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 selective secure constructions of SPE in the small universe settings. Their first construction is based on q-type 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 without random oracle. 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 adaptive 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 our constructions have efficient decryption function suggesting their practical applicability. Thus the primitives like WIBE and ABE resulting through black-box transformation of our constructions become more practical.

1 Introduction

The notion of Identity-Based Encryption (IBE) [7] was generalized by Katz et al. [21] to Predicate Encryption (PE). PE 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) [17] is another example of predicate encryption that emulates boolean function in the encrypted domain. One can view Broadcast Encryption (BE) [8] as a simpler form of ABE where the predicate evaluated is disjunction in the form of membership checking.

Katz et al. [20] recently introduced another primitive called Subset Predicate Encryption (SPE) that allows checking for *subset containment* in the encrypted domain. More formally, in an SPE, a key (SK) associated with a key-index set (Ω) can decrypt a ciphertext (CT) associated with data-index set (Θ) 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 input a set of identities Ω allowing a subset based testing during decryption. It is trivial to achieve subset containment check through multiple membership checks.

Thus, one may be tempted to use an efficient BE instantiation [8] to construct a small-universe 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. $\mathsf{SK}_\Omega = (\mathsf{SK}_{x_1}, \ldots, \mathsf{SK}_{x_\Bbbk})$ where $\Omega = (x_1, \ldots, x_\Bbbk)$. However, such a realization of SPE suffers from an obvious security issue. Given a ciphertext CT_Θ, an unprivileged user having secret key SK_Ω (for $\Omega \nsubseteq \Theta$), can easily derive a valid key by stripping the SK_Ω as long as $\Omega \cap \Theta \neq \phi$.

In their work, Katz et al. [20] discussed and then ruled out a few generic techniques to construct small-universe SPE from Inner-Product Encryption (IPE), Wildcard Identity-Based Encryption (WIBE) and Fuzzy Identity-Based Encryption (FIBE) due to the reason of inefficiency. They proposed two dedicated SPE constructions in the small universe settings. Both the constructions achieve 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 *inversion exponent* technique [9] and the second one utilized the *commutative blinding* technique [6]. 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 second construction is based on a static assumption (DBDH). The second construction of [20] can be easily modified to achieve selective security in large universe setting in the random oracle model.

Given the above results of [20], the main open question in the context of SPE is the following. Can we realize an adaptively secure SPE in the large universe setting without random oracle where security is based on some static assumption? 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 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 particular, the KeyGen of IBBE always takes a singleton set as input. However, trivially extending the KeyGen of IBBE to that of SPE may 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, the corresponding natural restriction in the context of SPE would be that none of the set of identities queried in the key extraction phase should be a subset of the challenge identity set.

A constant-size ciphertext IBBE was proposed in [13] based on q-type assumption in the random oracle model. Recently, Gong et al. [16] proposed integration of [13] and Déjà Q [26] towards selective secure IBBE with constant-size ciphertext under static subgroup decision assumptions. However, unlike the IBBE KeyGen that encodes a single identity, the KeyGen in SPE encodes a set Ω into a secret key of constant-size. We notice that the KeyGen of [16] can be tweaked appropriately to generate a constant-size secret key corresponding to a set. This way we arrive at our first construction SPE_1, a constant-size ciphertext SPE in the large universe setting without random oracle.

The security reduction, closely follows that of [16]. However, the reduction faces additional hurdles in order to properly simulate KeyGen of SPE. In the usual IBBE scenario, for a challenge ciphertext CT_{Θ^*}, adversary is not allowed to make secret key queries on $x \in \Theta^*$. In case of SPE, however, it is possible to have some $x \in \Omega \cap \Theta^*$. In other words, the simulator in our SPE security argument should be able to answer for key extraction queries which were naturally ruled out in IBBE security model considered in [16].

Our Déjà Q based security argument is able to achieve the following – (i) the effect of the terms encoding $x \in (\Theta^* \cap \ \Omega)$ gets nullified naturally and (ii) takes into consideration of the effect of availability of admissible $\mathsf{Aggregate}$ function [14] to adversary. This, however, comes with a restriction on the KeyGen queries (also due to the Déjà Q approach). Informally speaking, we need the sets that are queried for key extraction: $(\Omega_1, \Omega_2, \ldots, \Omega_q)$ to be *cover-free sets* i.e. for any $i \in [q]$, $\Omega_i \setminus (\bigcup_{j\in[q]\setminus\{i\}} \Omega_j) \neq \phi$.

While pairing-based adaptive secure IBBE achieveing constant size secret key as well as ciphertext remains still as an open problem; our above result indicates the limitations of the available techniques to argue even selective security for constant size ciphertext SPE.

Our main construction (SPE_2) achieves adaptive security in the prime order groups under SXDH with constant-size secret key. This construction resembles IBBE structure of [22] which extended JR-IBE [19] to achieve an efficient tag-based IBBE construction. We tweak the KeyGen algorithm of their IBBE $_1$ [22] to realize adaptive secure SPE in the large universe settings. Again, the non-triviality lies in the security argument. Precisely, in the security model of [22], for a challenge set $\Theta^* = (y_1, \ldots, y_\ell)$, the set of identities queried for key extraction should be strictly non-overlapping. However, in the security argument of (SPE_2), the query (Ω) adversary makes may contain some elements that also belong to the challenge set Θ^*.

We are able to realize the first large universe adaptive secure SPE without random oracle. Our construction is quite efficient too in terms of parameter size, encryption and decryption cost. For example, encryption does not require any pairing evaluation while decryption evaluates only 3 pairings. The only limitation is the obvious: ciphertext size depends on the size of the privileged set it is intended to.

We briefly discuss the effect of black-box transformations of Katz et al. [20] on our SPE_2 constructions. We achieve first adaptive secure CP-DNF (CP-ABE with DNF policy) evaluation with constant-size secret key. We present the comparison with state of the art in Table 1 and Table 2.

Organization of the Paper. In Section 2 we recall few definitions and present the notations that will be followed in this paper. In Section 3 we define the subset predicate encryption (SPE) and its security model. In Section 4 and in Section 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 $\mathsf{Adv}^i_{\mathcal{A}}(\lambda)$ to denote the advantage adversary $\mathcal{A}$ has in security game Game_i 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 an $(n+3)$-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 Type-1 bilinear pairing. 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 $e(g_i, g_j) = 1$ if $i \neq j$.

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 Type-3 bilinear pairing [15].

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 [26] as follows:

$\mathsf{DS1}$. $\{D, T_0\} \approx_{\epsilon_{\mathsf{DS1}}} \{D, T_1\}$ for $T_0 \hookleftarrow \mathsf{G}_{p_1}$ and $T_1 \hookleftarrow \mathsf{G}_{p_1p_2}$ given $D = (g_1, g_3, g_{12})$ where $g_1 \hookleftarrow \mathsf{G}_{p_1}^\times$, $g_3 \hookleftarrow \mathsf{G}_{p_3}^\times$ and $g_{12} \hookleftarrow \mathsf{G}_{p_1p_2}$. In other words, the advantage of any adversary $\mathcal{A}$ to solve the $\mathsf{DS1}$ is

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DS1}}(\lambda) = |\mathsf{Pr}[\mathcal{A}(D, T_0) \to 1] - \mathsf{Pr}[\mathcal{A}(D, T_1) \to 1]| \leq \epsilon_{\mathsf{DS1}}.$$

$\mathsf{DS1}$ is hard if advantage of $\mathcal{A}$ is negligible i.e. $\epsilon_{\mathsf{DS1}} \leq \mathsf{neg}(\lambda)$.

$\mathsf{DS2}$. $\{D, T_0\} \approx_{\epsilon_{\mathsf{DS2}}} \{D, T_1\}$ for $T_0 \hookleftarrow \mathsf{G}_{p_1p_3}$ and $T_1 \hookleftarrow \mathbb{G}$ given $D = (g_1, g_3, g_{12}, g_{23})$ where $g_1 \hookleftarrow \mathsf{G}_{p_1}^\times$, $g_3 \hookleftarrow \mathsf{G}_{p_3}^\times$, $g_{12} \hookleftarrow \mathsf{G}_{p_1p_2}$ and $g_{23} \hookleftarrow \mathsf{G}_{p_2p_3}$. In other words, the advantage of any adversary $\mathcal{A}$ to solve the $\mathsf{DS2}$ is

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DS2}}(\lambda) = |\mathsf{Pr}[\mathcal{A}(D, T_0) \to 1] - \mathsf{Pr}[\mathcal{A}(D, T_1) \to 1]| \leq \epsilon_{\mathsf{DS2}}.$$

$\mathsf{DS2}$ is hard if advantage of $\mathcal{A}$ is negligible i.e. $\epsilon_{\mathsf{DS2}} \leq \mathsf{neg}(\lambda)$.

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

Symmetric External Diffie-Hellman Assumption (SXDH). The SXDH assumption in group $(\mathsf{G}_1, \mathsf{G}_2)$ is: DDH in G_1 and DDH in G_2 is hard. We rewrite DDH in G_1 in the form of 1-Lin assumption below and call it $\mathsf{DDH}_{\mathsf{G}_1}$. The $\mathsf{DDH}_{\mathsf{G}_2}$ denotes the DDH problem in G_2.

