Function-Private Identity-Based Encryption: Hiding the Function in Functional Encryption

Dan Boneh[*] Ananth Raghunathan[†] Gil Segev[‡]

Abstract

We put forward a new notion, *function privacy*, in identity-based encryption and, more generally, in functional encryption. Intuitively, our notion asks that decryption keys reveal essentially no information on their corresponding identities, beyond the absolute minimum necessary. This is motivated by the need for providing *predicate privacy* in public-key searchable encryption. Formalizing such a notion, however, is not straightforward as given a decryption key it is always possible to learn some information on its corresponding identity by testing whether it correctly decrypts ciphertexts that are encrypted for specific identities.

In light of such an inherent difficulty, any meaningful notion of function privacy must be based on the *minimal* assumption that, from the adversary's point of view, identities that correspond to its given decryption keys are sampled from somewhat unpredictable distributions. We show that this assumption is in fact *sufficient* for obtaining a strong and realistic notion of function privacy. Loosely speaking, our framework requires that a decryption key corresponding to an identity sampled from any sufficiently unpredictable distribution is indistinguishable from a decryption key corresponding to an independently and uniformly sampled identity.

Within our framework we develop an approach for designing function-private identity-based encryption schemes, leading to constructions that are based on standard assumptions in bilinear groups (DBDH, DLIN) and lattices (LWE). In addition to function privacy, our schemes are also anonymous, and thus yield the first public-key searchable encryption schemes that are provably *keyword private*: A search key sk_w enables to identify encryptions of an underlying keyword w, while not revealing any additional information about w beyond the minimum necessary, as long as the keyword w is sufficiently unpredictable.

Keywords: Function privacy, identity-based encryption, functional encryption.

[*] Stanford University, Stanford, CA 94305, USA. Email: `dabo@cs.stanford.edu`.

[†] Stanford University, Stanford, CA 94305, USA. Email: `ananthr@stanford.edu`.

[‡] Stanford University, Stanford, CA 94305, USA. Email: `segev@stanford.edu`.

Contents

1 Introduction

Public-key searchable encryption is needed when a proxy is asked to route encrypted messages based on their content. For example, consider a payment gateway that needs to route transactions based on the transaction type. Transactions for benign items are routed for quick processing while transactions for sensitive items are routed for special processing. Similarly, consider an email gateway that routes emails based on the contents of the subject line. Urgent emails are routed to the user's mobile device, while less urgent mails are routed to the user's desktop. When the data is encrypted a simple design is to give such gateways full power to decrypt all ciphertexts, but this clearly exposes more information than necessary.

A better solution, called public-key searchable encryption (introduced by Boneh, Di Crescenzo, Ostrovsky and Persiano [BCO$^+$04]), is to give the gateway a trapdoor that enables it to learn the information it needs and nothing else. In recent years many elegant public-key searchable encryption systems have been developed [BCO$^+$04, GSW04, ABC$^+$08, BW07, SBC$^+$07, KSW08, BSNS08, CKR$^+$09, ABN10, AFV11] supporting a wide variety of search predicates.

Private searching. Beyond the standard notions of data privacy, it is often also necessary to guarantee *predicate privacy*, i.e., to keep the specific search predicate hidden from the gateway. For example, in the payment scenario it may be desirable to keep the list of sensitive items secret, and in the email scenario users may not want to reveal the exact criteria they use to classify an email as urgent. Consequently, we want the trapdoor given to the gateway to reveal as little as possible about the search predicate.

While this question has been considered before [SWP00, OS07, BSW09, SSW09], it is often noted that such a notion of privacy cannot be achieved in the public-key setting. For example, to test if an email from "spouse" is considered urgent the gateway could simply use the public key to create an email from the spouse and test if the trapdoor classifies it as urgent. More generally, the gateway can encrypt messages of its choice and apply the trapdoor to the resulting ciphertexts, thereby learning how the search functionality behaves on these messages. Hence, leaking some information about the search predicate is unavoidable.

As a concrete example, consider the case of keyword searching [BCO$^+$04]: A search key sk_w corresponds to a particular keyword w, and the search matches a ciphertext $\mathsf{Enc}(pk, m)$ if and only if $m = w$. In this case, it may be possible to formalize and realize a notion of "private keyword search" asking that a search key reveals no more information than what can be learned by invoking the search algorithm.

Function-private IBE: A new notion of security. Motivated by the challenge of hiding the search predicates in public-key searchable encryption, in this paper we introduce a new notion of security, *function privacy*, for identity-based encryption.[1] The standard notion of security for anonymous IBE schemes (e.g., [BF03, BW06, Gen06, GPV08, ABB10, BKP$^+$12]), asks that a ciphertext $c = \mathsf{Enc}(\mathrm{pp}, \mathrm{id}, m)$ reveals essentially no information on the pair (id, m) as long as a secret key $\mathrm{sk}_{\mathrm{id}}$ corresponding to the identity id is not explicitly provided (but secret keys corresponding to other identities may be provided). Our notion of function privacy takes a step forward by asking that it

[1] As observed by Abdalla et al. [ABC$^+$08], any anonymous IBE scheme can be used as a public-key searchable encryption scheme by defining the search key sk_w for a keyword w as the IBE secret key for the identity $\mathrm{id} = w$. A keyword w' is encoded as $c = \mathsf{Enc}(\mathrm{pp}, w', 0)$ and one tests if c matches the keyword w by invoking the IBE decryption algorithm on c with the secret key sk_w. The IBE anonymity property ensures that c reveals nothing else about the payload w'. For this reason we focus on *anonymous* IBE schemes, although we note that our notion of function privacy does not require anonymity.

should not be possible to learn any information, beyond the absolute minimum necessary, on the identity id corresponding to a given secret key $\mathrm{sk}_{\mathrm{id}}$.

Formalizing a realistic notion of function privacy, however, is not straightforward due to the actual functionality of identity-based encryption. Specifically, assuming that an adversary who is given a secret key $\mathrm{sk}_{\mathrm{id}}$ has some a-priori information that the corresponding identity id belongs to a small set S of identities (e.g., $S = \{\mathrm{id}_0, \mathrm{id}_1\}$), then the adversary can fully recover id: The adversary simply needs to encrypt a (possibly random) message m for each $\mathrm{id}' \in S$, and then run the decryption algorithm on the given secret key $\mathrm{sk}_{\mathrm{id}}$ and each of the resulting ciphertexts $c' = \mathsf{Enc}(\mathrm{pp}, \mathrm{id}', m)$ to identify the one that decrypts correctly. In fact, as long as the adversary has some a-priori information according to which the identity id is sampled from a distribution whose min-entropy is at most logarithmic in the security parameter, there is a non-negligible probability for a full recovery.

Our contributions. In light of the above inherent difficulty, any notion of function privacy for IBE schemes would have to be based on the *minimal* assumption that, from the adversary's point of view, identities that correspond to its given secret keys are sampled from distributions with a certain amount of min-entropy (which has to be at least super-logarithmic in the security parameter). Our work shows that this necessary assumption is in fact *sufficient* for obtaining a strong and meaningful indistinguishability-based notion of function privacy.

Our work formalizes this new notion of security (we note that we call it *function privacy* to emphasize the fact that $\mathrm{sk}_{\mathrm{id}}$ hides the functionality that it provides). Loosely speaking, our basic notion of function privacy requires that a secret key $\mathrm{sk}_{\mathrm{id}}$, where id is sampled from any sufficiently unpredictable (adversarially-chosen) distribution,[2] is indistinguishable from a secret key corresponding to an independently and uniformly sampled identity. In addition, we also consider a stronger notion of function privacy, to which we refer as *enhanced* function privacy. This enhanced notion addresses the fact that in various applications (such as searching on encrypted data), an adversary may obtain not only a secret key $\mathrm{sk}_{\mathrm{id}}$, but also an encryption $\mathsf{Enc}(\mathrm{pp}, \mathrm{id}, m)$ of some message m. Our notion of enhanced function privacy asks that even in such a scenario, it should not be possible to learn any unnecessary information on the identity id.

We refer the reader to Section 3 for the formal definitions, and for descriptions of simple attacks exemplifying that the anonymous IBE schemes presented in [BF03, GPV08, ABB10, KP11] do not even satisfy our basic notion of function privacy.[3]

Within our framework we develop an approach for designing identity-based encryption schemes that satisfy our notions of function private. Our approach leads to constructions that are based on standard assumptions in bilinear groups (DBDH, DLIN) and lattices (LWE). In particular, our schemes yield keyword searchable public-key encryption schemes that *do not reveal the keywords*: A search key sk_w reveals nothing about its corresponding keyword w beyond the minimum necessary, as long as the keyword w is chosen from a sufficiently unpredictable distribution.

[2] We emphasize that the distribution is allowed to depend on the public parameters of the scheme. This is in contrast to the setting of deterministic public-key encryption (DPKE) [BBO07], where similar inherent difficulties arise when formalizing notions of security. Nevertheless, our notion is inspired by that of [BBO07], and we refer the reader to Section 3 for an elaborate discussion (in particular, we discuss a somewhat natural DPKE-based approach for designing function-private IBE schemes which fails to satisfy our notion of security and only satisfies a weaker, less realistic, one).

[3] We note that other anonymous IBE schemes, such as [Gen06, BW06, BKP$^+$12] for which we were not able to find such simple attacks, can always be *assumed* to be function private based on somewhat non-standard entropy-based assumptions (such assumptions would essentially state that the schemes satisfy our definition). In this paper we are interested in schemes whose function privacy can be based on standard assumptions (e.g., DBDH, DLIN, LWE).

The bigger picture: Functional encryption and obfuscation. Our notion of function privacy for IBE naturally generalizes to functional encryption systems [BSW11, O'N10, BO12, GVW12, AGV$^+$13, GKP$^+$13], where we obtain an additional security requirement on such systems. Here, a functional secret key sk_f corresponding to a function f enables to compute $f(m)$ given an encryption $c = \mathsf{Enc}_{pk}(m)$. Functional encryption systems, however, need not be predicate private and sk_f may leak unnecessary information about f. Intuitively, we say that a functional encryption system is *function private* if such a functional secret key sk_f does not reveal information about f beyond what is already known and what can be obtained by running the decryption algorithm on test ciphertexts. This can be formalized within a suitable framework for program obfuscation (e.g., [Can97, BGI$^+$12, LPS04, GK05, Wee05, CKV$^+$10] and the references therein) by asking, for example, that any adversary that receives a functional secret key sk_f learns no more information than a simulator that has oracle access to the function f.

In this setting, our identity-based encryption schemes provide function privacy for the class of functions defined as

$$f_{\mathrm{id}^*}(\mathrm{id}, m) = \begin{cases} m \text{ if } \mathrm{id} = \mathrm{id}^* \\ \perp \text{ otherwise} \end{cases}$$

where id^* is sampled from an unpredictable distribution. A fascinating direction for future work is to extend our results to more general classes of functions.

Non-adaptive function privacy and deterministic encryption. The inherent difficulty discussed above in formalizing function privacy is somewhat similar to the one that arises in the context of deterministic public-key encryption (DPKE), introduced by Bellare, Boldyreva, and O'Neill [BBO07] (see also [BFO$^+$08a, BFO08b, BBN$^+$09, BS11, FOR12, MPR$^+$12, Wee12, RSV13]). In that setting one would like to capture as-strong-as-possible notions of security that can be satisfied by public-key encryption schemes whose encryption algorithms are deterministic. Similarly to our setting, if an adversary has some a-priori information that a ciphertext $c = \mathsf{Enc}_{pk}(m)$ corresponds to a plaintext m that is sampled from a low-entropy source (e.g., $m \in \{m_0, m_1\}$), then the plaintext can be fully recovered: The adversary simply needs to encrypt all "likely" plaintexts and to compare each of the resulting ciphertexts to c. Therefore, any notion of security for DPKE has to be based on the assumption that plaintexts are sampled from distributions with a certain amount of min-entropy (which has to be at least super-logarithmic in the security parameter).

However, unlike in our setting, in the setting of DPKE it is also necessary to limit the dependency of plaintexts on the public-key of the scheme.[4] In our setting, as the key-generation algorithm is allowed to be randomized, such limitations are not inherent: we allow adversaries to specify identity distributions in an adaptive manner after seeing the public parameters of the scheme.

This crucial difference between our setting and the setting of DPKE rules out, in particular, the following natural approach for designing anonymous IBE schemes providing function privacy: encapsulate all identities with a DPKE scheme, and then use any existing anonymous IBE scheme treating the ciphertexts of the DPKE scheme as its identities. That is, for encrypting to identity id, first encrypt id using a DPKE scheme and then treat the resulting ciphertext as an identity for an anonymous IBE system. This approach clearly preserves the standard security of the underlying IBE scheme. Moreover, as secret keys are now generated as sk_c, where $c = \mathsf{Enc}_{pk}(\mathrm{id})$ is a deterministic encryption of id, instead of as $\mathrm{sk}_{\mathrm{id}}$, one could hope that $\mathrm{sk}_{\mathrm{id}}$ does not reveal any unnecessary information on id as long as id is sufficiently unpredictable.

[4] Intuitively, the reason is that plaintexts distributions that can depend on the public key can use any deterministic encryption algorithm as a subliminal channel for leaking information on the plaintexts (consider, for example, sampling a uniform plaintext m for which the most significant bit of $c = \mathsf{Enc}_{pk}(m)$ agrees with that of m). We refer the reader to [BBO07, RSV13] for an in-depth discussion.

This approach, however, fails to satisfy our notion of function privacy and only satisfies a weaker,"non-adaptive", one.[5] Specifically, the notion of function privacy that is satisfied by such a two-tier construction is that secret keys do not reveal any unnecessary information on their corresponding identities as long as the identities are essentially independent of the public parameters of the scheme. We formalize this non-adaptive notion in Section 3, and present a generic transformation satisfying it in Section 6 based on any IBE scheme. In fact, observing that the DPKE-based construction described above never actually uses the decryption algorithm of the DPKE scheme, in our generic transformation we show that above idea can be realized without using a DPKE scheme. Instead, we only need to assume the existence of collision-resistant hash functions (and also use any pairwise independent family of permutations). We refer the reader to Section 6 for more details.

1.1 Our Approach: "Extract-Augment-Combine"

Our approach consists of three main steps: "extract", "augment", and "combine". We begin with a description of the main ideas underlying each step, and then provide an example using a concrete IBE scheme.

Given any anonymous IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, we use the exact same setup algorithm Setup, and our first step is to modify its key-generation algorithm KeyGen as follows: Instead of generating a secret key for an identity id, first apply a strong randomness extractor Ext to id using a randomly chosen seed s, then generate a secret key $\mathrm{sk}_{\mathrm{id}_s}$ for the identity $\mathrm{id}_s \overset{\mathsf{def}}{=} \mathsf{Ext}(\mathrm{id}, s)$, and output the pair $(s, \mathrm{sk}_{\mathrm{id}_s})$ as a secret for id in the new scheme. This steps clearly guarantees function privacy: As long as the identity id is sampled from a sufficiently unpredictable distribution,[6] the distribution (s, id_s) is statistically close to uniform, and therefore the pair $(s, \mathrm{sk}_{\mathrm{id}_s})$ reveals no information on the identity id.

This extraction step, however, may hurt the data privacy of the underlying scheme. For example, since randomness extractors are highly non-injective by definition, an adversary that is given a secret key $(s, \mathrm{sk}_{\mathrm{id}_s})$ may be able to find an identity id' such that $\mathsf{Ext}(\mathrm{id}, s) = \mathsf{Ext}(\mathrm{id}', s)$. In this case, the same secret key is valid for both id and id', contradicting the data privacy of the resulting scheme. Therefore, for overcoming this problem we make sure that the extractor is *at least* collision resistant: although many collisions exist, a computationally-bounded adversary will not be able to find one. This is somewhat natural to achieve in the random-oracle model [BR93], but significantly more challenging in the standard model.

An even more challenging problem is that the extraction step hurts the decryption of the underlying scheme. Specifically, when encrypting a message m for an identity id, the encryption algorithm does not know which seed s will be chosen (or was already chosen) when generating a secret key for id. In other words, the correctness of the decryption algorithm Dec should hold for any choice of seed s by the key-generation algorithm KeyGen, although s is not known to the encryption algorithm Enc. One possibility, is to modify the encryption algorithm such that it outputs an encryption of m for id_s for all possible seeds s. This clearly fails, as the number of seeds is inherently super-polynomial in the security parameter. We overcome this problem by augmenting ciphertexts of the underlying scheme with various additional pieces of information. These will enable the new decryption algorithm to combine the pieces in a particular way for generating an encryption of m for the identity id_s for any given s, and then simply apply the underlying decryption algorithm using the specific seed s chosen by the key-generation algorithm.[7]

[5] As discussed above, any DPKE becomes insecure once plaintext distributions (which here correspond to identity distributions) are allowed to depend on the public key of the scheme.

[6] Note that the new scheme assumes a slightly larger identity space compared to the underlying scheme.

