Embed-Augment-Recover: Function Private Predicate Encryption from Minimal Assumptions in the Public-Key Setting

Sikhar Patranabis and Debdeep Mukhopadhyay

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
sikhar.patranabis@iitkgp.ac.in, debdeep@cse.iitkgp.ernet.in

Abstract. We present a new class of public-key predicate encryption schemes that are provably function private in the standard model under well-known cryptographic assumptions, and assume predicate distributions satisfying realistic min-entropy requirements. More concretely, we present public-key constructions for identity-based encryption (IBE) and inner-product encryption (IPE) that are computationally function private in the standard model under a family of weaker variants of the DLIN assumption. Existing function private constructions in the public-key setting impose highly stringent requirements on the min-entropy of predicate distributions, thereby limiting their applicability in the context of real-world predicates. For example, the statistically function private constructions of Boneh, Raghunathan and Segev (CRYPTO'13 and ASIACRYPT'13) are inherently restricted to predicate distributions with min-entropy roughly proportional to λ, where λ is the security parameter. Our constructions allow relaxing this min-entropy requirement to $\omega(\log \lambda)$, while achieving a computational notion of function privacy against probabilistic polynomial-time adversaries, which suffices for most real-world applications. Our constructions also avoid the need for strong assumptions such as indistinguishability obfuscation.

Keywords: Predicate Encryption, Public-Key, Function Privacy, Computational Indistinguishability, Min-Entropy, Identity-Based Encryption, Inner-Product Encryption

1 Introduction

Predicate encryption schemes [1–3] in the public-key setting allow a single public-key to be associated with multiple secret-keys, where each secret-key corresponds to a Boolean predicate $f : \Sigma \longrightarrow \{0, 1\}$ over a pre-defined set of attributes Σ. A plaintext message in a predicate encryption scheme is an attribute-payload message pair $(I, M) \in \Sigma \times \mathcal{M}$, with $\mathcal{M}$ being the payload message space. A secret-key sk_f associated with a predicate f successfully decrypts a ciphertext C corresponding to a plaintext (I, M) and recovers the payload message M if and only if $f(I) = 1$. On the other hand, if $f(I) = 0$, attempting to decrypt C using sk_f returns $\perp$. A predicate encryption is said to be *attribute hiding* if the ciphertext C leaks no information about the underlying plaintext (I, M) to an adversary possessing benign secret-keys corresponding to predicates that do not trivially identify the attribute I.

Identity-Based Encryption. Identity-based encryption (IBE) [4–6] is the simplest sub-class of public-key predicate encryption. IBE supports a set of equality predicates of the form $f_{\mathsf{id}} : \Sigma \longrightarrow \{0,1\}$ defined as $f_{\mathsf{id}}(x) = 1$ if and only if $x = \mathsf{id}$. The attribute space in this case is a set of identities $\mathcal{ID}$, and each identity $\mathsf{id} \in \mathcal{ID}$ is associated with its own secret-key $\mathsf{sk}_{\mathsf{id}}$.

Inner-Product Encryption. Inner-product encryption (IPE) [2, 3, 7, 8] is a highly expressive sub-class of predicate encryption, supporting a set of predicates $f_{\overrightarrow{v}} : \Sigma \longrightarrow \{0,1\}$ over a vector space of attributes $\Sigma = \mathbb{F}_q^n$ (q being a λ-bit prime). Of particular interest is a specific form of IPE called zero-IPE [3] where for $\overrightarrow{v}, \overrightarrow{x} \in \Sigma$, we have $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$ if and only if $\langle \overrightarrow{v}, \overrightarrow{x} \rangle = 0$, where $\langle \overrightarrow{v}, \overrightarrow{x} \rangle$ denotes the inner-product of two vectors $\overrightarrow{v}$ and $\overrightarrow{x}$. IPE is powerful enough to encompass IBE and many other predicate encryption systems [3].

Searchable Encryption and Function Privacy. Predicate encryption provides a generic framework for searchable encryption supporting a wide range of query predicates including conjunctive, disjunctive, range and subset queries [9, 1–3]. For instance, a predicate encryption system can be used to realize a mail gateway that follows some special instructions to route encrypted mails based on their header information (e.g. if the mail is from the boss and needs to be treated as urgent). The mail gateway is given the secret-key corresponding to the predicate *is-urgent*, the mail header serves as the attribute, while the routing instructions can be used as the payload message. Another application could be a payment gateway that flags encrypted payments if they correspond to amounts beyond some pre-defined threshold X. The payment gateway is given the secret-key corresponding to the predicate *greater-than-X*, the payment amount itself serves as the attribute, while the flag signal is encoded as the payload message. The attribute hiding property of the predicate encryption scheme ensures that neither gateway learns any information about the plaintext data from the entire operation.

A natural question now arises: should the gateways in the aforementioned examples be able to learn the underlying predicate from the secret-keys given to them? The answer in most scenarios is *no* - the secret-key sk_f should ideally reveal nothing about the predicate f beyond the absolute minimum. This notion of predicate hiding security is commonly referred to as *function privacy*, and predicate encryption schemes satisfying this notion of security are described as *function private*.

1.1 Function Private Predicate Encryption in the Public-Key Setting

As pointed out by Boneh, Raghunathan and Segev in [10, 11], formalizing a realistic notion of function privacy in the context of public-key predicate encryption is, in general, not straightforward. Consider, for example, an adversary against an IBE scheme who is given a secret-key $\mathsf{sk}_{\mathsf{id}}$ corresponding to an identity id and has access to an encryption oracle. As long as the adversary has some apriori information that the identity id belongs to a set $\mathcal{S}$ such that $|\mathcal{S}|$ is at most polynomial in the security parameter λ, it can fully recover id from $\mathsf{sk}_{\mathsf{id}}$: it can simply resort to encrypting a random message M under each identity in $\mathcal{S}$, and decrypting using $\mathsf{sk}_{\mathsf{id}}$ to check for a correct recovery. Consequently, Boneh, Raghunathan and Segev [10, 11] consider a framework for function privacy under the minimal assumption that any predicate is sampled from a distribution with min-entropy at least super logarithmic in the security parameter λ. This rules out trivial attacks and results in a meaningful notion of function privacy in the public-key setting. However, their work leaves open the following important issues, which we address in this paper:

- The predicate encryption schemes proposed in [10, 11] are inherently restricted to satisfying a *statistical* notion of function privacy against computationally unbounded adversaries. For a vast majority of applications, a more relaxed *computational* notion of function privacy against probabilistically polynomial-time adversaries obviously suffices. Such a relaxation is crucial, since it potentially enables realizations of function-private predicate encryption based on weaker computational assumptions.

- Ideally, the function privacy guarantees of any public-key predicate encryption scheme should hold under the minimal assumption that the predicates are sampled from a distribution with min-entropy $k = \omega(\log \lambda)$ (where λ is the security parameter), so as to rule out trivial attacks. However, the statistically function private constructions in [10, 11] assume predicate distributions with min-entropy $k \geq \lambda$. This rather stringent assumption stems from their use of the universal hash lemma for arguing the statistical indistinguishability of secret-keys against unbounded adversaries, and limits the applicability of their constructions in the context of real-world predicates.

1.2 Our Contributions

In this paper, we present a new class of public-key predicate encryption schemes, that are computationally function private against probabilistic polynomial-time adversaries in the standard model, while making only the bare minimum assumptions on the min-entropy of the underlying predicate distributions. Our constructions are based on standard computational assumptions, and realize a relaxed notion of function privacy as compared to the constructions of Boneh, Raghunathan and Segev [10, 11], which is sufficient for most real-world applications. In order to prove function privacy, we adopt an indistinguishability-based framework that requires a secret-key corresponding to a predicate sampled from a distribution with min-entropy super logarithmic in the security parameter λ, to be *computationally indistinguishable* from another secret-key corresponding to a uniformly and independently sampled predicate.

Proposed Constructions. Our main result is a generic framework, denoted as *embed-augment-recover*, that takes an existing predicate encryption scheme and transforms it into a computationally function private one, *while retaining its original data privacy guarantees*. Our approach leads to the following constructions:

- We present a family of computationally function private identity-based encryption (IBE) schemes in the standard model. Our IBE constructions are based on the anonymous IBE scheme proposed by Gentry in [4]. Our schemes retain the selective data privacy guarantees of the original scheme, and are additionally computationally function private under progressively weaker variants of the well-known DLIN assumption. The detailed constructions of these schemes, along with the proofs of data and function privacy, are presented in Section 4.

- We then present a family of computationally function private inner-product encryption (IPE) schemes in the standard model. Our IPE constructions based on the seminal IPE scheme of Katz, Sahai and Waters [3]. Once again, our schemes retain the selectively attribute hiding property of the underlying scheme, and are computationally function private under progressively weaker variants of the DLIN assumption. The detailed constructions of these schemes, along with the proofs of data and function privacy, are presented in Section 5.

Advantages of Our Constructions. As compared to existing function private realizations of predicate encryption, our constructions offer the following advantages:

- **Relaxed Min-Entropy Requirements.** Our constructions relax the min-entropy assumptions on the underlying predicate distributions from $\omega(\lambda)$ in the constructions of Boneh, Raghunathan and Segev [10, 11], to the bare minimum requirement of $\omega(\log \lambda)$ for ruling out trivial breaches of function privacy. This is essentially enabled by relaxing the function privacy requirements from statistical privacy against unbounded adversaries to computational privacy against probabilistic polynomial-time adversaries.

- **Security in the Standard Model.** Agrawal et al. [12] have recently proposed a new universal composability-style definition of simulation-security for predicate encryption, capturing both data and function privacy. To the best of our knowledge, the only known public-key construction satisfying this *wishful* notion security is an IPE scheme proposed by Agrawal et al. themselves in [12]. This construction is, however, secure in the generic group model. Our constructions, on the other hand, achieve an indistinguishability-based notion of function privacy in the standard model.

- **Avoiding Strong Assumptions such as Obfuscation.** Certain existing approaches to achieving function privacy, such as that proposed by Iovino et al. in [13], are based on strong assumptions such as indistinguishability obfuscation.

In particular, the approach of Iovino et al. assumes the existence of a quasi-strong indistinguishability obfuscation algorithm over the class of all NC^1) circuits. While this makes their approach more generic, our constructions are based on practically realizable cryptographic primitives such as bilinear pairings, and entirely avoid such strong assumptions.

1.3 Overview of Our Approach: *Embed-Augment-Recover*

We present a three-step approach - embed, augment, recover - for achieving computationally function private predicate encryption schemes. We briefly describe the main ideas underlying each step, and exemplify them subsequently using a simple IBE scheme. We begin by briefly describing a public-key predicate encryption scheme.

Public-key Predicate Encryption. A public-key predicate encryption scheme for a class of predicates $\mathcal{F}$ over an attribute space Σ and a payload-message space $\mathcal{M}$ is a quadruple of probabilistic polynomial time algorithms $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. The Setup algorithm takes as input the security parameter λ, and generates the public parameter pp and the master-secret-key msk for the system. The key-generation algorithm, KeyGen takes as input the public parameter pp, the master-secret-key msk and a predicate $f \in \mathcal{F}$, and generates a secret-key sk_f corresponding to f. The Enc algorithm takes as input the public parameter pp, an attribute $I \in \Sigma$ and a payload-message $M \in \mathcal{M}$, and outputs the ciphertext $C = \mathsf{Enc}(\mathsf{pp}, I, M)$. The Dec algorithm takes as input the public parameter pp, a ciphertext C and a secret-key sk_f, and outputs either a payload-message $M \in \mathcal{M}$ or the symbol $\perp$.

Embed-Augment-Recover. Given a public-key predicate encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, we propose a generalized approach, termed as *embed-augment-recover*, to convert the same into a function-private one. The core idea of our approach is to base the computational function privacy of the modified scheme on a decisional hard problem. The steps to realize this idea are as follows:

1. **Embed:** The KeyGen algorithm is modified to *embed* a randomly sampled instance of the hard problem, along with a corresponding *check-point element*, in any secret-key sk_f. The check-point element ensures that decryption using sk_f succeeds if and only if the embedded instance is a *valid instance* of the hard problem. The necessary public parameters for sampling the random instance and the check-point element are included in pp generated by the Setup algorithm. The original master-secret-key msk is retained unchanged.

2. **Augment:** In order to synchronize encryption and decryption in the presence of the modified secret-keys, the ciphertext C generated by the Enc algorithm is *augmented* with additional random components, again derived using the updated public parameter pp. The challenge here is to ensure that this augmented ciphertext does not weaken the data privacy guarantees of the original predicate encryption scheme.

3. **Recover:** Finally, the Dec algorithm cleverly the additional ciphertext components to remove the *embedding* from the secret-key, and either recover the payload message M or return $\perp$.

