitting polynomials. Then, the rational function

$$\mathfrak{R} = \frac{P_{\Theta^*}(z)}{P_{\Omega_i}(z)} = \frac{\prod\limits_{y_j\in\Theta^*}(z+y_j)}{\prod\limits_{x_j\in\Omega_i}(z+x_j)} = \frac{\prod\limits_{y_j\in C_i'}(z+y_j)}{\prod\limits_{x_j\in C_i}(z+x_j)} = \mathfrak{A}\cdot\mathfrak{B} \tag{10}$$

where $\mathfrak{A} = \prod\limits_{y_j\in C_i'}(z+y_j) = P_{C_i'}(z)$ and $\mathfrak{B} = \frac{1}{\prod\limits_{x_j\in C_i}(z+x_j)}$. It is easy to see that, $\mathfrak{B}$ can be rewritten as $\frac{1}{\prod\limits_{x_j\in C_i}(z+x_j)} = \sum\limits_{x_j\in C_i} \frac{1}{\prod\limits_{\substack{x_k\in C_i\\ j\neq k}}(x_k-x_j)} \cdot \frac{1}{z+x_j}$ [21]. In other words, $\mathfrak{B} = \sum\limits_{x_j\in C_i} R_{j,i} \cdot \frac{1}{z+x_j}$ where $R_{j,i}$ are non-zero scalar values that can be computed from the set C_i. The rational fraction $\mathfrak{R}$ from Equation (10) therefore is

$$\mathfrak{R} = \sum_{x_j \in C_i} R_{j,i} \cdot \frac{P_{C'_i}(z)}{z + x_j}. \tag{11}$$

For any $i \in S_A$ and $\ell \in [m+q+1]$,

$$\begin{aligned}
\mathbf{P}[i,\ell] &= \tilde{\beta} \cdot \sum_{x_j \in C_i} R_{j,i} \cdot \tfrac{P_{C'_i}(z)}{z+x_j} && \text{(from Equation (9) and Equation (11))} \\
&= \tilde{\beta} \cdot \sum_{x_j \in C_i} R_{j,i} \left(K_{C'_i, x_j}(z) + \tfrac{t_j}{z+x_j} \right) && (t_j \text{ is scalar}) \\
&= \sum_{x_j \in C_i} \left(\tilde{R}_{j,i} K_{C'_i, x_j}(z) + \tfrac{R'_{j,i}}{z+x_j} \right) \\
& && (\tilde{R}_{j,i} = \tilde{\beta} \cdot R_{j,i},\ R'_{j,i} = \tilde{\beta} \cdot R_{j,i} \cdot t_j \text{ are scalars}) \\
&= \sum_{x_j \in C_i} \left(\sum_{k \in [0, \ell'_i]} \tilde{R}_{j,i} \cdot b_k^{j,i} z^k + \tfrac{R'_{j,i}}{z+x_j} \right) \\
& && (K_{C'_i, x_j}(z) = \sum_{k \in [0,\ell'_i]} b_k^{j,i} z^k \text{ is polynomial expansion}) \\
&= \sum_{x_j \in C_i} \left(\sum_{k \in [0, \ell'_i]} \tilde{R}'_{j,i,k} z^k + \tfrac{R'_{j,i}}{z+x_j} \right) && (\tilde{R}'_{j,i,k} = \tilde{R}_{j,i} \cdot b_k^{j,i} \text{ is scalar}). \\
&= \sum_{k \in [0,\ell'_i]} \hat{\tilde{R}}'_{j,i,k} z^k + \sum_{x_j \in C_i} \tfrac{R'_{j,i}}{z+x_j} && (\hat{\tilde{R}}'_{j,i,k} = \sum_{x_j \in C_i} \tilde{R}'_{j,i,k} \text{ is scalar}).
\end{aligned}$$

We rewrite Equation (9) as, for any $\ell \in [m+q+1]$,

$$\begin{aligned}
\mathbf{B}[i,\ell] &= z^i && \text{for } i \in [0,m], \\
\mathbf{P}[i,\ell] &= \sum_{k \in [0,\ell'_i]} \hat{\tilde{R}}'_{j,i,k} z^k + \sum_{x_j \in C_i} \frac{R'_{j,i}}{z + x_j} && \text{for } i \in [q].
\end{aligned} \tag{12}$$

Notice that, in Equation (12), each $\hat{\tilde{R}}'_{j,i,k} z^k$ in $\mathbf{P}[i,\ell]$ is in the linear span of $\{\mathbf{B}[i,\ell]\}_{j \in [0,m]}$ as $\ell'_i \le \ell \le m$. Therefore, elementary row operations remove such dependency to define a new matrix $\mathbf{A}' = \begin{pmatrix} \mathbf{B} \\ \mathbf{P}' \end{pmatrix}$ such that $\mathbf{A}$ and $\mathbf{A}'$ are similar matrices (i.e. $|\mathsf{det}(\mathbf{A})| = |\mathsf{det}(\mathbf{A}')|$) where for any $\ell \in [m+q+1]$, z takes value from the set $\{\alpha_1, \ldots, \alpha_{m+q+1}\}$ and each $[i,\ell]^{th}$ entry of $\mathbf{B}$ and $\mathbf{P}'$ are respectively

$$\begin{aligned}
\mathbf{B}[i,\ell] &= z^i && \text{for } i \in [0,m], \\
\mathbf{P}'[i,\ell] &= \sum_{x_j \in C_i} \frac{R'_{j,i}}{z + x_j} && \text{for } i \in [q].
\end{aligned} \tag{13}$$

To complete this proof, it is now sufficient to show the following claim holds.

Claim. The matrix $\mathbf{A}' = \begin{pmatrix} \mathbf{B} \\ \mathbf{P}' \end{pmatrix}$, where $\mathbf{B}$ and $\mathbf{P}'$ are as defined in Equation (13), is non-singular except with probability $(m+q)(m+q+1)/p_2$.

To prove that $\mathbf{A}'$ as defined above is non-singular, we start from the claim that the matrix $\mathbf{D}$ (in Equation (14)) is non-singular if all $x_j \neq x_t$ for $t, j \in [Q]$ for $t \neq j$ and all $\gamma_i \neq \gamma_k$ for $i, k \in [m+Q+1]$ for $i \neq k$. The non-singularity of $\mathbf{D}$ was proved in [18, Lemma 3] where Q was the number of key-queries (i.e. distinct x_j). We here set Q to be cardinality of $\left(\bigcup_{i \in S_A} \Omega_i \right)$ i.e. total number of distinct x_j that is queried as a part of some key query Ω_i.

$$\mathbf{D} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{m+Q+1} \\ \gamma_1^2 & \gamma_2^2 & \cdots & \gamma_{m+Q+1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1^m & \gamma_2^m & \cdots & \gamma_{m+Q+1}^m \\ \frac{1}{\gamma_1+x_1} & \frac{1}{\gamma_2+x_1} & \cdots & \frac{1}{\gamma_{m+Q+1}+x_1} \\ \frac{1}{\gamma_1+x_2} & \frac{1}{\gamma_2+x_2} & \cdots & \frac{1}{\gamma_{m+Q+1}+x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\gamma_1+x_Q} & \frac{1}{\gamma_2+x_Q} & \cdots & \frac{1}{\gamma_{m+Q+1}+x_Q} \end{pmatrix} \tag{14}$$

Performing elementary row operations on each $x_j \in C_i$ using $R'_{j,i}$ as scalar, one can get the polynomial $\mathbf{P}'[i, \ell]$ in Equation (13). One can perform such transformation if $Q \geq q$ which is the case due to the restriction we imposed on the relation between query sets. Precisely, the *cover-free* property of the query sets ensures that $Q \geq q$ as informally speaking each Ω_i contains some new x_j. In other words, we perform elementary row operations on $\mathbf{D}$ in Equation (14) to get to the matrix $\mathbf{D}'$ such that all the rows in $\mathbf{A}'$ are also present in $\mathbf{D}'$. Since, $\mathbf{D}$ is non-singular due to Lemma 3, $\mathbf{D}'$ is also non-singular and therefore all the $(m+Q+1)$ rows of $\mathbf{D}'$ are linearly independent. Let us denote the last Q rows of $\mathbf{D}'$ by $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_Q)$. Notice that, there are $(Q-q)$ many rows in $\mathbf{d}$ that are not present in $\mathbf{A}'$ (Equation (13)). We remove these rows to get a matrix $\tilde{\mathbf{D}}' \in \mathbb{Z}_{p_2}^{(m+q+1)\times(m+Q+1)}$ of rank $(m+q+1)$ as $\tilde{\mathbf{D}}'$ has $m+q+1$ many linear independent rows. Among the $m+Q+1$ many columns of $\tilde{\mathbf{D}}'$, $m+q+1$ many will be linearly independent as rank of $\tilde{\mathbf{D}}'$ is $m+q+1$. These columns form a *full-rank* matrix of order $(m+q+1) \times (m+q+1)$. Notice that, this matrix is exactly the same as $\mathbf{A}'$. This proves our claim that $\mathbf{A}'$ is non-singular except with probability $(m+q)(m+q+1)/p_2$.