[7] In fact, in some of our schemes the decryption algorithm combines the pieces to generate an encryption of a related

Our approach introduces the following two main challenges that we overcome in each of our constructions:

- Augmenting the ciphertexts of the underlying scheme with additional pieces of information may hurt the data privacy of the underlying scheme.
- Combining the additional pieces of information for generating an encryption for id_s for any given s requires using an extractor Ext that exhibits a particular interplay with the underlying encryption and decryption algorithms.

Our constructions in this paper are obtained by applying our "extract-augment-combine" approach to various known anonymous IBE schemes [BF03, GPV08, ABB10, KP11]. To do so, we overcome the two main challenges mentioned above in ways that are "tailored" specifically to each scheme. Using our approach we provide the following constructions (see also Table 1):

- In the random-oracle model [BR93] we give fully-secure constructions from pairings and lattices by building upon the systems of Boneh and Franklin [BF03] (based on the DBDH assumption) and of Gentry, Peikert and Vaikuntanathan [GPV08] (based on the LWE assumption).
- In the standard model we give selectively-secure constructions from pairings and lattices based on the constructions of Agrawal, Boneh and Boyen [ABB10] (based on the LWE assumption) and of Kurosawa and Phong [KP11] (based on the DLIN assumption), which we then generalize to a fully-secure construction (based on the DLIN assumption[8]).

In all instances our constructions are based on the same complexity assumptions as the underlying systems.

Scheme	**Model**	**Data Privacy**	**Function Privacy**
DBDH (Section 4.1)	Random Oracle	Full	Statistical
LWE1 (Section 4.2)	Random Oracle	Full	Statistical
DLIN1 (Section 5.1)	Standard	Selective	Statistical + Non-adaptive enhanced
LWE2 (Section 5.2)	Standard	Selective	Statistical
DLIN2 (Section 5.3)	Standard	Full	Statistical + Enhanced
CRH (Section 6)	Standard	Full	Non-adaptive statistical enhanced

Table 1: Our IBE schemes.

A concrete example. We conclude this section by exemplifying our approach using our DBDH-based construction in the random-oracle model (we refer the reader to Section 4.1 for a more formal description of the scheme and its proofs of data privacy and function privacy). The scheme is obtained by applying our approach to the anonymous IBE scheme of Boneh and Franklin [BF03].

- The setup algorithm in the scheme of Boneh and Franklin samples $\alpha \leftarrow \mathbb{Z}_p^*$, and lets $h = g^\alpha$, where g is a generator of a group $\mathbb{G}$ of prime order p. The public parameters are g and h, and the master secret key is α. Our scheme has exactly the same setup algorithm.

message m' from which m can be easily recovered (e.g., $m' = 2m$).

[8] We note that a similar generalization can also be applied to our selectively-secure LWE-based scheme in the standard model.

- The key-generation algorithm in the scheme of Boneh and Franklin computes a secret key for an identity id as $\mathrm{sk}_{\mathrm{id}} = H(\mathrm{id})^{\alpha}$, where H is a random oracle mapping identities into the group $\mathbb{G}$. As discussed above our first step is to extract from id. First, we use a random oracle mapping identities into $\mathbb{G}^{\ell}$ for some $\ell > 1$. Then, for $H(\mathrm{id}) = (h_1, \dots, h_\ell) \in \mathbb{G}^{\ell}$, we sample an extractor seed $s = (s_1, \dots, s_\ell) \leftarrow \mathbb{Z}_p^{\ell}$, and output the secret key $(s, (\mathsf{Ext}(H(\mathrm{id}), s)^{\alpha})$ where we use the specific extractor $\mathsf{Ext}((h_1, \dots, h_\ell), (s_1, \dots, s_\ell)) = \prod_{j=1}^{\ell} h_j^{s_j}$. Note that Ext is, in particular, collision resistant based on the discrete logarithm assumption in the group $\mathbb{G}$.

- An encryption of a message m for an identity id in the scheme of Boneh and Franklin is a pair (c_0, c_1), defined as $c_0 = g^r$ and $c_1 = \hat{e}(h, H(\mathrm{id}))^r \cdot m$. In our scheme, an encryption of a message m for an identity id consists of $\ell + 1$ components $(c_0, \dots, c_\ell)$ defined as $c_0 = g^r$, and $c_i = \hat{e}(h, h_i)^r \cdot m$ for every $i \in [\ell]$, where $H(\mathrm{id}) = (h_1, \dots, h_\ell)$. This is exactly using the encryption algorithm of Boneh and Franklin for separately encrypting m for each of the h_i's while re-using the same randomness r. The main technical challenge that is left is showing that such augmented ciphertexts still provide data privacy (the reader is referred to Section 4.1.1 for the proof of data privacy).

- Our decryption algorithm on input a ciphertext $c = (c_0, \dots, c_\ell)$, and a secret key $\mathrm{sk}_{\mathrm{id}} = (s_1, \dots, s_\ell, z)$, combines $c_1, \dots, c_\ell$ by computing

$$\prod_{i=1}^{\ell} c_i^{s_i} = \hat{e}(h, \prod_{i=1}^{\ell} h_i^{s_i})^r \cdot m^{s_1 + \cdots + s_\ell} = \hat{e}(h, \mathrm{id}_s)^r \cdot m^{s_1 + \cdots + s_\ell},$$

where $\mathrm{id}_s = \mathsf{Ext}(H(\mathrm{id}), s)$, as before. Note that the pair $(c_0, \prod_{i=1}^{\ell} c_i^{s_i})$ is exactly an encryption of the message $m' = m^{s_1 + \cdots + s_\ell}$ for the identity id_s in the scheme of Boneh and Franklin. This allows to invoke the decryption algorithm of Boneh and Franklin for recovering m', and then to easily recover m (as the s_i's are given in the clear).

1.2 Related Work

Searchable encryption has been studied in both the symmetric settings [SWP00, CGK+11, SSW09] and public-key settings [BCO+04, GSW04, ABC+08, BW07, SBC+07, KSW08, BSNS08, CKR+09, AFV11]. Public-key searching on encrypted data now supports equality testing, disjunctions and conjunctions, range queries, CNF/DNF formulas, and polynomial evaluation. These schemes, however, are not function private in that their secret searching keys reveal information about their corresponding predicates. Indeed, until this work, predicate privacy seemed impossible in the public-key settings.

The impossibility argument does not apply in the symmetric key settings where the encryptor and decryptor have a shared secret key. In this setting the entity searching over ciphertexts does not have the secret key and cannot (passively) test the searching key on ciphertexts of its choice. Indeed, in the symmetric-key setting predicate privacy is possible and a general solution to private searching on encrypted data was provided by Goldreich and Ostrovsky [GO96] in their construction of an oblivious RAM. More efficient constructions are known for equality testing [SWP00, CM05, CGK+11, CK10, vLSD+10, KPR12] and inner product testing [SSW09]. The latter enables CNF/DNF formulas, polynomial evaluation, and exact thresholds.

A closely related problem called *private stream searching* asks for the complementary privacy requirements: the data is available in the clear, but the search predicate must remain hidden. Constructions in these settings support efficient equality testing [OS07, BSW09] and can be viewed as a more expressive variant of private information retrieval.

1.3 Paper Organization

In Section 2 we introduce several standard definitions, computational assumptions, and tools. In Section 3 we formally define our notion of function privacy for identity-based encryption. In Section 4 we present a fully-secure DBDH-based scheme and a fully-secure LWE-based scheme in the random-oracle model. In Section 5 we present a selectively-secure DLIN-based scheme, a selectively-secure LWE-based scheme, and a fully-secure DLIN-based scheme in the standard model. In Section 6 we present a generic transformation that guarantees non-adaptive enhanced function privacy. Finally, in Section 7 we discuss several extensions and open problems.

2 Preliminaries

Notation. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \dots, n\}$, and by U_n the uniform distribution over the set $\{0,1\}^n$. For a random variable X we denote by $x \leftarrow X$ the process of sampling a value x according to the distribution of X. Similarly, for a finite set S we denote by $x \leftarrow S$ the process of sampling a value x according to the uniform distribution over S. We denote by $\mathbf{x}$ (and sometimes $\boldsymbol{x}$) a vector $(x_1, \dots, x_{|\mathbf{x}|})$. We denote by $\mathbf{X} = (X_1, \dots, X_T)$ a joint distribution of T random variables, and by $\mathbf{x} = (x_1, \dots, x_T)$ a sample drawn from $\mathbf{X}$. For two bit-strings x and y we denote by $x \| y$ their concatenation. A non-negative function $f : \mathbb{N} \to \mathbb{R}$ is negligible if it vanishes faster than any inverse polynomial. For a real number $x \in \mathbb{R}$ we define $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$ (i.e., the nearest integer to x). For a group $\mathbb{G}$ of order p with generator g and any $\mathbf{X} \in \mathbb{Z}_p^{n \times m}$, we denote the matrix whose (i,j)-th entry is $(g^{x_{i,j}})$ by $g^{\mathbf{X}}$.

2.1 Min-Entropy, Universal Hashing, and Randomness Extraction

The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. A *k-source* is a random variable X with $\mathbf{H}_\infty(X) \geq k$. A $(k_1, \dots, k_T)$*-source* is a random variable $\mathbf{X} = (X_1, \dots, X_T)$ where each X_i is a k_i-source. A (T,k)*-block-source* is a random variable $\mathbf{X} = (X_1, \dots, X_T)$ where for every $i \in [T]$ and $x_1, \dots, x_{i-1}$ it holds that $X_i|_{X_1 = x_1, \dots, X_{i-1} = x_{i-1}}$ is a k-source. The *statistical distance* between two random variables X and Y over a finite domain Ω is $\mathbf{SD}(X,Y) = \frac{1}{2}\sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$.

The following standard lemma states that conditioning on a random variable that obtains at most 2^v values can reduce the min-entropy of any other random variable by essentially at most v.

Lemma 2.1 ([Vad12, Lemma 6.30])**.** *Let (Z, X) be any two jointly distributed random variables such that $|\mathrm{Supp}(Z)| \leq 2^v$. Then, for any $\epsilon > 0$ it holds that*

$$\Pr_{z \leftarrow Z}[\mathbf{H}_\infty(X | Z = z) \geq \mathbf{H}_\infty(X) - v - \log(1/\epsilon)] \geq 1 - \epsilon.$$

Definition 2.2. A collection $\mathcal{H}$ of functions $H : U \to V$ is *universal* if for any $x_1, x_2 \in U$ such that $x_1 \neq x_2$ it holds that

$$\Pr_{H \leftarrow \mathcal{H}}[H(x_1) = H(x_2)] = \frac{1}{|V|}.$$

Lemma 2.3. *Let $\mathcal{H}$ be a universal collection of functions $H : U \to V$, and let $\mathbf{X} = (X_1, \dots, X_T)$ be (T,k)-block-source where $k \geq \log|V| + 2\log(1/\epsilon) + \Theta(1)$. Then, the distribution $(H_1, H_1(X_1), \dots, H_T, H_T(X_T))$, where $(H_1, \dots, H_T) \leftarrow \mathcal{H}^T$, is ϵT-close to the uniform distribution over $(\mathcal{H} \times V)^T$.*

Proof. We prove the lemma via an inductive claim showing that for every $i \in [T]$ the distributions $\mathcal{D}_i = (X_1, \dots, X_{i-1}, H_i, H_i(X_i), \dots, H_T, H_T(X_T))$ and $\mathcal{D}'_i = (X_1, \dots, X_{i-1}, H_i, U_i, \dots, H_T, U_T)$ are

$\epsilon(T-i+1)$-close, where $(H_i, \dots, H_T) \leftarrow \mathcal{H}^{T-i+1}$, and $(U_i, \dots, U_T)$ are $T-i+1$ independent copies of the uniform distribution over the set V. Starting with $i = T$, the fact that $\mathbf{X}$ is a (T,k)-block-source guarantees that $X_T|_{X_1=x_1,\dots,X_{T-1}=x_{T-1}}$ is a k-source for any $x_1, \dots, x_{T-1}$. An application of the leftover hash lemma [HIL$^+$99] implies that the distributions $\mathcal{D}_T = (X_1, \dots, X_{T-1}, H_T, H_T(X_T))$ and $\mathcal{D}'_T = (X_1, \dots, X_{T-1}, H_T, U_T)$ are ϵ-close.

Now assume that the inductive claim holds for some value $i+1 \leq T$, and we show that it holds also for i. Again, the fact that $\mathbf{X}$ is a (T,k)-block-source guarantees that $X_i|_{X_1=x_1,\dots,X_{i-1}=x_{i-1}}$ is a k-source for any $x_1, \dots, x_{i-1}$. An application of the leftover hash lemma [HIL$^+$99] implies that the distributions $(X_1, \dots, X_{i-1}, H_i, H_i(X_i))$ and $(X_1, \dots, X_{i-1}, H_i, U_i)$ are ϵ-close. In turn, this implies that the distributions $\mathcal{Z} = (X_1, \dots, X_{i-1}, H_i, H_i(X_i), H_{i+1}, U_{i+1}, \dots, H_T, U_T)$ and $\mathcal{D}'_i = (X_1, \dots, X_{i-1}, H_i, U_i, H_{i+1}, U_{i+1}, \dots, H_T, U_T)$ are also ϵ-close. Note that

$$\mathbf{SD}(\mathcal{D}_i, \mathcal{Z}) \leq \mathbf{SD}(\mathcal{D}_{i+1}, \mathcal{D}'_{i+1}) \leq \epsilon(T-i),$$

as applying the function H_i to X_i can only increase the statistical distance. Therefore,

$$\begin{aligned}\mathbf{SD}(\mathcal{D}_i, \mathcal{D}'_i) &\leq \mathbf{SD}(\mathcal{D}_i, \mathcal{Z}) + \mathbf{SD}(\mathcal{Z}, \mathcal{D}'_i) \\ &\leq \epsilon(T-i) + \epsilon \\ &= \epsilon(T-i+1).\end{aligned}$$

▮

Lemma 2.4. *Let $\mathcal{H}$ be a universal collection of functions $H : U \to V$, and let $\mathbf{X} = (X_1, \dots, X_T)$ be $(k_1, \dots, k_T)$-source where $k_i \geq i \cdot \log|V| + 3\log(1/\epsilon) + \Theta(1)$ for every $i \in [T]$. Then, the distribution $(H_1, H_1(X_1), \dots, H_T, H_T(X_T))$, where $(H_1, \dots, H_T) \leftarrow \mathcal{H}^T$, is $2\epsilon T$-close to the uniform distribution over $(\mathcal{H} \times V)^T$.*

Proof. We prove the lemma via an inductive claim showing that for every $i \in [T]$ the distributions $\mathcal{D} = (H_1, H_1(X_1), \dots, H_T, H_T(X_T))$ and $\mathcal{D}_i = (H_1, H_1(X_1), \dots, H_{i-1}, H_{i-1}(X_{i-1}), H_i, U_i, \dots, H_T, U_T)$ are $2\epsilon(T-i+1)$-close, where $(H_1, \dots, H_T) \leftarrow \mathcal{H}^T$, and $(U_i, \dots, U_T)$ are $T-i+1$ independent copies of the uniform distribution over the set V.

Starting with $i = T$, Lemma 2.1 guarantees that for any $h_1, \dots, h_{T-1} \in \mathcal{H}$, with probability at least $1-\epsilon$ over the choice of $(y_1, \dots, y_{T-1}) \leftarrow (h_1(X_1), \dots, h_{T-1}(X_{T-1}))$ it holds that

$$\begin{aligned}&\mathbf{H}_\infty(X_T | H_1 = h_1, \dots, H_{T-1} = h_{T-1}, h_1(X_1) = y_1, \dots, h_{T-1}(X_{T-1}) = y_{T-1}) \\ &\quad\geq \mathbf{H}_\infty(X_T | H_1 = h_1, \dots, H_{T-1} = h_{T-1}) - (T-1)\log|V| - \log(1/\epsilon) \\ &\quad= \mathbf{H}_\infty(X_T) - (T-1)\log|V| - \log(1/\epsilon) \\ &\quad\geq k_T - (T-1)\log|V| - \log(1/\epsilon) \\ &\quad= \log|V| + 2\log(1/\epsilon) + \Theta(1).\end{aligned}$$

Therefore, the leftover hash lemma [HIL$^+$99] implies that the two distributions $\mathcal{D} = (H_1, H_1(X_1), \dots, H_T, H_T(X_T))$ and $\mathcal{D}_T = (H_1, H_1(X_1), \dots, H_{T-1}, H_{T-1}(X_{T-1}), H_T, U_T)$ are 2ϵ-close.