An Example of Our Approach. We present an IBE scheme Π_1^{IBE}, that is achieved by applying our *embed-augment-recover* approach to Gentry's anonymous IBE scheme [4], which is selectively data private in the standard model. Our scheme retains the data privacy guarantees of Gentry's scheme, while also achieving computational function privacy under the well-known decisional linear (DLIN) assumption [14]. Let $\mathbb{G}$ be a group of prime order q and let g_1, g_2, g_3 be arbitrary generators for $\mathbb{G}$. The DLIN assumption is that the distribution ensembles:

$$\left\{\left(g_1, g_2, g_3, g_1^{a_1}, g_2^{a_2}, g_3^{a_1+a_2}\right)\right\}_{a_1,a_2 \xleftarrow{R} \mathbb{Z}_q^*} \text{ and } \left\{(g_1, g_2, g_3, g_1^{a_1}, g_2^{a_2}, g_3^{a_3})\right\}_{a_1,a_2,a_3 \xleftarrow{R} \mathbb{Z}_q^*}$$

are computationally indistinguishable, where $g_1, g_2, g_3 \xleftarrow{R} \mathbb{G}$. We now present the construction details for Π_1^{IBE}.

- Π_1^{IBE}.**Setup:** The setup algorithm in Gentry's scheme samples $s \xleftarrow{R} \mathbb{Z}_q^*$, where q is a λ-bit prime. The public parameters are (g, g^s, h), where g and h are randomly sampled generators of a bilinear group $\mathbb{G}$ of prime order q, while the master-secret-key is s. Our scheme additionally samples $x_1, x_2, x_3 \xleftarrow{R} \mathbb{Z}_q^*$ and $g_1, g_2, g_3 \xleftarrow{R} \mathbb{G}$. The modified public parameter pp and master-secret-key msk for our scheme are as follows:

$$\begin{aligned} \mathsf{pp} &= (g, g^s, h, g^{x_1}, g^{x_2}, g^{x_3}, g^{s\cdot x_1}, g^{s\cdot x_2}, g^{s\cdot x_3}, g_1, g_2, g_3, (g_1^{x_1} \cdot g_3^{x_3}), (g_2^{x_2} \cdot g_3^{x_3})) \\ \mathsf{msk} &= s \end{aligned}$$

  Note that the additional components of the public parameter pp are related to the DLIN assumption.

- Π_1^{IBE}.**KeyGen:** The key-generation algorithm in the Gentry's scheme computes a secret-key for an identity id as $\mathsf{sk}_{\mathsf{id}} = \left(y, (h \cdot g^{-y})^{1/(s-\mathsf{id})}\right)$, where $y \xleftarrow{R} \mathbb{Z}_q^*$. In our scheme, we augment the key generation process as follows. We additionally sample $y_1, y_2 \xleftarrow{R} \mathbb{Z}_q^*$, and output:

$$\mathsf{sk}_{\mathsf{id}} = \left(y, g_1^{y_1}, g_2^{y_2}, g_3^{y_1+y_2}, (g_1^{x_1} \cdot g_3^{x_3})^{y_1} \cdot (g_2^{x_2} \cdot g_3^{x_3})^{y_2} \cdot \left(h \cdot g^{-y}\right)^{1/(s-\mathsf{id})}\right)$$

  Note that we embed a randomly sampled valid DLIN instance $\left(g_1^{y_1}, g_2^{y_2}, g_3^{y_1+y_2}\right)$ in $\mathsf{sk}_{\mathsf{id}}$, along with an additional term of the form $(g_1^{x_1} \cdot g_3^{x_3})^{y_1} \cdot (g_2^{x_2} \cdot g_3^{x_3})^{y_2}$. This term serves as a check-point: it allows decryption only if the corresponding DLIN instance embedded in the secret-key is valid. Also observe that both the DLIN instance and the check-point element can be derived entirely from the additional elements in the public parameter pp.

- Π_1^{IBE}.**Enc:** An encryption of a message M for an identity id in Gentry's scheme is a tuple of the form $\left(g^{r\cdot(s-id)}, e\left(g,g\right)^{r}, M\cdot e\left(g,h\right)^{-r}\right)$, where $r \xleftarrow{R} \mathbb{Z}_q^*$. In our scheme, we augment the encryption process to produce the ciphertext:

$$C = \left(g^{r\cdot(s-id)}, g^{r\cdot x_1\cdot(s-id)}, g^{r\cdot x_2\cdot(s-id)}, g^{r\cdot x_3\cdot(s-id)}, e\left(g,g\right)^{r}, M\cdot e\left(g,h\right)^{-r}\right)$$

  Note that the augmented ciphertext in our scheme *retains unaltered* the ciphertext components of Gentry's scheme. The remaining ciphertext components are derived from the additional elements in the public parameter pp.

- Π_1^{IBE}.**Dec:** On input of a ciphertext $C = (c_0, c_1, c_2, c_3, c_4, c_5)$, and a secret-key $\mathsf{sk}_{\mathsf{id}} = (d_0, d_1, d_2, d_3, d_4)$, the decryption algorithm recovers the encrypted message M as:

$$M = \frac{c_5 \cdot c_4^{d_0} \cdot e\left(d_4, c_0\right)}{e\left(d_1, c_1\right)\cdot e\left(d_2, c_2\right)\cdot e\left(d_3, c_3\right)}$$

  Observe that at the core of the above computation is the original decryption procedure in the Gentry's scheme, with the additional components in the ciphertext and the secret-key canceling out each other via the pairing property. Note that this cancellation is subject to the condition that the DLIN instance originally embedded in $\mathsf{sk}_{\mathsf{id}}$ is valid, which is ensured by the check-point term in d_4.

A generalization of the above scheme is presented in greater detail in Section 4, along with proofs for data and function privacy.

1.4 Potential Extensions of Our Approach

Function Privacy for Private-Key Predicate Encryption. While we present our *embed-augment-recover* approach in the context of function privacy for public-key predicate encryption, it can readily be extended in the context of private-key predicate encryption. In the private-key setting, our approach works even if the underlying predicates are not necessarily sampled from distributions with at least super-logarithmic min-entropy. In particular, the core function privacy arguments for our constructions presented in this paper do not essentially rely on the unpredictability of the predicate distributions; this assumption is additionally made to rule out trivial attacks in the public-key setting. Consequently, our approach anticipates an expansion to the existing body of work in designing function private predicate encryption schemes in the private-key setting.

Function Privacy for Multi-Input Predicate Encryption. Multi-input predicate encryption (MIPE) introduced by Goldwasser et al. [15] is a generalization of functional encryption to the setting of multi-input predicates. An MIPE scheme has several encryption slots and each decryption key sk_f for a multi-input predicate f jointly decrypts the ciphertexts $\mathsf{Enc}(I_1), ..., \mathsf{Enc}(I_n)$ for all slots to obtain $f\left(I_1, ..., I_n\right)$ without revealing anything more about the encrypted attributes. In particular, this provides a framework to evaluate *bounded-norm* multi-input IPE: each predicate is specified by a collection of vectors $\overrightarrow{v}_1, \cdots, \overrightarrow{v}_n$, and takes as input a collection of vectors $\overrightarrow{x}_1, \cdots, \overrightarrow{x}_n$ to output $f_{\overrightarrow{v}_1, \cdots, \overrightarrow{v}_n}(\overrightarrow{x}_1, \cdots, \overrightarrow{x}_n) = \sum_{i=1}^{n} \langle \overrightarrow{v}_i, \overrightarrow{x}_i \rangle$. We point out that our technique can be easily generalized to obtain function private IPE schemes in the multi-input setting as follows: we first use our technique to obtain a function private IPE construction in the single-input setting, and then run n independent copies of this construction. The i^{th} copy is used to encrypt $\overrightarrow{x}_i$ in the i^{th} slot, while the new secret-key is the ensemble of the n secret-keys corresponding to $\overrightarrow{v}_1, \cdots, \overrightarrow{v}_n$. The decryption algorithm computes each inner product individually, and returns their sum. Although this means that the adversary also learns each individual inner product, this is an inherent leakage in the public-key setting and does not weaken the security guarantees.

1.5 Paper Organization

The remainder of this paper is organized as follows. Section 2 presents background material on predicate encryption and computational assumptions in bilinear groups. In Section 3, we formally define our framework for the computational function privacy of public-key predicate encryption. In Section 4, we present a family of selectively data private and computationally function private IBE schemes in the standard model. In Section 5, we present a family of selectively data private and computationally function private IPE schemes in the standard model. Finally, Section 6 concludes the paper and enumerates several extensions and open problems.

1.6 Notations Used

We write $x \xleftarrow{R} \chi$ to represent that an element x is sampled uniformly at random from a set $\mathcal{X}$. The output a of a deterministic algorithm $\mathcal{A}$ is denoted by $x \leftarrow \mathcal{A}$ and the output a' of a randomized algorithm $\mathcal{A}'$ is denoted by $x' \xleftarrow{R} \mathcal{A}'$. We refer to $\lambda \in \mathbb{N}$ as the security parameter, and denote by $\mathsf{exp}(\lambda)$, $\mathsf{poly}(\lambda)$ and $\mathsf{negl}(\lambda)$ any generic (unspecified) exponential function, polynomial function and negligible function in λ respectively. Note that a function $f : \mathbb{N} \rightarrow \mathbb{N}$ is said to be negligible in λ if for every positive polynomial p, $f(\lambda) < 1/p(\lambda)$ when λ is sufficiently large. Finally, for $a, b \in \mathbb{Z}$ such that $a \leq b$, we denote by $[a, b]$ the set of integers lying between a and b (both inclusive). The min-entropy of a random variable Y is called $\mathbf{H}_\infty(Y)$ and is evaluated as $\log(\max_y \Pr[Y = y])$; a random variable Y is said to be a k-source if $\mathbf{H}_\infty(Y) \geq k$. A (T, k)-block-source is a random variable $\mathbf{Y} = (Y_1, \cdots, Y_T)$ where for each $i \in [1, T]$ and $y_1, \cdots, y_{i-1}$, it holds that $\mathbf{H}_\infty(Y_i | Y_1 = y_1, \cdots, Y_{i-1} = y_{i-1}) \geq k$.

2 Preliminaries

Data Privacy of Public-Key Predicate Encryption. We recall the standard notion of *attribute hiding* data privacy for a predicate encryption scheme against an adaptive probabilistic polynomial-time adversary.

Definition 2.1 (Adaptively Data Private Predicate Encryption). A predicate encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *adaptively data private* if for any probabilistic polynomial-time adversary $\mathcal{A}$, the following holds:

$$\mathbf{Adv}^{\mathsf{DP}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \left| \Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{DP},\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{DP},\Pi,\mathcal{A}}(\lambda) = 1\right] \right| \leq \mathsf{negl}(\lambda)$$

where for each $\lambda \in \mathbb{N}$ and $b \in \{0,1\}$, the experiment $\mathsf{Expt}^{(b)}_{\mathsf{DP},\Pi,\mathcal{A}}(\lambda)$ is defined as:

1. $(\mathsf{pp}, \mathsf{msk}) \xleftarrow{R} \mathsf{Setup}\left(1^{\lambda}\right)$.
2. $\left(\left(I_0^*, M_0^*\right), \left(I_1^*, M_1^*\right), \mathsf{state}\right) \xleftarrow{R} \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{state})$, where $I_0^*, I_1^* \in \Sigma$ and $M_0^*, M_1^* \in \mathcal{M}$, subject to the restriction that for each predicate f_i with which $\mathcal{A}$ queries $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, we have $f_i\left(I_0^*\right) = f_i\left(I_1^*\right)$.
3. $C^* \xleftarrow{R} \mathsf{Enc}\left(\mathsf{pp}, I_b^*, M_b^*\right)$.
4. $b' \xleftarrow{R} \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}\left(C^*, \mathsf{state}\right)$, once again subject to the restriction that for each predicate f_i with which $\mathcal{A}$ queries $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, we have $f_i\left(I_0^*\right) = f_i\left(I_1^*\right)$.
5. Output b'.

The above notion of adaptive data privacy is referred to as DP throughout the rest of the paper. There also exists a *selective* variant of the above security notion, referred to as sDP throughout the rest of the paper, that requires the adversary to commit to the challenge pair of attributes $\left(I_0^*, I_1^*\right)$ before seeing the public parameters of the scheme.

The Generalized Decisional k-Linear Assumption (k-DLIN). Let $\mathbb{G}$ be a group of prime order q and let $g_1, \cdots, g_{k+1}, g_{k+2}$ be arbitrary generators for $\mathbb{G}$, for $k \geq 1$. The generalized decisional k-linear assumption is that the distribution ensembles:

$$\left\{\left(g_1, \cdots, g_{k+1}, g_{k+2}, g_1^{a_1}, \cdots, g_{k+1}^{a_{k+1}}, g_{k+2}^{\sum_{j=1}^{k+1} a_j}\right)\right\}_{a_1,\cdots,a_{k+1} \xleftarrow{R} \mathbb{Z}_q^*} \quad \text{and}$$

$$\left\{\left(g_1, \cdots, g_{k+1}, g_{k+2}, g_1^{a_1}, \cdots, g_{k+1}^{a_{k+1}}, g_{k+2}^{a_{k+2}}\right)\right\}_{a_1,\cdots,a_{k+1},a_{k+2} \xleftarrow{R} \mathbb{Z}_q^*}$$

are computationally indistinguishable, where $g_1, \cdots, g_{k+1}, g_{k+2} \xleftarrow{R} \mathbb{G}$.