- $\mathsf{DDH}_{\mathsf{G}_1}$: $\{D, T_0\} \approx_{\epsilon_{\mathsf{DDH}_{\mathsf{G}_1}}} \{D, T_1\}$ for $T_0 = g_1^s$ and $T_1 = g_1^{s+\hat{s}}$ given $D = (g_1, g_2, g_1^b, g_1^{bs})$ where $g_1 \hookleftarrow \mathsf{G}_1$, $g_2 \hookleftarrow \mathsf{G}_2$, $b \hookleftarrow \mathbb{Z}_p^\times$, $s, \hat{s} \hookleftarrow \mathbb{Z}_p$. In other words, the advantage of any adversary $\mathcal{A}$ to solve the $\mathsf{DDH}_{\mathsf{G}_1}$ is

  $$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}_{\mathsf{G}_1}}(\lambda) = |\mathsf{Pr}[\mathcal{A}(D, T_0) \to 1] - \mathsf{Pr}[\mathcal{A}(D, T_1) \to 1]| \leq \epsilon_{\mathsf{DDH}_{\mathsf{G}_1}}.$$

  $\mathsf{DDH}_{\mathsf{G}_1}$ is hard if advantage of $\mathcal{A}$ is negligible i.e. $\epsilon_{\mathsf{DDH}_{\mathsf{G}_1}} \leq \mathsf{neg}(\lambda)$.
- $\mathsf{DDH}_{\mathsf{G}_2}$: $\{D, T_0\} \approx_{\epsilon_{\mathsf{DDH}_{\mathsf{G}_2}}} \{D, T_1\}$ for $T_0 = g_2^{cr}$ and $T_1 = g_2^{cr+\hat{r}}$ given $D = (g_1, g_2, g_2^c, g_2^r)$ where $g_1 \hookleftarrow \mathsf{G}_1$, $g_2 \hookleftarrow \mathsf{G}_2$, $c, r, \hat{r} \hookleftarrow \mathbb{Z}_p$. In other words, the advantage of any adversary $\mathcal{A}$ to solve the $\mathsf{DDH}_{\mathsf{G}_2}$ is

  $$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}_{\mathsf{G}_2}}(\lambda) = |\mathsf{Pr}[\mathcal{A}(D, T_0) \to 1] - \mathsf{Pr}[\mathcal{A}(D, T_1) \to 1]| \leq \epsilon_{\mathsf{DDH}_{\mathsf{G}_2}}.$$

  $\mathsf{DDH}_{\mathsf{G}_2}$ is hard if advantage of $\mathcal{A}$ is negligible i.e. $\epsilon_{\mathsf{DDH}_{\mathsf{G}_2}} \leq \mathsf{neg}(\lambda)$.

3 Subset Predicate Encryption

We rephrase Subset Predicate Encryption (SPE) in terms of a predicate encryption [21] and formally model its security requirement.

3.1 Subset Predicate Encryption (SPE)

Let $\mathcal{ID}$ be the identity space. For a key-index set $\Omega \in \mathcal{X} \subset \mathcal{ID}$ and a data-index set $\Theta \in \mathcal{Y} \subset \mathcal{ID}$, 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 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 master secret key msk and public key mpk.
- KeyGen: It takes mpk, msk and key-index set $\Omega \in \mathcal{X}$ of size $k \leq m$ as input. It generates secret key $\mathsf{SK} \in \mathcal{SK}$ corresponding to key-index set Ω.
- $\mathsf{Encrypt}$: It takes mpk, data-index set $\Theta \in \mathcal{Y}$ of size $\ell \leq m$ as input. It generates encapsulation key $\kappa \in \mathcal{K}$ and 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 constructions [4, 22].

3.2 Security Notions

Adaptive CPA-Security **of SPE.** The security game for adaptive CPA-Security for SPE (SPE) is defined as following:

- **Setup**: The challenger gives mpk to adversary $\mathcal{A}$ and keeps msk as secret.
- **Query Phase-I**: Given a key-index Ω, challenger returns $\mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \Omega)$.
- **Challenge**: The adversary ($\mathcal{A}$) provides challenge data-index Θ^* (such that $\mathsf{R}_s(\Omega, \Theta^*) = 0$ for all previous key queries). Then challenger generates (κ_0, 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 $\mathsf{R}_s(\Omega, \Theta^*) = 0$, challenger returns $\mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \Omega)$.
- **Guess**: Adversary $(\mathcal{A})$ outputs its guess $\mathfrak{b}' \in \{0,1\}$ and wins if $\mathfrak{b} = \mathfrak{b}'$.

For any adversary $\mathcal{A}$,

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

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

4 SPE_1: Realizing Constant Size Ciphertext

We present 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 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 $N = p_1p_2p_3$. Then pick $\alpha, \beta \hookleftarrow N$, generators $g_1, u \hookleftarrow \mathsf{G}_{p_1}$ and $g_3 \hookleftarrow \mathsf{G}_{p_3}$. Choose $\mathsf{R}_{3,i} \hookleftarrow \mathsf{G}_{p_3}$ for all $i \in [m]$. Define the $\mathsf{msk} = (\alpha, \beta, u, g_3)$ and the public parameter is

$$\mathsf{mpk} = (g_1, g_1^\beta, \left(G_i = g_1^{\alpha^i}, U_i = u^{\alpha^i} \cdot \mathsf{R}_{3,i}\right)_{i\in[m]}, e(g_1, u)^\beta, \mathsf{H})$$

  where $\mathsf{H} : \mathsf{G}_\mathrm{T} \to \{0,1\}^{\mathsf{poly}(\lambda)}$ is a randomly chosen universal hash function.
- $\mathsf{KeyGen}(\mathsf{msk}, \Omega)$: Given a set Ω, such that $|\Omega| = \hbar \leq m$; define the polynomial $P_\Omega(z) = \prod_{x\in\Omega}(z + x) = d_0 + d_1z + d_2z^2 + \ldots + d_\hbar z^\hbar$, pick $X_3 \hookleftarrow \mathsf{G}_{p_3}$ and define secret key as

$$\mathsf{SK}_\Omega = u^{\frac{\beta}{P_\Omega(\alpha)}} \cdot X_3 = u^{\frac{\beta}{\prod_{x\in\Omega}(\alpha+x)}} \cdot X_3.$$

- $\mathsf{Encrypt}(\mathsf{mpk}, \Theta)$: Given a set Θ, such that $|\Theta| = \ell \leq m$; the polynomial $P_\Theta(z) = \prod_{y\in\Theta}(z + y) = c_0 + c_1z + c_2z^2 + \ldots + c_\ell z^\ell$. 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,

$$A = e(\mathsf{C}_0, \prod_{i\in[t]} U_i^{a_i}) = e(g_1^{s\beta}, u^{P_{\Theta\setminus\Omega}(\alpha)-a_0}) = e(g_1, u)^{s\beta\left(P_{\Theta\setminus\Omega}(\alpha)-a_0\right)},$$

$$B = e(\mathsf{C}_1, \mathsf{SK}_\Omega) = e(g_1^{sP_\Theta(\alpha)}, u^{\frac{\beta}{P_\Omega(\alpha)}} \cdot X_3) = e(g_1, u)^{s\beta P_{\Theta\setminus\Omega}(\alpha)}.$$

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

As we already have mentioned, one can view SPE as a generalization of IBBE [13]. Recently Gong et al. [16] used Déjà Q to prove their identity-based broadcast encryption selective secure in the standard model. The crux of their proof lies in the independence of the semi-functional component of the secret keys (SK_Ω) and semi-functional components of the related public parameters $(U_i)_{i\in[m]}$. To argue that, they showed corresponding matrix representation to be non-singular (in game G_5) during the hybrid argument of the proof of [16, Theorem 1]. The proof made an implicit natural assumption that none of the secret key queries get repeated. Otherwise, the matrix will have more than one identical rows that encode the same key-index. The matrix in such case is singular and the proof fails.

SPE, being a generalization of IBBE, allows key queries on sets where same key-index can appear in different key queries. Precisely, the adversary in case of SPE, can make key extraction queries on Ω_i and Ω_j for $\Omega_i \cap \Omega_j \neq 0$. This introduces a problem here due to dependency among the secret keys of SPE_1. As a result, the matrix in Game_6 (an intermediate game that we define in the hybrid argument to prove Theorem 1) might become singular. Here, we take a simple example to show this problem in light of admissible $\mathsf{Aggregate}$ [14].

In [14], an efficient algorithm called $\mathsf{Aggregate}$ was introduced. Given finite sets $S = (x_i)_{i\in I}$ and $H = \left(h^{\frac{1}{z+x_i}}\right)_{x_i\in S}$, $\mathsf{Aggregate}$ outputs $h^{\frac{1}{\prod_{x_i\in S}(z+x_i)}}$ where I is finite set of indices on S, z is the indeterminant, h is an element from cyclic group W and $x_i \in [\mathsf{ord}(h)]$. Note that this holds for any cyclic group (W) unless there exists distinct $x_i, x_j \in S$ but $x_i - x_j = 0 \bmod \mathsf{ord}(W)$.

Now, notice that the secret keys of SPE_1 allow collusion similar to [14, 16]. But such collusions did not create any problem in [14, 16] as their KeyGen takes singleton key-index. On the other hand, as $\mathsf{SPE}_1.\mathsf{KeyGen}$ takes set as input, collusion due to $\mathsf{Aggregate}$ creates the following problem. Suppose the adversary of

SPE_1 makes following three queries: $\Omega_1 = \{1,2\}$, $\Omega_2 = \{1,3\}$ and $\Omega_3 = \{2,3\}$. Given SK_{Ω_1} and SK_{Ω_2}, the adversary can easily compute SK_{Ω} using Aggregate function where $\Omega = \{1,2,3\}$. Moreover, given SK_{Ω_2} and SK_{Ω_3}, the adversary can also compute same key SK_{Ω} using Aggregate function. For the query sequence considered above, during the proof of Lemma 3 (in Section 4.2.1) which is at the core of the proof of indistinguishability of Game_5 and Game_6, the matrix $\mathbf{P}'$ (and subsequently $\mathbf{A}$ in Lemma 2) precisely would be singular. Notice that, given $(\mathsf{SK}_{\Omega_i})_{i\in I}$, one can use Aggregate in a cascading manner to get secret keys corresponding to other sets as well. We formally define the *claw due to* Aggregate as following: there exists $\Omega_i, \Omega_j, \Omega_k$[1] $\subset \mathcal{ID}$ such that adversary has acquired secret key on all three of them and $\mathsf{Aggregate}(\mathsf{SK}_{\Omega_i}, \mathsf{SK}_{\Omega_j}) = \mathsf{Aggregate}(\mathsf{SK}_{\Omega_j}, \mathsf{SK}_{\Omega_k})$. In case the query sequence has such a claw, the matrix $\mathbf{P}'$ becomes singular and the proof fails. The easiest work-around would be to ensure that no two queries have any element common i.e. $\Omega_i \cap \Omega_j = \phi$ for all distinct $i, j \in [q]$.