Now assume that the inductive claim holds for some value $i+1 \leq T$, and we show that it holds also for i. Again, Lemma 2.1 guarantees that for any $h_1, \dots, h_{i-1} \in \mathcal{H}$, with probability at least $1-\epsilon$ over the choice of $(y_1, \dots, y_{i-1}) \leftarrow (h_1(X_1), \dots, h_{i-1}(X_{i-1}))$ it holds that

$$\begin{aligned}&\mathbf{H}_\infty(X_i | H_1 = h_1, \dots, H_{i-1} = h_{i-1}, h_1(X_1) = y_1, \dots, h_{i-1}(X_{i-1}) = y_{i-1}) \\ &\quad\geq \mathbf{H}_\infty(X_i | H_1 = h_1, \dots, H_{i-1} = h_{i-1}) - (i-1)\log|V| - \log(1/\epsilon) \\ &\quad= \mathbf{H}_\infty(X_i) - (i-1)\log|V| - \log(1/\epsilon) \\ &\quad\geq k_i - (i-1)\log|V| - \log(1/\epsilon) \\ &\quad= \log|V| + 2\log(1/\epsilon) + \Theta(1).\end{aligned}$$

Therefore, the leftover hash lemma [HIL$^+$99] implies that the distributions $(H_1, H_1(X_1), \ldots, H_i, H_i(X_i))$ and $(H_1, H_1(X_1), \ldots, H_{i-1}, H_{i-1}(X_{i-1}), H_i, U_i)$ are 2ϵ-close. In turn, this implies that the distributions $\mathcal{D}_{i+1} = (H_1, H_1(X_1), \ldots, H_i, H_i(X_i), H_{i+1}, U_{i+1}, \ldots, H_T, U_T)$ and $\mathcal{D}_i = (H_1, H_1(X_1), \ldots, H_{i-1}, H_{i-1}(X_{i-1}), H_i, U_i, \ldots, H_T, U_T)$ are also 2ϵ-close. Therefore,

$$\begin{aligned} \mathbf{SD}(\mathcal{D}, \mathcal{D}_i) &\leq \mathbf{SD}(\mathcal{D}, \mathcal{D}_{i+1}) + \mathbf{SD}(\mathcal{D}_{i+1}, \mathcal{D}_i) \\ &\leq 2\epsilon(T-i) + 2\epsilon \\ &= 2\epsilon(T-i+1). \end{aligned}$$

∎

We also recollect the extended leftover hash lemma (cf. [DOR$^+$08] and [ABB10, Lemma 13]) in closely-related variants.

Lemma 2.5. *Let $m > (n+1) + \frac{\omega(\log\lambda)}{\log q}$ and let $q > 2$ be prime. Then, for all $\mathbf{v} \in \mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\intercal\mathbf{v})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\intercal\mathbf{v})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times k}$, and $\mathbf{R} \leftarrow \mathbb{Z}_q^{m\times k}$ for k polynomial in λ.*

The extended leftover hash lemma also holds with $\mathbf{R}$ is drawn uniformly with entries in $\{-1, 1\}$ (rather than $\mathbb{Z}_q$) at the expense of slightly larger m.

Lemma 2.6. *Let $m > (n+1)\log q + \omega(\log\lambda)$ and let $q > 2$ be prime. Then for all vectors $\mathbf{v} \in \mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\intercal\mathbf{v})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\intercal\mathbf{v})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times k}$, and $\mathbf{R} \leftarrow \{-1, 1\}^{m\times k}$ for k polynomial in λ.*

2.2 Identity-Based Encryption

An identity-based encryption (IBE) scheme [Sha84, BF03] is a quadruple $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of probabilistic polynomial-time algorithms. The setup algorithm, Setup, takes as input the security parameter 1^λ and outputs the public parameters pp of the scheme together with a corresponding master secret key msk. The encryption algorithm, Enc, takes as input the public parameters pp, an identity id, and a message m, and outputs a ciphertext $c = \mathsf{Enc}(\mathrm{pp}, \mathrm{id}, m)$. The key-generation algorithm, KeyGen, takes as input the master secret key msk and an identity id, and outputs a secret key $\mathrm{sk}_{\mathrm{id}}$ corresponding to id. The decryption algorithm, Dec, takes as input the public parameters pp, a ciphertext c, and a secret key $\mathrm{sk}_{\mathrm{id}}$, and outputs either a message m or the symbol $\perp$. For such a scheme we denote by $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ its identity space and message space, respectively.

Functionality. In terms of functionality, we require that the decryption algorithm is correct with all but a negligible probability. Specifically, for any security parameter $\lambda \in \mathbb{N}$, for any identity $\mathrm{id} \in \mathcal{ID}_\lambda$, and for any message $m \in \mathcal{M}_\lambda$ it holds that

$$\mathsf{Dec}(\mathrm{pp}, \mathsf{KeyGen}(\mathrm{msk}, \mathrm{id}), \mathsf{Enc}(\mathrm{pp}, \mathrm{id}, m)) = m$$

with probably at least $1 - \nu(\lambda)$ for a negligible function $\nu(\cdot)$, where the probability it taken over the internal randomness of the algorithm Setup, KeyGen, Enc, and Dec.

Data privacy. We consider the standard notion of anonymity and message indistinguishability under an adaptive chosen-identity chosen-plaintext attack (known as $\mathsf{anon\text{-}IND\text{-}ID\text{-}CPA}$ and abbreviated to DP in the rest of the paper). We also consider its "selective" variant that asks adversaries to announce ahead of time the challenge identities (known as $\mathsf{anon\text{-}IND\text{-}sID\text{-}CPA}$ and abbreviated to sDP in the rest of the paper).

Definition 2.7 (Data privacy – $\mathsf{anon\text{-}IND\text{-}ID\text{-}CPA}$)**.** An identity-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ over a identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ is *data private* if for any probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

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

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

1. $(\mathrm{msk}, \mathrm{pp}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $((\mathrm{id}^*_0, m^*_0), (\mathrm{id}^*_1, m^*_1), \mathsf{state}) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathrm{msk},\cdot)}(1^\lambda, \mathrm{pp})$, where $\mathrm{id}^*_0, \mathrm{id}^*_1 \in \mathcal{ID}_\lambda$ and $m^*_0, m^*_1 \in \mathcal{M}_\lambda$.
3. $c^* \leftarrow \mathsf{Enc}(\mathrm{pp}, \mathrm{id}^*_b, m^*_b)$.
4. $b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathrm{msk},\cdot)}(c^*, \mathsf{state})$, where $b' \in \{0,1\}$.
5. Denote by $\mathcal{S}$ the set of identities with which $\mathcal{A}$ queried $\mathsf{KeyGen}(\mathrm{msk}, \cdot)$.
6. If $\mathcal{S} \cap \{\mathrm{id}^*_0, \mathrm{id}^*_1\} = \emptyset$ then output b', and otherwise output $\perp$.

Definition 2.8 (Selective data privacy – $\mathsf{anon\text{-}IND\text{-}sID\text{-}CPA}$)**.** An identity-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ over a identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ is *selective data private* if for any probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

$$\mathbf{Adv}^{\mathsf{sDP}}_{\Pi,\mathcal{A}}(\lambda) \overset{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{sDP},\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{sDP},\Pi,\mathcal{A}}(\lambda) = 1\right]\right| \le \nu(\lambda),$$

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

1. $(\mathrm{id}^*_0, \mathrm{id}^*_1, \mathsf{state}_1) \leftarrow \mathcal{A}(1^\lambda)$, where $\mathrm{id}^*_0, \mathrm{id}^*_1 \in \mathcal{ID}_\lambda$.
2. $(\mathrm{msk}, \mathrm{pp}) \leftarrow \mathsf{Setup}(1^\lambda)$.
3. $(m^*_0, m^*_1, \mathsf{state}_2) \leftarrow \mathcal{A}(\mathsf{state}_1)$, where $m^*_0, m^*_1 \in \mathcal{M}_\lambda$.
4. $c^* \leftarrow \mathsf{Enc}(\mathrm{pp}, \mathrm{id}^*_b, m^*_b)$.
5. $b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathrm{msk},\cdot)}(c^*, \mathsf{state}_2)$, where $b' \in \{0,1\}$.
6. Denote by $\mathcal{S}$ the set of identities with which $\mathcal{A}$ queried $\mathsf{KeyGen}(\mathrm{msk}, \cdot)$.
7. If $\mathcal{S} \cap \{\mathrm{id}^*_0, \mathrm{id}^*_1\} = \emptyset$ then output b', and otherwise output $\perp$.

2.3 Computational Assumptions in Bilinear Groups

Our constructions in bilinear groups are based on the following computational assumptions.

The decisional bilinear Diffie-Hellman assumption (DBDH). Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^λ, and outputs $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e})$ where $\mathbb{G}$ and $\mathbb{G}_\mathrm{T}$ are groups of prime order p, $\mathbb{G}$ is generated by g, p is a λ-bit prime number, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathrm{T}$ is a non-degenerate efficiently computable bilinear map. The decisional bilinear Diffie-Hellman assumption is that the distributions $\left\{\left(g, g^a, g^b, g^c, \hat{e}(g,g)^{abc}\right)\right\}_{a,b,c\leftarrow\mathbb{Z}^*_p}$ and $\left\{\left(g, g^a, g^b, g^c, \hat{e}(g,g)^{d}\right)\right\}_{a,b,c,d\leftarrow\mathbb{Z}^*_p}$ are computationally indistinguishable, where $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e}) \leftarrow \mathsf{GroupGen}(1^\lambda)$.

The decisional linear assumption (DLIN). We rely on the matrix form of the decisional linear assumption, which is implied by the decisional linear assumption, as shown by Boneh, Halevi, Hamburg and Ostrovsky [BHH+08], and generalized by Naor and Segev [NS12]. Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^λ, and outputs a triplet $(\mathbb{G}, p, g)$ where $\mathbb{G}$ is a group of prime order p that is generated by $g \in \mathbb{G}$, and p is a λ-bit prime number. We denote by $\mathsf{Rk}_i(\mathbb{Z}_p^{a\times b})$ the set of all $a\times b$ matrices over $\mathbb{Z}_p$ of rank i. The matrix form of the decisional linear assumption is that for any integers a and b, and for any $2 \leq i < j \leq \min\{a, b\}$ the distributions $\{(\mathbb{G}, p, g, g^{\mathbf{X}})\}_{\mathbf{X}\leftarrow\mathsf{Rk}_i(\mathbb{Z}_p^{a\times b}),\lambda\in\mathbb{N}}$ and $\{(\mathbb{G}, p, g, g^{\mathbf{Y}})\}_{\mathbf{Y}\leftarrow\mathsf{Rk}_j(\mathbb{Z}_p^{a\times b}),\lambda\in\mathbb{N}}$ are computationally indistinguishable, where $(\mathbb{G}, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$.

2.4 Lattices

Probability distributions. The Gaussian distribution with mean 0 and variance σ^2 is the distribution on $\mathbb{R}$ having a density function $\frac{1}{\sigma\sqrt{2\pi}} \cdot \exp(-x^2/2\sigma^2)$.

For $\alpha \in \mathbb{R}^+$ and (implicit) $q \in \mathbb{Z}$, the distribution $\overline{\Psi}_\alpha$ is defined to be the *discretized* Gaussian distribution $\lfloor qX_\alpha \rceil \pmod q$ where X_α is a Gaussian with mean 0 and variance $\alpha^2/2\pi$ reduced modulo 1.

For a matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_m) \in \mathbb{Z}_q^{k\times m}$ of m vectors in $\mathbb{Z}_q^k$, $\|\mathbf{S}\| \stackrel{\mathsf{def}}{=} \max_{i\in[m]}(\|\mathbf{s}_i\|)$ and $\widetilde{\mathbf{S}} = (\widetilde{\mathbf{s}}_1, \ldots, \widetilde{\mathbf{s}}_m)$ denotes the Gram-Schmidt orthogonalization of $\mathbf{S}$.

Integer lattices. For q prime, $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$ define

$$\begin{aligned}
\Lambda_q(\mathbf{A}) &\stackrel{\mathsf{def}}{=} \left\{\mathbf{e} \in \mathbb{Z}^m \,\middle|\, \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ where } \mathbf{A}^\intercal\mathbf{s} = \mathbf{e} \,(\text{mod } q)\right\}\\
\Lambda_q^\perp(\mathbf{A}) &\stackrel{\mathsf{def}}{=} \{\mathbf{e} \in \mathbb{Z}^m \,|\, \mathbf{A}\mathbf{e} = \mathbf{0} \,(\text{mod } q)\}\\
\Lambda_q^{\mathbf{u}}(\mathbf{A}) &\stackrel{\mathsf{def}}{=} \{\mathbf{e} \in \mathbb{Z}^m \,|\, \mathbf{A}\mathbf{e} = \mathbf{u} \,(\text{mod } q)\}\,.
\end{aligned}$$

For a lattice Λ, let the Discrete Gaussian distribution over a lattice $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ denote the Gaussian distribution with probability mass $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) \stackrel{\mathsf{def}}{=} \exp\left(-\pi\frac{\|\mathbf{x}-\mathbf{c}\|^2}{\sigma^2}\right)$ restricted to $\mathbf{x} \in \Lambda$.

Sampling algorithms. We state the following relevant facts about lattices (see, for example, [ABB10] and references therein).

Lemma 2.9. *Let $q \geq 2$ and $m > n \log q$ be parameters that are polynomial in the security parameter, then:*

1. *There is an efficient algorithm* $\mathsf{TrapGen}$ *that outputs a pair $\mathbf{A}$ and $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ such that $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ and $\mathbf{T_A}$ is a basis for $\Lambda_q^\perp(\mathbf{A})$ satisfying $\|\widetilde{\mathbf{S}}\| \leq O(\sqrt{n \log q})$ and $\|\mathbf{S}\| \leq O(n \log q)$ with all but negligible probability.*
2. *For any $m_1, m_2 \in \mathbb{Z}^{\geq 0}$ and any $\mathbf{A} \in \mathbb{Z}_q^{n\times m_1}, \mathbf{C} \in \mathbb{Z}_q^{n\times m_2}$ there is an efficient algorithm* $\mathsf{ExtendBasis}$ *given a basis $\mathbf{T_B}$ for $\Lambda_q^\perp(\mathbf{B})$ produces a basis $\mathbf{T}$ for $\Lambda_q^\perp(\mathbf{A}|\mathbf{B}|\mathbf{C})$ such that $\|\widetilde{\mathbf{T}}\| = \|\widetilde{\mathbf{T_B}}\|$.*
3. $\Pr\left[\|\mathbf{x}\| > \sqrt{m}\sigma \,\middle|\, \mathbf{x} \leftarrow \mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma,\mathbf{c}}\right] \leq \text{negl}(n)$ *for any $\mathbf{c} \in \mathbb{R}^m$.*
4. *For any $\mathbf{u} \in \mathbb{Z}_q^n$ and any $\sigma > \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log m})$, there is an efficient algorithm* $\mathsf{SamplePre}$ *that returns $\mathbf{x}$ sampled statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}},\sigma}$. Additionally, the same algorithm* $\mathsf{SamplePre}$ *efficiently samples (given $\mathbf{T_A}$) from the distribution $\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}),\sigma,\mathbf{c}}$ for any $\mathbf{c} \in \mathbb{R}^m$.*

The learning with errors assumption (LWE). For a prime q, parameters $m = m(\lambda)$ and $n = n(\lambda)$, and any $\alpha = \alpha(\lambda)$ such that $\alpha q > 2\sqrt{n}$, it is hard for any polynomial time algorithm to distinguish between the distributions $\{(\mathbf{A}, \mathbf{A}^\mathsf{T}\mathbf{s} + \mathbf{x})\}_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\, \mathbf{s}\leftarrow\mathbb{Z}_q^n,\, \mathbf{x}\leftarrow\overline{\Psi}_\alpha^m}$ and $\{(\mathbf{A}, \mathbf{b})\}_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\, \mathbf{b}\leftarrow\mathbb{Z}_q^m}$. The problem of distinguishing the two distributions described above is the $\mathsf{LWE}_{q,\overline{\Psi}_\alpha}$ problem.

Regev in [Reg05] showed that any efficient adversary that solves the $\mathsf{LWE}_{q,\overline{\Psi}_\alpha}$ problem can be used to construct an efficient quantum algorithm for approximating SIVP and GapSVP in lattices to within $\tilde{O}(n/\alpha)$ factors. This is believed to be hard for appropriate polynomial choices of $m(\lambda)$ and $n(\lambda)$.

The following lemma [ABB10, Lemma 12] is used to bound the error term in the statement of the LWE assumption.

Lemma 2.10. *Let $\mathbf{e} \in \mathbb{Z}^m$ be a vector and let $\boldsymbol{\chi} \leftarrow \overline{\Psi}_\alpha^m$. Then the quantity $|\boldsymbol{\chi}^\mathsf{T}\mathbf{e}|$ treated as an integer between $[0, q-1]$ satisfies:*

$$|\boldsymbol{\chi}^\mathsf{T}\mathbf{e}| \leq \|\mathbf{e}\|_2 \cdot q\alpha\omega(\sqrt{\log m}) + \|\mathbf{e}\|_2\sqrt{m}/2$$

with an overwhelming probability in m.