This, at the same time ensures that the matrix $\mathbf{A}$ in Equation (8) is also non-singular. Due to the fact that each Ω_i where $i \in S_A$ will have at least one new x_j, it is evident that $\mathsf{det}(\mathbf{A}) \neq 0$ as long as $\alpha_s \neq \alpha_k \bmod p_2$ for $s, k \in [m+q+1]$ and $s \neq k$. Thus $\mathbf{A}$ is non-singular except with probability $(m+q)(m+q+1)/p_2$. $\square$

Lemma 3. $\mathsf{det}(\mathbf{D}) = \delta \cdot \frac{\prod_{1 \leq t < j \leq Q}(x_t - x_j) \prod_{1 \leq i < k \leq m+Q+1}(\gamma_i - \gamma_k)}{\prod_{k=1}^{m+Q+1} \prod_{t=1}^{Q}(\gamma_k + x_t)}$ *where* δ *is some non-zero scalar in* $\mathbb{Z}_{p_2}$ *where* $\mathbf{D}$ *is given in Equation* (14).

Proof. Gong et al. [18, Lemma 3] proved this statement. For the sake of completeness, we reproduce it here. The matrix in Equation (14) is of order $(m + Q + 1) \times (m + Q + 1)$.
Each of the monomials of $\mathcal{P} = \mathsf{det}(\mathbf{D}) \cdot \prod_{k=1}^{m+Q+1} \prod_{t=1}^{Q} (\gamma_k + x_t)$ is of degree

$$\frac{m(m+1)}{2} + Q(m + Q + 1) - Q = \frac{m(m+1)}{2} + Q(m + Q).$$

Notice that, $\mathsf{det}(\mathbf{D}) = 0$ if

- $\exists i, j \in [m + Q + 1]$ such that $\alpha_i = \alpha_j$ for $i \neq j$ then columns i and j are same.
- $\exists i, j \in [m + 2, m + Q + 1]$ such that $x_i = x_j \mod p_2$ for $i \neq j$ then rows i and j are same.

Therefore, the polynomial $\mathcal{P}$ must be a multiple of $\mathcal{T} = \prod_{1 \le t < j \le Q} (x_t - x_j) \cdot \prod_{1 \le i < k \le m+Q+1} (\gamma_i - \gamma_k)$ of degree $\frac{Q(Q-1)}{2} + \frac{(Q+m+1)(Q+m)}{2}$ same as $\mathsf{deg}(\mathcal{P})$.
The proof of lemma thus follows. □

5 SPE_2: An Adaptive Secure Construction

Our second and main construction is instantiated in the prime order bilinear groups and achieves adaptive security under the SXDH assumption.

5.1 Construction

SPE_2 is defined as the following four algorithms.

- $\mathsf{Setup}(1^\lambda, m)$: The asymmetric bilinear group generator outputs $(p, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_\mathrm{T}, e) \leftarrow \mathcal{G}_{\mathsf{abg}}(1^\lambda)$ where $\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_\mathrm{T}$ are cyclic groups of order p. Choose generators $g_1 \hookleftarrow \mathsf{G}_1$ and $g_2 \hookleftarrow \mathsf{G}_2$ and define $g_\mathrm{T} = e(g_1, g_2)$. Choose $\alpha_1, \alpha_2, c, d, (u_j, v_j)_{j \in [0,2m]} \hookleftarrow \mathbb{Z}_p$ and $b \hookleftarrow \mathbb{Z}_p^\times$. For $j \in \{0, \ldots, 2m\}$, define $g_1^{w_j} = g_1^{u_j + b v_j}$ and $g_1^w = g_1^{c+bd}$. Then define $\alpha = (\alpha_1 + b\alpha_2)$ and therefore $g_\mathrm{T}^\alpha = e(g_1, g_2)^{\alpha_1 + b\alpha_2}$. Define the $\mathsf{msk} = (g_2, g_2^c, \alpha_1, \alpha_2, d, (u_j, v_j)_{j \in [0,2m]})$ and the public parameter is defined as

$$\mathsf{mpk} = \left(g_1, g_1^b, \left(g_1^{w_j}\right)_{j \in [0,2m]}, g_1^w, g_\mathrm{T}^\alpha \right).$$

- $\mathsf{KeyGen}(\mathsf{msk}, \Omega)$: Given a set Ω, such that $|\Omega| = \Bbbk \le m$ choose $r \hookleftarrow \mathbb{Z}_p$. Compute the secret key as $\mathsf{SK}_\Omega = (\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4, \mathsf{K}_5)$ where

$$\mathsf{K}_1 = g_2^r, \mathsf{K}_2 = g_2^{cr}, \mathsf{K}_3 = g_2^{\alpha_1 + r \sum_{x \in \Omega} (u_0 + u_1 x + u_2 x^2 + \ldots + u_{2m} x^{2m})},$$

$$\mathsf{K}_4 = g_2^{dr}, \mathsf{K}_5 = g_2^{\alpha_2 + r \sum_{x \in \Omega} (v_0 + v_1 x + v_2 x^2 + \ldots + v_{2m} x^{2m})}.$$

– $\mathsf{Encrypt}(\mathsf{mpk}, \Theta)$: Given a set $\Theta = \{y_1, \ldots, y_\ell\}$ where $\ell \le m$, choose $s \hookleftarrow \mathbb{Z}_p$ and compute κ and $\mathsf{CT}_\Theta = (\mathsf{C}_0, \mathsf{C}_1, (\mathsf{C}_{2,i}, t_i)_{i \in [\ell]})$ where $(t_i)_{i\in[\ell]} \hookleftarrow \mathbb{Z}_p$ and

$$\kappa = e(g_1, g_2)^{\alpha s}, \mathsf{C}_0 = g_1^s, \mathsf{C}_1 = g_1^{bs}, \mathsf{C}_{2,i} = g_1^{s(w_0 + w_1 y_i + w_2 y_i^2 + \ldots + w_{2m} y_i^{2m} + w t_i)}.$$

– $\mathsf{Decrypt}((\mathsf{SK}_\Omega, \Omega), (\mathsf{CT}_\Theta, \Theta))$: Compute $\kappa = B/A$ where

$$A = e\left(\prod_{y_i \in \Omega} \mathsf{C}_{2,i}, \mathsf{K}_1\right), B = e\left(\mathsf{C}_0, \mathsf{K}_3 \prod_{y_i \in \Omega} \mathsf{K}_2^{t_i}\right) e\left(\mathsf{C}_1, \mathsf{K}_5 \prod_{y_i\in\Omega} \mathsf{K}_4^{t_i}\right).$$

Correctness. As $\Omega \subseteq \Theta$,

$$\begin{aligned}
A &= e\left(\prod_{y_i\in\Omega} \mathsf{C}_{2,i}, \mathsf{K}_1\right) \\
&= e\left(g_1^{s\sum_{y_i\in\Omega}(w_0 + w_1 y_i + w_2 y_i^2 + \ldots + w_{2m} y_i^{2m} + w t_i)}, g_2^r\right)
\end{aligned}$$

$$\begin{aligned}
B &= e\left(\mathsf{C}_0, \mathsf{K}_3 \prod_{y_i\in\Omega} \mathsf{K}_2^{t_i}\right) e\left(\mathsf{C}_1, \mathsf{K}_5 \prod_{y_i\in\Omega} \mathsf{K}_4^{t_i}\right), \\
&= e\left(\mathsf{C}_0, g_2^{\alpha_1 + r\sum_{y_i\in\Omega}(u_0 + u_1 y_i + u_2 y_i^2 + \ldots + u_{2m} y_i^{2m})} \cdot \prod_{y_i\in\Omega} g_2^{rct_i}\right) \\
&\qquad \cdot e\left(\mathsf{C}_1, g_2^{\alpha_2 + r\sum_{y_i\in\Omega}(v_0 + v_1 y_i + v_2 y_i^2 + \ldots + v_{2m} y_i^{2m})} \cdot \prod_{y_i\in\Omega} g_2^{rdt_i}\right) \\
&= e\left(\mathsf{C}_0, g_2^{\alpha_1 + r\sum_{y_i\in\Omega}(u_0 + u_1 y_i + u_2 y_i^2 + \ldots + u_{2m} y_i^{2m})} \cdot \prod_{y_i\in\Omega} g_2^{rct_i}\right) \\
&\qquad \cdot e\left(\mathsf{C}_0, g_2^{b\alpha_2 + rb\sum_{y_i\in\Omega}(v_0 + v_1 y_i + v_2 y_i^2 + \ldots + v_{2m} y_i^{2m})} \cdot \prod_{y_i\in\Omega} g_2^{rbdt_i}\right) \\
&= e\left(\mathsf{C}_0, g_2^{(\alpha_1 + b\alpha_2) + r\sum_{y_i\in\Omega}((u_0 + bv_0) + (u_1 + bv_1) y_i + \ldots + (u_{2m} + bv_{2m}) y_i^{2m})} \cdot \prod_{y_i\in\Omega} g_2^{r(c+bd)t_i}\right) \\
&= e\left(\mathsf{C}_0, g_2^{\alpha + r\sum_{y_i\in\Omega}(w_0 + w_1 y_i + w_2 y_i^2 + \ldots + w_{2m} y_i^{2m})} \cdot \prod_{y_i\in\Omega} g_2^{rwt_i}\right)
\end{aligned}$$

$$= e\left(g_1^s, g_2^{\alpha + r\sum\limits_{y_i \in \Omega}(w_0 + w_1 y_i + w_2 y_i^2 + \ldots + w_{2m} y_i^{2m} + w t_i)}\right)$$

Then $B/A = e(g_1^s, g_2^\alpha) = \kappa$.