We put a much weaker restriction on the adversary where we allow making key queries only on *cover-free* sets. Formally, after making a challenge query Θ^*, adversary $\mathcal{A}$ is allowed to make key extraction queries on $(\Omega_1, \Omega_2, \ldots, \Omega_q)$ adaptively with two restrictions. For all $i \in [q]$, the following must hold:

1. $\Omega_i \not\subset \Theta^*$,
2. $\Omega_i \setminus (\bigcup_{j\in[q]\setminus\{i\}} \Omega_j) \neq \phi$.

Notice that, the first is the natural restriction on the relation between challenge set Θ^* with secret key queries $\{\Omega_i\}_{i\in[q]}$. We say, SPE is selective* Ind-CPA secure (aka $\mathsf{s^*IND\text{-}CPA}$) if for any PPT adversary $\mathcal{A}$ that gives out the challenge Θ during **Init** and the queries it make following the above-mentioned restrictions, $\mathsf{Adv}^{\mathsf{SPE}}_{\mathcal{A},\mathsf{s^*IND\text{-}CPA}}(\lambda) \leq \mathsf{neg}(\lambda)$.

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 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 Θ^*, the natural restriction ensures $\Omega_i, \Omega_j \not\subset \Theta^*$ and therefore $\Omega \not\subset \Theta^*$. Naturally, the resulting Ω is a valid key-indexset. Thus, even if the Aggregate function is used to compute SK_{Ω} from SK_{Ω_i} and SK_{Ω_j}, it does not help the adversary in any way to break the security of the scheme. We reiterate that, we do not put any restriction on the relation between challenge Θ^* and secret-key queries Ω apart from the natural restriction mentioned above. This $\mathsf{s^*IND\text{-}CPA}$ model in this respect behaves exactly the same as sIND-CPA model.

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

$$\mathsf{Adv}^{\mathsf{SPE}_1}_{\mathcal{A},\mathsf{s^*IND\text{-}CPA}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{B}_1}(\lambda) + (m+q+2) \cdot \mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{B}_2}(\lambda) + \frac{((m+q)(m+q+1)+1)}{p_2} + 2^{-\lambda}.$$

[1] Atleast two of $\{\Omega_i, \Omega_j, \Omega_k\}$ are distinct.

Proof Sketch. 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 [25] porting of Déjà Q framework introduced by Chase and Meiklejohn [10]. Intuitively, in the first game Game_0, both the challenge ciphertext and secret keys are normal. We define three intermediate games ($\mathsf{Game}_1, \mathsf{Game}_2,$ and Game_3) to change the ciphertext to semi-functional in Game_4. 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 enough randomness into the semi-functional components of secret key and few related public parameters. Note that till this point, we mostly have followed [16]. Such a sub-sequence effectively introduces enough entropy in the semi-functional component such that we can replace it by pure random choice in Game_6. The structure here is more involved than [16] and we find a trick (namely key-queries on *cover-free* sets only) that is necessary and sufficient to complete the security argument. Finally, in Game_7, we show that semi-functional components as a whole supply enough entropy to hide encapsulation key κ. The detailed proof is given next. □

4.2.1 Formal Proof of Theorem 1

Proof. As already mentioned above, the proof is established via a hybrid argument. Intuitively, in the first game Game_0, both the challenge ciphertext and secret keys are normal. The Game_1 differs from the Game_0 as we introduce some simplifying natural restriction. The Game_2 reformulates the ciphertext to different representation that will be useful in later games. In Game_3, we replace challenge ciphertext component C_0 with a random G_{p_1} element. Then we introduce semi-functional component in the challenge ciphertext in Game_4. The secret keys are then changed to semi-functional in a series of games where the k^{th} game is denoted by its two sub-division $\mathsf{Game}_{5,k,0}$ and $\mathsf{Game}_{5,k,1}$. We perform another conceptual change this time on the semi-functional components of public parameters as well as the secret keys in Game_6. In the final game Game_7, we show that the semi-functional components supply enough entropy to hide the encapsulation key κ.

Aggregate Algorithm. Note that, the functionality of $\mathsf{Aggregate}$ [14] function essentially boils down to computing $R = \frac{1}{\prod\limits_{x_i \in S}(z+x_i)}$ using only *linear operations* given $(S, \hat{H})$ where $\hat{H} = \left(\frac{1}{z+x_i}\right)_{x_i \in S}$. Notice that this is a reversible process in the sense, given (S, R) as defined above, one can efficiently find out the linear transformation of $\hat{H}$ that resulted in R. Precisely, given (S, R) where $S = (x_i)_{i \in I}$ and $R = \frac{1}{\prod\limits_{x_i \in S}(z+x_i)}$, we can express R as following [18]. This representation will be required in the proof.

$$R = \frac{1}{\prod_{x_i \in S}(z+x_i)} = \sum_{x_i \in S} \frac{1}{\prod_{x_j \in S \setminus \{x_i\}} (x_j - x_i)} \cdot \frac{1}{z+x_i}. \tag{1}$$

Let the adversary $\mathcal{A}$ make challenge query on Θ^* and q many queries on the sets $(\Omega_1, \Omega_2, \dots, \Omega_q)$ where $\Omega_i \not\subseteq \Theta^*$ for all $i \in [q]$. Let us denote $\Theta^* = \{y_1, y_2, \dots, y_\ell\}$ and $\Omega_i = \{x_{i,1}, \dots, x_{i,k_i}\}$ 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 E_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 assumption $\mathsf{DS1}$ 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 assumption $\mathsf{DS2}$ by computing $\mathsf{gcd}((x - x'), N)$.

Therefore, $|\mathsf{Pr}[E_1] - \mathsf{Pr}[E_0]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DS1}}(\lambda) + \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DS2}}(\lambda)$.

Game_2. We perform a conceptual change to Game_1 here. Given the challenge $\Theta^* = \{y_1, \dots, y_\ell\}$, pick $\alpha, \tilde{\beta}, u \hookleftarrow \mathbb{Z}_N^2 \times \mathsf{G}_{p_1}$. Define polynomial $P_{\Theta^*}(z) = \prod_{y \in \Theta^*} (z+y)$. 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 is defined the same as Game_1. The secret keys corresponding to Ω_i is $\mathsf{SK}_{\Omega_i} = u^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot X_3$ for $i \in [q]$. The ciphertext is

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

where $U_0 = u \cdot \mathsf{R}_3$ for $\mathsf{R}_3 \hookleftarrow \mathsf{G}_{p_3}$. Note that, the $\beta = \tilde{\beta} \cdot P_{\Theta^*}(\alpha) \mod N$ replacement doesn't change the ciphertext distribution as $\tilde{\beta}$ is uniformly random and $P_{\Theta^*}(\alpha) \neq 0 \mod p_1$. Therefore, $\mathsf{Pr}[E_2] = \mathsf{Pr}[E_1]$.

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 ciphertext. Therefore, $\mathsf{Pr}[E_3] = \mathsf{Pr}[E_2]$.

Game_4. Here the subgroup decision assumption $\mathsf{DS1}$ is used to choose C_0 from the group $\mathsf{G}_{p_1 p_2}$ uniformly at random. The rest of the ciphertext and secret keys are generated similar to Game_3. Therefore, $|\mathsf{Pr}[E_4] - \mathsf{Pr}[E_3]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DS1}}(\lambda)$. We provide a proof sketch here. Given the problem instance $\mathsf{DS1}$, $\mathcal{B}$ chooses $\alpha, \tilde{\beta} \hookleftarrow \mathbb{Z}_N$. This allows $\mathcal{B}$ to compute all of mpk similar to Game_3. As it holds both α and $\tilde{\beta}$, $\mathcal{B}$ can answer any key extraction query. In the challenge phase it uses the target T of $\mathsf{DS1}$ problem instance to simulate C_0.

If T was from G_{p_1}, the C_0 is normal whereas if T was from $\mathsf{G}_{p_1 p_2}$, the 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. Now we change the secret keys SK_{Ω_i} for all $i \in [q]$ gradually to make them semi-functional. To do that, we also change the U_i in mpk for all $i \in [m]$ gradually by introducing G_{p_2} component. Precisely, we change the public parameter U_i from

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

and the secret key SK_{Ω_i} for each Ω_i is changed from

$$u^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot X_3 \quad \text{to} \quad u^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_2^{\sum_{j \in [m+q+1]} \frac{r_j \cdot \tilde{\beta} \cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} \cdot X'_3$$

for $r_1, \ldots, r_{m+q+1}, \alpha_1, \ldots, \alpha_{m+q+1} \hookleftarrow \mathbb{Z}_N$. This is done via intermediate games namely $\mathsf{Game}_{5,k,0}$ and $\mathsf{Game}_{5,k,1}$ for $k \in [m+q+1]$. We denote Game_4 by $\mathsf{Game}_{5,0,1}$ and Game_5 by $\mathsf{Game}_{5,m+q+2,0}$.

– In $\mathsf{Game}_{5,k,0}(k \in [0, m+q+1])$, the public parameter U_i for $i \in [m]$ is changed as follows. We also change U_0 similarly.