2.5 Programmable Hash Functions

We describe the following family of hash functions introduced by Hofheinz and Kiltz [HK12] (henceforth denoted $\mathcal{H}_{\mathsf{HK}}$). For every $\lambda \in \mathbb{N}$, prime $p = p(\lambda)$ and a parameter $n = n(\lambda)$, define the family (for implicit λ) $\mathcal{H}_{\mathsf{HK}} : \{H_h : \{0,1\}^n \to \mathbb{Z}_p\}_{h\in\mathbb{Z}_p^n}$ as:

$$H_h(x) \overset{\mathsf{def}}{=} 1 - \sum_{i=1}^{n} x_i h_i \,(\text{mod } p) \text{ for } x = (x_1, \ldots, x_n) \in \{0,1\}^n \text{ and } h = (h_1, \ldots, h_n) \in \mathbb{Z}_p^n. \quad (2.1)$$

For a parameter $Q = Q(\lambda)$ that is poylnomial in λ (which will refer to the number of queries when the hash functions are used in proofs) we consider a sub-family of hash functions $\mathcal{H}_{\mathsf{HK},Q}$. To sample $H_h \leftarrow \mathcal{H}_{\mathsf{HK},Q}$ proceed as follows: set $J = \Theta(Q^2)$ and sample $\eta_{i,j}$ for $i \in [n]$ and $j \in [J]$ uniformly and independently from $\{-1, 0, 1\}$. Set $h_i = \sum_{j\in[J]} \eta_{i,j}$ to define the hash function $H_h = 1 - \sum_{i=1}^n h_i x_i$ as in Equation (2.1). Such a hash function family is $(1, Q)$-programmable in the terminology of Hofheinz and Kiltz which implies the following lemma (implicit in the proof of [HK12, Theorem 6]).

Lemma 2.11. *For any polynomial $Q = Q(\lambda)$, polynomial $n = n(\lambda)$, and any $p = p(\lambda)$, for any $(Q+1)$-tuple of inputs $x^*, x^{(1)}, \ldots, x^{(Q)} \in \{0,1\}^n$, we have*

$$\Pr\Big[H(x^*) = 0 \wedge H\left(x^{(1)}\right) \neq 0 \wedge \cdots \wedge H\left(x^{(Q)}\right) \neq 0\Big] \geq \alpha_{\mathsf{HK}} = \Theta\left(\frac{1}{Q\sqrt{n}}\right),$$

where the probability is taken over the choice of $H \leftarrow \mathcal{H}_{\mathsf{HK},Q}$.

2.6 Two Simple Linear Algebra Facts

The following two simple facts are used in our proofs. For completeness we include their proofs, although they are standard.

Lemma 2.12. *Let n, m, and q be integers such that $m \geq n$ and q is prime. Then the probability that a uniformly chosen matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ has rank less than n is at most $2/q^{m-n+1}$.*

Proof. We view $\mathbf{A}$ as a set of n uniformly and independently sampled vectors $\mathbf{a}_i \leftarrow \mathbb{Z}_q^m$. The probability that $\mathbf{A}$ has rank less than m is bounded above by:

$$\sum_{i=0}^{n-1} \Pr[\mathbf{a}_{i+1} \in \mathrm{span}(\mathbf{a}_1, \ldots, \mathbf{a}_i)] = \sum_{i=0}^{n-1} \frac{1}{q^{m-i}} < \frac{1}{q^{m-n+1}} \cdot \left(\frac{1}{1 - \frac{1}{q}} \right) < \frac{2}{q^{m-n+1}}.$$

∎

Lemma 2.13. *Let m, n, k, and q be integers such that $m \geq n$ and q is prime and let $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ be a full-rank matrix. Then, for uniform $\mathbf{S} \leftarrow \mathbb{Z}_q^{m \times k}$, $\mathbf{BS}$ is distributed uniformly over $\mathbb{Z}_q^{n \times k}$.*

Proof. Let $\mathbf{B}$ be viewed as $\mathbf{B} = [\mathbf{b}_1 | \cdots | \mathbf{b}_m]$ for column vectors $\mathbf{b}_i \in \mathbb{Z}_q^n$. As $\mathsf{Rk}(\mathbf{B}) = n$, there are n columns that are linearly independent. Let $\mathbf{B}^* \in \mathbb{Z}_q^{n \times n}$ denote the submatrix of these n linearly independent columns. Consider fixing $m-n$ rows of $\mathbf{S}$ corresponding to the remaining $m-n$ columns and only consider a $n \times m$ submatrix $\mathbf{S}^*$ that correspond to $\mathbf{B}^*$. The matrix $\mathbf{S}^*$ has column vectors $\mathbf{s}_1^*, \ldots, \mathbf{s}_k^* \in \mathbb{Z}_q^n$.

Then $\mathbf{BS} = [\mathbf{B}^*\mathbf{s}_1^* + \mathbf{u}_1 | \cdots | \mathbf{B}^*\mathbf{s}_k^* + \mathbf{u}_k]$ where $\mathbf{u}_1, \ldots, \mathbf{u}_k$ are arbitrary vectors that depend on the values of the fixed rows. For any $i \in [k]$, as $\mathbf{B}^*$ is full-rank, there is a bijection between vectors $\mathbf{s}_i^*$ and $\mathbf{B}^*\mathbf{s}_i^*$. As $\mathbf{s}_i^*$ is distributed uniformly over $\mathbb{Z}_q^n$, so is $\mathbf{B}^*\mathbf{s}_i^*$ and $\mathbf{B}^*\mathbf{s}_i^* + \mathbf{u}_i$. This in turn implies that $\mathbf{BS}$ is distributed uniformly over all possible $n \times k$ matrices.

The above result holds true for every possible fixing of the $m - n$ rows corresponding to the columns not in $\mathbf{B}^*$ and therefore holds true for the uniform distribution over these values as well. ∎

3 Modeling Function Privacy for IBE

In this section we introduce our notions of function privacy for anonymous IBE schemes.[9] Recall that the standard notion of security for anonymous IBE schemes, $\mathsf{anon\text{-}IND\text{-}ID\text{-}CPA}$, asks that a ciphertext $c = \mathsf{Enc}(\mathrm{pp}, \mathrm{id}, m)$ reveals essentially no information on the pair (id, m) as long as a secret key $\mathrm{sk}_{\mathrm{id}}$ corresponding to the identity id is not explicitly provided (but secret keys corresponding to other identities may be provided). We refer to this notion of security as *data privacy*, and refer the reader to Section 2.2 for the formal definition. As discussed in Section 1, we put forward two main notions of function privacy: A basic notion that is formalized in Section 3.1, and an "enhanced" notion that is formalized in Section 3.2. We then also formalize non-adaptive relaxations of these two notions in Section 3.3.

Throughout this section we let T, k, and $k_1, \ldots, k_T$ be functions of the security parameter $\lambda \in \mathbb{N}$. In addition, we note that in the random-oracle model, all algorithms, adversaries, oracles, and distributions are given access to the random oracle.

3.1 Function Privacy

Our basic notion of function privacy asks that it should not be possible to learn any information, beyond the absolute minimum necessary, on the identity id corresponding to a given secret key $\mathrm{sk}_{\mathrm{id}}$. Specifically, our notion considers adversaries that are given the public parameters of the scheme, and can interact with a "real-or-random" function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$. This oracle takes as input any adversarially-chosen distribution over vectors of identities, and outputs secret keys either for identities sampled from the given distribution or for independently and uniformly distributed

[9] We focus on *anonymous* IBE schemes as our motivating application is public-key *searchable* encryption, to which anonymity is crucial [ABC+08].

identities[10]. We allow adversaries to adaptively interact with the real-or-random oracle, for any polynomial number of queries, as long as the distributions have a certain amount of min-entropy. At the end of the interaction, we ask that adversaries have only a negligible probability of distinguishing between the "real" and "random" modes of the oracle. The following definitions formally capture our basic notion of function privacy.

Definition 3.1 (Real-or-random function-privacy oracle)**.** The real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ takes as input triplets of the form $(\mathsf{mode}, \mathrm{msk}, \boldsymbol{ID})$, where $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, msk is a master secret key, and $\boldsymbol{ID} = (ID_1, \ldots, ID_T) \in \mathcal{ID}^T$ is a circuit representing a joint distribution over $\mathcal{ID}^T$. If $\mathsf{mode} = \mathsf{real}$ then the oracle samples $(\mathrm{id}_1, \ldots, \mathrm{id}_T) \leftarrow \boldsymbol{ID}$ and if $\mathsf{mode} = \mathsf{rand}$ then the oracle samples $(\mathrm{id}_1, \ldots, \mathrm{id}_T) \leftarrow \mathcal{ID}^T$ uniformly. It then invokes the algorithm $\mathsf{KeyGen}(\mathrm{msk}, \cdot)$ on each of $\mathrm{id}_1, \ldots, \mathrm{id}_T$ and outputs a vector of secret keys $(\mathrm{sk}_{\mathrm{id}_1}, \ldots, \mathrm{sk}_{\mathrm{id}_T})$.

Definition 3.2 (Function-privacy adversary)**.** Let $X \in \{(T, k)\text{-block}, (k_1, \ldots, k_T)\}$. An X-source function-privacy adversary $\mathcal{A}$ is an algorithm that is given as input a pair $(1^\lambda, \mathrm{pp})$ and oracle access to $\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode}, \mathrm{msk}, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, and to $\mathsf{KeyGen}(\mathrm{msk}, \cdot)$, and each of its queries to $\mathsf{RoR}^{\mathsf{FP}}$ is an X-source.

Definition 3.3 (Function privacy)**.** Let $X \in \{(T, k)\text{-block}, (k_1, \ldots, k_T)\}$. An identity-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is *X-source function private* if for any probabilistic polynomial-time X-source function-privacy adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

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

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

1. $(\mathrm{pp}, \mathrm{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $b \leftarrow \mathcal{A}^{\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode},\mathrm{msk},\cdot),\mathsf{KeyGen}(\mathrm{msk},\cdot)}(1^\lambda, \mathrm{pp})$.
3. Output b.

In addition, such a scheme is *statistically* X-source function private if the above holds for any *computationally-unbounded* X-source enhanced function-privacy adversary making a polynomial number of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle.

Multi-shot vs. single-shot adversaries. Note that Definition 3.3 considers adversaries that query the function-privacy oracle for any polynomial number of times. In fact, as adversaries are also given access to the key-generation oracle, this "multi-shot" definition is polynomially equivalent to its "single-shot" variant in which adversaries query the real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ at most once. This is proved via a straightforward hybrid argument, where the hybrids are constructed such that only one query is forwarded to the function-privacy oracle, and all other queries are answered using the key-generation oracle.

Known schemes that are not function private. To exercise our notion of function privacy we demonstrate that the anonymous IBE schemes of Boneh and Frankin [BF03], Gentry, Peikert and Vaikuntanathan [GPV08], Agrawal, Boneh and Boyen [ABB10], and Kurosawa and Phong [KP11] are not function private. We present simple and efficient attacks showing that the schemes

[10] We note that the resulting notion of security is polynomially equivalent to the one obtained by using a "left-or-right" oracle instead of a "real-or-random" oracle, as for example, in the case of semantic security for public-key encryption schemes.

[BF03, GPV08] do not satisfy Definition 3.3, and note that almost identical attacks can be carried on [ABB10, KP11]. As discussed in Section 1, other anonymous IBE schemes such as [Gen06, BW06] for which we were not able to find such simple attacks, can always be *assumed* to be function private based on somewhat non-standard entropy-based assumptions (such assumptions would essentially state that the schemes satisfy our definition). In this paper we are interested in schemes whose function privacy can be based on standard assumptions.

The Boneh-Franklin scheme uses a random oracle $H : \mathcal{ID} \to \mathbb{G}$ and the secret key for id is $\mathrm{sk}_{\mathrm{id}} = H(\mathrm{id})^{\alpha}$ where $\alpha \leftarrow \mathbb{Z}_p$ is the master secret. The public parameters are g and $h = g^{\alpha}$ for some generator g of $\mathbb{G}$. Consider an adversary that queries the real-or-random oracle with the circuit of the distribution that samples a uniformly distributed id for which the most significant bit of $\hat{e}(g^{\alpha}, H(\mathrm{id}))$ is 0. Clearly, this distribution has almost full entropy, and can be described by a circuit of polynomial size given the public parameters.[11] Then, given $\mathrm{sk}_{\mathrm{id}} = H(\mathrm{id})^{\alpha}$ the adversary outputs 0 if the most significant bit of $\hat{e}(g, \mathrm{sk}_{\mathrm{id}})$ is 0 and outputs 1 otherwise. Since $\hat{e}(g, \mathrm{sk}_{\mathrm{id}}) = \hat{e}(g^{\alpha}, H(\mathrm{id}))$ it is easy to see that the adversary has advantage 1/2 in distinguishing the real mode from the rand mode, thereby breaking function privacy. In Section 4.1 we present a modification of this scheme which is function private.

In the scheme of Gentry, Peikert and Vaikuntanathan, the public parameters consist of a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and the master secret key is a short basis for the lattice $\Lambda_q^{\perp}(\mathbf{A})$. A secret key corresponding to an identity id is a short vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{Ae} = H(\mathrm{id}) \in \mathbb{Z}_q^n$, where $H : \mathcal{ID} \to \mathbb{Z}_q^n$ is a random oracle. Consider an adversary that queries the real-or-random oracle with the circuit of the distribution that samples a uniformly distributed id for which the most significant bit of $H(\mathrm{id})$ is 0. Then, given $\mathrm{sk}_{\mathrm{id}} = \mathbf{e}$ the adversary outputs 0 if the most significant bit of $\mathbf{Ae}$ is 0 and outputs 1 otherwise. Since $\mathbf{Ae} = H(\mathrm{id})$ it is easy to see that the adversary has advantage 1/2 in distinguishing the real mode from the rand mode, thereby breaking function privacy. In Section 4.2 we present a modification of this scheme which is function private.

3.2 Enhanced Function Privacy

We now put forward a stronger notion of function privacy, to which we refer as *enhanced* function privacy. Recall that our basic notion of function privacy asks that it should not be possible to learn any information, beyond the absolute minimum necessary, on the identity id corresponding to a given secret key $\mathrm{sk}_{\mathrm{id}}$. However, in various applications (such as searching on encrypted data), an adversary may obtain not only a secret key $\mathrm{sk}_{\mathrm{id}}$, but also an encryption $\mathsf{Enc}(\mathrm{pp}, \mathrm{id}, m)$ of some message m. Our notion of enhanced function privacy asks that even in such a scenario, it should not be possible to learn any unnecessary information on the identity id.

It is easy to observe that not any function-private IBE scheme is also enhanced function private. For example, given any function-private anonymous IBE scheme Π consider the scheme $\widetilde{\Pi}$ that is obtained by modifying Π's encryption algorithm as follows: In order to encrypt a message m for id, use Π's encryption algorithm for encrypting the pair (m, id) for id. The scheme $\widetilde{\Pi}$ preserves the function privacy and anonymity of Π, but it is clearly not enhanced function private.

We formalize the notion of enhanced function privacy by considering adversaries that interact not only with the key-generation and the real-or-random function-privacy oracles (as in Definition 3.3), but also with a function-privacy encryption oracle. This oracle, denoted $\mathsf{Enc}^{\mathsf{FP}}$, shares a state with the real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ and takes as inputs queries of the form (i, j, m) where i and j are integers, and m is a message. On input such a query, denote by $(\mathrm{id}_{i,1}, \ldots, \mathrm{id}_{i,T})$ the vector of identities that was sampled by the real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ when

[11] More specifically, rejection sampling can be used to obtain a sufficiently good approximation.

answering the adversary's i^{th} real-or-random query.[12] The function-privacy encryption oracle $\mathsf{Enc}^{\mathsf{FP}}$ then responds with $c \leftarrow \mathsf{Enc}(\mathrm{pp}, \mathrm{id}_{i,j}, m)$.

Definition 3.4 (Enhanced function privacy)**.** Let $X \in \{(T,k)\text{-block}, (k_1,\dots,k_T)\}$. An identity-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is X-source *enhanced* function private if for any probabilistic polynomial-time X-source function-privacy adversary $\mathcal{A}$ there exists a negligible function $\nu(\lambda)$ such that

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

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

1. $(\mathrm{pp}, \mathrm{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $b \leftarrow \mathcal{A}^{\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode},\mathrm{msk},\cdot,\cdot),\mathsf{Enc}^{\mathsf{FP}}(\mathrm{pp},\cdot,\cdot,\cdot),\mathsf{KeyGen}(\mathrm{msk},\cdot)}(1^\lambda, \mathrm{pp})$.
3. Output b.

Multi-shot vs. single-shot adversaries. We note that Definition 3.4 is polynomially equivalent to its "single-shot" variant in which adversaries query the real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ at most once (see the discussion following Definition 3.3). In this case the function-privacy encryption oracle $\mathsf{Enc}^{\mathsf{FP}}$ can be simplified to take as inputs queries of the form (j, m) instead of queries of the form (i, j, m) (since only the case $i = 1$ is possible).

3.3 Non-Adaptive Function Privacy

We now put forward non-adaptive relaxations of our notions of functions privacy. These relaxations ask that it should not be possible to learn any unnecessary information on the identity id corresponding to a given secret key $\mathrm{sk}_{\mathrm{id}}$, as long as id is not allowed to depend on the public parameters of the IBE scheme. As discussed in Section 3.1, such a non-adaptive notion is inspired by the notions of security for deterministic public-key encryption (DPKE) [BBO07].