Note that 1-DLIN is essentially equivalent to the well-known DLIN assumption [14]. Moreover, the k-DLIN assumption implies the $(k+1)$-DLIN assumption for all $k \geq 1$, but the reverse is not necessarily true. In other words, the k-DLIN assumption family is a family of progressively weaker assumptions.

3 Computational Function Privacy of Public-Key Predicate Encryption

In this section, we formally define the indistinguishability-based framework for computational function privacy of predicate encryption in the public-key setting. We consider adversaries that have access to the public parameters of the scheme, as well as a secret-key generation oracle. The adversary can also adaptively interact with a *real-or-random* function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$. This oracle takes as input any adversarially-chosen distribution over the class of predicates $\mathcal{F}$, subject to certain min-entropy requirements. It outputs a secret-key either for a predicate sampled from the given distribution, or for a predicate sampled uniformly at random from $\mathcal{F}$. At the end of the interaction, the adversary should be able to distinguish between these *real* and *random* modes of operation of $\mathsf{RoR}^{\mathsf{FP}}$ with only negligible probability.

Definition 3.1 (Real-or-Random Function Privacy Oracle). A real-or-random function privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ takes as input triplets of the form $(\mathsf{mode}, msk, \mathbf{F})$, where $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, msk is the master-secret-key of the predicate encryption scheme, and $\mathbf{F}$ is a circuit representing a distribution over the class of predicates $\mathcal{F}$. If $\mathsf{mode} = \mathsf{real}$, the oracle samples $f \xleftarrow{R} \mathbf{F}$, while if $\mathsf{mode} = \mathsf{rand}$, it samples $f \xleftarrow{R} \mathcal{F}$. It then responds with $\mathsf{sk}_f \xleftarrow{R} \mathsf{KeyGen}\,(msk, f)$.

Definition 3.2 (Computational Function Privacy). A predicate encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *computationally function private* if for any probabilistic polynomial-time adversary $\mathcal{A}$, the following holds:

$$\mathbf{Adv}^{\mathsf{FP}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathrm{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi,\mathcal{A}}(\lambda) = 1\right] \right| \leq \mathsf{negl}(\lambda)$$

where for each $\lambda \in \mathbb{N}$ and $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, the experiment $\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi,\mathcal{A}}(\lambda)$ is defined as follows:

1. $(\mathsf{pp}, \mathsf{msk}) \xleftarrow{R} \mathsf{Setup}\left(1^\lambda\right)$.
2. $b \xleftarrow{R} \mathcal{A}^{\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode},\mathsf{msk},\cdot),\mathsf{KeyGen}(\mathsf{msk},\cdot)}\left(1^\lambda, \mathsf{pp}\right)$, subject to the restriction that each $\mathbf{F}_i$ with which $\mathcal{A}$ queries $\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode}, \mathsf{msk}, \cdot)$ represents a distribution with min-entropy $k = \omega(\log\lambda)$.
3. Output b.

Note that our definitions are generic, and may be suitably adopted for IBE, IPE and other classes of predicate encryption.

Min-Entropy Requirements. In our definitions for computational function privacy, the adversary is allowed to adaptively issue a polynomial number of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle, as long as the queries correspond to distributions with min-entropy $k = \omega(\log\lambda)$. In the absence of this restriction, an adversary can trivially distinguish the between the real and random modes of the oracle by encrypting a message under each $f \in \mathbf{F}$, and then using the secret-key sk_f output by the oracle to check for successful decryption. In particular, the following restrictions are pertinent to the forthcoming discussions:

- A function privacy adversary against an IBE scheme is allowed to query the real-or-random oracle with a circuit $\mathbf{ID}^* \in \mathcal{ID}$ if and only if $\mathbf{ID}^*$ represents a k-source such that $k = \omega(\log\lambda)$.

- A function privacy adversary against an IPE scheme is allowed to query the real-or-random oracle with a circuit $\mathbf{V}^* = (V_1^*, \cdots, V_n^*) \in \mathbb{F}_q^n$ if and only if $\mathbf{V}^*$ is an (n,k)-block source such that $k = \omega(\log \lambda)$. Additionally, the component-wise distributions $\{V_i^*\}_{i \in [1,n]}$ should be mutually uncorrelated; otherwise, the adversary can create circuits corresponding to vectorial distributions with arbitrary inter-component correlations to trivially distinguish the real and random modes of operation of the oracle [11].

4 Computationally Function Private Identity-Based Encryption from the k-DLIN Assumption

In this section, we present an IBE scheme, denoted as Π_k^{IBE}, that is computationally function private under the k-DLIN assumption. The simplest version of the scheme (for $k = 1$) has already been illustrated in Section 1.3. In the forthcoming discussion, we present the generalized version of the scheme for any $k \geq 1$, along with detailed proofs for data and function privacy. As stated earlier, this scheme is obtained by applying our *embed-augment-recover* methodology to the anonymous IBE scheme proposed by Gentry in [4].

Gentry's IBE Scheme. We briefly recall the original IBE scheme of Gentry. Let $\mathsf{GroupGen}(1^\lambda)$ be a probabilistic polynomial-time algorithm that takes as input a security parameter λ, and outputs the tuple $(\mathbb{G}, \mathbb{G}_T, q, g, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of of order q (q being a λ-bit prime), g is a generator for $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map. Gentry's IBE scheme $\Pi_{\mathrm{G}}^{\mathrm{IBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is defined over the identity space $\mathcal{ID} = \mathbb{Z}_q^*$ and the message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ as follows:

- $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$.**Setup:** The setup algorithm samples $(\mathbb{G}, \mathbb{G}_T, q, g, e) \xleftarrow{R} \mathsf{GroupGen}(1^\lambda)$ on input the security parameter 1^λ. It also samples $s \xleftarrow{R} \mathbb{Z}_q^*$ and $h \xleftarrow{R} \mathbb{G}$, and outputs the public parameter pp and the master-secret-key msk as:

$$\mathsf{pp} = (g, g^s, h) \ , \ \mathsf{msk} = s$$

- $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$.**KeyGen:** On input the public parameter pp, the master-secret-key msk and an identity $\mathsf{id} \in \mathcal{ID}$, the key generation algorithm samples $y \xleftarrow{R} \mathbb{Z}_q^*$ and outputs the secret-key $\mathsf{sk}_{\mathsf{id}} = (d_0, d_1)$ where:

$$d_0 = y \ , \ d_1 = \left(h \cdot g^{-y}\right)^{1/(s - \mathsf{id})}$$

- $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$.**Enc:** On input the public parameter pp, an identity $\mathsf{id} \in \mathcal{ID}$ and a message $M \in \mathcal{M}$, the encryption algorithm samples $r \xleftarrow{R} \mathbb{Z}_q^*$ and outputs the ciphertext $C = (c_0, c_1, c_2)$ where:

$$c_0 = g^{r \cdot (s - id)} \ , \ c_1 = e(g, g)^r \ , \ c_2 = M \cdot e(g, h)^{-r}$$

- $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$.**Dec:** On input a ciphertext $C = (c_0, c_1, c_2)$ and a secret-key $\mathsf{sk}_{\mathsf{id}} = (d_0, d_1)$, the decryption algorithm computes:

$$M' = c_1^{d_0} \cdot c_2 \cdot e\left(d_1, c_0\right)$$

If $M' \in \mathcal{M}$, the decryption algorithm outputs M', else it outputs $\perp$.

While the above scheme is selectively data private, its function privacy cannot be based on any standard cryptographic assumption to the best of our knowledge.

4.1 Our Function Private IBE Scheme

We now present the construction for our function private IBE scheme Π_k^{IBE}. For the ease of understanding, we highlight the principal alterations made to Gentry's scheme as per our *embed-augment-recover* approach.

- Π_k^{IBE}.**Setup:** The setup algorithm samples $(\mathbb{G}, \mathbb{G}_T, q, g, e) \xleftarrow{R} \mathsf{GroupGen}(1^\lambda)$ on input the security parameter 1^λ. It also samples $s, x_1, x_2, \cdots, x_{k+2} \xleftarrow{R} \mathbb{Z}_q^*$ as well as $g_1, g_2, \cdots, g_{k+2}, h \xleftarrow{R} \mathbb{G}$. It outputs the public parameter pp and the master-secret-key msk as:

$$\mathsf{pp} = \left(g, g^s, h, \{g^{x_j}, g^{s \cdot x_j}\}_{j \in [1,k+1]}, g_1, \cdots, g_{k+2}, \{g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}\}_{j \in [1,k+1]}\right)$$
$$\mathsf{msk} = s$$

- Π_k^{IBE}.**KeyGen:** On input the public parameter pp, the master-secret-key msk and an identity $\mathsf{id} \in \mathcal{ID}$, the key generation algorithm samples $y, y_1, y_2 \cdots, y_{k+1} \xleftarrow{R} \mathbb{Z}_q^*$ and outputs the secret-key $\mathsf{sk}_{\mathsf{id}} = (d_0, d_1, \cdots, d_{k+3})$ where:

$$d_0 = y \quad , \quad d_j = g_j^{y_j} \text{ for } j \in [1, k+1]$$

$$d_{k+2} = g_{k+2}^{\sum_{j=1}^{k+1} y_j} \quad , \quad d_{k+3} = \left(\prod_{j=1}^{k+1} \left(g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}\right)^{y_j}\right) \cdot \left(h \cdot g^{-y}\right)^{1/(s-\mathsf{id})}$$

Observe that we embed a valid k-DLIN instance $\left(g_1^{y_1}, \cdots, g_{k+1}^{y_{k+1}}, g_{k+2}^{\sum_{j=1}^{k+1} y_j}\right)$ in the original secret-key of Gentry's scheme, along with the check-point element $\left(\prod_{j=1}^{k+1} \left(g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}\right)^{y_j}\right)$.

- Π_k^{IBE}.**Enc:** On input the public parameter pp, an identity $\mathsf{id} \in \mathcal{ID}$ and a message $M \in \mathcal{M}$, the encryption algorithm samples $r \xleftarrow{R} \mathbb{Z}_q^*$ and outputs the ciphertext $C = (c_0, c_1, \cdots, c_{k+4})$ where:

$$c_0 = g^{r \cdot (s - id)} \quad , \quad c_j = g^{r \cdot x_j \cdot (s - id)} \text{ for } j \in [1, k+2]$$
$$c_{k+3} = e\left(g, g\right)^r \quad , \quad c_{k+4} = M \cdot e\left(g, h\right)^{-r}$$

Observe that the augmented ciphertext retains unaltered the components of the original ciphertext in Gentry's scheme.

- Π_k^{IBE}.**Dec:** On input a ciphertext $C = (c_0, \cdots, c_{k+4})$ and a secret-key $\mathsf{sk}_{\mathsf{id}} = (d_0, \cdots, d_{k+3})$, the decryption algorithm computes:

$$M = \frac{c_{k+4} \cdot c_{k+3}^{d_0} \cdot e\left(d_{k+3}, c_0\right)}{\prod_{j=1}^{k+2} e\left(d_j, c_j\right)}$$

  If $M' \in \mathcal{M}$, the decryption algorithm outputs M', else it outputs $\perp$.