Remark 2. Our SPE_2 construction has a *pair encoding* [4] embedded implicitly. One can utilize the generic technique of Chen et al. [13] to get corresponding predicate encryption. Compared to SPE_2, the generic construction has significantly larger (almost double) public parameter and ciphertext and requires one additional pairing during decryption. On the other hand, the secret key in generic construction contains one less group element. As an alternative, Chen and Gong [14] compiler can be applied on such pair encoding to construct SPE with adaptive CPA-security under k-Linear assumption [13]. For SXDH/1-Linear, the resulting construction and SPE_2 will have similar parameter size and computational complexity. However, for $k \geq 2$, all of mpk, SK and CT grow in size. For example, for 2-Lin, corresponding construction of SPE makes all of mpk, SK and CT almost double with an associated increase in computational cost.

5.2 Security

Theorem 2. *For any adversary $\mathcal{A}$ of SPE construction* SPE_2 *in the* $\mathsf{IND\text{-}ID\text{-}CPA}$ *security model that makes at most q many secret key queries, there exist adversary $\mathcal{B}_1$, $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\mathcal{A},\mathsf{SPE}_2}(\lambda) \leq \mathsf{Adv}^{\mathsf{DDH}_{\mathbb{G}_1}}_{\mathcal{B}_1}(\lambda) + q \cdot \mathsf{Adv}^{\mathsf{DDH}_{\mathbb{G}_2}}_{\mathcal{B}_2}(\lambda) + 2/p.$$

Proof Idea. We propose a hybrid argument based proof that uses dual system proof technique [28] at its core. This hybrid argument follows the proof strategy of [25]. In this sequence of game-based argument, in the first game, (Game_0) both the challenge ciphertext and secret keys are normal. The ciphertext is changed first to semi-functional in Game_1. Then all the keys are changed to semi-functional via a series of games $(\mathsf{Game}_{2,k})_k$ for $k \in [q]$. Precisely, in any $\mathsf{Game}_{2,k}$ where $k \in [q]$, all the previous (i.e. $1 \leq j \leq k$) secret keys are semi-functional whereas all the following (i.e. $k < j \leq q$) secret keys are normal. We continue this till $\mathsf{Game}_{2,q}$ where all the keys are semi-functional. In the final game Game_3, the encapsulation key κ is replaced by a uniformly random choice from $\mathcal{K}$. We show that the semi-functional components of challenge ciphertext and secret keys in Game_3 supply enough entropy to hide the encapsulation key κ; hence it is distributionally same as a random choice from $\mathcal{K}$. Note that, we denote Game_1 by $\mathsf{Game}_{2,0}$. Suppose we use X_i to denote the event that adversary wins Game_i. Then, for any efficient adversary $\mathcal{A}$ of the $\mathsf{IND\text{-}ID\text{-}CPA}$ security model, we compute its advantage using Lemmas 4 to 6 thereby completing the proof of Theorem 2.

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\mathcal{A},\mathsf{SPE}_2}(\lambda) &= |\mathsf{Pr}[X_0] - 1/2| \\ &= |\mathsf{Pr}[X_0] - \mathsf{Pr}[X_3]|\end{aligned}$$

$$\begin{aligned}
&\leq |\mathsf{Pr}[X_0] - \mathsf{Pr}[X_1]| + |\mathsf{Pr}[X_1] - \mathsf{Pr}[X_{2,0}]| \\
&\qquad + \sum_{k\in[q]} |\mathsf{Pr}[X_{2,k-1}] - \mathsf{Pr}[X_{2,k}]| + |\mathsf{Pr}[X_{2,q}] - \mathsf{Pr}[X_3]| \\
&\leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{DDH}_{\mathsf{G}_1}}(\lambda) + \sum_{k\in[q]} \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{DDH}_{\mathsf{G}_2}}(\lambda) + 2/p.
\end{aligned}$$

We already have mentioned that our large universe SPE_2 construction uses IBBE [25] as a starting point. Therefore, before giving the formal presentation of the above mentioned lemmas, let us first recall the crucial tactics Ramanna and Sarkar [25] used to prove their IBBE adaptive CPA-secure. The crux of the proof of IBBE in [25] is a linear map that reflects the relation between tags $(t_1, \ldots, t_\ell)$ which encoded the challenge data-index set $\Theta^* = \{y_1, \ldots, y_\ell\}$ and semi-functional component (π) in the secret key SK_x that encoded queried key-index (i.e. the identity) x. This scenario occurs when a normal secret key is translated into corresponding semi-functional form. At this point, [25] showed that such linear map is non-singular following Attrapadung and Libert [6]. Such a property of the linear map effectively ensures that the semi-functional component of the key has enough entropy to hide the encapsulation key κ.
Next, we define the semi-functional ciphertext and semi-functional secret keys.

5.2.1 Semi-functional Algorithms

– $\mathsf{SFKeyGen}(\mathsf{msk}, \Omega)$: Let the normal secret key be $\mathsf{SK}'_\Omega = (\mathsf{K}'_1, \mathsf{K}'_2, \mathsf{K}'_3, \mathsf{K}'_4, \mathsf{K}'_5) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \Omega)$ where r is the randomness used in KeyGen. Choose $\hat{r}, \pi \hookleftarrow \mathbb{Z}_p$. Compute the semi-functional *trapdoor* as $\mathsf{SK}_\Omega = (\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4, \mathsf{K}_5)$ such that

$$\begin{aligned}
&\mathsf{K}_1 = \mathsf{K}'_1 = g_2^r, \mathsf{K}_2 = \mathsf{K}'_2 \cdot g_2^{\hat{r}} = g_2^{cr+\hat{r}}, \\
&\mathsf{K}_3 = \mathsf{K}'_3 \cdot g_2^{\hat{r}\pi} = g_2^{\alpha_1 + r\sum_{x\in\Omega}(u_0+u_1x+u_2x^2+\ldots+u_{2m}x^{2m}) + \hat{r}\pi}, \\
&\mathsf{K}_4 = \mathsf{K}'_4 \cdot g_2^{-\hat{r}b^{-1}} = g_2^{dr-\hat{r}b^{-1}}, \\
&\mathsf{K}_5 = \mathsf{K}'_5 \cdot g_2^{-\hat{r}\pi b^{-1}} = g_2^{\alpha_2 + r\sum_{x\in\Omega}(v_0+v_1x+v_2x^2+\ldots+v_{2m}x^{2m}) - \hat{r}\pi b^{-1}}.
\end{aligned}$$

– $\mathsf{SFEncrypt}(\mathsf{mpk}, \mathsf{msk}, \Theta)$: Let the normal *encapsulation key* and normal ciphertext be $(\kappa', \mathsf{CT}'_\Theta) \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{msk}, \Theta)$ where s is the randomness and $(t_i)_{i\in[\ell]}$ are the random tags used in $\mathsf{Encrypt}$ such that $\mathsf{CT}'_\Theta = (\mathsf{C}'_0, \mathsf{C}'_1, (\mathsf{C}'_{2,i}, t_i)_{i\in[\ell]})$. Choose $s \hookleftarrow \mathbb{Z}_p$. Compute the semi-functional encapsulation key κ and semi-functional ciphertext $\mathsf{CT}_\Theta = (\mathsf{C}_0, \mathsf{C}_1, (\mathsf{C}_{2,i}, t_i)_{i\in[\ell]})$ as follows:

$$\begin{aligned}
&\kappa = \kappa' \cdot g_{\mathrm{T}}^{\alpha_1\hat{s}} = e(g_1, g_2)^{\alpha s + \alpha_1\hat{s}}, \mathsf{C}_0 = \mathsf{C}'_0 \cdot g_1^{\hat{s}} = g_1^{s+\hat{s}}, \mathsf{C}_1 = g_1^{bs}, \\
&\mathsf{C}_{2,i} = \mathsf{C}'_{2,i} \cdot g_1^{\hat{s}(u_0+u_1y_i+u_2y_i^2+\ldots+u_{2m}y_i^{2m}+ct_i)}, \\
&\quad = g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_{2m}y_i^{2m}+wt_i)+\hat{s}(u_0+u_1y_i+u_2y_i^2+\ldots+u_{2m}y_i^{2m}+ct_i)}.
\end{aligned}$$