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

The secret key SK_{Ω_i} for $i \in [q]$ on the other hand is changed as follows.

$$\begin{aligned} &u^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_2^{\sum_{j \in [k-1]} \frac{r_j \cdot \tilde{\beta} \cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} X'_3 \\ &\qquad \to u^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot \boxed{g_2^{\frac{r \cdot \tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}}} \cdot g_2^{\sum_{j \in [k-1]} \frac{r_j \cdot \tilde{\beta} \cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} X'_3. \end{aligned} \tag{2}$$

– In $\mathsf{Game}_{5,k,1}(k \in [0, m+q+1])$, the parameters $\{U_i\}_{i \in [0,m]}$ and secret key $\{\mathsf{SK}_{\Omega_i}\}_{i \in [q]}$ distributions are respectively given by,

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

$$\mathsf{SK}_{\Omega_i} = u^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_2^{\sum_{j \in [k]} \frac{r_j \cdot \tilde{\beta} \cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} X'_3.$$

Notice that, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_{5,k,0}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_{5,k,1}}(\lambda)$ as α mod p_2 is uniformly random to the view of the adversary $\mathcal{A}$ since public parameters $g_1^{\beta}, (G_i)_{i \in [m]}$ contains information related to α mod p_1 only; and by CRT, they do not leak any information regarding α mod p_2. The semi-functional components of C_1 and κ_0 is completely determined by semi-functional component of C_0.

Since C_0 is chosen uniformly at random from $\mathsf{G}_{p_1 p_2}$, it is completely independent of $\alpha \bmod p_2$. Therefore, the changes between $\mathsf{Game}_{5,k,0}$ and $\mathsf{Game}_{5,k,1}$ is invisible to any $\mathcal{A}$.

Now in Lemma 1, we prove that under $\mathsf{DS2}$ assumption, $\mathsf{Game}_{5,k-1,1}$ and $\mathsf{Game}_{5,k,0}$ are indistinguishable for any $k \in [m+q+1]$. As a result we have $|\mathsf{Pr}[E_5] - \mathsf{Pr}[E_4]| \leq (m+q+1) \cdot \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DS2}}(\lambda)$.

Lemma 1. *There exists PPT adversary* $\mathcal{B}$ *such that,* $|\mathsf{Pr}[E_{5,k-1,1}] - \mathsf{Pr}[E_{5,k,0}]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DS2}}(\lambda)$.

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 $\beta = \tilde{\beta} \cdot P_{\Theta^*}(\alpha) \bmod N$, $G_i = g_1^{\alpha^i}$. 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 $X'_3 \hookleftarrow \mathsf{G}_{p_3}$ and sets

$$\mathsf{SK}_{\Omega_i} = T^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_{23}^{\sum_{j \in [k-1]} \frac{\hat{r}_j \cdot \tilde{\beta} \cdot P_{\Theta^*}(\alpha_j)}{P_{\Omega_i}(\alpha_j)}} X'_3.$$

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 $\kappa_0 = \mathsf{H}(e(\mathsf{C}_0, U_0))$. 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 $T \in \mathsf{G}_{p_1 p_3}$, 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 $\mathsf{Game}_{5,k,0}$ as can be seen in Equation (2). □

Game_6. Here, we replace the G_{p_2} components of $(U_i)_{i \in [0,m]}$ and $(\mathsf{SK}_{\Omega_i})_{i \in [q]}$ with randomly chosen elements $z_0, z_1, \ldots, z_{m+q}$ respectively. Precisely, for all $i \in [0, m]$ and for all $j \in [q]$,

$$U_i = u^{\alpha^i} \cdot g_2^{z_i} \cdot \hat{R}_{3,i}, \mathsf{SK}_{\Omega_j} = u^{\frac{\tilde{\beta} \cdot P_{\Theta^*}(\alpha)}{P_{\Omega_i}(\alpha)}} \cdot g_3^{z_{m+j}} \hat{X}_{3,j}.$$

This change between Game_5 and Game_6 can be represented as a linear system $\mathbf{z} = \mathbf{A}\mathbf{r}$ in Equation (3). To argue that such a change will be invisible to the adversary, it is enough to argue that the matrix $\mathbf{A}$ is non-singular as this ensures $\mathbf{z}$ to be a random vector in the span of $\mathbf{A}$. However, this does not hold always as some x_i can repeat across multiple key-queries as each key-indices are set. This is where our restriction of *cover-free sets* is essential in this proof. Even after putting the restriction, the matrix $\mathbf{A}$ still is quite complicated in nature and direct computation of determinant is troublesome. We solve this problem by showing one can get $\mathbf{A}$ (or some similar matrix $\mathbf{A}'$) from another non-singular matrix via row operations in Lemma 2 and Lemma 3.

$$\begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_m \\ z_{m+1} \\ \vdots \\ z_{m+q} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{m+q+1} \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_{m+q+1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^m & \alpha_2^m & \cdots & \alpha_{m+q+1}^m \\ \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_1)}{P_{\Omega_1}(\alpha_1)} & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_2)}{P_{\Omega_1}(\alpha_2)} & \cdots & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_{m+q+1})}{P_{\Omega_1}(\alpha_{m+q+1})} \\ \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_1)}{P_{\Omega_2}(\alpha_1)} & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_2)}{P_{\Omega_2}(\alpha_2)} & \cdots & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_{m+q+1})}{P_{\Omega_2}(\alpha_{m+q+1})} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_1)}{P_{\Omega_q}(\alpha_1)} & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_2)}{P_{\Omega_q}(\alpha_2)} & \cdots & \frac{\tilde{\beta}\cdot P_{\Theta^*}(\alpha_{m+q+1})}{P_{\Omega_q}(\alpha_{m+q+1})} \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{m+q+1} \end{pmatrix}. \tag{3}$$

Lemma 2. *The matrix* $\mathbf{A}$ *in Equation* (3) *is non-singular.*

Proof. From Equation (3), we denote $\mathbf{A} = \begin{pmatrix} \mathbf{B} \\ \mathbf{P} \end{pmatrix}$ where $\mathbf{B} \in \mathbb{Z}_{p_2}^{(m+1)\times(m+q+1)}$ is the first $(m+1)$ rows of $\mathbf{A}$ and $\mathbf{P} \in \mathbb{Z}_{p_2}^{q\times(m+q+1)}$ is last q rows of $\mathbf{A}$. Each entry of $\mathbf{B}$ and $\mathbf{P}$ are respectively evaluation of following polynomials with 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:

$$\begin{aligned} \mathbf{B}[i,\ell] &= z^i && \text{for } i \in [0,m], \\ \mathbf{P}[i,\ell] &= \frac{\tilde{\beta}\cdot P_{\Theta^*}(z)}{P_{\Omega_i}(z)} && \text{for } i \in [q]. \end{aligned} \tag{4}$$

We simplify the $\mathbf{P}[i,\ell]$ polynomial in Equation (4) next. Due to natural restriction, for all queries, $\Omega_i \not\subset \Theta^*$. Therefore, the polynomial $P_{\Omega_i}(z) \nmid P_{\Theta^*}(z)$ for all $i \in [q]$ where z is indeterminant. However, both the polynomials $P_{\Omega_i}(z)$ and $P_{\Theta^*}(z)$ are splitting polynomial. Precisely, $P_{\Theta^*}(z) = \prod_{j\in[\ell]} (z+y_j)$ and $P_{\Omega_i}(z) = \prod_{j\in[\hbar_i]} (z+x_j)$. Then, the rational function,

$$\mathfrak{R} = \frac{P_{\Theta^*}(z)}{P_{\Omega_i}(z)} = \frac{\prod_{y_j\in\Theta^*}(z+y_j)}{\prod_{x_j\in\Omega_i}(z+x_j)} = \frac{\prod_{y_j\in C_i'}(z+y_j)}{\prod_{x_j\in C_i}(z+x_j)} = \mathfrak{A}\cdot\mathfrak{B} \tag{5}$$

where $\mathfrak{A} = \prod_{y_j \in C'_i} (z + y_j) = P_{C'_i}(z)$ and $\mathfrak{B} = \frac{1}{\prod_{x_j \in C_i} (z+x_j)}$. Due to Equation (1), $\mathfrak{B} = \frac{1}{\prod_{x_j \in C_i} (z+x_j)} = \sum_{x_j \in C_i} \frac{1}{\prod_{\substack{x_k \in C_i \\ j \neq k}} (x_k - x_j)} \cdot \frac{1}{z+x_j}$. In other words, $\mathfrak{B} = \sum_{x_j \in C_i} \mathsf{R}_{j,i} \cdot \frac{1}{z+x_j}$ where $\mathsf{R}_{j,i}$ are non-zero scalar values that can be computed from the set C_i. The polynomial $\mathfrak{R}$ from Equation (5) therefore is

$$\mathfrak{R} = \sum_{x_j \in C_i} \mathsf{R}_{j,i} \cdot \frac{P_{C'_i}(z)}{z + x_j}. \tag{6}$$

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

$$\begin{aligned}
\mathbf{P}[i,\ell] &= \tilde{\beta} \cdot \sum_{x_j \in C_i} \mathsf{R}_{j,i} \cdot \frac{P_{C'_i}(z)}{z+x_j} && \text{(from Equation (4) and Equation (6))} \\
&= \tilde{\beta} \cdot \sum_{x_j \in C_i} \mathsf{R}_{j,i} \left(K_{C'_i, x_j}(z) + \frac{t_j}{z+x_j} \right) && (t_j \text{ is scalar}) \\
&= \sum_{x_j \in C_i} \left(\tilde{R}_{j,i} K_{C'_i, x_j}(z) + \frac{R'_{j,i}}{z+x_j} \right) \\
& && (\tilde{R}_{j,i} = \tilde{\beta} \cdot \mathsf{R}_{j,i},\ R'_{j,i} = \tilde{\beta} \cdot \mathsf{R}_{j,i} \cdot t_j \text{ scalar}) \\
&= \sum_{x_j \in C_i} \left(\sum_{k \in [0, \ell'_i]} \tilde{R}_{j,i} \cdot b_k^{j,i} z^k + \frac{R'_{j,i}}{z+x_j} \right) \\
& && (K_{C'_i, x_j}(z) = \sum_{k \in [0, \ell'_i]} b_k^{j,i} z^k \text{ polynomial expansion}) \\
&= \sum_{x_j \in C_i} \left(\sum_{k \in [0, \ell'_i]} \tilde{R}'_{j,i,k} z^k + \frac{R'_{j,i}}{z+x_j} \right) && (\tilde{R}'_{j,i,k} = \tilde{R}_{j,i} \cdot b_k^{j,i} \text{ scalar}). \\
&= \sum_{k \in [0, \ell'_i]} \hat{\tilde{R}}'_{j,i,k} z^k + \sum_{x_j \in C_i} \frac{R'_{j,i}}{z+x_j} && (\hat{\tilde{R}}'_{j,i,k} = \sum_{x_j \in C_i} \tilde{R}'_{j,i,k} \text{ scalar}).
\end{aligned}$$