Correctness. Consider a ciphertext $C = (c_0, \cdots, c_{k+4})$ corresponding to a message M under an identity id, and a secret-key $\mathsf{sk}_{\mathsf{id}} = (d_0, \cdots, d_{k+3})$ corresponding to the same identity id. Then, we have:

$$\begin{aligned} M &= M \cdot e\left(g, h\right)^{-r} \cdot e\left(g, g\right)^{r \cdot y} \cdot e\left(\left(h \cdot g^{-y}\right)^{1/(s - \mathsf{id})}, g^{r \cdot (s - id)}\right) \\ &\quad \cdot \left(\frac{e\left(\left(\prod_{j=1}^{k+1}\left(g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}\right)^{y_j}\right), g^{r \cdot (s - id)}\right)}{\left(\prod_{j=1}^{k+1} e\left(g_j^{y_j}, \left(g^{x_j}\right)^{r \cdot (s - id)}\right)\right) \cdot e\left(g_{k+2}^{\sum_{j=1}^{k+1} y_j}, \left(g^{x_{k+2}}\right)^{r \cdot (s - id)}\right)}\right) \\ &= M \cdot \frac{\left(\prod_{j=1}^{k+1} e\left(g_j^{x_j \cdot y_j}, g^{r \cdot (s - id)}\right)\right) \cdot e\left(g_{k+2}^{x_{k+2} \cdot \sum_{j=1}^{k+1} y_j}, g^{r \cdot (s - id)}\right)}{\left(\prod_{j=1}^{k+1} e\left(g_j^{y_j}, \left(g^{x_j}\right)^{r \cdot (s - id)}\right)\right) \cdot e\left(g_{k+2}^{\sum_{j=1}^{k+1} y_j}, \left(g^{x_{k+2}}\right)^{r \cdot (s - id)}\right)} \\ &= M \end{aligned}$$

Therefore as long as the ciphertext and the secret-key correspond to the same identity, the message is recovered correctly. Again, when the ciphertext and the secret-key correspond to two different identities, say id and id' respectively, the decryption algorithm computes:

$$M' = M \cdot e\left(g, h\right)^{-r} \cdot e\left(g, g\right)^{r \cdot y} \cdot e\left(\left(h \cdot g^{-y}\right)^{1/\left(s - \mathsf{id}'\right)}, g^{r \cdot (s - id)}\right)$$

We may assume here that $\mathcal{M}$ is a small subset of $\mathbb{G}_T$, namely $|\mathcal{M}| < |\mathbb{G}_T|^{1/2}$. This is not very serious since the space of valid messages in reality is expected to be significantly smaller than $|\mathbb{G}_T|^{1/2}$. This restriction ensures that the probability of M' still lying in $\mathcal{M}$, for $y, r \xleftarrow{R} \mathbb{Z}_q^*$, is negligible in the security parameter λ. This completes the proof of correctness for our generalized IBE scheme Π_k^{IBE}.

4.2 Security of Our IBE Scheme

In this section, we formally prove the selective data privacy and computational function privacy of our IBE scheme Π_k^{IBE}.

Selective Data Privacy. We state the following theorem for the selective data privacy of Π_k^{IBE}:

Theorem 4.1 *Our IBE scheme Π_k^{IBE} is selectively data private in the standard model if Gentry's scheme $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$ is selectively data private in the standard model.*

Proof. Let $\mathcal{A}$ be any probabilistic polynomial-time adversary such that:

$$\mathbf{Adv}^{\mathsf{sDP}}_{\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{sDP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{sDP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] \right| = \epsilon$$

where $\epsilon > \mathsf{negl}(\lambda)$. We construct a polynomial-time algorithm $\mathcal{B}$ such that:

$$\mathbf{Adv}^{\mathsf{sDP}}_{\Pi_{\mathrm{G}}^{\mathrm{IBE}},\mathcal{B}}(\lambda) = \left| \Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{sDP},\Pi_{\mathrm{G}}^{\mathrm{IBE}},\mathcal{B}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{sDP},\Pi_{\mathrm{G}}^{\mathrm{IBE}},\mathcal{B}}(\lambda) = 1\right] \right| = \epsilon$$

$\mathcal{B}$ interacts with $\mathcal{A}$ in the selective data privacy experiment as follows:

- ***Init:*** $\mathcal{A}$ commits to the challenge identity pair $(\mathsf{id}_0^*, \mathsf{id}_1^*)$. $\mathcal{B}$ also commits to the same identity pair.

- ***Setup:*** $\mathcal{B}$ begins by obtaining the public parameter $\mathsf{pp} = (g, g^s, h)$ for $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$. It then samples $x_1, \cdots, x_{k+2} \xleftarrow{R} \mathbb{Z}_q^*$ and $g_1, g_2, \cdots, g_{k+2} \xleftarrow{R} \mathbb{G}$ and provides $\mathcal{A}$ with the modified public parameter:

$$\mathsf{pp}' = \left(g, g^s, h, \{g^{x_j}, g^{s \cdot x_j}\}_{j\in[1,k+1]}, g_1, \cdots, g_{k+2}, \{g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}\}_{j\in[1,k+1]}\right)$$

- ***Secret-Key Queries:*** When $\mathcal{A}$ issues a secret-key query for $\mathsf{id}_i \in \mathcal{ID}$, $\mathcal{B}$ forwards the query to the key-generation oracle for $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$, and receives $\mathsf{sk}_{\mathsf{id}_i} = (d_0, d_1)$. It then samples $y_1, y_2, \cdots, y_{k+1} \xleftarrow{R} \mathbb{Z}_q^*$ and responds to $\mathcal{A}$ with the secret-key:

$$\mathsf{sk}'_{\mathsf{id}_i} = \left(d_0, g_1^{y_1}, g_2^{y_2}, \cdots, g_{k+1}^{y_{k+1}}, g_{k+2}^{\sum_{j=1}^{k+1} y_j}, \left(\prod_{j=1}^{k+1} \left(g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}\right)^{y_j} \right) \cdot d_1 \right)$$

- ***Challenge:*** $\mathcal{A}$ outputs the challenge message pair (M_0^*, M_1^*). $\mathcal{B}$ outputs the same challenge message pair and receives the challenge ciphertext $C^* = (c_0^*, c_1^*, c_2^*)$ for $\Pi_{\mathrm{G}}^{\mathrm{IBE}}$. It then computes the challenge ciphertext for $\mathcal{A}$ as:

$$C'^* = (c_0^*, (c_0^*)^{x_1}, (c_0^*)^{x_2} \cdots, (c_0^*)^{x_{k+2}}, c_1^*, c_2^*)$$

- ***Output:*** Finally, $\mathcal{B}$ outputs the same bit b' as output by $\mathcal{A}$.

It is easy to see that $\mathcal{B}$'s simulation is perfect and hence, it has the same advantage ϵ as $\mathcal{A}$ in the selective data privacy game. This completes the proof of Theorem 4.1.

Computational Function Privacy. We state the following theorem for the computational function privacy of Π_k^{IBE}:

Theorem 4.2 *Our IBE scheme Π_k^{IBE} is function private under the k-*DLIN *assumption for identities sampled uniformly from k-sources with $k = \omega(\log \lambda)$.*

Proof. The proof follows directly from the following claim:

Claim 4.1 *For any probabilistic polynomial-time adversary $\mathcal{A}$, the following holds:*

$$\left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] \right| \leq \mathsf{negl}(\lambda)$$

To prove this claim, we assume the contrary. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary such that:

$$\left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] \right| = \epsilon$$

where $\epsilon > \mathsf{negl}(\lambda)$. We construct an algorithm $\mathcal{B}$ that solves an instance of the k-DLIN problem with non-negligible advantage $\epsilon' \geq \epsilon/2$. $\mathcal{B}$ receives as input a k-DLIN instance $\left(g_1, \cdots, g_{k+2}, g_1^{a_1}, \cdots, g_{k+2}^{a_{k+2}}\right)$ and interacts with $\mathcal{A}$ as follows:

- ***Setup:*** $\mathcal{B}$ samples $s, x_1, \cdots, x_{k+2} \xleftarrow{R} \mathbb{Z}_q^*$ and $g_1, \cdots, g_{k+2}, h \xleftarrow{R} \mathbb{G}$. It outputs the public parameter pp and the master-secret-key msk as:

$$\begin{aligned} \mathsf{pp} &= \left(g, g^s, h, \{g^{x_j}, g^{s \cdot x_j}\}_{j \in [1,k+1]}, g_1, \cdots, g_{k+2}, \{g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}\}_{j \in [1,k+1]}\right) \\ \mathsf{msk} &= s \end{aligned}$$

- ***Secret-Key Queries:*** When $\mathcal{A}$ issues a secret-key query for some identity id_i, $\mathcal{B}$ responds with $\mathsf{sk}_{\mathsf{id}_i} = \Pi_k^{\mathrm{IBE}}.\mathsf{KeyGen}\left(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_i\right)$.

- ***Real-or-Random Query:*** Suppose $\mathcal{A}$ queries the real-or-random oracle with $\mathbf{ID}^*$ - a circuit representing a k-source over the identity space $\mathcal{ID}$ such that $k = \omega\left(\log \lambda\right)$. $\mathcal{B}$ uniformly samples $\mathsf{mode} \xleftarrow{R} \{\mathsf{real}, \mathsf{rand}\}$. If $\mathsf{mode} = \mathsf{real}$, it samples $\mathsf{id}^* \xleftarrow{R} \mathbf{ID}^*$; otherwise, it samples $\mathsf{id}^* \xleftarrow{R} \mathcal{ID}$. It then responds with the secret-key $\mathsf{sk}_{\mathsf{id}^*}$ as:

$$\mathsf{sk}_{\mathsf{id}^*} = \left(y, g_1^{a_1}, \cdots, g_{k+2}^{a_{k+2}}, \left(\prod_{j=1}^{k+2} \left(g_j^{a_j}\right)^{x_j}\right) \cdot \left(h \cdot g^{-y}\right)^{1/(s - \mathsf{id}^*)}\right)$$

  where $g_1^{a_1}, \cdots, g_{k+2}^{a_{k+2}}$ are part of its input instance and $y \xleftarrow{R} \mathbb{Z}_q^*$.

- ***Output:*** Finally, $\mathcal{B}$ outputs the same bit b as output by $\mathcal{A}$.

Observe that the requirement that $\mathbf{ID}^*$ is a circuit representing a k-source over the identity space $\mathcal{ID}$, such that $k = \omega\left(\log \lambda\right)$, ensures that the polynomial-time adversary $\mathcal{A}$ cannot trivially determine if $\mathsf{id}^* \in \mathbf{ID}^*$ by exhaustively encrypting a random message M under each identity in $\mathbf{ID}^*$, and decrypting using $\mathsf{sk}_{\mathsf{id}^*}$ to check for a correct recovery. We now state and prove the following claims:

Claim 4.2 *When $a_{k+2} = \sum_{j=1}^{k+1} a_j$, the joint distribution of* mode *and the secret-key* $\mathsf{sk}_{\mathsf{id}^*}$ *in the simulation by the algorithm $\mathcal{B}$ is computationally indistinguishable from that in the experiment* $\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda)$.

Proof. Note that the sampling of id^* from either $\mathbf{ID}^*$ or $\mathcal{ID}$ by $\mathcal{B}$ is consistent with its random choice of mode. Additionally, when $a_{k+2} = \sum_{j=1}^{k+1} a_j$, the secret-key $\mathsf{sk}_{\mathsf{id}^*}$ takes the form:

$$\mathsf{sk}_{\mathsf{id}^*} = \left(y, g_1^{a_1}, \cdots, g_{k+2}^{\sum_{j=1}^{k+1} a_j}, \left(\prod_{j=1}^{k+1} \left(g_j^{x_j} \cdot g_{k+2}^{x_{k+2}} \right)^{a_j} \right) \cdot \left(h \cdot g^{-y} \right)^{1/(s-\mathsf{id}^*)} \right)$$

which is exactly what the response of the real-or-random oracle in the experiment $\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda)$ should be. This completes the proof of Claim 4.2.

Claim 4.3 *When a_{k+2} is uniformly random in $\mathbb{Z}_q^*$, the distribution of the secret-key $\mathsf{sk}_{\mathsf{id}^*}$ is independent of $\mathcal{B}$'s choice of mode.*

Proof. Note that $\mathcal{B}$'s choice of mode essentially determines whether the identity id^* is sampled from the adversarially-chosen distribution $\mathbf{ID}^*$ or uniformly at random from $\mathcal{ID}$. Consequently, the only component of $\mathsf{sk}_{\mathsf{id}^*}$ that should be distributed in accordance with mode is the final component $d_{k+3}^* = \delta \cdot (h \cdot g^{-y})^{1/(s-\mathsf{id}^*)}$, where $\delta = \prod_{j=1}^{k+2} \left(g_j^{a_j} \right)^{x_j}$. Now, let $g_j = g^{z_j}$ for $j \in [1, k+2]$, and let $h_j = g_j^{x_j} \cdot g_{k+2}^{x_{k+2}}$ for $j \in [1, k+1]$. Now, consider the following system of equations, determined by the public parameters pp and the secret-key $\mathsf{sk}_{\mathsf{id}^*}$:

$$\begin{aligned} \log_g h_1 &= x_1 \cdot z_1 + x_{k+2} \cdot z_{k+2} \\ &\vdots \\ \log_g h_{k+1} &= x_{k+1} \cdot z_{k+1} + x_{k+2} \cdot z_{k+2} \\ \log_g \delta &= \sum_{j=1}^{k+2} a_j \cdot x_j \cdot z_j \end{aligned}$$

Since a_{k+2} is uniformly random in $\mathbb{Z}_q^*$, with all but negligible probability, we have that $a_{k+2} \neq \sum_{j=1}^{k+1} a_j$, *which makes the aforementioned system of equations linearly independent*. Hence, the *conditional distribution* of δ (where the conditioning is on $\mathcal{B}$'s choice of mode and everything else in $\mathcal{A}$'s view) is uniform. In other words, δ acts as a perfect one-time pad in the only component of $\mathsf{sk}_{\mathsf{id}^*}$ that contains id^*, thus making it independent of mode [1]. This completes the proof of Claim 4.3.