5.2.2 Intuition of Proof Recall that, both $\ell, k \leq m$ where $k = |\Omega|$ and $\ell = |\Theta^*|$ for queried key-index set Ω and challenge data-index set Θ^*. In the following, we consider the upper bound i.e. $k = \ell = m$. As mentioned earlier, the crux of the proof lies in establishing that the tags used in the challenge ciphertext (i.e. $t_1, \ldots, t_m$) and that used in the secret key (π) are independent. Following [25], these tags basically imitate the polynomials used to define the challenge ciphertext and the secret key in the semi-functional domain. Informally speaking, [25] defined a polynomial $\sigma(z) = w_0 + w_1 z + w_2 z^2 + \ldots + w_m z^m$ and computed the tags in the ciphertext as $t_i = \sigma(y_i)$ for all $y_i \in \Theta^*$ and the semi-functional secret key tag $\pi = \sigma(x)$. Then, they proved the required independence by representing these polynomial evaluations as a linear system of equations involving an invertible Vandermonde matrix of order $(m+1) \times (m+1)$. The independence followed from the fact that all y_i in the challenge data-index set Θ^* were distinct and the queried key-index (element) $x \notin \Theta^*$. Informally speaking, their proof required a Vandermonde matrix of rank $(m+1)$ as they dealt with $(m+1)$ of $\sigma(\cdot)$ evaluations namely $(t_1, \ldots, t_m, \pi)$.

In our case, observe that π will imitate the polynomial in secret key which basically is the summation of $\hbar = m$ polynomial evaluations each encoding an element of queried key-index set (i.e. $x \in \Omega$). Thus, we are dealing with $2m$ evaluations of the corresponding polynomial ($t_1, \ldots, t_m$ to encode Θ^* and m more for encoding Ω). Our construction, therefore, requires a $2m$ degree polynomial (i.e. $w_0 + w_1 z + w_2 z^2 + \ldots + w_{2m} z^{2m}$) to ensure the independence of the tags used. The details are given in the proof.

5.2.3 Sequence of Games

The idea is to change each game only by a small margin and prove indistinguishability of two consecutive games.

Lemma 4. (Game_0 *to* Game_1) *For any efficient adversary $\mathcal{A}$ that makes at most q key queries, there exists a PPT algorithm $\mathcal{B}$ such that* $|\mathsf{Pr}[X_0] - \mathsf{Pr}[X_1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH}_{\mathsf{G}_1}}(\lambda)$.

Proof. The solver $\mathcal{B}$ is given the $\mathsf{DDH}_{\mathsf{G}_1}$ problem instance $D = (g_1, g_2, g_1^b, g_1^{bs})$ and the target $T = g_1^{s+\hat{s}}$ where $\hat{s} = 0$ or chosen uniformly at random from $\mathbb{Z}_p^\times$.

Setup. $\mathcal{B}$ chooses $\alpha_1, \alpha_2, (u_i, v_i)_{i\in[0,2m]}, c, d \hookleftarrow \mathbb{Z}_p$. As both α_1 and α_2 are available to $\mathcal{B}$, it can generate $g_T^\alpha = e(g_1^{\alpha_1} \cdot (g_1^b)^{\alpha_2}, g_2)$. Hence, $\mathcal{B}$ outputs the public parameter mpk. Notice that the master secret key msk is available to $\mathcal{B}$.

Phase-I Queries. Since $\mathcal{B}$ knows msk, it can answer with normal secret keys on any query of Ω.

Challenge. Given the challenge data-index set $\Theta^* = (y_1, \ldots, y_\ell)$ for $\ell \leq m$, $\mathcal{B}$ chooses $(t_i)_{i\in[\ell]} \hookleftarrow \mathbb{Z}_p$. It then computes the challenge as κ_0 and $\mathsf{CT}_{\Theta^*} = (\mathsf{C}_0, \mathsf{C}_1, (\mathsf{C}_{2,i}, t_i)_{i\in[\ell]})$ using the problem instance as follows.

$$\kappa_0 = e(\mathsf{C}_0, g_2)^{\alpha_1} \cdot e(\mathsf{C}_1, g_2)^{\alpha_2}, \mathsf{C}_0 = T, \mathsf{C}_1 = g_1^{bs},$$

$$\mathsf{C}_{2,i} = \mathsf{C}_0^{u_0 + u_1 y_i + u_2 y_i^2 + \ldots + u_{2m} y_i^{2m} + c t_i} \cdot \mathsf{C}_1^{v_0 + v_1 y_i + v_2 y_i^2 + \ldots + v_{2m} y_i^{2m} + d t_i}$$

where $i \in [\ell]$. $\mathcal{B}$ then chooses $\kappa_1 \hookleftarrow \mathcal{K}$ and returns $(\kappa_{\mathfrak{b}}, \mathsf{CT}_{\Theta^*})$ as the challenge for $\mathfrak{b} \hookleftarrow \{0, 1\}$.

Phase-II Queries. Same as Phase-I queries.
Guess. $\mathcal{A}$ output $\mathfrak{b}' \in \{0,1\}$. $\mathcal{B}$ outputs 1 if $\mathfrak{b} = \mathfrak{b}'$ and 0 otherwise.

Notice that, if $\hat{s}$ in $\mathsf{DDH}_{\mathsf{G}_1}$ problem instance is 0, then the challenge ciphertext CT_{Θ^*} is normal. Otherwise the challenge ciphertext CT_{Θ^*} is semi-functional. If $\mathcal{A}$ can distinguish these two scenarios, the solver $\mathcal{B}$ will use it to break $\mathsf{DDH}_{\mathsf{G}_1}$ problem. Thus, $|\mathsf{Pr}[X_0] - \mathsf{Pr}[X_1]| \leq \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{DDH}_{\mathsf{G}_1}}(\lambda)$. □

Lemma 5. ($\mathsf{Game}_{2,k-1}$ *to* $\mathsf{Game}_{2,k}$) *For any efficient adversary $\mathcal{A}$ that makes at most q key queries, there exists a PPT algorithm $\mathcal{B}$ such that* $|\mathsf{Pr}[X_{2,k-1}] - \mathsf{Pr}[X_{2,k}]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH}_{\mathsf{G}_2}}(\lambda)$.

Proof. The solver $\mathcal{B}$ is given the $\mathsf{DDH}_{\mathsf{G}_2}$ problem instance $D = (g_1, g_2, g_2^c, g_2^r)$ and the target $T = g_2^{cr+\hat{r}}$ where $\hat{r} = 0$ or chosen uniformly random from $\mathbb{Z}_p^\times$.

Setup. $\mathcal{B}$ chooses $b \hookleftarrow \mathbb{Z}_p^\times$, $\alpha, \alpha_1, w, (p_i, q_i, w_i)_{i\in[0,2m]} \hookleftarrow \mathbb{Z}_p$. It sets $\alpha_2 = b^{-1}(\alpha - \alpha_1)$, $d = b^{-1}(w - c)$, $u_i = p_i + cq_i$, $v_i = b^{-1}(w_i - u_i)$. Note that, as c explicitly is unknown to $\mathcal{B}$, all but α_2 assignment has been done implicitly. The public parameters mpk are generated as $(g_1, g_1^b, g_1^{w_i}, g_1^w, g_{\mathrm{T}}^\alpha)$ where $g_{\mathrm{T}} = e(g_1, g_2)$. Here note that not all of msk is available to $\mathcal{B}$. Still we show that, even without knowing $(d, (u_i, v_i)_{i\in[0,2m]})$ explicitly, $\mathcal{B}$ can simulate the game.