Hence, we represent Equation (4) as, for any $\ell \in [m+q+1]$,

$$\begin{aligned}
\mathbf{B}[i,\ell] &= z^i && \text{for } i \in [0,m], \\
\mathbf{P}[i,\ell] &= \sum_{k \in [0, \ell'_i]} \hat{\tilde{R}}'_{j,i,k} z^k + \sum_{x_j \in C_i} \frac{R'_{j,i}}{z + x_j} && \text{for } i \in [q].
\end{aligned} \tag{7}$$

Notice that, in Equation (7), $\hat{\tilde{R}}'_{j,i,k} z^k$ in $\mathbf{P}[i,\ell]$ is in linear span of $\mathbf{B}[i,\ell]$ for $i \in [0,m]$. Therefore, elementary row operations removes such dependency to define a new matrix $\mathbf{A}' = \begin{pmatrix} \mathbf{B} \\ \mathbf{P}' \end{pmatrix}$ such that $|\mathsf{det}(\mathbf{A})| = |\mathsf{det}(\mathbf{A}')|$ where for any $\ell \in [m+q+1]$, z takes value from $\{\alpha_1, \ldots, \alpha_{m+q+1}\}$, each $[i,\ell]^{th}$ entry of $\mathbf{B}$ and $\mathbf{P}'$ are respectively

$$\begin{aligned}
\mathbf{B}[i,\ell] &= z^i && \text{for } i \in [0,m], \\
\mathbf{P}'[i,\ell] &= \sum_{x_j \in C_i} \frac{R'_{j,i}}{z + x_j} && \text{for } i \in [q].
\end{aligned} \tag{8}$$

Lemma 3. *The matrix* $\mathbf{A}' = \begin{pmatrix} \mathbf{B} \\ \mathbf{P}' \end{pmatrix}$*, where* $\mathbf{B}$ *and* $\mathbf{P}'$ *are as defined in Equation* (8)*, is non-singular.*

To prove that $\mathbf{A}'$ as defined above is non-singular, we start from the claim that the matrix $\mathbf{D}$ (in Equation (9)) is non-singular if all $x_j \neq x_l$ for $l, j \in [Q]$ for $l \neq j$ and all $\gamma_i \neq \gamma_k$ for $i, k \in [m + Q + 1]$ for $i \neq k$. This result was proved in [16, Lemma 3] where Q was number of key-queries (i.e. distinct x_j). We here set Q to be cardinality of $\left(\bigcup_{i \in [q]} \Omega_i \right)$ i.e. total number of distinct x_j that is queried as a part of some key query Ω_i.

Lemma 4. $\mathsf{det}(\mathbf{D}) = \delta \cdot \frac{\prod_{1 \leq l < j \leq Q}(x_l - x_j) \prod_{1 \leq i < k \leq m+Q+1}(\gamma_i - \gamma_k)}{\prod_{k=1}^{m+Q+1} \prod_{l=1}^{Q}(\gamma_k + x_l)}$ *where* δ *is some non-zero scalar in* $\mathbb{Z}_{p_2}$ *where* $\mathbf{D}$ *is given in Equation* (9)*.*

$$\mathbf{D} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{m+Q+1} \\ \gamma_1^2 & \gamma_2^2 & \cdots & \gamma_{m+Q+1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1^m & \gamma_2^m & \cdots & \gamma_{m+Q+1}^m \\ \frac{1}{\gamma_1 + x_1} & \frac{1}{\gamma_2 + x_1} & \cdots & \frac{1}{\gamma_{m+Q+1} + x_1} \\ \frac{1}{\gamma_1 + x_2} & \frac{1}{\gamma_2 + x_2} & \cdots & \frac{1}{\gamma_{m+Q+1} + x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\gamma_1 + x_Q} & \frac{1}{\gamma_2 + x_Q} & \cdots & \frac{1}{\gamma_{m+Q+1} + x_Q} \end{pmatrix} \tag{9}$$

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 (8). One can perform such transformation if $Q \geq q$ which is the case due to the restriction we imposed on relation between query sets. Precisely, the *cover-free* property of the query sets ensures that $Q \geq q$ as informally speaking each Ω_i should contain some new x_j. In other words, we perform elementary row operations on $\mathbf{D}$ in Equation (9) to get to 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 4, $\mathbf{D}'$ 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 (8)). Next 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 $\mathbf{A}'$. Therefore, the matrix $\mathbf{A}'$ is non-singular. This ensures that the matrix $\mathbf{A}$ in Equation (3) is also non-singular.

Due to the fact that each Ω_i will have some new x_j (in the $\mathsf{s^*IND\text{-}CPA}$ model $(\Omega_i)_{i\in[q]}$ are *cover-free sets*) and Lemma 2, it is evident that $\mathsf{det}(\mathbf{A}) \neq 0$ as long as $\alpha_l \neq \alpha_k \bmod p_2$ for $l,k \in [m+q+1]$ and $l \neq k$. Therefore, $|\mathsf{Pr}[E_6] - \mathsf{Pr}[E_5]| \leq (m+q)(m+q+1)/p_2$. □

Game_7. Here 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 $u \cdot g_2^{z_0} \cdot \mathsf{R}_3$. As we saw in the last game, z_0 is an 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 $z_0 \bmod p_2$. Due to left-over hash lemma, $\kappa_0 = \mathsf{H}(e(\mathsf{C}_0, U_0))$ is at most $2^{-\lambda}$ distant from 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 $|\mathsf{Pr}[E_7] - \mathsf{Pr}[E_6]| \leq 1/p_2 + 2^{-\lambda}$. κ_0 now is a random choice and it hides $\mathfrak{b}$ completely i.e. $\mathsf{Pr}[E_7] = 1/2$.

This completes the proof of Theorem 1. □

Remark 2. Here, we mention that, the value is Q is in general lower-bounded by $\log q$ and upper-bounded by mq. By putting our restriction, we push the lower-bound up to $q+t$ (where $t = \min_i |\Omega_i|$) and ensure each key-index Ω_i to have a distinct x_j that was not present in rest of the key-indices.

Proof of Lemma 4. Gong et al. [16, Lemma 3] proved this statement. For the sake of completeness, we reproduce it here. The matrix in Equation (9) is of order $(m+Q+1) \times (m+Q+1)$.
Each of the monomials of $\mathcal{P} = \mathsf{det}(\mathbf{D}) \cdot \prod_{k=1}^{m+Q+1} \prod_{l=1}^{Q} (\gamma_k + x_l)$ 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 is 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 is same.

Therefore, the polynomial $\mathcal{P}$ must be a multiple of $\mathcal{T} = \prod_{1\leq l<j\leq Q}(x_l - x_j) \cdot \prod_{1\leq i<k\leq m+Q+1}(\gamma_i - \gamma_k)$ 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 SXDH assumption.

5.1 Construction

SPE_2 is defined as 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 α_1, α_2, c, d, $(u_j, v_j)_{j\in[0,m]} \hookleftarrow \mathbb{Z}_p$ and $b \hookleftarrow \mathbb{Z}_p^\times$. For $j \in \{0, \ldots, m\}$, define $g_1^{w_j} = g_1^{u_j+bv_j}$ 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 $\mathsf{msk} = (g_2, g_2^c, \alpha_1, \alpha_2, d, (u_j, v_j)_{j\in[0,m]})$ and the public parameter is defined as
$$\mathsf{mpk} = \left(g_1, g_1^b, \left(g_1^{w_j}\right)_{j\in[0,m]}, g_1^w, g_{\mathrm{T}}^\alpha\right).$$
- $\mathsf{KeyGen}(\mathsf{msk}, \Omega)$: Given a set Ω, such that $|\Omega| = \Bbbk \leq m$ 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
$$\mathsf{K}_1 = g_2^r, \mathsf{K}_2 = g_2^{cr}, \mathsf{K}_3 = g_2^{\alpha_1 + r\sum\limits_{x\in\Omega}(u_0+u_1x+u_2x^2+\ldots+u_mx^m)},$$
$$\mathsf{K}_4 = g_2^{dr}, \mathsf{K}_5 = g_2^{\alpha_2 + r\sum\limits_{x\in\Omega}(v_0+v_1x+v_2x^2+\ldots+v_mx^m)}.$$
- $\mathsf{Encrypt}(\mathsf{mpk}, \Theta)$: Given a set Θ, such that $|\Theta| = \ell \leq m$. 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
$$\kappa = e(g_1, g_2)^{\alpha s}, \mathsf{C}_0 = g_1^s, \mathsf{C}_1 = g_1^{bs}, \mathsf{C}_{2,i} = g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_my_i^m+wt_i)}.$$
- $\mathsf{Decrypt}((\mathsf{SK}_\Omega, \Omega), (\mathsf{CT}_\Theta, \Theta))$: Computes $\kappa = B/A$ 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 $\Omega \subseteq \Theta$,
$$\begin{aligned}
B &= e\left(\mathsf{C}_0, \mathsf{K}_3 \prod_{y_i\in\Omega} \mathsf{K}_2^{t_i}\right) e\left(\mathsf{C}_1, \mathsf{K}_5 \prod_{y_i\in\Omega} \mathsf{K}_4^{t_i}\right),\\
&= e\left(\mathsf{C}_0, g_2^{\alpha_1 + r\sum\limits_{y_i\in\Omega}(u_0+u_1y_i+u_2y_i^2+\ldots+u_my_i^m)} \cdot \prod_{y_i\in\Omega} g_2^{rct_i}\right)\\
&\qquad \cdot e\left(\mathsf{C}_1, g_2^{\alpha_2 + r\sum\limits_{y_i\in\Omega}(v_0+v_1y_i+v_2y_i^2+\ldots+v_my_i^m)} \cdot \prod_{y_i\in\Omega} g_2^{rdt_i}\right)\\
&= e\left(\mathsf{C}_0, g_2^{\alpha_1 + r\sum\limits_{y_i\in\Omega}(u_0+u_1y_i+u_2y_i^2+\ldots+u_my_i^m)} \cdot \prod_{y_i\in\Omega} g_2^{rct_i}\right)
\end{aligned}$$