[1] This is an adaptation of the same argument used in the proof of security for the well-known Cramer-Shoup cryptosystem [16], albeit in the context of the k-DLIN assumption

It now follows from Claims 4.2 and 4.3 that the advantage ϵ' of $\mathcal{B}$ in solving the k-DLIN instance may be quantified as:

$$\begin{aligned}
\epsilon' = \Bigg| \Pr\left[\mathcal{B}\left(g_1, \cdots, g_{k+2}, g_1^{a_1}, \cdots, g_{k+1}^{a_{k+1}}, g_{k+2}^{\sum_{j=1}^{k+1} a_j}\right) = 1\right] \\
- \Pr\left[\mathcal{B}\left(g_1, \cdots, g_{k+2}, g_1^{a_1}, \cdots, g_{k+1}^{a_{k+1}}, g_{k+2}^{a_{k+2}}\right) = 1\right] \Bigg| \\
&= \left| \Pr\left[\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| \\
&= \frac{1}{2}\left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| + \frac{1}{2}\left| \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| \\
&\geq \frac{1}{2}\left| \left(\Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right) - \left(\Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right) \right| \\
&= \frac{1}{2}\left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi_k^{\mathrm{IBE}},\mathcal{A}}(\lambda) = 1\right] \right| \\
&= \epsilon/2
\end{aligned}$$

where the probability is taken over all possible choices of $a_1, \cdots, a_{k+2} \xleftarrow{R} \mathbb{Z}_q^*$ and all possible choices of $g_1, \cdots, g_{k+2} \xleftarrow{R} \mathbb{G}$. This completes the proof of Theorem 4.2.

5 Computationally Function Private Inner-Product Encryption from the k-DLIN Assumption

In this section, we present a zero-IPE scheme, denoted as Π_k^{IPE}, that is computationally function private under the k-DLIN assumption. This scheme is obtained by applying our *embed-augment-recover* methodology to the zero-IPE scheme of Katz, Sahai and Waters (KSW)[3]. As in the original scheme, our scheme is defined over the set of attributes $\Sigma = \mathbb{Z}_N^n$ (N being a product of three primes q_1, q_2 and q_3), and the class of vectorial predicates $\mathcal{F} = \{f_{\overrightarrow{v}} \mid \overrightarrow{v} \in \mathbb{Z}_N^n\}$, such that for $I = (I_1, \cdots, I_n) \in \mathbb{Z}_N^n$, we have $f_{\overrightarrow{v}}(I) = 1$ if and only if $\langle \overrightarrow{v}, I \rangle = 0 \bmod N$.

Construction Overview. Let $\mathbb{G}$ be a bilinear group of order $N = q_1 q_2 q_3$ (each of q_1, q_2 and q_3 being λ-bit primes), and let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ denote the subgroups of $\mathbb{G}$ of order q_1, q_2 and q_3, respectively. Also, let $\hat{e} : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ be an efficiently computable non-degenerate bilinear map, where $\mathbb{G}_T$ is also a group of order N. Note that if g is the generator for $\mathbb{G}$, then the element $g_1 = g^{q_2 \cdot q_3}$ is a generator for $\mathbb{G}_1$, the element $g_2 = g^{q_1 \cdot q_3}$ is a generator for $\mathbb{G}_2$, and the element $g_3 = g^{q_1 \cdot q_2}$ is a generator for $\mathbb{G}_3$. Furthermore, for any elements $h_1 \in \mathbb{G}_1$, $h_2 \in \mathbb{G}_2$ and $h_3 \in \mathbb{G}_3$, we have $\hat{e}(h_1, h_2) = \hat{e}(h_2, h_3) = \hat{e}(h_1, h_3) = 1$. Also, let $\mathsf{GroupGen}'(1^\lambda)$ be a probabilistic polynomial-time algorithm that takes as input a security parameter λ, and outputs the tuple $(\mathbb{G}, \mathbb{G}_T, q_1, q_2, q_3, g_1, g_2, g_3, \hat{e})$. Finally, the payload message space $\mathcal{M}$ is assumed to be a small subset of $\mathbb{G}_T$, namely $|\mathcal{M}| < |\mathbb{G}_T|^{1/2}$. Our function private zero-IPE scheme uses the three subgroups for three distinct roles:

- The subgroup $\mathbb{G}_2$ is used to encode the vectors $\overrightarrow{v}$ and I in the secret-key and the ciphertexts, respectively, and to compute the inner product $\langle \overrightarrow{v}, I \rangle$ in the exponent of a bilinear map computation.

- The subgroup $\mathbb{G}_1$ serves a dual purpose in our scheme. On one hand, it has the effect of *masking* the inner product computation in $\mathbb{G}_2$, and preventing the adversary from improperly manipulating the computation in any way to reveal information about the underlying attributes. In particular, it is pivotal in ensuring the non-malleability of the secret-keys and ciphertexts generated by the scheme. On the other hand, it is in the $\mathbb{G}_1$ subgroup that we incorporate our *embed-augment-recover* methodology to achieve computational function privacy.

- The subgroup $\mathbb{G}_3$ serves as an additional layer of masking for the other subgroups. In particular, random elements sampled from $\mathbb{G}_3$ are multiplied with various components in both the secret-keys as well as the ciphertexts to hide possible information leakages from the subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$.

Alterations to the KSW Scheme. We apply our *embed-augment-recover* approach to the KSW scheme to achieve computational function privacy. We provide an informal overview of the alterations made to the original scheme; the detailed construction is presented subsequently:

- **Secret-Key Generation:** The KSW scheme generates secret-keys of the form $\mathsf{sk}_{\overrightarrow{v}} = \left(d_0, \{d_{1,i}, d_{2,i}\}_{i\in[1,n]}\right)$, where the tuple $(d_{1,i}, d_{2,i})$ corresponds to the i^{th} component of the predicate vector $\overrightarrow{v}$. In our scheme, we modify the key-generation algorithm to generate secret-keys of the form:

  $$\mathsf{sk}'_{\overrightarrow{v}} = \left(d_0, \{d_{1,i}^j, d_{2,i}^j\}_{i\in[1,n],j\in[0,k+2]}\right)$$

  such that for each $i \in [1,n]$, the components $\{d_{1,i}^j\}_{j\in[0,k+2]}$ and $\{d_{2,i}^j\}_{j\in[0,k+2]}$ are obtained by embedding independent and uniformly random valid k-DLIN instances in the original secret-key components components $d_{1,i}$ and $d_{2,i}$, respectively. All embeddings are performed in the group $\mathbb{G}_1$; consequently, the function privacy guarantees of our zero-IPE scheme follow from the hardness of the k-DLIN problem in the group $\mathbb{G}_1$.

- **Encryption:** As in our IBE construction, the modified encryption algorithm in our IPE construction retains unaltered the original ciphertext components from the KSW scheme. The ciphertext is augmented to include additional components that synchronize decryption with the modified secret-keys. These components naturally belong to the group $\mathbb{G}_1$ (same as the k-DLIN instances embedded in the modified secret-keys), and are additionally masked using uniformly random elements from $\mathbb{G}_3$. The masking ensures that the augmented ciphertexts do not weaken the data privacy guarantees of the original scheme.

- **Decryption:** Finally, the decryption algorithm uses the aforementioned additional ciphertext components to remove the effect of embedding the k-DLIN instances in the secret-key, and recovers the payload message M or returns $\perp$.

5.1 Construction Details for Our Zero-IPE Scheme

We now present the detailed construction for our zero-IPE scheme Π_k^{IPE}. Due to space constraints, we avoid presenting the original IPE scheme of Katz, Sahai and Waters (the reader is referred to [3] for details of the original construction). However, we highlight the alterations made to the original scheme for ease of understanding.

- Π_k^{IPE}.**Setup:** The setup algorithm samples the following:
  - $(\mathbb{G}, \mathbb{G}_T, q_1, q_2, q_3, g_1, g_2, g_3, \hat{e}) \xleftarrow{R} \mathsf{GroupGen}'(1^\lambda)$
  - $\{x_{1,j}, x_{2,j} \xleftarrow{R} \mathbb{Z}_{q_1}^*\}_{j\in[1,k+2]}$ and $\{g_{1,j}, g_{2,j} \xleftarrow{R} \mathbb{G}_1\}_{j\in[1,k+2]}$
  - $\{h_{1,i}, h_{2,i} \xleftarrow{R} \mathbb{G}_1\}_{i\in[1,n]}$ and $\{R_{1,i}^j, R_{2,i}^j \xleftarrow{R} \mathbb{G}_3\}_{i\in[1,n],j\in[0,k+2]}$
  - $h \xleftarrow{R} \mathbb{G}_1$, $\gamma \xleftarrow{R} \mathbb{Z}_{q_1}^*$ and $R_3 \xleftarrow{R} \mathbb{G}_3$

  Next, it sets:

$$
\begin{aligned}
&Q = g_2 \cdot R_3 \\
&S_{1,i}^0 = h_{1,i} \cdot R_{1,i}^0 \ , \ S_{2,i}^0 = h_{2,i} \cdot R_{2,i}^0 \ \text{for } i \in [1,n] \\
&S_{1,i}^j = h_{1,i}^{x_{1,j}} \cdot R_{1,i}^j \ , \ S_{2,i}^j = h_{2,i}^{x_{2,j}} \cdot R_{2,i}^j \ \text{ for } i \in [1,n], j \in [1,k+2]
\end{aligned}
$$

  and outputs the public parameter pp and the master-secret-key msk as:

$$
\begin{aligned}
\mathsf{pp} = \Big(& g_1, g_3, Q, \{S_{1,i}^j, S_{2,i}^j\}_{i\in[1,n],j\in[0,k+2]}, \hat{e}(g_1,h)^\gamma, \\
& \{g_{1,j}, g_{2,j}\}_{j\in[1,k+2]}, \{g_{1,j}^{x_{1,j}} \cdot g_{1,k+2}^{x_{1,k+2}}, g_{2,j}^{x_{2,j}} \cdot g_{2,k+2}^{x_{2,k+2}}\}_{j\in[1,k+1]} \Big) \\
\mathsf{msk} = \big(& q_1, q_2, q_3, g_2, \{h_{1,i}, h_{2,i}\}_{i\in[1,n]}, h^\gamma \big)
\end{aligned}
$$

- Π_k^{IPE}.**KeyGen:** On input the public parameter pp, the master-secret-key msk and a vector $\overrightarrow{v} = (v_1, \cdots, v_n)$, the key generation algorithm samples the following:
  - $\{z_{1,i}, z_{2,i} \xleftarrow{R} \mathbb{Z}_{q_1}^*\}_{i\in[1,n]}$
  - $\{y_{1,i}^j, y_{2,i}^j \xleftarrow{R} \mathbb{Z}_{q_1}^*\}_{i\in[1,n],j\in[1,k+1]}$
  - $Q_4 \xleftarrow{R} \mathbb{G}_2$, $R_5 \xleftarrow{R} \mathbb{G}_3$
  - $f_1, f_2 \xleftarrow{R} \mathbb{Z}_{q_2}^*$

  As in the original scheme, it first sets:

$$
d_0 = Q_4 \cdot R_5 \Big/ \left(h^\gamma \cdot \prod_{i=1}^{n} h_{1,i}^{z_{1,i}} \cdot h_{2,i}^{z_{2,i}} \right)
$$

Next, it sets:

$$d_{1,i}^{0} = g_1^{z_{1,i}} \cdot g_2^{f_1 \cdot v_i} \Bigg/ \left(\prod_{j=1}^{k+1} \left(g_{1,j}^{x_{1,j}} \cdot g_{1,k+2}^{x_{1,k+2}} \right)^{y_{1,i}^{j}} \right) \text{ for } i \in [1,n]$$

$$d_{2,i}^{0} = g_1^{z_{2,i}} \cdot g_2^{f_2 \cdot v_i} \Bigg/ \left(\prod_{j=1}^{k+1} \left(g_{2,j}^{x_{2,j}} \cdot g_{2,k+2}^{x_{2,k+2}} \right)^{y_{2,i}^{j}} \right) \text{ for } i \in [1,n]$$

Finally, it sets the following additional components:

$$d_{1,i}^{j} = g_{1,j}^{y_{1,i}^{j}} \quad , \quad d_{2,i}^{j} = g_{2,j}^{y_{2,i}^{j}} \text{ for } i \in [1,n], j \in [1,k+1]$$

$$d_{1,i}^{k+2} = g_{1,k+2}^{\sum_{j=1}^{k+1} y_{1,i}^{j}} \quad , \quad d_{2,i}^{k+2} = g_{2,k+2}^{\sum_{j=1}^{k+1} y_{2,i}^{j}} \text{ for } i \in [1,n]$$

and outputs the secret-key $\mathsf{sk}_{\overrightarrow{v}}$ as:

$$\mathsf{sk}_{\overrightarrow{v}} = \left(d_0, \{d_{1,i}^{j}, d_{2,i}^{j}\}_{i \in [1,n], j \in [0,k+2]} \right)$$

Note that the additional embeddings in the secret-key component $d_{1,i}^{0}$ and $d_{2,i}^{0}$ for $i \in [1,n]$ check-point the validity of the corresponding k-DLIN instances.