On one hand, this definition is weaker than those presented in Sections 3.1 and 3.2. However, on the other, it may still suffice for various applications (see [BBO07]), and in Section 6 we show that it can be obtained generically from any anonymous IBE scheme and any family of collision-resistant hash functions (this is a more refined variant of the simple DPKE-based construction described in Section 3.1). In fact, this generic construction satisfies the non-adaptive relaxation of *enhanced* function privacy. For simplicity, in what follows we present the definition of non-adaptive *enhanced* function privacy, and note that the non-enhanced definition follows easily by not providing adversaries with access to the function-privacy encryption oracle $\mathsf{Enc}^{\mathsf{FP}}$.

Definition 3.5 (Non-adaptive enhanced function privacy)**.** Let $X \in \{(T,k)\text{-block}, (k_1,\dots,k_T)\}$. An identity-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is *X-source non-adaptive enhanced function private* if for any probabilistic polynomial-time X-source function-privacy adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

$$\mathbf{Adv}^{\mathsf{NA\text{-}EFP}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{NA\text{-}EFP},\Pi,\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{NA\text{-}EFP},\Pi,\mathcal{A}}(\lambda)=1\right]\right| \le \nu(\lambda),$$

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

[12] If the adversary made less than i real-or-random queries then the function-privacy encryption oracle $\mathsf{Enc}^{\mathsf{FP}}$ responds with $\perp$.

1. $(\boldsymbol{ID}, \mathsf{state}) \leftarrow \mathcal{A}(1^\lambda)$.
2. $(\mathrm{pp}, \mathrm{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
3. $(\mathrm{sk}_{\mathrm{id}_1}, \ldots, \mathrm{sk}_{\mathrm{id}_T}) \leftarrow \mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode}, \mathrm{msk}, \boldsymbol{ID})$.
4. $b \leftarrow \mathcal{A}^{\mathsf{Enc}^{\mathsf{FP}}(\mathrm{pp},\cdot,\cdot,\cdot),\mathsf{KeyGen}(\mathrm{msk},\cdot)}(\mathsf{state}, (\mathrm{sk}_{\mathrm{id}_1}, \ldots, \mathrm{sk}_{\mathrm{id}_T}))$.
5. Output b.

4 Function-Private Schemes in the Random-Oracle Model

4.1 A DBDH-Based Scheme

In this section we present an IBE scheme based on the DBDH assumption in the random-oracle model. The scheme is based on the IBE of Boneh and Franklin [BF03] by applying our "extract-augment-combine" approach, as discussed and exemplified in Section 1.1. The scheme is described below, and its proofs of data privacy and function privacy are presented in Sections 4.1.1 and 4.1.2, respectively.

The scheme. Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^λ, and outputs $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e})$ where $\mathbb{G}$ and $\mathbb{G}_\mathrm{T}$ are groups of prime order p, $\mathbb{G}$ is generated by g, p is a λ-bit prime number, and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathrm{T}$ is a non-degenerate efficiently computable bilinear map. The scheme $\mathcal{IBE}_{\mathsf{DBDH}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is parameterized by the security parameter $\lambda \in \mathbb{N}$. For any such $\lambda \in \mathbb{N}$ we denote by $\mathcal{ID}_\lambda$ and $\mathcal{M}_\lambda$ the identity space and the message space, respectively.

- **Setup:** On input 1^λ the setup algorithm samples $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e}) \leftarrow \mathsf{GroupGen}(1^\lambda)$ and $\alpha \leftarrow \mathbb{Z}_p^*$, and lets $h = g^\alpha$. It outputs the public parameters $\mathrm{pp} = (H, g, h)$ and the master secret key $\mathrm{msk} = \alpha$, where $H : \mathcal{ID}_\lambda \to \mathbb{G}^\ell$ is a hash function (modeled as a random oracle) for $\ell \geq \frac{2 \log |\mathcal{ID}_\lambda| + \omega(\log \lambda)}{\log p}$.
- **Key generation:** On input the master secret key $\mathrm{msk} = \alpha$ and a identity $\mathrm{id} \in \mathcal{ID}_\lambda$, the key-generation algorithm computes $H(\mathrm{id}) = (h_1, \ldots, h_\ell)$ and samples $s_1, \ldots, s_\ell \leftarrow \mathbb{Z}_p$. It then outputs the secret key $\mathrm{sk}_{\mathrm{id}} = \left(s_1, \ldots, s_\ell, \left(\prod_{j=1}^{\ell} h_j^{s_j}\right)^\alpha\right)$.
- **Encryption:** On input the public parameters $\mathrm{pp} = (H, g, h)$, an identity $\mathrm{id} \in \mathcal{ID}_\lambda$, and a message $\mathfrak{m} \in \mathbb{G}_\mathrm{T}$, the encryption algorithm computes $H(\mathrm{id}) = (h_1, \ldots, h_\ell)$ and samples $r \leftarrow \mathbb{Z}_p$. It then outputs the ciphertext $c = (c_0, \ldots, c_\ell)$, where $c_0 = g^r$ and $c_i = \hat{e}(h, h_i)^r \cdot \mathfrak{m}$ for every $i \in [\ell]$.
- **Decryption:** On input the public parameters $\mathrm{pp} = (H, g, h)$, a ciphertext $c = (c_0, \ldots, c_\ell)$, and a secret key $\mathrm{sk} = (s_1, \ldots, s_\ell, z)$, the decryption algorithm computes $d = \left(\prod_{i \in [\ell]} c_i^{s_i}\right) / \hat{e}(c_0, z)$, and outputs $\mathfrak{m} = d^{(s_1 + \cdots + s_\ell)^{-1}}$.

Correctness. Consider a message $\mathfrak{m}$, an encryption $(c_0, \ldots, c_\ell)$ of $\mathfrak{m}$ under identity id, and a secret key $(s_1, \ldots, s_\ell, z)$ corresponding to id. Then, we have

$$\begin{aligned} d = \frac{\prod_{i\in[\ell]} c_i^{s_i}}{\hat{e}(c_0, z)} &= \frac{\prod_{i\in[\ell]} \hat{e}(h, h_i)^{r\cdot s_i} \cdot \mathfrak{m}^{s_i}}{\hat{e}\left(c_0, \prod_{i\in[\ell]} h_i^{\alpha\cdot s_i}\right)} \\ &= \frac{\prod_{i\in[\ell]} \hat{e}(g^\alpha, h_i)^{r\cdot s_i}}{\hat{e}\left(g^r, \prod_{i\in[\ell]} h_i^{\alpha\cdot s_i}\right)} \cdot \mathfrak{m}^{s_1+\cdots+s_\ell} = \frac{\prod_{i\in[\ell]} \hat{e}(g, h_i)^{r\alpha\cdot s_i}}{\prod_{i\in[\ell]} \hat{e}(g^r, h_i)^{\alpha\cdot s_i}} \cdot \mathfrak{m}^{s_1+\cdots+s_\ell} \\ &= \mathfrak{m}^{s_1+\cdots+s_\ell}. \end{aligned}$$

Therefore, as long as $s_1+\cdots+s_\ell \neq 0 \pmod{p}$ (an event which occurs with probability $1-1/p$ over the randomness of KeyGen), the message is indeed correctly reconstructed by computing $d^{(s_1+\cdots+s_\ell)^{-1}}$.

Security. In Sections 4.1.1 and 4.1.2 we prove the following theorem:

Theorem 4.1. *In the random-oracle model the scheme $\mathcal{IBE}_{\mathsf{DBDH}}$ is data private based on the DBDH assumption, and is statistically function private for:*

1. *(T, k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq \lambda + \omega(\log \lambda)$.*
2. *$(k_1, \ldots, k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1, \ldots, k_T)$ such that $k_i \geq i \cdot \lambda + \omega(\log \lambda)$ for every $i \in [T]$.*

4.1.1 Proof of Data Privacy

Lemma 4.2. *The scheme $\mathcal{IBE}_{\mathsf{DBDH}}$ is data private based on the DBDH assumption in the random-oracle model.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial time adversary. For each $b \in \{0, 1\}$, we consider experiment $\mathsf{Expt}_0^{(b)}$ that is identical to $\mathsf{Expt}_{\mathsf{DP},\mathcal{IBE}_{\mathsf{DBDH}},\mathcal{A}}^{(b)}$ in Definition 2.7. Then, for each $i \in [\ell]$, we define experiment $\mathsf{Expt}_i^{(b)}$ (for $1 \leq i \leq \ell$) is identical to Expt_0 except in step (3) where the challenge ciphertext c^* is now $(c_0^*, u_1, \ldots, u_i, c_{i+1}^*, \ldots, c_\ell^*)$ for independently and uniformly sampled elements $u_1, \ldots, u_i$. In particular, we note that in $\mathsf{Expt}_\ell^{(0)}$ and $\mathsf{Expt}_\ell^{(1)}$, c_0 is chosen uniformly from $\mathbb{G}$ (and independent of $(\mathrm{id}_0^*, \mathfrak{m}_0^*)$ and $(\mathrm{id}_1^*, \mathfrak{m}_1^*)$) and the adversary's view is independent of b. Therefore $\mathsf{Expt}_\ell^{(0)} = \mathsf{Expt}_\ell^{(1)}$.

Claim 4.3. *Based on the DBDH assumption, for any $0 \leq i \leq \ell - 1$ and $b \in \{0, 1\}$, it holds that*

$$\left|\Pr\left[\mathsf{Expt}_i^{(b)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_{i+1}^{(b)}(\lambda) = 1\right]\right| \leq \mathrm{negl}(\lambda).$$

Proof. Assume the contrary. Denote by Q_T and Q_H the number of secret key and random oracle queries of a probabilistic polynomial time adversary $\mathcal{A}$ in experiment $\mathsf{Expt}_j^{(b)}$ for $j \in \{i, i+1\}$ such that

$$\left|\Pr\left[\mathsf{Expt}_i^{(b)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_{i+1}^{(b)}(\lambda) = 1\right]\right| > \epsilon(\lambda),$$

for some non-negligible $\epsilon(\lambda)$. We construct an algorithm $\mathcal{B}$ that solves the DBDH problem with advantage at least $\epsilon' = \epsilon/e(Q_T + 1)$. The algorithm $\mathcal{B}$ is given $(g, g_a = g^a, g_b = g^b, g_c = g^c, v)$ and interacts with $\mathcal{A}$ as follows:

- **Setup:** Algorithm $\mathcal{B}$ sets up $\mathrm{pp} = (g, g_a)$.
- ***H*-queries:** Algorithm $\mathcal{B}$ maintains a list of tuples $L = (\mathrm{id}_j, \mathbf{h}_j, \boldsymbol{\alpha}_j, \gamma_j)$ for $j \in \{1, \ldots, Q_H\}$ where $\mathbf{h}_j = \left(h_1^{(j)}, \ldots, h_\ell^{(j)}\right)$ is a vector of elements in $\mathbb{G}$ and $\boldsymbol{\alpha}_j = \left(\alpha_1^{(j)}, \ldots, \alpha_\ell^{(j)}\right)$ is a vector of $\mathbb{Z}_p$ elements. For each query $\mathrm{id} \in \mathcal{ID}$, $\mathcal{B}$ responds as follows:
  1. If $\mathrm{id} = \mathrm{id}_j \in L$ already for some id_j, algorithm $\mathcal{B}$ returns $H(\mathrm{id}) = \mathbf{h}_j$.
  2. Otherwise, $\mathcal{B}$ generates a random coin γ_j such that $\Pr[\gamma_j = 0] = 1/(Q_T + 1)$.
  3. Algorithm $\mathcal{B}$ picks a random $\boldsymbol{\alpha_j} \in \mathbb{Z}_p^n$.

     If $\gamma_j = 0$, $\mathcal{B}$ computes $h_{i+1}^{(j)} \leftarrow g_b \cdot g^{\alpha_{i+1}^{(j)}}$

     If $\gamma_j = 1$, $\mathcal{B}$ computes $h_{i+1}^{(j)} \leftarrow g^{\alpha_{i+1}^{(j)}}$

     Algorithm $\mathcal{B}$ sets the rest of the components of the $\mathbf{h}_j$ vector as follows: $h_\zeta^{(j)} = g^{\alpha_\zeta^{(j)}}$ for $\zeta \in [\ell] \backslash \{i+1\}$.
  4. Algorithm $\mathcal{B}$ adds the tuple $(\mathrm{id}_j, \mathbf{h}_j, \boldsymbol{\alpha_j}, \gamma_j)$ to the list L at position j.
- **Secret key queries:** When $\mathcal{A}$ issues a query for identity $\mathrm{id} \in \mathcal{ID}$, algorithm $\mathcal{B}$ responds as follows:
  1. Algorithm $\mathcal{B}$ computes $H(\mathrm{id})$ as above to obtain $(w, \mathbf{h}, \boldsymbol{\alpha}, \gamma)$. If $\gamma = 0$, algorithm $\mathcal{B}$ *aborts* and outputs a uniform bit.
  2. Else, we have $\gamma = 1$ and therefore $h_i = g^{\alpha_i}$ for each $i \in [n]$. Algorithm $\mathcal{B}$ chooses random values $s_1, \ldots, s_\ell \leftarrow \mathbb{Z}_p$, computes $z = \prod_{i=1}^n g_a^{s_i \alpha_i}$, and outputs $\mathrm{sk}_{\mathrm{id}} = (s_1, \ldots, s_\ell, z)$. Observe that $g_a = g^a$ and z is well-formed for the public parameters pp.
- **Challenge:** When $\mathcal{A}$ outputs two identities and two messages $(\mathrm{id}_0^*, \mathfrak{m}_0^*)$ and $(\mathrm{id}_1^*, \mathfrak{m}_1^*)$ on which to be challenged, $\mathcal{B}$ does the following:
  1. Depending on the bit b, it computes $H(\mathrm{id}_b^*) = \mathbf{h}$ as above and retrieves $(\mathrm{id}_b^*, \mathbf{h}, \boldsymbol{\alpha}, \gamma)$ from table L. If $\gamma = 1$, the algorithm $\mathcal{B}$ *aborts* and outputs a uniform bit.
  2. If $\gamma = 0$, it proceeds to set $c_0 = g_c$, and $(c_1^*, \ldots, c_i^*) = (u_1, \ldots, u_i)$ to uniform and independently chosen elements in $\mathbb{G}_\mathrm{T}$. Next, it sets $c_{i+1}^* = v \cdot \hat{e}(g_a, g_c)^{\alpha_{i+1}} \cdot \mathfrak{m}_b^*$. Finally, it sets c_ζ^* to $\hat{e}(g_a, g_c)^{\alpha_\zeta} \cdot \mathfrak{m}_b^*$, for $i + 2 \le \zeta \le \ell$.
  3. It returns $c^* = (c_0^*, c_1^*, \ldots, c_\ell^*)$ as the challenge ciphertext.
- **Output:** At the end of the experiment on receiving the bit b' as output, the adversary $\mathcal{B}$ outputs the bit b'

It is easy to see from the construction that in the challenge phase, if $\mathcal{B}$ is given a DBDH tuple, i.e., $v = \hat{e}(g, g)^{abc}$, then

$$\begin{aligned} c_{i+1}^* &= \hat{e}\,(g, g)^{abc} \cdot \hat{e}\,(g, g)^{ac\alpha_{i+1}} \cdot \mathfrak{m}_b^* = \hat{e}\,(g^a, g)^{c(b+\alpha_{i+1})} \cdot \mathfrak{m}_b^* \\ &= \hat{e}\left(g^a, g^{(b+\alpha_{i+1})}\right)^c \cdot \mathfrak{m}_b^* = \hat{e}\,(g^a, h_{i+1})^c \cdot \mathfrak{m}_b^* \end{aligned}$$

is well-formed and therefore, the challenge $(c_0^*, c_1^*, \ldots, c_\ell^*)$ is identically distributed to the challenge in $\mathsf{Expt}_i^{(b)}$. If $\mathcal{B}$ is given a random tuple, i.e., v is uniform over $\mathbb{G}_\mathrm{T}$, then in addition to $c_1^*, \ldots, c_i^*$, we have c_{i+1}^* is also distributed uniformly over $\mathbb{G}_\mathrm{T}$ and thus, $(c_0^*, c_1^*, \ldots, c_\ell^*)$ is identically distributed to the challenge in $\mathsf{Expt}_{i+1}^{(b)}$.

To complete the proof, it suffices to bound the probability of $\mathcal{B}$ aborting the simulation (denoted by Abort). We define two events: $\mathsf{Abort_T}$ the event that $\mathcal{B}$ aborts in one of the secret key queries, and $\mathsf{Abort_C}$ the event that $\mathcal{B}$ aborts during the challenge phase.

Without loss of generality, we assume that $\mathcal{A}$ does not ask for the secret key of the same identity twice. The probability that a secret key query causes $\mathcal{B}$ to abort is $1/(Q_T+1)$. To see this, note that γ_i is independent of $\mathcal{A}$'s view and $\mathcal{B}$ only aborts when $\gamma_i = 0$. As $\mathcal{A}$ makes at most Q_T secret key queries, the probability that $\mathcal{B}$ does not abort as a result of all secret key queries is at least $(1 - 1/(Q_T+1))^{Q_T} \geq 1/e$. Thus, $\Pr[\mathsf{Abort_T}] \leq 1 - 1/e$.