Phase-I Queries. Given the j^{th} key query on Ω_j s.t. $|\Omega_j| = \hbar_j \leq m$,

– If $j > k$: $\mathcal{B}$ has to return a normal key. We already have mentioned that $(d, (u_i, v_i)_{i\in[0,2m]})$ of msk are unavailable to $\mathcal{B}$. Thus $\mathcal{B}$ simulates the normal secret keys as follows.
  $\mathcal{B}$ chooses $r_j \hookleftarrow \mathbb{Z}_p$. Computes the secret key $\mathsf{SK}_{\Omega_j} = (\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4, \mathsf{K}_5)$ where,

$$\mathsf{K}_1 = g_2^{r_j}, \mathsf{K}_2 = (g_2^c)^{r_j},$$

$$\mathsf{K}_3 = g_2^{\alpha_1} \cdot \mathsf{K}_1^{\sum\limits_{x\in\Omega_j}(p_0+p_1x+p_2x^2+\ldots+p_{2m}x^{2m})} \cdot \mathsf{K}_2^{\sum\limits_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_{2m}x^{2m})},$$
$$= g_2^{\alpha_1 + r_j \sum\limits_{x\in\Omega_j}(u_0+u_1x+u_2x^2+\ldots+u_{2m}x^{2m})},$$

$$\mathsf{K}_4 = \mathsf{K}_1^{b^{-1}w} \cdot \mathsf{K}_2^{-b^{-1}} = g_2^{dr_j},$$

$$\mathsf{K}_5 = g_2^{b^{-1}\alpha} \cdot \mathsf{K}_1^{b^{-1}\sum\limits_{x\in\Omega_j}(w_0+w_1x+w_2x^2+\ldots+w_{2m}x^{2m})} \cdot \mathsf{K}_3^{-b^{-1}}$$
$$= g_2^{b^{-1}\alpha + r_j b^{-1}\sum\limits_{x\in\Omega_j}(w_0+w_1x+w_2x^2+\ldots+w_{2m}x^{2m})} \cdot g_2^{-b^{-1}(\alpha_1 + r_j \sum\limits_{x\in\Omega_j}(u_0+u_1x+u_2x^2+\ldots+u_{2m}x^{2m}))},$$
$$= g_2^{\alpha_2 + r_j \sum\limits_{x\in\Omega_j}(v_0+v_1x+v_2x^2+\ldots+v_{2m}x^{2m})}.$$

  Notice that SK_{Ω_j} is identically distributed to the output of $\mathsf{KeyGen}(\mathsf{msk}, \Omega_j)$. Hence $\mathcal{B}$ has managed to simulate the normal secret key without knowing the msk completely.

- If $j < k$: $\mathcal{B}$ has to return a semi-functional secret key. It first creates normal secret keys as above and chooses $\hat{r}, \pi \hookleftarrow \mathbb{Z}_p$ to create semi-functional secret keys following $\mathsf{SFKeyGen}$.
- If $j = k$: $\mathcal{B}$ will use $\mathsf{DDH}_{\mathsf{G}_2}$ problem instance to simulate the secret key. It sets,

$$\mathsf{K}_1 = g_2^r,\ \mathsf{K}_2 = T = g_2^{cr+\hat{r}} = \mathsf{K}_2' \cdot g_2^{\hat{r}},$$
$$\mathsf{K}_3 = g_2^{\alpha_1} \cdot \mathsf{K}_1^{\sum_{x\in\Omega_j}(p_0+p_1x+p_2x^2+\ldots+p_{2m}x^{2m})} \cdot \mathsf{K}_2^{\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_{2m}x^{2m})},$$
$$= g_2^{\alpha_1+r\sum_{x\in\Omega_j}(u_0+u_1x+u_2x^2+\ldots+u_{2m}x^{2m})+\hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_{2m}x^{2m})},$$
$$= \mathsf{K}_3' \cdot g_2^{\hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_{2m}x^{2m})}.$$
$$\mathsf{K}_4 = \mathsf{K}_1^{b^{-1}w} \cdot \mathsf{K}_2^{-b^{-1}} = g_2^{dr} \cdot g_2^{-b^{-1}\hat{r}} = \mathsf{K}_4' \cdot g_2^{-b^{-1}\hat{r}}.$$
$$\mathsf{K}_5 = g_2^{b^{-1}\alpha} \cdot \mathsf{K}_1^{b^{-1}\sum_{x\in\Omega_j}(w_0+w_1x+w_2x^2+\ldots+w_{2m}x^{2m})} \cdot \mathsf{K}_3^{-b^{-1}},$$
$$= g_2^{\alpha_2+r\sum_{x\in\Omega_j}(v_0+v_1x+v_2x^2+\ldots+v_{2m}x^{2m})} \cdot g_2^{-b^{-1}\hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_{2m}x^{2m})}$$
$$= \mathsf{K}_5' \cdot g_2^{-b^{-1}\hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_{2m}x^{2m})}.$$

  Here, $\mathcal{B}$ has implicitly set $\pi = \sum_{x\in\Omega_j}(q_0 + q_1x + q_2x^2 + \ldots + q_{2m}x^{2m})$.

  Notice that if $\hat{r} = 0$ then the key is normal; otherwise it is semi-functional secret key.

Challenge. Given the challenge set Θ^*, of size $\ell \leq m$, $\mathcal{B}$ chooses $s, \hat{s} \hookleftarrow \mathbb{Z}_p$. It then defines the challenge as κ_0 and $\mathsf{CT}_{\Theta^*} = (\mathsf{C}_0, \mathsf{C}_1, (\mathsf{C}_{2,i}, t_i)_{i\in[\ell]})$ such that,

$$\kappa_0 = g_{\mathrm{T}}^{(\alpha s+\alpha_1\hat{s})},\ \mathsf{C}_0 = g_1^{s+\hat{s}},\ \mathsf{C}_1 = g_1^{bs},$$
$$\mathsf{C}_{2,i} = g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_{2m}y_i^{2m}+wt_i)+\hat{s}(u_0+u_1y_i+u_2y_i^2+\ldots+u_{2m}y_i^{2m}+ct_i)},$$
$$= g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_{2m}y_i^{2m}+wt_i)+\hat{s}(p_0+p_1y_i+p_2y_i^2+\ldots+p_{2m}y_i^{2m})} \cdot g_1^{c\hat{s}(q_0+q_1y_i+q_2y_i^2+\ldots+q_{2m}y_i^{2m}+t_i)}.$$

However, g_1^c is not available to $\mathcal{B}$. We here implicitly set $t_i = -(q_0 + q_1y_i + q_2y_i^2 + \ldots + q_{2m}y_i^{2m})$ for each $i \in [\ell]$.

Then, $\mathsf{C}_{2,i} = g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_{2m}y_i^{2m}+wt_i)+\hat{s}(p_0+p_1y_i+p_2y_i^2+\ldots+p_{2m}y_i^{2m})}$ where i^{th} element of the challenge set Θ^* is denoted by y_i. $\mathcal{B}$ then chooses $\kappa_1 \hookleftarrow \mathcal{K}$ and returns $\left(\kappa_{\mathfrak{b}}, \mathsf{C}_0, \mathsf{C}_1, (\mathsf{C}_{2,i}, t_i)_{i\in[\ell]}\right)$ as the challenge. Notice that, the challenge $(\kappa_0, \mathsf{CT}_{\Theta^*})$ is identically distributed to the output of $\mathsf{SFEncrypt}(\mathsf{mpk}, \mathsf{msk}, \Theta^*)$ (i.e. it is semi-functional).

Phase-II Queries. Same as Phase-I queries.

Guess. $\mathcal{A}$ outputs $\mathfrak{b}' \in \{0, 1\}$. $\mathcal{B}$ outputs 1 if $\mathfrak{b} = \mathfrak{b}'$ and 0 otherwise.

As noted earlier, if $\hat{r}$ in $\mathsf{DDH}_{\mathsf{G}_2}$ problem instance is 0, then the k^{th} secret key is normal. Otherwise the k^{th} secret key is semi-functional. Note that, the challenge ciphertext here is constructed semi-functional.

However, we need to argue that the tags $(t_i)_{i\in[\ell]}$ output as part of the challenge ciphertext are uniformly random to the view of adversary $\mathcal{A}$ who has got hold of the semi-functional k^{th} secret key containing π. As we already have defined the semi-functional distributions in Section 5.2.1, the tags that are used in the semi-functional secret key and semi-functional ciphertext, should also be uniformly random and independent. We confirm this in Lemma 7 and conclude that the challenge ciphertext and the k^{th} secret key are properly simulated.

If $\mathcal{A}$ can distinguish normal and semi-functional secret keys, the solver $\mathcal{B}$ will use it to break $\mathsf{DDH}_{\mathsf{G}_2}$ problem. Thus, $|\Pr[X_{2,k-1}] - \Pr[X_{2,k}]| \leq \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{DDH}_{\mathsf{G}_2}}(\lambda)$. $\square$

Lemma 6. ($\mathsf{Game}_{2,q}$ *to* Game_3) *For any efficient adversary $\mathcal{A}$ that makes at most q key queries,* $|\Pr[X_{2,q}] - \Pr[X_3]| \leq 2/p$.

Proof. In $\mathsf{Game}_{2,q}$, all the queried secret keys and the challenge ciphertext are transformed into semi-functional. To argue that the challenge encapsulation key κ is identically distributed to uniformly random G_{T} element, we perform a conceptual change on the parameters of $\mathsf{Game}_{2,q}$.