- Π_k^{IPE}.**Enc:** On input the public parameter pp, an attribute $I = (I_1, \cdots, I_n) \in \mathbb{Z}_N^n$ and a payload message $M \in \mathcal{M}$, the encryption algorithm samples the following:
  - $r, \alpha, \beta \xleftarrow{R} \mathbb{Z}_N^*$
  - $\{Q_{6,i}^{j}, Q_{7,i}^{j} \xleftarrow{R} \mathbb{G}_2\}_{i \in [1,n], j \in [1,k+2]}$
  - $\{R_{8,i}^{j}, R_{9,i}^{j} \xleftarrow{R} \mathbb{G}_3\}_{i \in [1,n], j \in [0,k+2]}$

  where the $Q_{6,i}^{j}$ and $Q_{7,i}^{j}$ values are sampled using Q from the public parameter. It then sets the following:

$$c_0 = g_1^r$$

$$c_{1,i}^{0} = \left(S_{1,i}^{0}\right)^r \cdot Q^{\alpha \cdot I_i} \cdot R_{8,i}^{0} \ , \ c_{2,i}^{0} = \left(S_{2,i}^{0}\right)^r \cdot Q^{\beta \cdot I_i} \cdot R_{9,i}^{0} \text{ for } i \in [1,n]$$

$$c_{1,i}^{j} = \left(S_{1,i}^{j}\right)^r \cdot Q_{6,i}^{j} \cdot R_{8,i}^{j} \quad , \quad c_{2,i}^{j} = \left(S_{2,i}^{j}\right)^r \cdot Q_{7,i}^{j} \cdot R_{9,i}^{j} \text{ for } i \in [1,n], j \in [1,k+2]$$

  Finally, it sets $c_3 = M \cdot \left(\hat{e}\left(g_1, h\right)^{\gamma}\right)^r$ and outputs the ciphertext C as:

$$C = \left(c_0, \{c_{1,i}^{j}, c_{2,i}^{j}\}_{i \in [1,n], j \in [0,k+2]}, c_3 \right)$$

  Observe that the *augmented* ciphertext retains unaltered the ciphertext components of the original scheme. Also, notice that the additional ciphertext components $\{c_{1,i}^{j}, c_{2,i}^{j}\}_{i \in [1,n], j \in [1,k+2]}$ are masked using random elements from both $\mathbb{G}_2$ and $\mathbb{G}_3$. The double masking is necessary to ensure data privacy, as will be explained in the subsequent discussion.

- Π_k^{IPE}.**Dec:** On input a ciphertext $C = \left(c_0, \{c_{1,i}^j, c_{2,i}^j\}_{i\in[1,n],j\in[0,k+2]}\right)$ and a secret-key $\mathsf{sk}_{\overrightarrow{v}} = \left(d_0, \{d_{1,i}^j, d_{2,i}^j\}_{i\in[1,n],j\in[0,k+2]}\right)$, the decryption algorithm computes:

$$M' = c_3 \cdot \hat{e}\left(d_0, c_0\right) \cdot \left(\prod_{i=1}^{n}\prod_{j=0}^{k+2} \hat{e}\left(d_{1,i}^j, c_{1,i}^j\right) \cdot \hat{e}\left(d_{2,i}^j, c_{2,i}^j\right)\right)$$

If $M' \in \mathcal{M}$, the decryption algorithm outputs M', else it outputs $\perp$.

Correctness. To see that correctness holds for our zero-IPE scheme, let C and $\mathsf{sk}_{\overrightarrow{v}}$ be as described in Section 5. Then we have:

$$\begin{aligned}
M' &= c_3 \cdot \hat{e}\left(d_0, c_0\right) \cdot \left(\prod_{i=1}^{n}\prod_{j=0}^{k+2} \hat{e}\left(d_{1,i}^j, c_{1,i}^j\right) \cdot \hat{e}\left(d_{2,i}^j, c_{2,i}^j\right)\right) \\
&= M \cdot \left(\prod_{i=1}^{n} \hat{e}\left(g_2^{f_1\cdot v_i}, g_2^{\alpha\cdot I_i}\right) \cdot \hat{e}\left(g_2^{f_2\cdot v_i}, g_2^{\beta\cdot I_i}\right)\right) \\
&\qquad \cdot \left(\frac{\left(\hat{e}\left(g_1, h\right)^{\gamma}\right)^r \cdot \prod_{i=1}^{n} \hat{e}\left(g_1^{z_{1,i}}, h_{1,i}^r\right) \cdot \hat{e}\left(g_1^{z_{2,i}}, h_{2,i}^r\right)}{\hat{e}\left(h^{\gamma}, g_1^r\right) \cdot \hat{e}\left(\prod_{i=1}^{n} h_{1,i}^{z_{1,i}} \cdot h_{2,i}^{z_{2,i}}, g_1^r\right)}\right) \\
&\qquad \cdot \prod_{i=1}^{n} \left(\frac{\left(\prod_{j=1}^{k+1} \hat{e}\left(g_{1,j}^{y_{1,i}^j}, \left(h_{1,i}^{x_{1,j}}\right)^r\right)\right) \cdot \hat{e}\left(g_{1,k+2}^{\sum_{j=1}^{k+1} y_{1,i}^j}, \left(h_{1,i}^{x_{1,k+2}}\right)^r\right)}{\hat{e}\left(\prod_{j=1}^{k+1}\left(g_{1,j}^{x_{1,j}} \cdot g_{1,k+2}^{x_{1,k+2}}\right)^{y_{1,i}^j}, h_{1,i}^r\right)}\right) \\
&\qquad \cdot \prod_{i=1}^{n} \left(\frac{\left(\prod_{j=1}^{k+1} \hat{e}\left(g_{2,j}^{y_{2,i}^j}, \left(h_{2,i}^{x_{2,j}}\right)^r\right)\right) \cdot \hat{e}\left(g_{2,k+2}^{\sum_{j=1}^{k+1} y_{2,i}^j}, \left(h_{2,i}^{x_{2,k+2}}\right)^r\right)}{\hat{e}\left(\prod_{j=1}^{k+1}\left(g_{2,j}^{x_{2,j}} \cdot g_{2,k+2}^{x_{2,k+2}}\right)^{y_{2,i}^j}, h_{2,i}^r\right)}\right) \\
&= M \cdot \prod_{i=1}^{n} \hat{e}\left(g_2, g_2\right)^{(\alpha f_1 + \beta f_2)\cdot v_i \cdot I_i} \\
&= M \cdot \hat{e}\left(g_2, g_2\right)^{(\alpha f_1 + \beta f_2 \bmod q_2)\cdot\langle \overrightarrow{v}, I\rangle}
\end{aligned}$$

where α, β are uniformly random in $\mathbb{Z}_N^*$ and f_1, f_2 are uniformly random in $\mathbb{Z}_{q_2}^*$. Now, if $\langle \overrightarrow{v}, I\rangle = 0 \bmod N$, then we have $M' = M$. If $\langle \overrightarrow{v}, I\rangle \neq 0 \bmod N$, there are two cases: if $\langle \overrightarrow{v}, I\rangle \neq 0 \bmod q_2$, then with all but negligible probability (over random choice of α, β, f_1, f_2), M' does not lie in $\mathbb{G}_T$ (since $\mathcal{M}$ is a small subset of $\mathbb{G}_T$). Otherwise, we have $\langle \overrightarrow{v}, I\rangle = 0 \bmod q_2$, in which case M' will always be equal to M. However, this would amount to revealing a non-trivial factor of N, and hence, must occur with negligible probability.

5.2 Security of Our IPE Scheme

In this section, we formally prove the selective data privacy and computational function privacy of our zero-IPE scheme Π_k^{IPE}.

Selective Data Privacy. The original KSW scheme was proven to be selectively data private in [3]. We state the following theorem for the data privacy of Π_k^{IPE}:

Theorem 5.1 *Our zero-IPE scheme Π_k^{IPE} retains the selective data privacy guarantees of the original* KSW *scheme.*

Proof. Suppose there exists a probabilistic polynomial-time adversary $\mathcal{A}$ against Π_k^{IPE} with non-negligible advantage ϵ in the selective data privacy game. We construct a polynomial-time selective data privacy adversary $\mathcal{B}$ against the original KSW zero-IPE scheme with the same advantage ϵ. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

- ***Init:*** $\mathcal{A}$ commits to the challenge vector pair $\left(\overrightarrow{v^*}_0, \overrightarrow{v^*}_1\right)$. $\mathcal{B}$ also commits to the same challenge vector pair.

- ***Setup:*** $\mathcal{B}$ obtains the public parameter pp from the challenger for the original KSW scheme, and modifies it by including the additional components related to the k-DLIN problem, as highlighted in the construction of Π_k^{IPE}. In particular, observe that the $S_{1,i}^0$ and $S_{1,i}^1$ values for $i \in [1,n]$ are already provided to $\mathcal{B}$ by the challenger as part of pp. $\mathcal{B}$ now additionally samples $\{x_{1,j}, x_{2,j}\}_{j\in[1,k+2]} \xleftarrow{R} \mathbb{G}_1$. For $i \in [1,n]$ and $j \in [1,k+2]$, it samples $R_{1,i}^j, R_{2,i}^j \xleftarrow{R} \mathbb{G}_3$ and sets $S_{1,i}^j = \left(S_{1,i}^0\right)^{x_{1,j}} \cdot R_{1,i}^j$ and $S_{2,i}^j = \left(S_{2,i}^0\right)^{x_{2,j}} \cdot R_{2,i}^j$, respectively. To simulate the remaining components, $\mathcal{B}$ samples $\{g_{1,j}, g_{2,j}\}_{j\in[1,k+2]} \xleftarrow{R} \mathbb{G}_1$, and appropriately combines them with the set of values $\{x_{1,j}, x_{2,j}\}_{j\in[1,k+2]}$ sampled earlier. In other words, $\mathcal{B}$ can ensure that the modified public parameter has the same distribution as in the real world from $\mathcal{A}$'s point of view, *without the knowledge of the master-secret-key of the original* KSW *scheme*. The modified public parameter is then provided to $\mathcal{A}$.

- ***Secret-Key Queries:*** Suppose $\mathcal{A}$ issues a secret-key query for $\overrightarrow{v} \in \mathbb{Z}_N^n$. $\mathcal{B}$ forwards the same query vector to the key-generation oracle for the KSW scheme, and receives $\mathsf{sk}_{\overrightarrow{v}} = \left(d_0, \{d_{1,i}^0, d_{2,i}^0\}_{i\in[1,n]}\right)$ as response. It then uses the public parameter components $\{g_{1,j}, g_{2,j}\}_{j\in[1,k+2]}$, and the values $\{x_{1,j}, x_{2,j}\}_{j\in[1,k+2]}$ sampled during setup, to create and embed a pair of independent and uniformly random valid k-DLIN instances in $\mathsf{sk}_{\overrightarrow{v}}$, along with the corresponding checkpointing elements (as described in the key-generation procedure for Π_k^{IPE}). Since the embedding again requires no knowledge of the master-secret-key of the original scheme, the modified secret-key generated by $\mathcal{B}$ has the same distribution from $\mathcal{A}$'s point of view as in the real world.

- ***Challenge:*** $\mathcal{A}$ outputs the challenge message pair (M_0^*, M_1^*). $\mathcal{B}$ queries the the challenger for the KSW scheme with the same challenge message pair, and receives the challenge ciphertext $C^* = \left(c_0^*, \{{c^*}_{1,i}^0, {c^*}_{2,i}^0\}_{i\in[1,n]}, c_3^*\right)$. The challenge ciphertext for $\mathcal{A}$ retains each of these components as is. It remains to simulate the additional ciphertext components $\{{c^*}_{1,i}^j, {c^*}_{2,i}^j\}_{i\in[1,n],j\in[1,k+2]}$. $\mathcal{B}$ samples
  - $\{{Q^*}_{1,i}^j, {Q^*}_{2,i}^j \xleftarrow{R} \mathbb{G}_2\}_{i\in[1,n],j\in[1,k+2]}$
  - $\{{R^*}_{1,i}^j, {R^*}_{2,i}^j \xleftarrow{R} \mathbb{G}_3\}_{i\in[1,n],j\in[1,k+2]}$

  and sets:

$$
\begin{aligned}
{c^*}_{1,i}^j &= \left({c^*}_{1,i}^0\right)^{x_{1,j}} \cdot {Q^*}_{1,i}^j \cdot {R^*}_{1,i}^j \text{ for } i \in [1,n], j \in [1,k+2] \\
{c^*}_{2,i}^j &= \left({c^*}_{2,i}^0\right)^{x_{1,j}} \cdot {Q^*}_{2,i}^j \cdot {R^*}_{2,i}^j \text{ for } i \in [1,n], j \in [1,k+2]
\end{aligned}
$$

  Since the additional ciphertext components generated by $\Pi_k^{\mathrm{IPE}}.\mathsf{Enc}$ in the real world are also masked using random elements from both $\mathbb{G}_2$ and $\mathbb{G}_3$, the ciphertext simulated by $\mathcal{B}$ is indistinguishable from a real world ciphertext, in $\mathcal{A}$'s view. This in turn justifies double masking the additional ciphertext components in $\Pi_k^{\mathrm{IPE}}.\mathsf{Enc}$.