The algorithm $\mathcal{B}$ will abort during the challenge phase if $\mathcal{A}$ is able to produce id_b^* with the property that $\gamma = 1$ for that corresponding entry in L. Since $\mathcal{A}$ cannot query for a secret key of id_b^*, γ is set independently of $\mathcal{A}$'s view. With probability $\Pr[\gamma = 0] = 1/(Q_T+1)$, algorithm $\mathcal{B}$ does not abort and therefore, $\Pr[\mathsf{Abort_C}] \leq 1 - 1/(Q_T+1)$.

The two events $\overline{\mathsf{Abort_C}}$ and $\overline{\mathsf{Abort_T}}$ are independent because $\mathcal{A}$ cannot ask for secret key queries corresponding to id_b^*. Thus the probability of abort is at most $1 - \Pr\left[\overline{\mathsf{Abort_T}} \wedge \overline{\mathsf{Abort_C}}\right] \leq 1 - 1/e(Q_T+1)$.

Therefore, the advantage of $\mathcal{B}$ (where the probability is taken over choices of uniform $a, b, c \leftarrow \mathbb{G}$ and $v \leftarrow \mathbb{G}_\mathrm{T}$) is:

$$\begin{aligned}
\epsilon' &= \left|\Pr\left[\mathcal{B}\left(g, g^a, g^b, g^c, \hat{e}(g,g)^{abc}\right) = 1\right] - \Pr\left[\mathcal{B}\left(g, g^a, g^b, g^c, v\right) = 1\right]\right| \\
&= \left|\Pr\left[\mathcal{B}\left(g, g^a, g^b, g^c, \hat{e}(g,g)^{abc}\right) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right] + \frac{1}{2} \cdot \Pr[\mathsf{Abort}]\right. \\
&\qquad\qquad \left. - \left(\Pr\left[\mathcal{B}\left(g, g^a, g^b, g^c, v\right) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right] + \frac{1}{2} \cdot \Pr[\mathsf{Abort}]\right)\right| \\
&= \left|\Pr\left[\mathsf{Expt}_i^{(b)}(\mathcal{A}) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right] - \Pr\left[\mathsf{Expt}_{i+1}^{(b)}(\mathcal{A}) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right]\right| \\
&= \left|\Pr\left[\mathsf{Expt}_i^{(b)}(\mathcal{A}) = 1\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right] - \Pr\left[\mathsf{Expt}_{i+1}^{(b)}(\mathcal{A}) = 1\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right]\right| \\
&\geq \epsilon \cdot \frac{1}{e(Q_T+1)},
\end{aligned}$$

as required. The derivation uses the fact that the abort condition is independent of the view of the adversary. To see this, we can consider an identical simulation without an embedded DBDH challenge that does not abort until the entire interaction is done with the adversary, then chooses bits γ_i and decides to abort *aposteriori*. The two simulations are identical as far as the adversary is concerned. ∎

To conclude the proof of Lemma 4.2, we compute:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{DP}}_{\mathcal{IBE}_{\mathsf{DBDH}},\mathcal{A}}(\lambda) &= \left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{DP},\mathcal{IBE},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{DP},\mathcal{IBE},\mathcal{A}}(\lambda) = 1\right]\right| \\
&= \left|\Pr\left[\mathsf{Expt}_0^{(0)}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_0^{(1)}(\mathcal{A}) = 1\right]\right| \qquad (4.1) \\
&\leq \sum_{i=1}^{\ell} \left|\Pr\left[\mathsf{Expt}_{i-1}^{(0)}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_i^{(0)}(\mathcal{A}) = 1\right]\right| \\
&\qquad + \sum_{i=1}^{\ell} \left|\Pr\left[\mathsf{Expt}_{i-1}^{(1)}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_i^{(1)}(\mathcal{A}) = 1\right]\right| \\
&\qquad\qquad + \left|\Pr\left[\mathsf{Expt}_\ell^{(0)}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_\ell^{(1)}(\mathcal{A}) = 1\right]\right| \qquad (4.2) \\
&\leq 2ne(Q_T+1) \cdot \epsilon' + 0 \qquad (4.3) \\
&\leq \mathrm{negl}(\lambda), \qquad (4.4)
\end{aligned}$$

where (4.1) follows from the definition of the experiments, (4.2) follows from the triangle inequality, (4.3) follows from Claim 4.3, and (4.4) follows from the hardness of DBDH and the fact that Q_T is polynomial in n. ∎

4.1.2 Proof of Function Privacy

Lemma 4.4. *The scheme $\mathcal{IBE}_{\mathsf{DBDH}}$ is statistically function private in the random-oracle model for:*

1. *(T, k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq \lambda + \omega(\log \lambda)$.*
2. *$(k_1, \ldots, k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1, \ldots, k_T)$ such that $k_i \geq i \cdot \lambda + \omega(\log \lambda)$ for every $i \in [T]$.*

Proof. Let $X \in \{(T, k)\text{-block}, (k_1, \ldots, k_T)\}$, and let $\mathcal{A}$ be a computationally unbounded X-source function-privacy adversary that makes a polynomial number $Q_{\mathsf{RoR}} = Q_{\mathsf{RoR}}(\lambda)$ of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{DBDH}},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{DBDH}},\mathcal{A}}$ (we refer the reader to Definition 3.3 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$, respectively.

We first observe that since the hash function $H : \mathcal{ID}_\lambda \to \mathbb{G}^\ell$ is modeled as a random oracle, we can restrict ourselves to the above distributions conditioned on the event in which H is injective on the identity space $\mathcal{ID}_\lambda$. Indeed, since $\mathbb{G}$ is a group of order p where p is a λ-bit prime number, our choice of $\ell = \ell(\lambda) \geq \frac{2 \log |\mathcal{ID}_\lambda| + \omega(\log \lambda)}{\log p}$ implies that

$$\begin{aligned}\Pr_H[H \text{ is injective on } \mathcal{ID}_\lambda] &\geq 1 - \frac{|\mathcal{ID}_\lambda|^2}{p^\ell} \\ &= 1 - 2^{-\omega(\log \lambda)}.\end{aligned}$$

Assuming that H is injective guarantees that for any X-source $\boldsymbol{ID} = (ID_1, \ldots, ID_T)$ over $(\mathcal{ID}_\lambda)^T$ it holds that $H(\boldsymbol{ID}) \stackrel{\mathsf{def}}{=} (H(ID_1), \ldots, H(ID_T))$ is an X-source over $(\mathbb{G}^\ell)^T$. From this point on we fix a function H which is injective over $\mathcal{ID}_\lambda$, and show that the two distributions $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$ are statistically close for any such function H.

Next, as the adversary $\mathcal{A}$ is computationally unbounded, we assume without loss of generality that $\mathcal{A}$ public parameters in our scheme uniquely determine the master secret key $\mathrm{msk} = \alpha$, such queries can be internally simulated by $\mathcal{A}$. Moreover, as discussed in Section 3.1, it suffices to focus on adversaries $\mathcal{A}$ that query the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once. From this point on we fix the value of $\alpha \in \mathbb{Z}_p$ chosen by the setup algorithm, and show that the two distributions $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$ are statistically close for any such α.

Denote by $\boldsymbol{ID} = (ID_1, \ldots, ID_T)$ the random variable corresponding to the X-source with which $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. Having already fixed H and α, we can assume that

$$\mathsf{View}_{\mathsf{mode}} = \left(\left(s_{1,1}, \ldots, s_{1,\ell}, \prod_{j=1}^{\ell} h_{1,j}^{s_{1,j}} \right), \ldots, \left(s_{T,1}, \ldots, s_{T,\ell}, \prod_{j=1}^{\ell} h_{T,j}^{s_{T,j}} \right) \right)$$

for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, where $(\mathrm{id}_1, \ldots, \mathrm{id}_T) \leftarrow (ID_1, \ldots, ID_T)$ for $\mathsf{mode} = \mathsf{real}$, $(\mathrm{id}_1, \ldots, \mathrm{id}_T)$ is uniformly distributed over $(\mathcal{ID}_\lambda)^T$ for $\mathsf{mode} = \mathsf{rand}$, $H(\mathrm{id}_i) = (h_{i,1}, \ldots, h_{i,\ell})$ for every $i \in [T]$, and $s_{i,j} \leftarrow \mathbb{Z}_p$ for every $i \in [T]$ and $j \in [\ell]$. For $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ we prove that the distribution $\mathsf{View}_{\mathsf{mode}}$ is statistically-close to uniform.

Note that the collection of functions $\{f_{s_1,\ldots,s_\ell} : \mathbb{G}^\ell \to \mathbb{G}\}_{s_1,\ldots,s_\ell \in \mathbb{Z}_p}$ defined by $f_{s_1,\ldots,s_\ell}(h_1,\ldots,h_\ell) = \prod_{j=1}^{\ell} h_j^{s_j}$ is universal. This enables us to directly apply Lemma 2.3 (in case $\boldsymbol{ID}$ is a (T,k)-block-source) and Lemma 2.4 (in case $\boldsymbol{ID}$ is a $(k_1,\ldots,k_T)$-source), implying that the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ. The same clearly holds also for $\mathsf{View}_{\mathsf{rand}}$, as the uniform distribution over $(\mathcal{ID}_\lambda)^T$ is, in particular, a (T,k)-block-source and a $(k_1,\ldots,k_T)$-source. ∎

4.2 An LWE-Based Scheme

In this section we present an IBE scheme based on the LWE assumption in the random-oracle model. The scheme is based the IBE scheme of Gentry, Peikert, and Vaikuntanathan [GPV08] by applying our "extract-augment-combine" approach described in Section 1.1. In what follows, before formally describing our scheme, we discuss the main challenges in applying our approach to the IBE scheme of Gentry et al. (we refer to their scheme as the GPV scheme).

In the GPV scheme, the public parameters consist of a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and the master secret key is a short basis $\mathbf{T_A}$ for the lattice $\Lambda_q^\perp(\mathbf{A})$. A secret key corresponding to an identity id is a short vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{Ae} = H(\mathrm{id}) \in \mathbb{Z}_q^n$. Thus, a natural application of our "extract" step for generating a secret key corresponding to an identity id, would be to view $H(\mathrm{id})$ as a matrix over $\mathbb{Z}_q^{n\times\ell}$, sample a uniform vector $\mathbf{s} \in \mathbb{Z}_q^\ell$, and output a short vector $\mathbf{e}$ such that $\mathbf{Ae} = H(\mathrm{id})\cdot\mathbf{s} \in \mathbb{Z}_q^n$. As long as the matrix $H(\mathrm{id}) - H(\mathrm{id}')$ is of full rank for all identities id and id$'$, the map $H(\mathrm{id}) \mapsto H(\mathrm{id})\cdot\mathbf{s}$ is a collection of universal functions over the choice of uniform $\mathbf{s} \in \mathbb{Z}_q^\ell$. Therefore, in particular, such a short vector $\mathbf{e}$ reveals essentially no information on id so long as id is sufficiently unpredictable.

The main difficulty, however, is to guarantee the correctness of decryption in the "augment" and "combine" steps. In the GPV scheme, ciphertexts are decrypted by computing an inner-product with the vector $\mathbf{e}$, while carefully making sure (during encryption) that the added noise term (which guarantees data privacy) does not overwhelm the rest of the ciphertext. Applying a similar idea in our scheme runs into trouble because the entries of the vector $\mathbf{s}$ are not small and the therefore the noise term grows too large.

We overcome this difficulty by augmenting the public parameters with matrices $\mathbf{B}_1,\ldots,\mathbf{B}_d$ (where d is chosen such that q is a d-bit prime) that allow us to compute inner products with *low-norm* vectors over $\mathbb{Z}_q^\ell$ that correspond to the bit representation of a uniform $\mathbf{s}$. Using such low-norm vectors ensures that the noise terms do not overwhelm the message, and our "combine" step then produces an encryption of $\mathfrak{m}\cdot\left(\sum_{i\in[d]}\|\mathbf{s}_i\|_1\right)$.[13] By choosing our parameters appropriately, we can guarantee that this remains an encryption of the original $\mathfrak{m}$ and thus enables decryption. We note that the idea of representing $\mathbf{s}$ as its bit-vectors is inspired by that of Agrawal, Freeman, and Vaikuntanathan [AFV11].

The scheme. The scheme $\mathcal{IBE}_{\mathsf{LWE1}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is parameterized by the security parameter $\lambda \in \mathbb{N}$. Let $\mathcal{ID}_\lambda$ denote the identity space. The scheme additionally has lattice parameters m, n and q, a parameter $\ell \in \mathbb{N}$ related to randomness extraction, and $d \in \mathbb{N}$ such that q is a d-bit prime.

- **Setup:** On input 1^λ the setup algorithm picks parameters m, n, q and α as stated in the formulation of the $\mathsf{LWE}_{q,\overline{\Psi}_\alpha}$ assumption (see Section 2.4). The algorithm samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ with a trapdoor $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ for $\Lambda_q^\perp(\mathbf{A})$ by using the algorithm $\mathsf{TrapGen}$ (as described in Section 2.4). In addition, it samples $\mathbf{B}_1,\ldots,\mathbf{B}_d \leftarrow \mathbb{Z}_q^{n\times\ell}$ and a hash function $H : \mathcal{ID}_\lambda \to \mathbb{Z}_q^{n\times\ell}$

[13] Here $\|\cdot\|_1$ denotes the ℓ_1-norm of a vector.

(modeled as a random oracle). It outputs the public parameters pp $= (\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_d)$ and the master secret key $msk = \mathbf{T}_{\mathbf{A}}$.

- **Key generation:** On input the public parameters pp and an identity id $\in \mathcal{ID}_\lambda$ the algorithm samples $\mathbf{s} \leftarrow \mathbb{Z}_q^\ell$ and parses $H(\text{id})$ as a matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times\ell}$. It represents $\mathbf{s} = \sum_{i\in[d]} 2^{i-1}\cdot\mathbf{s}_i \pmod q$ where the $\mathbf{s}_i$'s are vectors over $\{0,1\}^\ell$. Running algorithm $\mathsf{SamplePre}$ with the lattice trapdoor $\mathbf{T}_{\mathbf{A}}$ it samples $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{e} = \left(\mathbf{H}\mathbf{s} + \sum_{i\in[d]} \mathbf{B}_i\mathbf{s}_i\right) \pmod q$. It outputs $\mathrm{sk}_{\text{id}} = (\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^\ell \times \mathbb{Z}^m$.
- **Encryption:** On input the the public parameters pp, an identity id $\in \mathcal{ID}_\lambda$, and a message $\mathfrak{m} \in \{0,1\}$, the algorithm samples $\mathbf{r} \leftarrow \mathbb{Z}_q^n$ and computes $H(\text{id}) = \mathbf{H} \in \mathbb{Z}_q^{n\times\ell}$. Next, it chooses (low-norm) error vectors $\boldsymbol{\chi}_0 \leftarrow \overline{\Psi}_\alpha^m$ and $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_d \leftarrow \overline{\Psi}_\alpha^\ell$. Let $\mathbf{1}$ denote the all-ones vector over $\mathbb{Z}_q^\ell$. It outputs
$$\mathsf{Enc}(\text{pp}, \text{id}, m) = \left(\mathbf{A}^\mathsf{T}\mathbf{r} + \boldsymbol{\chi}_0, \left\{\left(2^{i-1}\cdot\mathbf{H} + \mathbf{B}_i\right)^\mathsf{T}\mathbf{r} + \boldsymbol{\chi}_i + \mathfrak{m}\cdot\frac{q}{2\ell d}\cdot\mathbf{1}\right\}_{i\in[d]}\right) \in \mathbb{Z}_q^m \times (\mathbb{Z}_q^\ell)^d.$$
- **Decryption:** On input the public parameters pp, a ciphertext $(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_d)$, and a secret key $(\mathbf{s}, \mathbf{e})$, the algorithm represents $\mathbf{s} = \sum_{i\in[d]} 2^{i-1}\cdot\mathbf{s}_i \pmod q$ and outputs 0 if $|(\mathbf{c}_0^\mathsf{T}\mathbf{e} - \sum_{i\in[d]} \mathbf{c}_i{}^\mathsf{T}\mathbf{s}_i) \pmod q| < \frac{q}{10}$ and 1 otherwise.

Parameter selection. For the scheme, n is polynomial in the security parameter λ, and we set $m = n\cdot\omega(\log n)$, $q = m^{2.5}\cdot\omega(\sqrt{\log n})$, $\alpha = \frac{1}{m^2\cdot\omega(\sqrt{\log n})}$, and $\ell \geq n + \frac{2\log|\mathcal{ID}_\lambda| + \log n + \omega(\log\lambda)}{\log q}$.