Setup. Choose $b \hookleftarrow \mathbb{Z}_p^{\times}$, $\alpha_1, \alpha, c, w, (u_i, w_i)_{i\in[0,2m]} \hookleftarrow \mathbb{Z}_p$. Set $\alpha_2 = b^{-1}(\alpha - \alpha_1)$, $d = b^{-1}(w - c)$, $v_i = b^{-1}(w_i - u_i)$. The public parameters are generated as $(g_1, g_1^b, g_1^{w_i}, g_1^w, g_{\mathrm{T}}^{\alpha})$ where $g_{\mathrm{T}} = e(g_1, g_2)$. Notice that g_{T} is independent of α_1 as α was chosen independently.

Phase-I Queries. Given key query on Ω, choose $r, \hat{r}, \pi' \hookleftarrow \mathbb{Z}_p$. Compute the secret key $\mathsf{SK}_\Omega = (\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4, \mathsf{K}_5)$ as follows.

$$\mathsf{K}_1 = g_2^r, \mathsf{K}_2 = g_2^{cr+\hat{r}}, \mathsf{K}_3 = g_2^{\pi'} \cdot g_2^{r \sum\limits_{x\in\Omega}(u_0+u_1x+u_2x^2+\ldots+u_{2m}x^{2m})},$$

$$\mathsf{K}_4 = g_2^{dr-\hat{r}b^{-1}}, \mathsf{K}_5 = g_2^{b^{-1}(\alpha-\pi')} \cdot g_2^{r \sum\limits_{x\in\Omega}(v_0+v_1x+v_2x^2+\ldots+v_{2m}x^{2m})}.$$

The reduction sets $\pi' = \alpha_1 + \hat{r}\pi$. We first argue that π is uniformly random which ensures that the secret key constructed here is semi-functional. Therefore, if $\hat{r} = 0$, π can take any uniformly random value from $\mathbb{Z}_p$. On the other hand, if $\hat{r} \neq 0$, due to the independent random choice of both π' and α_1, π is uniformly random and independent. Therefore no matter what value $\hat{r}$ takes, π is uniformly random and independent. As a result, the secret keys are simulated properly.

Now, we show that the secret key $\mathsf{SK}_\Omega = (\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4, \mathsf{K}_5)$ is independent of α. Notice that both K_3 and K_5 are generated using randomly chosen π' that is independent of α_1 as long as $\hat{r} \neq 0$ and none of the other key components contain α_1. The secret key SK_Ω therefore, is independent of α_1 if $\hat{r} \neq 0$. This happens with probability $1 - 1/p$.

Challenge. On challenge Θ^*, choose $s, \hat{s} \hookleftarrow \mathbb{Z}_p$ and $(t_i)_{i\in[\ell]} \hookleftarrow \mathbb{Z}_p$. Compute the ciphertext $\mathsf{CT}_\Theta = (\kappa_0, \mathsf{C}_0, \mathsf{C}_1, (\mathsf{C}_{2,i}, t_i)_{i\in[\ell]})$ where,

$$\kappa_0 = e(g_1, g_2)^{\alpha s + \alpha_1 \hat{s}} = g_{\mathrm{T}}^{\alpha s} \cdot g_{\mathrm{T}}^{\alpha_1 \hat{s}}, \mathsf{C}_0 = g_1^{s+\hat{s}}, \mathsf{C}_1 = g_1^{bs},$$

$$\mathsf{C}_{2,i} = g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_{2m}y_i^{2m}+wt_i)+\hat{s}(u_0+u_1y_i+u_2y_i^2+\ldots+u_{2m}y_i^{2m}+ct_i)}.$$

Phase-II Queries. Same as Phase-I queries.
Guess. $\mathcal{A}$ outputs $\mathfrak{b}' \in \{0,1\}$. Output 1 if $\mathfrak{b} = \mathfrak{b}'$ and 0 otherwise.

All the scalars used in mpk and $(\mathsf{SK}_{\Omega_i})_{i\in[q]}$ are independent of α_1 as we already have seen. Notice that none of the ciphertext components but κ_0 contain α_1. The entropy due to α_1 thus makes κ_0 random as long as $\hat{s} \neq 0$. In fact, this allows the replacement of κ_0 by a uniform random choice $\kappa_1 \hookleftarrow \mathcal{K}$ provided $\hat{s} \neq 0$. Recall that, this exactly is the situation of Game_3. Thus, $|\mathsf{Pr}[X_{2,q}] - \mathsf{Pr}[X_3]| \leq \mathsf{Pr}[\hat{r} = 0] + \mathsf{Pr}[\hat{s} = 0] \leq 2/p$.
Notice that, $\kappa_{\mathfrak{b}}$ output in Game_3 completely hides $\mathfrak{b}$. Thus, for any adversary $\mathcal{A}$, the advantage $\mathsf{Pr}[X_3] = 1/2$. □

Lemma 7. *Let $(q_1, \ldots, q_{2m}) \hookleftarrow \mathbb{Z}_p^{2m}$. Consider two non-empty sets Ω_k, Θ^* as described in the proof of Lemma 5 above. If $\Omega_k \not\subset \Theta^*$, then π is independent of all the tags $(t_i)_{i\in[\ell]}$ where $\pi = \sum_{x\in\Omega_k} (q_0 + q_1x + q_2x^2 + \ldots + q_{2m}x^{2m})$ and $t_i = -(q_0 + q_1y_i + q_2y_i^2 + \ldots + q_{2m}y_i^{2m})$ for all $y_i \in \Theta^*$.*

Proof. Let us define $A = \Omega_k \setminus \Theta^*$, $B = \Omega_k \cap \Theta^*$ and $C = \Theta^* \setminus \Omega_k$ with $|A| = \delta$, $|B| = \eta$ and $|C| = \gamma$. Clearly, $\delta+\eta = \hbar$ and $\eta+\gamma = \ell$. Furthermore, let us define $f(z) = q_0 + q_1z + q_2z^2 + \ldots + q_{2m}z^{2m}$. Observe that, the polynomial evaluations $\{f(z)\}_{z\in A\uplus B\uplus C}$ result in a system of linear equations $\boldsymbol{f} = \mathbf{M}\boldsymbol{q}$ described in Equation (15) where $\mathbf{M}$ is a Vandermonde matrix of order $(\delta+\eta+\gamma)\times 2m$ with $(\delta+\eta+\gamma) \leq 2m$. Note that, $\mathbf{M}$ contains a sub-matrix $\mathbf{M}'$ that is defined by the first $(\delta+\eta+\gamma)$ columns of $\mathbf{M}$. In other words, $\mathbf{M}'$ is a square vandermonde matrix of order $(\delta+\eta+\gamma)\times(\delta+\eta+\gamma)$ with determinant $\prod_{x\neq x'\in A\uplus B\uplus C} (x-x') \neq 0$.
Thus $\mathbf{M}'$ is a full rank matrix of rank $(\delta+\eta+\gamma)$ and the rank of $\mathbf{M} \geq (\delta+\eta+\gamma)$. As order of $\mathbf{M}$ is $(\delta+\eta+\gamma)\times 2m$, (row-)rank of $\mathbf{M} = (\delta+\eta+\gamma)$. Thus, all the rows of $\mathbf{M}$ are linearly independent.

$$\begin{pmatrix} f(a_1) \\ \vdots \\ f(a_\delta) \\ f(b_1) \\ \vdots \\ f(b_\eta) \\ f(c_1) \\ \vdots \\ f(c_\gamma) \end{pmatrix} = \begin{pmatrix} 1 & a_1 & (a_1)^2 & \cdots & (a_1)^{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_\delta & (a_\delta)^2 & \cdots & (a_\delta)^{2m} \\ 1 & b_1 & (b_1)^2 & \cdots & (b_1)^{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & b_\eta & (b_\eta)^2 & \cdots & (b_\eta)^{2m} \\ 1 & c_1 & (c_1)^2 & \cdots & (c_1)^{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & c_\gamma & (c_\gamma)^2 & \cdots & (c_\gamma)^{2m} \end{pmatrix} \cdot \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ \vdots \\ q_{2m} \end{pmatrix} \tag{15}$$

Notice that, $\{f(b_1), \ldots, f(b_\eta), f(c_1), \ldots, f(c_\gamma)\} = \{t_1, \ldots, t_\ell\}$ and $\pi = f(a_1) + \ldots + f(a_\delta) + f(b_1) + \ldots + f(b_\eta)$. For easier comprehension, we rename the vectors in the above system of linear equation (Equation (15)) as the following:

$$\begin{pmatrix} f(a_1) \\ \vdots \\ f(a_\delta) \\ f(b_1) \\ \vdots \\ f(b_\eta) \\ f(c_1) \\ \vdots \\ f(c_\gamma) \end{pmatrix} = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_\delta \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_\eta \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_\gamma \end{pmatrix} \cdot \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ \vdots \\ q_{2m} \end{pmatrix} \tag{16}$$