- ***Output:*** Finally, $\mathcal{B}$ outputs the same bit b' as output by $\mathcal{A}$.

It is easy to see that $\mathcal{B}$'s simulation is perfect and hence, it has the same advantage ϵ as $\mathcal{A}$ in the selective data privacy game. This completes the proof of Theorem 5.1.

Computational Function Privacy. We state the following theorem for the computational function privacy of Π_k^{IPE}:

Theorem 5.2 *Our zero-IPE scheme Π_k^{IPE} is computationally function private under the k-DLIN assumption for predicate vectors sampled uniformly from (n,k)-block sources with $k = \omega(\log \lambda)$.*

Proof. We present the detailed proof for the above theorem. Our aim is to show that any probabilistic poly-time adversary $\mathcal{A}$ cannot distinguish between the real and random modes of operation of the function privacy oracle, provided that the oracle is queried with circuits that sample sufficiently unpredictable distributions over the space of predicates. In particular, such distributions should be (n,k)-block sources over $\mathbb{Z}_N^n$, such that each component of a vector $\overrightarrow{v}$ sampled from an adversarially chosen distribution has a min-entropy of $k = \omega(\log \lambda)$, and is uncorrelated with all other components. As already mentioned, this restriction is a bare minimum to rule out trivial breaches of function privacy. We define a series of hybrid experiments $\mathsf{Expt}^{\mathsf{mode},m}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$ for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $m \in [0,n]$ as follows:

- $\mathsf{Expt}^{\mathsf{mode},0}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$ is identical to $\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$.

- $\mathsf{Expt}^{\mathsf{mode},m}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$ for $m \in [1,n]$ is identical to $\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$ except that, in the secret-key $\mathsf{sk}_{\vec{v^*}} = \left(d_0^*, \{{d^*}^j_{1,i}, {d^*}^j_{2,i}\}_{i\in[1,n],j\in[0,k+2]}\right)$ generated by the real-or-random oracle, the distribution of the components $\{{d^*}^j_{1,i}, {d^*}^j_{2,i}\}_{i\in[1,m],j\in[0,k+2]}$ is statistically independent of mode.

Quite evidently, for $m = n$, the entire secret-key $\mathsf{sk}_{\vec{v^*}}$ is independent of mode. In other words, the following holds:

$$\left|\Pr\left[\mathsf{Expt}^{\mathsf{real},n}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand},n}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda) = 1\right]\right| = 0$$

We now state and prove the following claim:

Claim 5.1 *For any probabilistic polynomial-time adversary $\mathcal{A}$, for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and for $m \in [0, n-1]$, the following holds:*

$$\left|\Pr\left[\mathsf{Expt}^{\mathsf{mode},m}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mode},m+1}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

To prove this claim, we assume the contrary. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary such that:

$$\left|\Pr\left[\mathsf{Expt}^{\mathsf{mode},m}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mode},m+1}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda) = 1\right]\right| = \epsilon > \mathsf{negl}(\lambda)$$

for some $m \in [0, n-1]$. Also, let $\mathbb{G}$ be a bilinear group of order $N = q_1q_2q_3$ (each of q_1, q_2 and q_3 being λ-bit primes), and let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ denote the subgroups of $\mathbb{G}$ of order q_1, q_2 and q_3, respectively. Also, let $\hat{e} : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ be an efficiently computable non-degenerate bilinear map, where $\mathbb{G}_T$ is also a group of order N. We construct an algorithm $\mathcal{B}$ that can distinguish between the ensembles:

$$\left(\left(\{g_{1,j}\}_{j\in[1,k+2]}, \{g_{1,j}^{a_j}\}_{j\in[1,k+1]}, g_{1,k+2}^{\sum_{j=1}^{k+1} a_j}\right), \left(\{g_{2,j}\}_{j\in[1,k+2]}, \{g_{2,j}^{a'_j}\}_{j\in[1,k+1]}, g_{2,k+2}^{\sum_{j=1}^{k+1} a'_j}\right)\right)$$
$$\text{and} \quad \left(\left(\{g_{1,j}\}_{j\in[1,k+2]}, \{g_{1,j}^{a_j}\}_{j\in[1,k+2]}\right), \left(\{g_{2,j}\}_{j\in[1,k+2]}, \{g_{2,j}^{a'_j}\}_{j\in[1,k+2]}\right)\right)$$

with probability $1/2 + \epsilon$, where the probability is over random choice of $\{a_j, a'_j \xleftarrow{R} \mathbb{Z}^*_{q_1}\}_{j\in[1,k+2]}$, and over random choice of $\{g_{1,j}, g_{2,j} \xleftarrow{R} \mathbb{G}_1\}_{j\in[1,k+2]}$. Observe that $\mathcal{B}$ can in turn be trivially used to construct another algorithm that has advantage at least ϵ in solving a given instance of the k-DLIN problem in the group $\mathbb{G}_1$.

- ***Setup:*** $\mathcal{B}$ uniformly samples $\{x_{1,j}, x_{2,j} \xleftarrow{R} \mathbb{Z}^*_{q_1}\}_{j\in[1,k+2]}$, $\{h_{1,i}, h_{2,i} \xleftarrow{R} \mathbb{G}_1\}_{i\in[1,n]}$ and $\{R^j_{1,i}, R^j_{2,i} \xleftarrow{R} \mathbb{G}_3\}_{i\in[1,n],j\in[0,k+2]}$. It additionally samples $h \xleftarrow{R} \mathbb{G}_1$, $\gamma \xleftarrow{R} \mathbb{Z}^*_{q_1}$ and $R_3 \xleftarrow{R} \mathbb{G}_3$, and sets:

$$\begin{aligned}
Q &= g_2 \cdot R_3\\
S^0_{1,i} &= h_{1,i} \cdot R^0_{1,i}\ ,\ S^0_{2,i} = h_{2,i} \cdot R^0_{2,i} \text{ for } i \in [1,n]\\
S^j_{1,i} &= h_{1,i}^{x_{1,j}} \cdot R^j_{1,i}\ ,\ S^j_{2,i} = h_{2,i}^{x_{2,j}} \cdot R^j_{2,i} \text{ for } i \in [1,n], j \in [1,k+2]
\end{aligned}$$

Finally, it sets the public parameter pp and the master-secret-key msk as:

$$\mathsf{pp} = \Big(g_1, g_3, Q, \{S_{1,i}^j, S_{2,i}^j\}_{i\in[1,n], j\in[0,k+2]}, \hat{e}\,(g_1, h)^{\gamma},$$
$$\{g_{1,j}, g_{2,j}\}_{j\in[1,k+2]}, \{g_{1,j}^{x_{1,j}} \cdot g_{1,k+2}^{x_{1,k+2}}, g_{2,j}^{x_{2,j}} \cdot g_{2,k+2}^{x_{2,k+2}}\}_{j\in[1,k+1]}\Big)$$
$$\mathsf{msk} = \big(q_1, q_2, q_3, g_2, \{h_{1,i}, h_{2,i}\}_{i\in[1,n]}, h^{\gamma}\big)$$

$\mathcal{B}$ provides pp to $\mathcal{A}$. Observe that pp is distributed exactly as in the real world.

- ***Secret-Key Queries:*** When $\mathcal{A}$ issues a secret-key query for $\overrightarrow{v} \in \mathbb{Z}_N^n$, $\mathcal{B}$ responds with $\mathsf{sk}_{\overrightarrow{v}} = \Pi_k^{\mathrm{IPE}}.\mathsf{KeyGen}\,(\mathsf{pp}, \mathsf{msk}, \overrightarrow{v})$.

- ***Real-or-Random Query:*** Suppose $\mathcal{A}$ queries the real-or-random oracle with an (n,k)-block source $\mathbf{V}^* = (V_1^*, \cdots, V_n^*)$ over $\mathbb{Z}_N^n$ such that $k = \omega\,(\log\lambda)$. $\mathcal{B}$ samples $\mathsf{mode} \xleftarrow{R} \{\mathsf{real}, \mathsf{rand}\}$. For each $i \in [1,n]$, $\mathcal{B}$ samples $v_i^* \xleftarrow{R} V_i^*$ if $\mathsf{mode} = \mathsf{real}$, or $v_i^* \xleftarrow{R} \mathbb{Z}_N$ if $\mathsf{mode} = \mathsf{rand}$. The vector $\overrightarrow{v^*} = (v_1^*, \cdots, v_n^*)$ is the challenge vector that $\mathcal{B}$ uses to respond to the query from $\mathcal{A}$. $\mathcal{B}$ now sets the various components of the secret-key $\mathsf{sk}_{\overrightarrow{v^*}}$ as described next.

  Recall that in both the experiments $\mathsf{Expt}^{\mathsf{mode},m}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$ and $\mathsf{Expt}^{\mathsf{mode},m+1}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$, the secret-key elements corresponding to the first m components of $\overrightarrow{v^*}$ should be statistically independent of mode, while the elements corresponding to the last $(n-m-1)$ components of $\overrightarrow{v^*}$ should be well-formed with respect to mode. We present the details of how this may be achieved by $\mathcal{B}$. $\mathcal{B}$ samples $\{z_{1,i}, z_{2,i} \xleftarrow{R} \mathbb{Z}_{q_1}^*\}_{i\in[1,n]}$, $\{y_{1,i}^j, y_{2,i}^j \xleftarrow{R} \mathbb{Z}_{q_1}^*\}_{i\in[1,n], j\in[1,k+2]}$, $Q_4 \xleftarrow{R} \mathbb{G}_2$, $R_5 \xleftarrow{R} \mathbb{G}_3$ and $f_1, f_2 \xleftarrow{R} \mathbb{Z}_{q_2}^*$. It then sets the following:

$$d_0^* = Q_4 \cdot R_5 \Bigg/ \left(\prod_{i=1}^{n} h_{1,i}^{z_{1,i}} \cdot h_{2,i}^{z_{2,i}}\right)$$
$$d^{*0}_{1,i} = g_1^{z_{1,i}} \cdot g_2^{f_1\cdot v_i} \Bigg/ \left(\prod_{j=1}^{k+2} \big(g_{1,j}^{x_{1,j}}\big)^{y_{1,i}^j}\right) \text{ for } i \in [1,m]$$
$$d^{*0}_{2,i} = g_1^{z_{2,i}} \cdot g_2^{f_2\cdot v_i} \Bigg/ \left(\prod_{j=1}^{k+2} \big(g_{2,j}^{x_{2,j}}\big)^{y_{2,i}^j}\right) \text{ for } i \in [1,m]$$
$$d^{*0}_{1,i} = g_1^{z_{1,i}} \cdot g_2^{f_1\cdot v_i} \Bigg/ \left(\prod_{j=1}^{k+1} \left(g_{1,j}^{x_{1,j}} \cdot g_{1,k+2}^{x_{1,k+2}}\right)^{y_{1,i}^j}\right) \text{ for } i \in [m+2,n]$$
$$d^{*0}_{2,i} = g_1^{z_{2,i}} \cdot g_2^{f_2\cdot v_i} \Bigg/ \left(\prod_{j=1}^{k+1} \left(g_{2,j}^{x_{2,j}} \cdot g_{2,k+2}^{x_{2,k+2}}\right)^{y_{2,i}^j}\right) \text{ for } i \in [m+2,n]$$

$\mathcal{B}$ then sets the remaining secret-key components as:

$$d^{*j}_{1,i} = g_{1,j}^{y^j_{1,i}} \quad , \quad d^{*j}_{2,i} = g_{2,j}^{y^j_{2,i}} \text{ for } i \in [1,n] \setminus \{m+1\}, j \in [1,k+1]$$
$$d^{*k+2}_{1,i} = g_{1,k+2}^{y^{k+2}_{1,i}} \quad , \quad d^{*k+2}_{2,i} = g_{2,k+2}^{y^{k+2}_{2,i}} \text{ for } i \in [1,m]$$
$$d^{*k+2}_{1,i} = g_{1,k+2}^{\sum_{j=1}^{k+1} y^j_{1,i}} \quad , \quad d^{*k+2}_{2,i} = g_{2,k+2}^{\sum_{j=1}^{k+1} y^j_{2,i}} \text{ for } i \in [m+2,n]$$