Correctness. Consider a ciphertext $(\mathbf{c}_0, \ldots, \mathbf{c}_d)$ and the corresponding secret key $(\mathbf{s}, \mathbf{e})$ generated by running algorithms for encryption and secret key generation for the same identity. To see correctness of the decryption algorithm, observe that $\sum_{i\in[d]} \mathbf{c}^\mathsf{T}\mathbf{s}_i = \sum_{i\in[d]} \mathbf{r}^\mathsf{T}\left(2^{i-1}\cdot\mathbf{H} + \mathbf{B}_i + \boldsymbol{\chi}_i{}^\mathsf{T}\right)\mathbf{s}_i + \sum_{i\in[d]} \mathfrak{m}\cdot\frac{q}{2n\log q}\cdot\mathbf{1}^\mathsf{T}\mathbf{s}_i$ which equals $\mathbf{r}^\mathsf{T}\left(\mathbf{H}\mathbf{s} + \sum_{i\in[d]}\mathbf{B}_i\mathbf{s}_i\right)$ plus error term $\sum_{i\in[d]} \boldsymbol{\chi}_i{}^\mathsf{T}\mathbf{s}_i$ plus message term $\mathfrak{m}\cdot\frac{q}{2\ell d}\cdot\left(\sum_{i\in[d]}\mathbf{1}^\mathsf{T}\mathbf{s}_i\right)$. Note that $\mathbf{e}$ is constructed such that $\mathbf{c}_0{}^\mathsf{T}\mathbf{e} = \mathbf{r}^\mathsf{T}\mathbf{A}\mathbf{e} + \boldsymbol{\chi}_0{}^\mathsf{T}\mathbf{e}$ and $\mathbf{r}^\mathsf{T}\mathbf{A}\mathbf{e}$ cancels the corresponding term with $\mathbf{r}^\mathsf{T}$ from earlier. To bound the error terms, we have that with overwhelming probability

$$\begin{align}
\left|\boldsymbol{\chi}_0{}^\mathsf{T}\mathbf{e} - \sum_{i\in[d]}\boldsymbol{\chi}_i{}^\mathsf{T}\mathbf{s}_i\right| &\leq \left(\sqrt{m}/2 + q\alpha\omega(\sqrt{\log m})\right)\left(\|\mathbf{e}\|_2 + \sum_{i\in[d]}\|\mathbf{s}_i\|_2\right) \tag{4.5}\\
&\leq \left(\sqrt{m}/2 + q\alpha\omega(\sqrt{\log m})\right)\left(\sqrt{m}\cdot d\sqrt{m} + \sqrt{m}\cdot\|\widetilde{\mathbf{T}_{\mathbf{A}}}\|\cdot\omega(\sqrt{\log m})\right) \tag{4.6}\\
&\leq \sqrt{m}(1/2+1)\cdot\left(md + m^{1.5}\right)\omega(\sqrt{\log m}) \tag{4.7}\\
&\leq \tilde{O}(m^2) < q^{4/5}. \tag{4.8}
\end{align}$$

Equation (4.5) follows from Lemma 2.10, Equation (4.6) follows from the bound on $\|\mathbf{e}\|_2$ in Lemma 2.9, Equation (4.7) follows from the quality of $\|\widetilde{\mathbf{T}_{\mathbf{A}}}\|$ from Lemma 2.9, and Equation (4.8) follows from collecting terms and observing that $d \approx \log q$.

1. If $\mathfrak{m} = 0$, the message term is 0 and from Equation (4.8), Dec successfully decrypts the message.

2. If $\mathfrak{m} = 1$, then the message term $\frac{q}{2\ell d} \cdot \left(\sum_{i\in[d]} \|\mathbf{s}_i\|_1\right)$ where $\|\cdot\|_1$ denotes the ℓ_1 norm of a vector. Observe that for a majority of the lower-order bits of $\mathbf{s} \leftarrow \mathbb{Z}_q^\ell$, the corresponding vectors $\mathbf{s}_i$ are drawn uniformly from $\{0,1\}^\ell$. Applying a standard Chernoff bound implies that $\Pr[\|\mathbf{s}_i\|_1 < \ell/2 - \Gamma]$ is negligible in n for any $\Gamma \geq \omega(\log n)\sqrt{\ell}$. Thus, setting $\Gamma = 3\ell/10$ and observing that this bound holds for at least $d/2$ of the $\mathbf{s}_i$'s implies that the term $\frac{q}{2\ell d}\left(\sum_{i\in[d]} \|\mathbf{s}_i\|_1\right)$ is bounded below by $q/5$ with overwhelming probability. Therefore, Dec successfully decrypts the message with overwhelming probability.

Security. In Sections 4.2.1 and 4.2.2 we prove the following theorem:

Theorem 4.5. *In the random-oracle model the scheme $\mathcal{IBE}_{\mathsf{LWE1}}$ is data private based on the LWE assumption, and is statistically function private for:*

1. *(T,k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq n\log q + \omega(\log\lambda)$.*
2. *$(k_1,\ldots,k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1,\ldots,k_T)$ such that $k_i \geq i\cdot n\log q + \omega(\log\lambda)$ for every $i \in [T]$.*

Proof overview. The function privacy of the scheme follows quite naturally from our "extract" step, as discussed in Section 1.1. The proof of data privacy is inspired by the proof of the GPV scheme [GPV08], extended to deal with the extraction and bit-representation issues discussed above. Briefly, the proof of the GPV scheme uses the fact that to answer a key-generation query, without actually knowing a short basis for $\Lambda_q^\perp(\mathbf{A})$, it is possible to construct an appropriate short vector $\mathbf{e}$ by programming the random oracle at $H(\mathrm{id})$.

We use a similar approach that is adapted to deal with the augmented ciphertext that includes additional information using the public parameters $\mathbf{B}_1,\ldots,\mathbf{B}_d$. To do so, we consider a larger LWE challenge $(\mathbf{A}\,|\,\mathbf{H}_1\,|\,\cdots\,|\,\mathbf{H}_d)$ and we construct the augmented public parameters $\mathbf{B}_1,\ldots,\mathbf{B}_d$ appropriately for a programmed output $\mathbf{H}^*$ of the random oracle on the challenge identity id* (specifically, we set $\mathbf{B}_i = \mathbf{H}_i - 2^{i-1}\cdot\mathbf{H}^*$). This allows us to map the LWE challenge vector to either a well-formed ciphertext or a random ciphertext.

4.2.1 Proof of Data Privacy

Lemma 4.6. *The scheme $\mathcal{IBE}_{\mathsf{LWE1}}$ is data private based on the LWE assumption in the random-oracle model.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial time adversary. Experiment Expt_0 is identical to $\mathsf{Expt}^{(0)}_{\mathsf{DP},\mathcal{IBE}_{\mathsf{LWE1}},\mathcal{A}}$ in Definition 2.7. Experiment Expt_1 is identical to Expt_0 except in step (3). The challenger replaces a well-constructed challenge ciphertext with independently and uniformly sampled $(\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_d) \leftarrow \mathbb{Z}_q^m \times (\mathbb{Z}_q^\ell)^d$. Experiment Expt_2 is identical to $\mathsf{Expt}^{(1)}_{\mathsf{DP},\mathcal{IBE}_{\mathsf{DLIN1}},\mathcal{A}}$ in Definition 2.7. Now we can state the following claim.

Claim 4.7. *Based on the LWE assumption, it holds that* $|\Pr[\mathsf{Expt}_0(\lambda)=1] - \Pr[\mathsf{Expt}_1(\lambda)=1]| \leq \mathrm{negl}(\lambda)$.

Proof. Denote by Q_T and Q_H the number of secret key and random oracle queries of a probabilistic polynomial time adversary $\mathcal{A}$ in experiment Expt_0 and Expt_1 such that

$$|\Pr[\mathsf{Expt}_0(\lambda)=1] - \Pr[\mathsf{Expt}_1(\lambda)=1]| > \epsilon(\lambda),$$

for some non-negligible $\epsilon(\lambda)$. We construct an algorithm $\mathcal{B}$ that solves the LWE problem with advantage at least $\epsilon' = \epsilon/(Q_H + 1)$. The algorithm $\mathcal{B}$ is given $(\mathbf{E}, \mathbf{f}) \in \mathbb{Z}_q^{n \times (m+\ell d)} \times \mathbb{Z}_q^{(m+\ell d)}$ and interacts with $\mathcal{A}$ to decide whether $\mathbf{f}$ comes from the uniform distribution or from the distribution $\mathbf{E}^{\intercal}\mathbf{s} + \boldsymbol{\chi}$ for $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\boldsymbol{\chi} \leftarrow \overline{\Psi}_\alpha^{m+\ell d}$. In the proof that follows we assume without loss of generality that all Q_H random oracle queries are distinct. The algorithm $\mathcal{B}$ chooses a random integer $i^* \in [1, Q_H]$ and proceeds as follows:

- **Setup:** The algorithm $\mathcal{B}$ parses the matrix $\mathbf{E}$ as $d+1$ matrices $(\mathbf{A} \,|\, \mathbf{H}_1 \,|\, \cdots \,|\, \mathbf{H}_d) \in \mathbb{Z}_q^{n \times m} \times (\mathbb{Z}_q^{n \times \ell})^d$. Next, it chooses a random $\mathbf{H}^* \in \mathbb{Z}_q^{n \times \ell}$ and sets $\mathbf{B}_i = \mathbf{H}_i - 2^{i-1} \cdot \mathbf{H}^*$. It publishes $\text{pp} = (\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_d)$.

  As the matrix $\mathbf{E}$ is drawn uniformly from $\mathbb{Z}_q^{n \times (m+\ell d)}$, the public parameters are distributed uniformly as in the real scheme.

- ***H*-queries:** The algorithm $\mathcal{B}$ maintains a list of tuples $L = (\text{id}_j, \mathbf{H}_j, \mathbf{s}_j, \mathbf{e}_j)$ (for $j \in [Q_H]$) to answer hash queries. Here $\mathbf{H}_j \in \mathbb{Z}_q^{n \times \ell}$ and $\mathbf{s} \in \mathbb{Z}_p^\ell$. For each distinct query $\text{id} \in \mathcal{ID}$, algorithm $\mathcal{B}$ responds as follows:

  1. If id is the i^*-th query, add $(\text{id}_{i^*}, \mathbf{H}^*, \times, \times)$, where × denotes any junk/random value.
  2. Otherwise, to create entry j, $\mathcal{B}$ samples a discrete Gaussian vector $\mathbf{e}_j \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sqrt{m}}$. Next, it samples uniform $\mathbf{s}_j \leftarrow \mathbb{Z}_p^\ell$ (split into bit-vectors $\{\mathbf{s}_i^{(j)}\}_{i \in [d]}$) and solves for $\mathbf{H}_j \in \mathbb{Z}_q^{n \times \ell}$ such that $\mathbf{H}_j \mathbf{s}_j = \mathbf{A}\mathbf{e}_j - \sum_{i \in [d]} \mathbf{B}_i \mathbf{s}_i^{(j)} \pmod{q}$.
  3. The algorithm $\mathcal{B}$ adds the tuple $(\text{id}_j, \mathbf{H}_j, \mathbf{s}_j, \mathbf{e}_j)$ to the list L and returns $\mathbf{H}_j$.

  We need to argue that the random oracle output $\mathbf{H}_j$ sampled above is distributed as in the real scheme. To see that, we skip ahead to the proof of function privacy (see Section 4.2.2). We show that in the real scheme for random $\mathbf{s} \leftarrow \mathbb{Z}_p^\ell$ and random $\mathbf{H} \in \mathbb{Z}_q^{n \times \ell}$, $\mathbf{Hs}$ (and therefore $\mathbf{Hs} + \sum_{i \in [d]} \mathbf{B}_i \mathbf{s}_i$) is statistically close to a uniform vector $\mathbf{v} \in \mathbb{Z}_q^n$. Additionally, from [GPV08, Corollary 5.4], with overwhelming probability over the choice of $\mathbf{A}$, for $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sqrt{m}}$, the syndrome $\mathbf{Ae} \pmod{q}$ is statistically close to uniform over $\mathbb{Z}_q^n$. Now, we look at two distributions $\mathcal{D}_{\mathsf{real}} = (\mathbf{s}, \mathbf{H}, \mathbf{Hs} + \sum_{i \in [d]} \mathbf{B}_i \mathbf{s}_i)$ and $\mathcal{D}_{\mathsf{sim}} = (\mathbf{s}, \mathbf{H}_j, \mathbf{v})$. Observe that $\mathbf{s}$ is distributed identically in both distributions, and $\mathbf{v}$ is statistically close to $\mathbf{Hs}$. Therefore, it suffices to show that we can sample $\mathbf{H}_j$ uniformly from $\mathbb{Z}_q^{n \times \ell}$ conditioned on $\mathbf{H}_j \mathbf{s} = \mathbf{v}$. This is easily done by observing that the rows of $\mathbf{H}_j$ can be sampled independently and uniformly over the $(\ell - 1)$-dimensional subspace derived from the above constraint.

- **Secret key queries:** When $\mathcal{A}$ issues a query for sk_{id}, the algorithm $\mathcal{B}$ responds as follows (without loss of generality we can assume that id was one of the H-queries):

  1. If $\text{id} = \text{id}_{i^*}$, then *abort*, algorithm $\mathcal{B}$ aborts and outputs a uniform bit.
  2. Otherwise, algorithm $\mathcal{B}$ finds entry j in L such that $\text{id} = \text{id}_j$ and return $\text{sk}_{\text{id}} = (\mathbf{s}_j, \mathbf{e}_j)$.

  Along the lines of the proof in [GPV08, Lemma 5.2], the distribution of $\mathbf{e}_j$ in $\mathcal{D}_{\mathsf{sim}}$ (which is $\mathbf{e}_j \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sqrt{m}}$ conditioned on $\mathbf{A}\mathbf{e}_j = \mathbf{H}_j \mathbf{s}_j + \sum_{i \in [d]} \mathbf{B}_i \mathbf{s}_i \pmod{q}$) is statistically close to the distribution of $\mathbf{e}_j$ in $\mathcal{D}_{\mathsf{real}}$ (output by the algorithm SamplePre). Also note that $\mathbf{s}_i$ is sampled identically in both distributions.

- **Challenge:** Eventually $\mathcal{A}$ returns two tuples of identities and messages $(\text{id}_0, \mathfrak{m}_0)$ and $(\text{id}_1, \mathfrak{m}_1)$ on which to be challenged, and $\mathcal{B}$ does the following:

  1. It computes $H(\text{id}_0) = \mathbf{H}$ as above. If this is not the i^*-th entry query to the random oracle, $\mathcal{B}$ aborts and outputs a uniform bit.

2. If $\mathcal{B}$ does not abort, observe that $H(\mathrm{id}_0) = \mathbf{H}^*$. The algorithm $\mathcal{B}$ parses $\mathbf{f}$ from the LWE challenge as $(\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_d) \in \mathbb{Z}_q^m \times (\mathbb{Z}_q^\ell)^d$ and outputs $(\mathbf{f}_0, \mathbf{f}_1 + \mathfrak{m}_0 \frac{q}{2\ell d} \cdot \mathbf{1}, \ldots, \mathbf{f}_d + \mathfrak{m}_0 \frac{q}{2\ell d} \cdot \mathbf{1})$ as the challenge ciphertext.

- **Output:** If $\mathcal{A}$ at the end of the simulation outputs a bit b, $\mathcal{B}$ outputs the same bit b.

It is easy to see from the construction that in the challenge phase, if $\mathcal{B}$ is given an LWE instance, i.e., $\mathbf{f} = \mathbf{E}^\intercal \mathbf{r} + \boldsymbol{\chi}$ for some random $\mathbf{r} \in \mathbb{Z}_q^n$ and error term $\boldsymbol{\chi} \leftarrow \overline{\Psi}_\alpha^{m+\ell d}$, then

$$(\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_d) = \left(\mathbf{A}^\intercal \mathbf{r} + \boldsymbol{\chi}_1, \left\{ \left(2^{i-1} \cdot \mathbf{H}^* + \mathbf{B}_i\right)^\intercal \mathbf{r} + \boldsymbol{\chi}_i + \mathfrak{m}_0 \cdot \frac{q}{2\ell d} \cdot \mathbf{1} \right\}_{i \in [d]} \right),$$

(where $\boldsymbol{\chi} = (\boldsymbol{\chi}_0, \boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_d)$) is a well-formed ciphertext corresponding to id_0 and therefore, the challenge is distributed as in Expt_0.

Also, in a rather straightforward manner, if $\mathcal{B}$ is given a random tuple, i.e., $\mathbf{f}$ is uniformly chosen from $\mathbb{Z}_q^{m+\ell d}$, then the challenge ciphertext $(\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_d)$ is identically distributed to the challenge in Expt_1.

Thus, to complete the proof of Claim 4.7, it suffices to bound the probability of $\mathcal{B}$ aborting the simulation. It follows in a straightforward manner that the probability that $\mathcal{B}$ does not abort during the simulation is at least $1/(Q_H + 1)$ if the view of the adversary is independent of i^*. To see this, consider an identical game where $\mathcal{B}$ does not choose an index i^* and hence does not embed the LWE challenge. Such a game can can answer *all* hash and secret key queries correctly. It is easy to see that as far as the adversary is concerned, the two simulations are identical (so long as it does not abort). Therefore, the index i^* is hidden perfectly from the adversary $\mathcal{A}$.

Finally, note that as the challenge ciphertext is distributed correctly in each of Expt_b, $b \in \{0, 1\}$, the advantage of $\mathcal{B}$ is identical to that of $\mathcal{A}$ conditioned on $\mathcal{B}$ not aborting. As the view of $\mathcal{A}$ is independent of the abort condition, this completes the proof.

∎

Claim 4.8. *Based on the LWE assumption, it holds that* $|\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \leq \mathrm{negl}(\lambda)$.