$$\cdot\, e\left(\mathsf{C}_0, g_2^{b\alpha_2 + rb\sum_{y_i\in\Omega}(v_0+v_1y_i+v_2y_i^2+\ldots+v_my_i^m)} \cdot \prod_{y_i\in\Omega} g_2^{rbdt_i}\right)$$

$$= e\left(\mathsf{C}_0, g_2^{(\alpha_1+b\alpha_2)+r\sum_{y_i\in\Omega}((u_0+bv_0)+(u_1+bv_1)y_i+\ldots+(u_m+bv_m)y_i^m)} \cdot \prod_{y_i\in\Omega} g_2^{r(c+bd)t_i}\right)$$

$$= e\left(\mathsf{C}_0, g_2^{\alpha+r\sum_{y_i\in\Omega}(w_0+w_1y_i+w_2y_i^2+\ldots+w_my_i^m)} \cdot \prod_{y_i\in\Omega} g_2^{rwt_i}\right)$$

$$= e\left(g_1^s, g_2^{\alpha+r\sum_{y_i\in\Omega}(w_0+w_1y_i+w_2y_i^2+\ldots+w_my_i^m+wt_i)}\right)$$

$$A = e\left(\prod_{y_i\in\Omega}\mathsf{C}_{2,i}, \mathsf{K}_1\right)$$

$$= e\left(g_1^{s\sum_{y_i\in\Omega}(w_0+w_1y_i+w_2y_i^2+\ldots+w_my_i^m+wt_i)}, g_2^r\right)$$

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

Remark 3. We observe that our SPE_2 construction has a *pair encoding* [3] embedded. One can utilize the generic technique of Chen et al. [11] to get corresponding predicate encryption. The public parameter and ciphertext size, however, will be significantly larger than that of SPE_2. Precisely, both the public parameter and ciphertext contain additional m G_1 elements. Although the secret key requires one less G_2 element, the decryption is costlier as it takes one extra pairing evaluation. In addition, one can apply such pair encoding on framework by Chen and Gong [12] to generalize our SPE_2 construction further in terms of security.

5.2 Security

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

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

Proof Sketch. We propose a hybrid argument based proof that uses dual system proof technique [24] at its core. This hybrid argument follows the proof strategy of [22]. 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 uniform 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 random choice from $\mathcal{K}$. Note that, we denote Game_1 by $\mathsf{Game}_{2,0}$.

We first recall the crucial tactics [22] used to prove their IBBE adaptive CPA-secure as we already have mentioned that our large-universe SPE_2 construction uses IBBE [22] as a starting point. The crux of the proof of IBBE in [22] is a linear map that reflects the relation between tags $(t_1, \ldots, t_\ell)$ which encoded $(y_1, \ldots, y_\ell)$ respectively and semi-functional component (π) in the secret key SK_x that encoded queried key-index x. This scenario occurs when a normal secret key is translated into corresponding semi-functional form. At this point, [22] showed that such linear map is non-singular following Attrapadung and Libert [5]. Such a property of the linear map effectively ensures that semi-functional component of the key has enough entropy to hide the encapsulation key κ.

However, following their proof technique verbatim does not work out as the semi-functional component π no longer encodes only one identity rather it has to encode multiple identities belonging to the queried set Ω. Let us consider a case where, $x \in (\Theta^* \cap \Omega)$, i.e. $\exists j \in [\ell], x = y_j$. In other words $\exists j \in [\ell]$ such that tag t_j encodes $y_j (= x)$ where $x \in \Omega$. As the semi-functional component π, that encodes queried set Ω, will also contain some information about x (i.e. y_j), it is not clear if $(t_1, \ldots, t_\ell)$ and π are still independent.

The novelty in our proof technique is that we proceed in a different manner where we argue independence of $(t_1, \ldots, t_\ell)$ and π^* as well as the independence of $\hat{\pi}$ and π^* where π^* encodes $x^* \in \Omega \setminus \Theta^*$ and $\hat{\pi}$ encodes all $x \in \Omega \setminus \{x^*\}$. Notice that such a x^* will always exist as $\Omega \not\subset \Theta^*$. This therefore ensures that the linear map reflecting the relation between $(t_1, \ldots, t_\ell)$ and π to be non-singular.

Now, 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

  $\mathsf{K}_1 = \mathsf{K}'_1 = g_2^r, \mathsf{K}_2 = \mathsf{K}'_2 \cdot g_2^{\hat{r}} = g_2^{cr+\hat{r}},$
  $\mathsf{K}_3 = \mathsf{K}'_3 \cdot g_2^{\hat{r}\pi} = g_2^{\alpha_1 + r \sum_{x \in \Omega} (u_0 + u_1 x + u_2 x^2 + \ldots + u_m x^m) + \hat{r}\pi},$
  $\mathsf{K}_4 = \mathsf{K}'_4 \cdot g_2^{-\hat{r}b^{-1}} = g_2^{dr - \hat{r}b^{-1}},$

$$\mathsf{K}_5 = \mathsf{K}'_5 \cdot g_2^{-\hat{r}\pi b^{-1}} = g_2^{\alpha_2 + r \sum_{x\in\Omega}(v_0+v_1x+v_2x^2+\ldots+v_mx^m) - \hat{r}\pi b^{-1}}.$$

– $\mathsf{SFEncrypt}(\mathsf{mpk}, \mathsf{msk}, \Theta)$: 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:

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

□

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

Lemma 5. (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* $\left|\mathsf{Adv}^0_{\mathcal{A}}(\lambda) - \mathsf{Adv}^1_{\mathcal{A}}(\lambda)\right| \leq \mathsf{Adv}^{\mathsf{DDH}_{\mathsf{G}_1}}_{\mathcal{B}}(\lambda)$.

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 random from $\mathbb{Z}_p^\times$.

Setup. $\mathcal{B}$ chooses $\alpha_1, \alpha_2, (u_i, v_i)_{i\in[0,m]}, c, d \hookleftarrow \mathbb{Z}_p$. As both α_1 and α_2 are available to $\mathcal{B}$, it can generate $g_{\mathrm{T}}^{\alpha} = e(g_1^{\alpha_1} \cdot (g_1^b)^{\alpha_2}, g_2)$. 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 the msk, it can answer with normal secret keys on any query of Ω.

Challenge. Given the challenge 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.

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

$$\mathsf{C}_{2,i} = \mathsf{C}_0^{u_0+u_1y_i+u_2y_i^2+\ldots+u_my_i^m+ct_i} \cdot \mathsf{C}_1^{v_0+v_1y_i+v_2y_i^2+\ldots+v_my_i^m+dt_i}$$

where $i \in [\ell]$. $\mathcal{B}$ then chooses $\kappa_1 \hookleftarrow \mathcal{K}$ and returns $(\kappa_{\mathfrak{b}}, \mathsf{CT}_{\Theta^*})$ as the challenge ciphertext 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, $\left|\mathsf{Adv}^0_{\mathcal{A}}(\lambda) - \mathsf{Adv}^1_{\mathcal{A}}(\lambda)\right| \leq \epsilon_{\mathsf{DDH}_{\mathsf{G}_1}}$. □

Lemma 6. ($\mathsf{Game}_{2,k-1}$ *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* $|\mathsf{Adv}_{\mathcal{A}}^{2,k-1}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{2,k}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH}_{\mathsf{G}_2}}(\lambda)$.

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 $b \hookleftarrow \mathbb{Z}_p^\times, \alpha, \alpha_1, w, (p_i, q_i, w_i)_{i\in[0,m]} \hookleftarrow \mathbb{Z}_p$. 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 $(d, (u_i, v_i)_{i\in[0,m]})$ 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 $(d, (u_i, v_i)_{i\in[0,m]})$ 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,
  $$\mathsf{K}_1 = g_2^{r_j}, \mathsf{K}_2 = (g_2^c)^{r_j},$$
  $$\mathsf{K}_3 = g_2^{\alpha_1} \cdot \mathsf{K}_1^{\sum\limits_{x\in\Omega_j}(p_0+p_1x+p_2x^2+\ldots+p_mx^m)} \cdot \mathsf{K}_2^{\sum\limits_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_mx^m)},$$
  $$= g_2^{\alpha_1 + r_j \sum\limits_{x\in\Omega_j}(u_0+u_1x+u_2x^2+\ldots+u_mx^m)},$$
  $$\mathsf{K}_4 = \mathsf{K}_1^{b^{-1}w} \cdot \mathsf{K}_2^{-b^{-1}} = g_2^{dr_j},$$
  $$\mathsf{K}_5 = g_2^{b^{-1}\alpha} \cdot \mathsf{K}_1^{b^{-1}\sum\limits_{x\in\Omega_j}(w_0+w_1x+w_2x^2+\ldots+w_mx^m)} \cdot \mathsf{K}_3^{-b^{-1}}$$
  $$= g_2^{b^{-1}\alpha + r_jb^{-1}\sum\limits_{x\in\Omega_j}(w_0+w_1x+w_2x^2+\ldots+w_mx^m)} \cdot g_2^{-b^{-1}(\alpha_1 + r_j\sum\limits_{x\in\Omega_j}(u_0+u_1x+u_2x^2+\ldots+u_mx^m))},$$
  $$= g_2^{\alpha_2 + r_j\sum\limits_{x\in\Omega_j}(v_0+v_1x+v_2x^2+\ldots+v_mx^m)}.$$
  Notice that SK_{Ω_j} is identically distributed to 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,
  $$\mathsf{K}_1 = g_2^r,\ \mathsf{K}_2 = T = g_2^{cr+\hat{r}} = \mathsf{K}_2' \cdot g_2^{\hat{r}},$$
  $$\mathsf{K}_3 = g_2^{\alpha_1} \cdot \mathsf{K}_1^{\sum\limits_{x\in\Omega_j}(p_0+p_1x+p_2x^2+\ldots+p_mx^m)} \cdot \mathsf{K}_2^{\sum\limits_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_mx^m)},$$