It is straightforward to see that the secret-key elements corresponding to the last $(n-m-1)$ components of $\overrightarrow{v^*}$ are distributed exactly as in the real world. We now demonstrate that the secret-key elements corresponding to the first m components are distributed independent of mode. Since mode essentially determines the distribution of each v_i^*, it is enough to show that the components $d^{*0}_{1,i}$ and $d^{*0}_{2,i}$ for $i \in [1,m]$ are distributed independent of v_i^*. Let $g_{1,j} = g_1^{z'_{1,j}}$ for $j \in [1,k+2]$, and let $h'_{1,j} = g_{1,j}^{x_{1,j}} \cdot g_{1,k+2}^{x_{1,k+2}}$ for $j \in [1,k+1]$. Also, let $\delta_{1,i} = \prod_{j=1}^{k+2} \left(g_{1,j}^{x_{1,j}}\right)^{y^j_{1,i}}$ in the secret-key component $d^{*0}_{1,i}$ for $i \in [1,m]$. Now, consider the following system of equations, determined by the public parameters pp and the secret-key component $d^{*0}_{1,i}$ for $i \in [1,m]$:

$$\begin{aligned}
\log_{g_1} h'_{1,1} &= x_{1,1} \cdot z'_{1,1} + x_{1,k+2} \cdot z'_{1,k+2} \\
&\vdots \\
\log_{g_1} h'_{1,k+1} &= x_{1,k+1} \cdot z'_{1,k+1} + x_{1,k+2} \cdot z'_{1,k+2} \\
\log_{g_1} \delta_{1,i} &= \sum_{j=1}^{k+2} y^j_{1,i} \cdot x_{1,j} \cdot z'_{1,j}
\end{aligned}$$

Since each $y^{k+2}_{1,i}$ for $i \in [1,m]$ is uniformly random, the event that $y^{k+2}_{1,i} \neq \sum_{j=1}^{k+1} y^j_{1,i}$ occurs except with negligible probability, *which makes the aforementioned system of equations linearly independent*. Hence, the *conditional distribution* of $\delta_{1,i}$ (where the conditioning is on $\mathcal{B}$'s choice of mode and everything else in $\mathcal{A}$'s view) is uniform. In other words, $\delta_{1,i}$ acts as a perfect one-time pad in $d^{*0}_{1,i}$ for $i \in [1,m]$. A similar argument is applicable in the case of $d^{*0}_{2,i}$ when each $y^{k+2}_{2,i}$ for $i \in [1,m]$ is uniformly random.

$\mathcal{B}$ now embeds its input k-DLIN-instance pair in the secret key elements corresponding to the $(m+1)^{\text{th}}$ component of $\overrightarrow{v^*}$. In particular, it sets:

$$d^{*0}_{1,m+1} = g_1^{z_{1,m+1}} \cdot g_1^{f_1 \cdot v_{m+1}} \Big/ \left(\prod_{j=1}^{k+2} \left(g_{1,j}^{a_j}\right)^{x_{1,j}} \right)$$
$$d^{*0}_{2,m+1} = g_1^{z_{2,m+1}} \cdot g_2^{f_2 \cdot v_{m+1}} \Big/ \left(\prod_{j=1}^{k+2} \left(g_{2,j}^{a'_j}\right)^{x_{2,j}} \right)$$
$$d^{*j}_{1,m+1} = g_{1,j}^{a_j} \quad , \quad d^{*j}_{2,m+1} = g_{2,j}^{a'_j} \text{ for } j \in [1,k+2]$$

where $\{g_{1,j}^{a_j}\}_{j\in\{1,k+2\}}$ and $\{g_{2,j}^{a'_j}\}_{j\in\{1,k+2\}}$ are parts of its input instances. $\mathcal{B}$ finally responds to $\mathcal{A}$ with the secret-key $\mathsf{sk}_{\vec{v^*}}$ as:

$$\mathsf{sk}_{\vec{v^*}} = \left(d_0^*, \{d^{*j}_{1,i}, d^{*j}_{2,i}\}_{i\in[1,n],j\in[0,k+2]}\right)$$

- ***Output:*** Finally, $\mathcal{B}$ outputs the same bit b as output by $\mathcal{A}$.

It is again easy to see that when $a_{k+2} = \sum_{j=1}^{k+1} a_j$ and $a'_{k+2} = \sum_{j=1}^{k+1} a'_j$, the components of the secret-key $\mathsf{sk}_{\vec{v^*}}$ corresponding to v^*_{m+1} are well-formed. On the other hand, when both a_{k+2} and a'_{k+2} are uniformly random in $\mathbb{Z}^*_{q_1}$, the components of the secret-key $\mathsf{sk}_{\vec{v^*}}$ corresponding to v^*_{m+1} are statistically independent of mode. To see this, again let $g_{1,j} = g_1^{z'_{1,j}}$ for $j \in [1, k+2]$, and let $h'_{1,j} = g_{1,j}^{x_{1,j}} \cdot g_{1,k+2}^{x_{1,k+2}}$ for $j \in [1, k+1]$. Also, let $\delta_{1,m+1} = \prod_{j=1}^{k+2} \left(g_{1,j}^{a_j}\right)^{x_{1,j}}$ in the secret-key component $d^{*0}_{1,m+1}$. If a_{k+2} is uniformly random in $\mathbb{Z}^*_{q_1}$, we may assume that $a_{k+2} \neq \sum_{j=1}^{k+1} a_j$, since this occurs except with negligible probability. Then, the following system of equations, determined by the public parameters pp and the secret-key component $d^{*0}_{1,m+1}$, is linearly independent:

$$\begin{aligned}
\log_{g_1} h'_{1,1} &= x_{1,1} \cdot z'_{1,1} + x_{1,k+2} \cdot z'_{1,k+2} \\
&\vdots \\
\log_{g_1} h'_{1,k+1} &= x_{1,k+1} \cdot z'_{1,k+1} + x_{1,k+2} \cdot z'_{1,k+2} \\
\log_{g_1} \delta_{1,m+1} &= \sum_{j=1}^{k+2} a_j \cdot x_{1,j} \cdot z'_{1,j}
\end{aligned}$$

Hence, the *conditional distribution* of $\delta_{1,m+1}$ (where the conditioning is on $\mathcal{B}$'s choice of mode and everything else in $\mathcal{A}$'s view) is uniform. In other words, $\delta_{1,m+1}$ acts as a perfect one-time pad in $d^{*0}_{1,m+1}$. A similar argument is applicable in the case of $d^{*0}_{2,m+1}$ when a'_{k+2} is uniformly random.

The final distribution of the secret-key $\mathsf{sk}_{\vec{v^*}} = \left(d_0^*, \{d^{*j}_{1,i}, d^{*j}_{2,i}\}_{i\in[1,n],j\in[0,k+2]}\right)$, generated by $\mathcal{B}$ as the response to the real-or-random oracle query from $\mathcal{A}$, may be summarized as follows:

- When $a_{k+2} = \sum_{j=1}^{k+1} a_j$ and $a'_{k+2} = \sum_{j=1}^{k+1} a'_j$, the distribution of the components $\{d^{*j}_{1,i}, d^{*j}_{2,i}\}_{i\in[1,m],j\in[0,k+2]}$ is statistically independent of mode, while the remaining components are well-formed with respect to mode. In other words, $\mathsf{sk}_{\vec{v^*}}$ is identically distributed to the response of the real-or-random oracle in the experiment $\mathsf{Expt}^{\mathsf{mode},m}_{\mathsf{FP},\Pi_k^{\mathrm{IPE}},\mathcal{A}}(\lambda)$.

- When both a_{k+2} and a'_{k+2} are uniformly random in $\mathbb{Z}^*_{q_1}$, the distribution of the components $\{d^{*j}_{1,i}, d^{*j}_{2,i}\}_{i\in[1,m+1],j\in[0,k+2]}$ is statistically independent of mode, while the remaining components are well-formed with respect to mode. In other words, $\mathsf{sk}_{\vec{v}^*}$ is identically distributed to the response of the real-or-random oracle in the experiment $\mathsf{Expt}^{\mathsf{mode},m+1}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)$.

It follows readily that $\mathcal{B}$ has the same advantage ϵ as $\mathcal{A}$ in solving its input k-DLIN instance pair. This completes the proof of Claim 5.1.

Consequently, for $\mathsf{mode} \xleftarrow{R} \{\mathsf{real}, \mathsf{rand}\}$, we have:

$$
\begin{aligned}
\mathbf{Adv}^{\mathsf{FP}}_{\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda) &= \left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] \right| \\
&\leq \left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{real},n}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] \right| \\
&\quad + \left| \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand},n}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] \right| \\
&\quad\quad + \left| \Pr\left[\mathsf{Expt}^{\mathsf{real},n}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand},n}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] \right| \\
&\leq 2\left| \Pr\left[\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mode},n}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] \right| \\
&\leq 2\sum_{m=0}^{n-1} \left| \Pr\left[\mathsf{Expt}^{\mathsf{mode},m}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mode},m+1}_{\mathsf{FP},\Pi^{\mathrm{IPE}}_k,\mathcal{A}}(\lambda)=1\right] \right| \\
&\leq \mathsf{negl}(\lambda)
\end{aligned}
$$

This completes the proof of Theorem 5.2.

6 Extensions and Open Problems

In this paper, we presented a new class of public-key predicate encryption schemes that are provably function private in the standard model, under well-known cryptographic assumptions. A large class of existing function private constructions in the public-key setting impose highly stringent requirements on the min-entropy of predicate distributions, thereby limiting their applicability in the context of real-world predicates. Our constructions, on the other hand, are function private for predicate distributions that satisfy more realistic min-entropy requirements. Our main result is a generic framework, denoted as *embed-augment-recover*, that takes an existing predicate encryption scheme and transforms it into a computationally function private one while retaining its original data privacy guarantees. Our approach leads to public-key constructions for identity-based encryption (IBE) and inner-product encryption (IPE) that are computationally function private in the standard model under a family of weaker variants of the DLIN assumption. In this section, we present some interesting open problems that arise from our work.

Hidden Vector Encryption and Polynomial Evaluation. Boneh and Waters [1] proposed hidden vector encryption (HVE), a pre-cursor to IPE, that supports search using conjunctive, range and comparison-based query predicates. In HVE, attributes correspond to vectors over an alphabet Σ, while secret-keys correspond to predicate vectors over the augmented alphabet $\Sigma_\star = \Sigma \cup \{\star\}$ containing the wild card character $\star$. Decryption succeeds if the attribute matches the predicate vector in every coordinate that is not $\star$. We note that although IPE can be used to realize HVE [3], our computational function privacy definitions do not naturally extend to HVE. In particular, the presence of the wild card character $\star$ in the predicate vectors of HVE trivially violates our min-entropy requirements, making it difficult to *hide* their presence in the secret-key. It is thus open to formalize function privacy definitions for HVE, and to achieve constructions satisfying such definitions. It is also open to formalize security definitions and realize constructions for function private encryption schemes that support arbitrary polynomial evaluation predicate.

Generalization of Our Approach. In this work, we have applied our *embed-augment-recover* approach to transform certain existing public-key predicate encryption schemes that are not function private, into computationally function private ones. An interesting open problem is to explore whether our approach can be generalized for *any* public-key predicate encryption scheme, or if there are any specific properties of existing predicate encryption schemes that make them amenable to transformation using our approach. A starting point in this direction could be to explore the applicability of our approach to public-key predicate encryption schemes based on lattices, such as the IPE scheme in [2]. It would also be interesting to explore if our approach can be used to design public-key encryption schemes supporting a set of predicates beyond inner-products. Iovino et al. [13] have demonstrated that computational function privacy from standard assumptions seems unattainable for a very generalized class of predicates, such as the class of all NC^1) circuits. This motivates exploring the limits of our techniques in terms of the range of predicates for which they are applicable.

References

1. Dan Boneh and Brent Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, pages 535–554, 2007.
2. Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional Encryption for Inner Product Predicates from Learning with Errors. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 21–40, 2011.
3. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. *J. Cryptology*, 26(2):191–224, 2013.
4. Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 445–464, 2006.

5. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology–EUROCRYPT 2005*, pages 440–456. Springer, 2005.
6. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. *J. Cryptology*, 21(3):350–391, 2008.
7. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 733–751, 2015.
8. Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, pages 333–362, 2016.
9. Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public Key Encryption with Keyword Search. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 506–522, 2004.
10. Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-Private Identity-Based Encryption: Hiding the Function in Functional Encryption. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 461–478, 2013.
11. Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-Private Subspace-Membership Encryption and Its Applications. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, pages 255–275, 2013.
12. Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. On the practical security of inner product functional encryption. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 777–798, 2015.
13. Vincenzo Iovino, Qiang Tang, and Karol Zebrowski. On the power of public-key function-private functional encryption. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings*, pages 585–593, 2016.
14. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 41–55, 2004.
15. Vipul Goyal, Aayush Jain, and Adam O'Neill. Multi-input functional encryption with unbounded-message security. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, pages 531–556, 2016.
16. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, pages 13–25, 1998.