Thus, to argue the independence of π and $t_1, \ldots, t_\ell$, it is sufficient to show that $\{\mathbf{b}_1, \ldots, \mathbf{b}_\eta, \mathbf{c}_1, \ldots, \mathbf{c}_\gamma, \sum_{i\in[\delta]} \mathbf{a}_i + \sum_{i\in[\eta]} \mathbf{b}_i\}$ are linearly independent vectors. Suppose that, there exists $\tau_1, \ldots, \tau_\eta, \xi_1, \ldots, \xi_\gamma, \beta \in \mathbb{N}$ such that,

$$\tau_1\mathbf{b}_1 + \ldots + \tau_\eta\mathbf{b}_\eta + \xi_1\mathbf{c}_1 + \ldots + \xi_\gamma\mathbf{c}_\gamma + \beta(\sum_{i\in[\delta]} \mathbf{a}_i + \sum_{i\in[\eta]} \mathbf{b}_i) = 0.$$

Then, $\beta \sum_{i\in[\delta]} \mathbf{a}_i + \sum_{i\in[\eta]} (\beta + \tau_i)\mathbf{b}_i + \sum_{i\in[\gamma]} \xi_i\mathbf{c}_i = 0$. Since, $\mathbf{M}$ in Equation (15) (and therefore in Equation (16)) is a full (row-)rank matrix, $\{\mathbf{a}_1, \ldots, \mathbf{a}_\delta, \mathbf{b}_1, \ldots, \mathbf{b}_\eta, \mathbf{c}_1, \ldots, \mathbf{c}_\gamma\}$ are linearly independent. Thus, $\beta = 0$, all the τ_i are 0 and all the ξ_i are also 0. This proves that, $\{\mathbf{b}_1, \ldots, \mathbf{b}_\eta, \mathbf{c}_1, \ldots, \mathbf{c}_\gamma, \sum_{i\in[\delta]} \mathbf{a}_i + \sum_{i\in[\eta]} \mathbf{b}_i\}$ are linearly independent vectors. As $(q_1, \ldots, q_{2m}) \hookleftarrow \mathbb{Z}_p^{2m}$, the independence of vectors ensures that π is independent of all the tags $(t_i)_{i\in[\ell]}$.

□

5.3 Applications

Katz et al. [23] described a few black-box transformations from small-universe SPE to well known cryptographic protocols. We use those transformations on our SPE_2 to construct new adaptively secure WIBE, WKD-IBE and DNF-ABE protocols. We define a function called Encode that formalizes the black-box transformations [23] and use Encode to construct protocol Π of different functionalities. Given a data-index (resp. key-index), Π.Encrypt (resp. Π.KeyGen) runs Encode to get modified data-index (resp. key-index) followed by SPE_2.Encrypt (resp. SPE_2.KeyGen) on the modified data-index (resp. key-index) to construct ciphertext (resp. secret key). The Π.Decrypt runs SPE_2.Decrypt on given ciphertext and secret key.

IBE with Wildcard. The generic transformation of [23] allows construction of WIBE [1] and WKD-IBE [2]. Recall that, in WIBE the wildcard $*$ is present in the data-index and in WKD-IBE the wildcard $*$ is present in the key-index. To construct a WIBE or WKD-IBE for n-bit identities, we define Encode to convert the n-bit identities into $2n$-bit identities (i.e. identity space $\mathcal{U}' = \{0,1\}^{2n}$). Precisely, Encode takes $z \in \{0,1,*\}^n$ as input and outputs $T_z^W \in \{0,1\}^{2n}$ in case of WIBE and $T_z^D \in \{0,1\}^{2n}$ in case of WKD-IBE.

$$T_z^W[2i, 2i+1] = \begin{cases} 10 & \text{if } z[i] = 1 \\ 01 & \text{if } z[i] = 0 \\ 11 & \text{if } z[i] = * \end{cases} \quad \text{and} \quad T_z^D[2i, 2i+1] = \begin{cases} 10 & \text{if } z[i] = 1 \\ 01 & \text{if } z[i] = 0 \\ 00 & \text{if } z[i] = * \end{cases}.$$

Then $S_z^{(Q)} = \left\{ i \in \mathcal{U}' : T_z^Q[i] = 1 \right\}$ where $Q \in \{W, D\}$. The KeyGen and $\mathsf{Encrypt}$ of WIBE and WKD-IBE are simply $\mathsf{SPE_2.KeyGen}$ and $\mathsf{SPE_2.Encrypt}$ running on such set $S_z^{(Q)}$ respectively. The correctness and security of the constructions follow from that of $\mathsf{SPE_2}$. Note that, other than the way $*$ is processed, the encodings for WIBE and WKD-IBE are the same. Therefore, any SPE based WIBE or WKD-IBE for n-bit identity will have equal number of components in mpk. This also holds for secret key size $|\mathsf{SK}|$ and number of pairing required during $\mathsf{Decrypt}$. Because of the way $*$ is encoded in T^W and T^D, it's obvious that number of bits set to 1 in an $2n$-bit data index will be in the range $[n, 2n]$ and n respectively. This is clearly visible when Tables 1 and 2 are compared together with respect to SPE-based transformation.

CP-ABE. Most interesting application of SPE is that it can realize a secure CP-ABE for DNF formula with constant-size key [23]. Let $F = (C_1 \vee C_2 \vee \cdots C_t)$ be a DNF formula where each C_i is conjunction of attributes i.e. $C_i = \bigwedge_j I_{i,j}$ where $I_{i,j} \in \mathcal{U}$. Since, C_i is conjunction of literals, we can interpret C_i by a set that contains all those literals i.e. $C_i = \{I_{i,j}\}_j \subset \mathcal{U}$. Then an attribute set A satisfies the formula F if $\exists k \in [t]$ such that $C_k \subseteq \mathsf{A}$.

This functionality is achieved by associating the clauses C_i as well as A to corresponding *revocation list* i.e. $\mathcal{U} \setminus C_i$ and $\mathcal{U} \setminus \mathsf{A}$ and perform the subset predicate evaluation: $\mathcal{U} \setminus \mathsf{A} \subseteq \mathcal{U} \setminus C_i$ where $\mathcal{U}$ denotes the attribute universe of size n. Precisely, Encode takes input $Z \in \{C_1, \cdots, C_t, \mathsf{A}\}$ and outputs $S_Z = \{i \in \mathcal{U}' : T_Z[i] = 1\}$ where T_Z is defined as follows:

$$T_Z[i] = \begin{cases} 0 & \text{if } i \in Z, \\ 1 & \text{if } i \notin Z. \end{cases}$$

The KeyGen and $\mathsf{Encrypt}$ of CP-DNF is simply $\mathsf{SPE_2.KeyGen}$ and $\mathsf{SPE_2.Encrypt}$ running on such set S_Z respectively. The correctness and security of the construction follow from that of $\mathsf{SPE_2}$.

Above transformations give us new constructions of WIBE, WKD-IBE and CP-DNF schemes that we denote by $\mathsf{SPE_2}$-WIBE, $\mathsf{SPE_2}$-WKD and $\mathsf{SPE_2}$-DNF respectively. We compare them with existing constructions in terms of size of public key ($|\mathsf{mpk}|$), secret key ($|\mathsf{SK}|$) and ciphertext ($|\mathsf{CT}|$), and also in terms of the number of pairing evaluations ($[\mathsf{P}]$) required in $\mathsf{Decrypt}$. It is evident in Tables 1 and 2 that even if $\mathsf{SPE_2}$-WIBE and $\mathsf{SPE_2}$-WKD-IBE doubles the public parameter size, in terms of performance both are comparable to available selective secure realizations. In other words, SPE_2 leads us to adaptively CPA-secure WIBE and WKD-IBE constructions at a price for which previous constructions only achieved selectively CPA security. The only other adaptively CPA-secure WIBE and WKD-IBE constructions in the standard model are due to Abdalla et al. [1, 2] who used Water's IBE [27] to this end. We call these constructions as Wat-WIBE and Wat-WKD-IBE respectively in Tables 1 and 2. SPE_2-WIBE performs better than Wat-WIBE in all respect but the public parameter size. SPE_2-WKD, on the other hand, performs better in terms of both parameter size and secret key size than Wat-WKD. Precisely, SPE_2-WIBE achieves $\mathcal{O}(n)$ sized master public key and ciphertext, constant-size secret key and only three pairing evaluations during Decrypt achieving an $\mathcal{O}(n)$ improvement over Wat-WIBE [1] on almost all the fronts. On the other hand, SPE_2-WKD-IBE achieves $\mathcal{O}(n)$ sized master public key and constant-size secret key as opposed to $\mathcal{O}(n^2)$ sized master public key and secret key in Wa-WKD-IBE [2]. In case of CP-DNF as can be seen in Table 3, ours is the only scheme that achieves adaptive security under static assumption and still enjoys constant-size secret key.