$$= g_2^{\alpha_1 + r\sum_{x\in\Omega_j}(u_0+u_1x+u_2x^2+\ldots+u_mx^m) + \hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_mx^m)},$$
$$= \mathsf{K}_3' \cdot g_2^{\hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_mx^m)}.$$
$$\mathsf{K}_4 = \mathsf{K}_1^{b^{-1}w} \cdot \mathsf{K}_2^{-b^{-1}} = g_2^{dr} \cdot g_2^{-b^{-1}\hat{r}} = \mathsf{K}_4' \cdot g_2^{-b^{-1}\hat{r}}.$$
$$\mathsf{K}_5 = g_2^{b^{-1}\alpha} \cdot \mathsf{K}_1^{b^{-1}\sum_{x\in\Omega_j}(w_0+w_1x+w_2x^2+\ldots+w_mx^m)} \cdot \mathsf{K}_3^{-b^{-1}},$$
$$= g_2^{\alpha_2 + r\sum_{x\in\Omega_j}(v_0+v_1x+v_2x^2+\ldots+v_mx^m)} \cdot g_2^{-b^{-1}\hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_mx^m)}$$
$$= \mathsf{K}_5' \cdot g_2^{-b^{-1}\hat{r}\sum_{x\in\Omega_j}(q_0+q_1x+q_2x^2+\ldots+q_mx^m)}.$$

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

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

$$\mathsf{C}_{2,i} = g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_my_i^m+wt_i)+\hat{s}(u_0+u_1y_i+u_2y_i^2+\ldots+u_my_i^m+ct_i)},$$
$$= g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_my_i^m+wt_i)+\hat{s}(p_0+p_1y_i+p_2y_i^2+\ldots+p_my_i^m)} \cdot g_1^{c\hat{s}(q_0+q_1y_i+q_2y_i^2+\ldots+q_my_i^m+t_i)}.$$

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

Then, $\mathsf{C}_{2,i} = g_1^{s(w_0+w_1y_i+w_2y_i^2+\ldots+w_my_i^m+wt_i)+\hat{s}(p_0+p_1y_i+p_2y_i^2+\ldots+p_my_i^m)}$ 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 ciphertext. Notice that, the challenge ciphertext $(\kappa_0, \mathsf{CT}_{\Theta^*})$ is identically distributed to the output of $\mathsf{SFEncrypt}(\mathsf{mpk}, \mathsf{msk}, \Theta^*)$. Hence, the ciphertext is semi-functional.

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.

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. The challenge ciphertext is also constructed semi-functional.

However, we need to argue that the tags $(t_i)_{i\in[\ell]}$ output as the challenge ciphertext component are uniformly random to the view of adversary $\mathcal{A}$ who has got hold of the semi-functional k^{th} secret key containing π. This is because, according to 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.

Recall that, $\pi = \sum_{x\in\Omega_k}(q_0+q_1x+q_2x^2+\ldots+q_mx^m)$ and $t_i = -(q_0+q_1y_i+q_2y_i^2+\ldots+q_my_i^m)$ for all $y_i \in \Theta^*$. As $\Omega_k \not\subset \Theta^*$, due to natural restriction of the security game, there exists an $x^* \in \Omega_k$ but $x^* \notin \Theta^*$. Then, $\pi = \sum_{x\in\Omega_k}(q_0+q_1x+$

$q_2x^2+\ldots+q_mx^m) = \sum_{\substack{x\in\Omega_k\\ x\neq x^*}} (q_0+q_1x+q_2x^2+\ldots+q_mx^m)+(q_0+q_1(x^*)+q_2(x^*)^2+\ldots+q_m(x^*)^m)$. Let us denote $\pi^* = (q_0+q_1(x^*)+q_2(x^*)^2+\ldots+q_m(x^*)^m)$ and $\hat{\pi} = \sum_{x_i\in\Omega_k\setminus\{x^*\}} \pi_i$ where $\pi_i = (q_0+q_1x_i+q_2x_i^2+\ldots+q_mx_i^m)$.

Next we argue that π^* is independent of all the tags $(t_i)_{i\in[\ell]}$. The relation between π^* and $(t_1,t_2,\ldots,t_\ell)$ can be expressed as the following linear system of equations $\mathbf{t} = \mathbf{V}\mathbf{q}$.

$$\begin{pmatrix}\pi^*\\ t_1\\ t_2\\ \vdots\\ t_\ell\end{pmatrix} = \begin{pmatrix}1 & x^* & (x^*)^2 & \cdots & (x^*)^m\\ 1 & y_1 & (y_1)^2 & \cdots & (y_1)^m\\ 1 & y_2 & (y_2)^2 & \cdots & (y_2)^m\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & y_\ell & (y_\ell)^2 & \cdots & (y_\ell)^m\end{pmatrix} \cdot \begin{pmatrix}q_0\\ q_1\\ q_2\\ \vdots\\ q_m\end{pmatrix} \tag{10}$$

Notice that $\mathbf{V}$ is Vandermonde matrix of rank $(\ell+1)$ as $x^*\notin\Theta^*=\{y_1,y_2,\ldots,y_\ell\}$. The vector $\mathbf{q}$ is completely hidden from adversary $\mathcal{A}$ and was chosen uniformly at random. Therefore, π^* is independent of $(t_1,t_2,\ldots,t_\ell)$ and uniformly random in the view of $\mathcal{A}$.

Recall that, $\pi = \hat{\pi}+\pi^*$ where $\hat{\pi}$ is linear combination of $(\Bbbk-1)$ many m-degree polynomials as $|\Omega_k| = \Bbbk$. The collection $\pi_1,\ldots,\pi_{\Bbbk-1}$ and π^* also result in a full rank matrix as each encodes m-degree polynomial evaluated on distinct $\Bbbk$ points. This effectively ensures that π^* is independent of $\hat{\pi}$ as well. Thus, $\pi = \hat{\pi}+\pi^*$ is now a one-time-pad evaluation in the view of $\mathcal{A}$. Hence, π is uniformly random and independent choice from $\mathbb{Z}_p$. This completes the proof as $(\pi,(t_i)_{i\in[\ell]})$ are uniformly random quantities. Thereby, the ciphertext and k^{th} secret key is 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, $\left|\mathsf{Adv}_{\mathcal{A}}^{2,k-1}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{2,k}(\lambda)\right| \leq \epsilon_{\mathsf{DDH}_{\mathsf{G}_2}}$. $\square$

Lemma 7. ($\mathsf{Game}_{2,q}$ *to* Game_3) *For any efficient adversary $\mathcal{A}$ that makes at most q key queries,* $\left|\mathsf{Adv}_{\mathcal{A}}^{2,q}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{3}(\lambda)\right| \leq 2/p$.

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 $b \hookleftarrow \mathbb{Z}_p^\times, \alpha_1, \alpha, c, w, (u_i,w_i)_{i\in[0,m]} \hookleftarrow \mathbb{Z}_p$. It sets $\alpha_2 = b^{-1}(\alpha-\alpha_1)$, $d = b^{-1}(w-c)$, $v_i = b^{-1}(w_i-u_i)$. 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.

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

$\mathsf{K}_4 = g_2^{dr - \hat{r}b^{-1}}$, $\mathsf{K}_5 = g_2^{b^{-1}(\alpha - \pi')} \cdot g_2^{r \sum_{x \in \Omega} (v_0 + v_1 x + v_2 x^2 + \ldots + v_m x^m)}$.
The reduction sets $\pi' = \alpha_1 + \hat{r}\pi$. 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.
Here the point of focus is 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,

$\kappa_0 = e(g_1, g_2)^{\alpha s + \alpha_1 \hat{s}} = g_{\mathrm{T}}^{\alpha s} \cdot g_{\mathrm{T}}^{\alpha_1 \hat{s}}, \mathsf{C}_0 = g_1^{s + \hat{s}}, \mathsf{C}_1 = g_1^{bs},$
$\mathsf{C}_{2,i} = g_1^{s(w_0 + w_1 y_i + w_2 y_i^2 + \ldots + w_m y_i^m + w t_i) + \hat{s}(u_0 + u_1 y_i + u_2 y_i^2 + \ldots + u_m y_i^m + c t_i)}.$

Phase-II Queries. Same as Phase-I queries.
Guess. $\mathcal{A}$ output $\mathfrak{b}' \in \{0,1\}$. 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, $\left|\mathsf{Adv}_{\mathcal{A}}^{2,q}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{3}(\lambda)\right| \leq \mathsf{Pr}[\hat{r} = 0] + \mathsf{Pr}[\hat{s} = 0] \leq 2/p$.
Notice that, $\kappa_{\mathfrak{b}}$ output in Game_3 completely hides $\mathfrak{b}$. Thus, for any adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\mathcal{A}}^{3}(\lambda) = 0$. □

5.3 Applications

Katz et al. [20] described a few black-box transformations from SPE to well known cryptographic protocols. We can perform those transformations on our adaptive-secure SPE_2 construction. Note that, all these transformations were designed for small-universe SPE. We therefore restrict our large-universe SPE_2 construction to small universe. This is done by considering the universes $\mathcal{U} = \{1, \cdots, n\}$ and $\mathcal{U}' = \{1, \cdots, \mathfrak{n}\}$ where $\mathcal{U}$ is universe for protocol to be designed and $\mathcal{U}'$ is the universe for underlying SPE_2 for some $\mathfrak{n} \in \mathbb{N}$. Note that, we formalize the black-box transformation [20] as a function called Encode.

WIBE. The generic transformation of [20] allows construction of WIBE [1] which supports presence of wildcard in the data-index. Here, any index (key-index, data-index alike) will be first processed bit-wise into a ordered set of double size (i.e. $\mathfrak{n} = 2n$). Informally, Encode expands $z \in \{0,1,*\}^n$ to $T \in \{0,1\}^{\mathfrak{n}}$ where $T[2i-1]$ stores z_i and $T[2i]$ stores $\bar{z}_i$ if $z_i \in \{0,1\}$. In case of $z_i = *$, both $T[2i-1]$ and $T[2i]$ stores 1. Then $S^{(z)}$ is defined as the set that stores all indexes that are set in T. The WIBE KeyGen and $\mathsf{Encrypt}$ is defined as $\mathsf{SPE}_2.\mathsf{KeyGen}$ and

$\mathsf{SPE}_2.\mathsf{Encrypt}$ running on such set S respectively. We can achieve a WKD-IBE [2] in a similar way with the exception that now, the wildcard is present in the key-index.

CP-ABE. As [20] mentions, the most interesting black-box transformation of SPE is that it can achieve a secure CP-ABE (though restricted to DNF formula only) with constant-size key. Intuitively, an attribute set A can satisfy a DNF formula $C_1 \vee C_2 \vee \cdots C_t$ where each C_j represents a conjunction over some subset of the attributes if $\exists j \in [t]$ such that $C_j \subseteq \mathsf{A}$. This is done by associating the clauses C_j as well as A to corresponding *revocation list* i.e. $\mathcal{U} \setminus C_j$ and $\mathcal{U} \setminus \mathsf{A}$ and perform the subset predicate evaluation: $\mathcal{U} \setminus \mathsf{A} \subseteq \mathcal{U} \setminus C_j$ 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 $S^{(Z)} = \left\{i \in \mathcal{U}' : T^{(Z)}[i] = 1\right\}$ where for all $i \in \{1, 2, \cdots, \mathfrak{n}\}$ (here $\mathfrak{n} = n$).

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

We now compare the black-box transformation [20] applied on SPE_2 in terms of performance to previous WIBE and DNF schemes (both dedicated and due to black-box transformation [20]). From Table 1, we see that both adaptive secure BBG-WIBE and Wa-WIBE attain much bigger secret key size. Although, other parameter sizes are quite competitive to ours, Wa-WIBE is proved secure under parameterized assumption. In case of the second one however, the all the parameters blow up. Our construction not only attains similar parameter size as the selective secure constructions due to black-box transformation [20], is also proved adaptive secure under standard assumption. In case of DNF in Table 2, ours is the only scheme that achieve adaptive security and still enjoy constant-size key and constant number of pairing evaluations during decryption. Again, as compared to black-box transformation [20], our parameter sizes are quite competitive. We denote size of public key by $|\mathsf{mpk}|$, size of secret key by $|\mathsf{SK}|$, size of ciphertext by $|\mathsf{CT}|$, number of primitive operations required in $\mathsf{Decrypt}$. Here n denotes depth of hierarchy, ℓ is bit-length of identity in Wa-IBE [23], γ is number of disjunctive clauses in a DNF formula and $[\mathsf{P}]$ denotes number of pairing operations.

WIBE Schemes	$\lvert\mathsf{mpk}\rvert$	$\lvert\mathsf{SK}\rvert$	$\lvert\mathsf{CT}\rvert$	Decrypt	Security	Assumption
BBG-WIBE [1]	$(n+4)\mathsf{G}$	$(n+2)\mathsf{G}$	$(n+2)\mathsf{G} + \mathsf{G}_\mathrm{T}$	$2[\mathsf{P}]$	adaptive	n-BDHI
Wa-WIBE [1]	$((\ell+1)n+3)\mathsf{G}$	$(n+1)\mathsf{G}$	$((\ell+1)n+2)\mathsf{G} + \mathsf{G}_\mathrm{T}$	$(n+1)[\mathsf{P}]$	adaptive	DBDH
SPE-1 [20]	$(2n+2)\mathsf{G}_1 + \mathsf{G}_\mathrm{T}$	$\mathsf{G}_2 + \mathbb{Z}_p$	$(2n+1)\mathsf{G}_1 + \mathsf{G}_\mathrm{T}$	$1[\mathsf{P}]$	selective	q-BDHI
SPE-2 [20]	$(2n+1)\mathsf{G}_1 + 2\mathsf{G}_2$	$\mathsf{G}_1 + \mathsf{G}_2$	$2n\mathsf{G}_1 + \mathsf{G}_2 + \mathsf{G}_\mathrm{T}$	$2[\mathsf{P}]$	selective	DBDH
SPE_2	$(2n+6)\mathsf{G}_1 + \mathsf{G}_\mathrm{T}$	$5\mathsf{G}_2$	$(n+2)\mathsf{G}_1 + \mathsf{G}_\mathrm{T} + n\mathbb{Z}_p$	$3[\mathsf{P}]$	adaptive	SXDH

Table 1 Comparison of efficient standard model WIBE schemes.

DNF Schemes	$\lvert\mathsf{mpk}\rvert$	$\lvert\mathsf{SK}\rvert$	$\lvert\mathsf{CT}\rvert$	Decrypt	Security	Assumption
SPE-1 [20]	$(n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$\mathsf{G}_2+\mathbb{Z}_p$	$\gamma((n+1)\mathsf{G}_1+\mathsf{G}_\mathrm{T})$	1[P]	selective	q-BDHI
SPE-2 [20]	$(n+1)\mathsf{G}_1+2\mathsf{G}_2$	$\mathsf{G}_1+\mathsf{G}_2$	$\gamma(2n\mathsf{G}_1+\mathsf{G}_2+\mathsf{G}_\mathrm{T})$	2[P]	selective	DBDH
SPE_2	$(n+3)\mathsf{G}_1+\mathsf{G}_\mathrm{T}$	$5\mathsf{G}_2$	$\gamma((n+2)\mathsf{G}_1+\mathsf{G}_\mathrm{T}+n\mathbb{Z}_p)$	3[P]	adaptive	SXDH

Table 2 Comparison of efficient standard model DNF schemes.

6 Conclusion

We presented two large universe constructions of 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 of $\mathcal{O}(|\Theta^*|)$. 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.

References

1. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: ICALP. LNCS, vol. 4052, pp. 300–311. Springer (2006)
2. Abdalla, M., Kiltz, E., Neven, G.: Generalized key delegation for hierarchical identity-based encryption. In: ESORICS. LNCS, vol. 4734, pp. 139–154. Springer (2007)
3. 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)
4. Attrapadung, N.: *Dual System Encryption Framework in Prime-Order Groups*. IACR Cryptology ePrint Archive 2015, 390 (2015)
5. Attrapadung, N., Libert, B.: Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In: PKC. LNCS, vol. 6056, pp. 384–402. Springer (2010)
6. Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: EUROCRYPT. LNCS, vol. 3027, pp. 223–238. Springer (2004)
7. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: CRYPTO. LNCS, vol. 2139, pp. 213–229. Springer (2001)
8. 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)
9. Boyen, X.: General ad hoc encryption from exponent inversion IBE. In: EUROCRYPT. LNCS, vol. 4515, pp. 394–411. Springer (2007)
10. Chase, M., Meiklejohn, S.: Déjà Q: Using dual systems to revisit q-type assumptions. In: EUROCRYPT. LNCS, vol. 8441, pp. 622–639. Springer (2014)

11. 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)
12. Chen, J., Gong, J.: ABE with tag made easy. In: ASIACRYPT. LNCS, vol. 10625, pp. 35–65. Springer (2017)
13. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: ASIACRYPT. LNCS, vol. 4833, pp. 200–215. Springer (2007)
14. Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: Pairing. LNCS, vol. 4575, pp. 39–59. Springer (2007)
15. 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
16. Gong, J., Libert, B., Ramanna, S.C.: Compact IBBE and Fuzzy IBE from Simple Assumptions. In: SCN. LNCS, vol. 11035, pp. 563–582. Springer (2018)
17. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM CCS. pp. 89–98 (2006)
18. Jao, D., Yoshida, K.: Boneh-boyen signatures and the strong diffie-hellman problem. In: Pairing. LNCS, vol. 5671, pp. 1–16. Springer (2009)
19. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. Journal of Cryptology 30(4), 1116–1156 (2017)
20. Katz, J., Maffei, M., Malavolta, G., Schröder, D.: Subset predicate encryption and its applications. In: CANS. LNCS, Springer (2017)
21. 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)
22. Ramanna, S.C., Sarkar, P.: Efficient adaptively secure IBBE from the SXDH assumption. IEEE IT 62(10), 5709–5726 (2016)
23. Waters, B.: Efficient identity-based encryption without random oracles. In: EUROCRYPT. LNCS, vol. 3494, pp. 114–127. Springer (2005)
24. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: CRYPTO. LNCS, vol. 5677, pp. 619–636. Springer (2009)
25. Wee, H.: Dual system encryption via predicate encodings. In: TCC. LNCS, vol. 8349, pp. 455–479. Springer (2014)
26. Wee, H.: Déjà Q: Encore! un petit IBE. In: TCC-II. LNCS, vol. 9563, pp. 237–258. Springer (2016)