WIBE Schemes	$\|\mathsf{mpk}\|$	$\|\mathsf{SK}\|$	$\|\mathsf{CT}\|$	Decrypt	Security	Assumption
BBG-WIBE [1]	$(n+4)\mathsf{G}$	$(n+2)\mathsf{G}$	$(n+2)\mathsf{G}+\mathsf{G}_\mathrm{T}$	2[P]	selective	n-BDHI
Wat-WIBE [1]	$((n+1)n+3)\mathsf{G}$	$(n+1)\mathsf{G}$	$((n+1)n+2)\mathsf{G}+\mathsf{G}_\mathrm{T}$	$(n+1)$[P]	adaptive	DBDH
SPE-1-WIBE [23]	$(2n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$\mathsf{G}_2+\mathbb{Z}_p$	$(2n+1)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	1[P]	selective	q-BDHI
SPE-2-WIBE [23]	$(2n+1)\mathsf{G}_1+2\mathsf{G}_2$	$\mathsf{G}_1+\mathsf{G}_2$	$2n\mathsf{G}_1+\mathsf{G}_2+\mathsf{G}_\mathrm{T}$	2[P]	selective	DBDH
SPE_2-WIBE	$(4n+8)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$5\mathsf{G}_2$	$(2n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}+2n\mathbb{Z}_p$	3[P]	adaptive	SXDH

Table 1 Comparison of efficient standard model WIBE schemes for n-bit identity.

WKD-IBE Schemes	$\|\mathsf{mpk}\|$	$\|\mathsf{SK}\|$	$\|\mathsf{CT}\|$	Decrypt	Security	Assumption
BBG-WKD [2]	$(n+4)\mathsf{G}$	$(n+2)\mathsf{G}$	$2\mathsf{G}+\mathsf{G}_\mathrm{T}$	2[P]	selective	n-BDHE
Wat-WKD [2]	$((n+1)n+3)\mathsf{G}$	$(n(n+1)+2)\mathsf{G}$	$2\mathsf{G}+\mathsf{G}_\mathrm{T}$	2[P]	adaptive	DBDH
SPE-1-WKD [23]	$(2n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$\mathsf{G}_2+\mathbb{Z}_p$	$(n+1)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	1[P]	selective	q-BDHI
SPE-2-WKD [23]	$(2n+1)\mathsf{G}_1+2\mathsf{G}_2$	$\mathsf{G}_1+\mathsf{G}_2$	$n\mathsf{G}_1+\mathsf{G}_2+\mathsf{G}_\mathrm{T}$	2[P]	selective	DBDH
SPE_2-WKD	$(4n+8)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$5\mathsf{G}_2$	$(n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}+n\mathbb{Z}_p$	3[P]	adaptive	SXDH

Table 2 Comparison of efficient standard model WKD-IBE schemes for n-bit identity.

DNF Schemes	$\|\mathsf{mpk}\|$	$\|\mathsf{SK}\|$	$\|\mathsf{CT}\|$	Decrypt	Security	Assumption
SPE-1-DNF [23]	$(n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$\mathsf{G}_2+\mathbb{Z}_p$	$\gamma((n+1)\mathsf{G}_1+\mathsf{G}_\mathrm{T})$	1[P]	selective	q-BDHI
SPE-2-DNF [23]	$(n+1)\mathsf{G}_1+2\mathsf{G}_2$	$\mathsf{G}_1+\mathsf{G}_2$	$\gamma(2n\mathsf{G}_1+\mathsf{G}_2+\mathsf{G}_\mathrm{T})$	2[P]	selective	DBDH
SPE_2-DNF	$(2n+4)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$5\mathsf{G}_2$	$\gamma((n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}+n\mathbb{Z}_p)$	3[P]	adaptive	SXDH

Table 3 Comparison of efficient standard model DNF schemes. Here size of the universe is n and γ is the number of disjunctive clauses in a DNF formula.

6 Conclusion

We presented two constructions of large universe subset predicate encryption (SPE). Both the constructions achieve constant-size secret key and efficient decryption. First construction achieves constant-size ciphertext as well and is proven selectively secure in a restricted model. Our second and main construction achieves adaptive security in the asymmetric prime order bilinear group setting under the SXDH assumption. The ciphertext size in this construction is linear in the set size it is intended to. It is an interesting open problem to design an SPE with constant-size ciphertext without the kind of restriction we imposed in the selective security model so is any improvement of our second construction in terms of the ciphertext size.

Acknowledgement. We are grateful to Atsushi Takayasu for pointing out an error in the original security argument of SPE_2.

References

1. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: International Colloquium on Automata, Languages, and Programming. LNCS, vol. 4052, pp. 300–311. Springer (2006)
2. Abdalla, M., Kiltz, E., Neven, G.: Generalized key delegation for hierarchical identity-based encryption. In: European Symposium on Research in Computer Security. LNCS, vol. 4734, pp. 139–154. Springer (2007)
3. Ateniese, G., Camenisch, J., Hohenberger, S., de Medeiros, B.: Practical group signatures without random oracles. IACR Cryptology ePrint Archive 2005, 385 (2005), `https://eprint.iacr.org/2005/385`
4. Attrapadung, N.: Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In: EUROCRYPT. LNCS, vol. 8441, pp. 557–577. Springer (2014)
5. Attrapadung, N.: Dual System Encryption Framework in Prime-Order Groups. IACR Cryptology ePrint Archive 2015, 390 (2015), `https://eprint.iacr.org/2015/390`
6. Attrapadung, N., Libert, B.: Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In: International Workshop on Public Key Cryptography. LNCS, vol. 6056, pp. 384–402. Springer (2010)
7. Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: EUROCRYPT. LNCS, vol. 3027, pp. 223–238. Springer (2004)
8. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: CRYPTO. LNCS, vol. 2139, pp. 213–229. Springer (2001)
9. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: CRYPTO. LNCS, vol. 3621, pp. 258–275. Springer (2005)
10. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Theory of Cryptography Conference. LNCS, vol. 3378, pp. 325–341. Springer (2005)

11. Boyen, X.: General ad hoc encryption from exponent inversion IBE. In: EUROCRYPT. LNCS, vol. 4515, pp. 394–411. Springer (2007)
12. Chase, M., Meiklejohn, S.: Déjà Q: Using dual systems to revisit q-type assumptions. In: EUROCRYPT. LNCS, vol. 8441, pp. 622–639. Springer (2014)
13. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: EUROCRYPT. LNCS, vol. 9057, pp. 595–624. Springer (2015)
14. Chen, J., Gong, J.: ABE with tag made easy. In: ASIACRYPT. LNCS, vol. 10625, pp. 35–65. Springer (2017)
15. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: ASIACRYPT. LNCS, vol. 4833, pp. 200–215. Springer (2007)
16. Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: International Conference on Pairing-Based Cryptography. LNCS, vol. 4575, pp. 39–59. Springer (2007)
17. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics 156(16), 3113 – 3121 (2008), applications of Algebra to Cryptography
18. Gong, J., Libert, B., Ramanna, S.C.: Compact IBBE and Fuzzy IBE from Simple Assumptions. In: International Conference on Security and Cryptography for Networks. LNCS, vol. 11035, pp. 563–582. Springer (2018)
19. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security. pp. 89–98. ACM (2006)
20. Håstad, J., Impagliazzo, R., Levin, L., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999)
21. Jao, D., Yoshida, K.: Boneh-Boyen signatures and the strong Diffie-Hellman problem. In: International Conference on Pairing-Based Cryptography. LNCS, vol. 5671, pp. 1–16. Springer (2009)
22. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. Journal of Cryptology 30(4), 1116–1156 (2017)
23. Katz, J., Maffei, M., Malavolta, G., Schröder, D.: Subset predicate encryption and its applications. In: International Conference on Cryptology and Network Security. LNCS, vol. 11261, pp. 115–134. Springer (2017)
24. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: EUROCRYPT. LNCS, vol. 4965, pp. 146–162. Springer (2008)
25. Ramanna, S.C., Sarkar, P.: Efficient adaptively secure IBBE from the SXDH assumption. IEEE Transactions on Information Theory 62(10), 5709–5726 (2016)
26. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: EUROCRYPT. LNCS, vol. 3494, pp. 457–473. Springer (2005)
27. Waters, B.: Efficient identity-based encryption without random oracles. In: EUROCRYPT. LNCS, vol. 3494, pp. 114–127. Springer (2005)
28. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: CRYPTO. LNCS, vol. 5677, pp. 619–636. Springer (2009)
29. Wee, H.: Dual system encryption via predicate encodings. In: Theory of Cryptography Conference. LNCS, vol. 8349, pp. 455–479. Springer (2014)
30. Wee, H.: Déjà Q: Encore! un petit IBE. In: Theory of Cryptography Conference-II. LNCS, vol. 9563, pp. 237–258. Springer (2016)