The proof of the above claim is identical to the proof of Claim 4.7 except that the challenger uses $(\mathrm{id}_1, \mathfrak{m}_1)$ to embed the LWE challenge. To complete the proof of the theorem,

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{DP}}_{\mathcal{IBE}_{\mathsf{LWE1}},\mathcal{A}}(\lambda) \\
&\qquad = \left| \Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{DP},\mathcal{IBE}_{\mathsf{LWE1}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{DP},\mathcal{IBE}_{\mathsf{LWE1}},\mathcal{A}}(\lambda) = 1\right] \right| \\
&\qquad = |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \\
&\qquad = |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]| + |\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \\
&\qquad \leq \mathrm{negl}(\lambda), \qquad\qquad\qquad\qquad\qquad\qquad \text{(from Claims 4.7 and 4.8)}
\end{aligned}$$

as required.

∎

4.2.2 Proof of Function Privacy

Lemma 4.9. *The scheme $\mathcal{IBE}_{\mathsf{LWE1}}$ is statistically function private in the random-oracle model for:*

1. *(T, k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq n \log q + \omega(\log \lambda)$.*

2. $(k_1, \ldots, k_T)$*-sources for any* $T = \mathrm{poly}(\lambda)$ *and* $(k_1, \ldots, k_T)$ *such that* $k_i \geq i \cdot n \log q + \omega(\log \lambda)$ *for every* $i \in [T]$.

Proof. Let $X \in \{(T,k)\text{-block}, (k_1, \ldots, k_T)\}$, and let $\mathcal{A}$ be a computationally unbounded X-source function-privacy adversary that makes a polynomial number $Q_{\mathsf{RoR}} = Q_{\mathsf{RoR}}(\lambda)$ of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{LWE1}},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{LWE1}},\mathcal{A}}$ (we refer the reader to Definition 3.3 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$, respectively.

We first observe that since the hash function $H : \mathcal{ID}_\lambda \to \mathbb{Z}_q^{n \times \ell}$ is modeled a a random oracle, we can restrict ourselves to the above distributions conditioned on the event in which for any two distinct identities $\mathrm{id}_1, \mathrm{id}_2 \in \mathcal{ID}_\lambda$ the matrix $H(\mathrm{id}_1) - H(\mathrm{id}_2)$ is of rank n. Specifically, we show that this event (denote $\mathsf{FullRankDiff}$) occurs with an overwhelming probability over the uniform choice of the function H. Indeed, for a uniformly sampled matrix $\mathbf{A} \leftarrow\in \mathbb{Z}_q^{n \times \ell}$ Lemma 2.12 states that $\Pr[\mathsf{Rk}(\mathbf{A}) = n] > 1 - 2/q^{\ell - n + 1}$. Therefore, our choice of $\ell \geq n + \frac{2 \log |\mathcal{ID}_\lambda| + \log n + \omega(\log \lambda)}{\log q}$ implies that

$$\begin{aligned}\Pr_H[\exists\ \mathrm{id}_1 \neq \mathrm{id}_2 : \mathsf{rank}(H(\mathrm{id}_1) - H(\mathrm{id}_2)) < n] &\leq \frac{|\mathcal{ID}_\lambda|^2 \cdot 2}{q^{\ell - n + 1}} \\ &\leq 2^{-\omega(\log \lambda)}.\end{aligned}$$

The event $\mathsf{FullRankDiff}$ guarantees, in particular, that H is injective on the identity space. Thus, for any X-source $\boldsymbol{ID} = (ID_1, \ldots, ID_T)$ over $(\mathcal{ID}_\lambda)^T$ it holds that $H(\boldsymbol{ID}) \overset{\mathsf{def}}{=} (H(ID_1), \ldots, H(ID_T))$ is an X-source over $(\mathbb{Z}_q^{n \times \ell})^T$. In addition, the event $\mathsf{FullRankDiff}$ implies that the collection of functions $\{f_{\mathbf{s}} : \mathbb{Z}_q^{n \times \ell} \to \mathbb{Z}_q^n\}_{\mathbf{s} \in \mathbb{Z}_q^\ell}$ defined by $f_{\mathbf{s}}(\mathbf{A}) = \mathbf{A}\mathbf{s}$ is universal over the set $\{H(id) : \mathrm{id} \in \mathcal{ID}_\lambda\}$.[14]

From this point on we fix a function H such that the event $\mathsf{FullRankDiff}$ occurs. In addition, we also fix the public parameters pp, and the master secret key msk, of the scheme, and show that the two distributions $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$ are statistically close for any such H, pp, and msk. Next, as the adversary $\mathcal{A}$ is computationally unbounded, we assume without loss of generality that $\mathcal{A}$ does not query the $\mathsf{KeyGen}(\mathrm{msk}, \cdot)$ oracle. In addition, as discussed in Section 3.1, we can assume that $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once.

Denote by $\boldsymbol{ID} = (ID_1, \ldots, ID_T)$ the random variable corresponding to the X-source with which $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. Having already fixed H, pp, and msk, we can assume that

$$\mathsf{View}_{\mathsf{mode}} = ((\mathbf{s}_1, \mathbf{H}_1 \mathbf{s}_1), \ldots, (\mathbf{s}_T, \mathbf{H}_T \mathbf{s}_T))$$

for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, where $(\mathrm{id}_1, \ldots, \mathrm{id}_T) \leftarrow (ID_1, \ldots, ID_T)$ for $\mathsf{mode} = \mathsf{real}$, $(\mathrm{id}_1, \ldots, \mathrm{id}_T)$ is uniformly distributed over $(\mathcal{ID}_\lambda)^T$ for $\mathsf{mode} = \mathsf{rand}$, $\mathbf{H}_i = H(\mathrm{id}_i)$ for every $i \in [T]$, and $\mathbf{s}_i \leftarrow \mathbb{Z}_q^\ell$ for every $i \in [T]$. For $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ we prove that the distribution $\mathsf{View}_{\mathsf{mode}}$ is statistically-close to uniform.

As discussed above, $(H(ID_1), \ldots, H(ID_T))$ is an X-source, and the collection of functions $\{f_{\mathbf{s}} : \mathbb{Z}_q^{n \times \ell} \to \mathbb{Z}_q^n\}_{\mathbf{s} \in \mathbb{Z}_q^\ell}$ defined by $f_{\mathbf{s}}(\mathbf{A}) = \mathbf{A}\mathbf{s}$ is universal over the set $\{H(\mathrm{id}) : \mathrm{id} \in \mathcal{ID}_\lambda\}$. This enables us to directly apply Lemma 2.3 (in case $\boldsymbol{ID}$ is a (T,k)-block-source) and Lemma 2.4 (in case $\boldsymbol{ID}$ is a $(k_1, \ldots, k_T)$-source), implying that the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ. The same clearly holds also for $\mathsf{View}_{\mathsf{rand}}$, as the uniform distribution over $(\mathcal{ID}_\lambda)^T$ is, in particular, a (T,k)-block-source and a $(k_1, \ldots, k_T)$-source. ∎

[14] For any distinct id_1 and id_2, the fact that the matrix $H(\mathrm{id}_1) - H(\mathrm{id}_2)$ is of rank n implies that the kernel of $H(\mathrm{id}_1) - H(\mathrm{id}_2)$ is of dimension $\ell - n$. Therefore, $\Pr_{\mathbf{s} \leftarrow \mathbb{Z}_q^\ell}[H(\mathrm{id}_1)\mathbf{s} = H(\mathrm{id}_2)\mathbf{s}] = \frac{q^{\ell - n}}{q^\ell} = \frac{1}{q^n}$.

5 Function-Private Schemes in the Standard Model

5.1 A Selectively-Secure DLIN-Based Scheme

In this section we present an IBE scheme based on the DLIN assumption in the standard model. For emphasizing the main ideas underlying our approach, we present here a *selectively* data private scheme, and refer the reader to Section 5.3 for its extension to *full* data privacy. The scheme is based on the DLIN-based IBE of Kurosawa and Phong [KP11], which is an adaptation of the LWE-based IBE of Agrawal, Boneh and Boyen [ABB10] to bilinear groups. The scheme is obtained by applying our "extract-augment-combine" approach, as discussed in Section 1.1. The scheme is described below, and its proofs of data privacy and function privacy are presented in Sections 5.1.1 and 5.1.2, respectively.

The scheme. Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^λ, and outputs $(\mathbb{G}, \mathbb{G}_T, p, g, \hat{e})$ where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order p, $\mathbb{G}$ is generated by g, p is a λ-bit prime number, and $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear map. The scheme $\mathcal{IBE}_{\mathsf{DLIN1}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is parameterized by the security parameter $\lambda \in \mathbb{N}$. For any such $\lambda \in \mathbb{N}$, the scheme has parameters $m \geq 3$ and $\ell \geq 2$, identity space $\mathcal{ID}_\lambda = \mathbb{Z}_p^\ell$, and message space $\mathcal{M}_\lambda = \mathbb{G}_T$.

- **Setup:** On input 1^λ the setup algorithm samples $(\mathbb{G}, \mathbb{G}_T, p, g, \hat{e}) \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{B} \leftarrow \mathbb{Z}_p^{2\times m}$, and $\mathbf{u} \leftarrow \mathbb{Z}_p^2$. It outputs $\mathrm{pp} = \left(g, g^{\mathbf{A}_0}, g^{\mathbf{A}_1}, \ldots, g^{\mathbf{A}_\ell}, \mathbf{B}, g^{\mathbf{u}}\right)$ and $\mathrm{msk} = (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{u})$.

- **Key generation:** On input the master secret key msk and an identity $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_\ell) \in \mathbb{Z}_p^\ell$, the algorithm samples $s_1, \ldots, s_\ell \leftarrow \mathbb{Z}_p$ and computes
$$\mathbf{F}_{\mathbf{id},(s_1,\ldots,s_\ell)} = \left[\mathbf{A}_0 \middle| \left(\sum_{i\in[\ell]} s_i \mathbf{A}_i\right) + \left(\sum_{i\in[\ell]} s_i \cdot \mathrm{id}_i\right)\mathbf{B}\right] \in \mathbb{Z}_p^{2\times 2m}.$$
Then, it samples $\mathbf{v} \leftarrow \mathbb{Z}_p^{2m}$ such that $\mathbf{F}_{\mathbf{id},(s_1,\ldots,s_\ell)} \cdot \mathbf{v} = \mathbf{u} \pmod p$ and sets $\mathbf{z} = g^{\mathbf{v}} \in \mathbb{G}^{2m}$. It outputs $\mathrm{sk}_{\mathbf{id}} = (s_1, \ldots, s_\ell, \mathbf{z})$.

- **Encryption:** On input the public parameters pp, an identity $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_\ell) \in \mathbb{Z}_p^\ell$, and a message $\mathfrak{m} \in \mathbb{G}_T$, the algorithm samples $\mathbf{r} \leftarrow \mathbb{Z}_p^2$. It sets $\mathbf{c}_0^\mathsf{T} = g^{\mathbf{r}^\mathsf{T}\mathbf{A}_0} \in \mathbb{G}^{1\times m}$, $\mathbf{c}_i^\mathsf{T} = g^{\mathbf{r}^\mathsf{T}[\mathbf{A}_i + \mathrm{id}_i \mathbf{B}]} \in \mathbb{G}^{1\times m}$ for all $i \in [\ell]$, $c_{\ell+1} = \hat{e}(g,g)^{\mathbf{r}^\mathsf{T}\mathbf{u}} \cdot \mathfrak{m} \in \mathbb{G}_T$, and outputs $(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_\ell, c_{\ell+1}) \in \mathbb{G}^{(\ell+1)m} \times \mathbb{G}_T$.

- **Decryption:** On input a ciphertext $c = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_\ell, c_{\ell+1})$ and a secret key $\mathrm{sk} = (s_1, \ldots, s_\ell, \mathbf{z})$, the decryption algorithm outputs
$$\mathfrak{m} = c_{\ell+1} \cdot \hat{e}\left(\begin{bmatrix} \mathbf{c}_0 \\ \prod_{i\in[\ell]} \mathbf{c}_i^{s_i} \end{bmatrix}, \begin{matrix} | \\ \mathbf{z} \\ | \end{matrix}\right)^{-1}.$$

Correctness. Note that
$$\mathbf{d}^\mathsf{T} = \left[\mathbf{c}_0^\mathsf{T} \middle| \prod_{i\in[\ell]} (\mathbf{c}_i^\mathsf{T})^{s_i}\right] = g^{\mathbf{r}^\mathsf{T}\left[\mathbf{A}_0 \middle| \sum_{i\in[\ell]} s_i \mathbf{A}_i + \left(\sum_{i\in[\ell]} s_i \cdot \mathrm{id}_i\right)\mathbf{B}\right]} = g^{\mathbf{r}^\mathsf{T}\mathbf{F}_{\mathbf{id},(s_1,\ldots,s_\ell)}}.$$

We have $\hat{e}(\mathbf{d}, \mathbf{z}) = \hat{e}(g,g)^{\mathbf{r}^\mathsf{T}\mathbf{F}_{\mathbf{id},(s_1,\ldots,s_\ell)}\cdot\mathbf{v}} = \hat{e}(g,g)^{\mathbf{r}^\mathsf{T}\mathbf{u}}$. Therefore, dividing $c_{\ell+1}$ by $\hat{e}(\mathbf{d}, \mathbf{z})$ eliminates the term $\hat{e}(g,g)^{\mathbf{r}^\mathsf{T}\mathbf{u}}$ which recovers $\mathfrak{m}$ correctly.

Security. In Sections 5.1.1 and 5.1.2 we prove the following theorem:

Theorem 5.1. *The scheme $\mathcal{IBE}_{\mathsf{DLIN1}}$ is selectively data private based on the DLIN assumption, and is function private for:*

1. *(T, k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq \lambda + \omega(\log \lambda)$.*
2. *$(k_1, \ldots, k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1, \ldots, k_T)$ such that $k_i \geq i \cdot \lambda + \omega(\log \lambda)$ for every $i \in [T]$.*

Proof overview. The function privacy of the scheme follows quite naturally from our "extract" step, as discussed in Section 1.1. To prove selective data privacy under the DLIN assumption, given the challenge identity $\mathbf{id}^*$, we set up the public parameters $\{g^{\mathbf{A}_i}\}_{i\in[\ell]}$, $\mathbf{B}$, and $g^{\mathbf{u}}$ such that the matrix $\mathbf{G}_{\mathbf{id},\mathbf{s}} \stackrel{\mathsf{def}}{=} \left[\left(\sum_{i\in[\ell]} s_i \mathbf{A}_i\right) + \left(\sum_{i\in[\ell]} s_i \cdot \mathrm{id}_i\right)\mathbf{B}\right]$ is equipped with a 'punctured' trapdoor. This trapdoor allows us to sample a vector such that $\mathbf{F}_{\mathbf{id},\mathbf{s}} \cdot \mathbf{v} = \mathbf{u}$ whenever $\mathbf{G}_{\mathbf{id},\mathbf{s}}$ contains a non-zero scalar multiple of $\mathbf{B}$. This occurs whenever $\sum_{i\in[\ell]} s_i(\mathrm{id}_i - \mathrm{id}_i^*) \neq 0$. Thus, with all but a negligible probability, we can simulate the adversary's key-generation queries with specially chosen matrices as above.

To embed the DLIN challenge, the first two rows of the DLIN challenge is used to constitute the public parameter $g^{\mathbf{A}_0}$. The third row is either linearly dependent on the first two rows or chosen uniformly at random and independently. This third row of the challenge is embedded into the augmented challenge ciphertext that is either well-formed or uniform and independent of the adversary's view depending on the DLIN challenge. This is done by choosing secret matrices $\mathbf{R}_i^*$ and having $\mathbf{A}_i = \mathbf{A}_0\mathbf{R}_i^* - \mathrm{id}_i^*\mathbf{B}$. This generalizes the ideas of [ABB10, KP11] to fit our "extract-augment-combine" approach and provide function privacy.

5.1.1 Proof of (Selective) Data Privacy

Lemma 5.2. *The scheme $\mathcal{IBE}_{\mathsf{DLIN1}}$ is selectively-secure data private based on the DLIN assumption in the standard model.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary. We consider a series of experiments that interacts with the adversary as follows. Experiment Expt_0 is identical to $\mathsf{Expt}^{(0)}_{\mathsf{sDP},\mathcal{IBE}_{\mathrm{DLIN1}},\mathcal{A}}$ in Definition 2.8. Experiment Expt_1 is identical to Expt_0 except in step (3). The experiment replaces a well-constructed challenge ciphertext with independently and uniformly sampled $(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_\ell, c_{\ell+1}) \leftarrow \mathbb{G}^{(\ell+1)m} \times \mathbb{G}_{\mathrm{T}}$. Experiment Expt_2 is identical to $\mathsf{Expt}^{(1)}_{\mathsf{sDP},\mathcal{IBE}_{\mathsf{DLIN1}},\mathcal{A}}$ in Definition 2.8. Now we can state the following claim.

Claim 5.3. *Based on the DLIN assumption, it holds that* $|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]| \leq \mathrm{negl}(\lambda)$.

Proof. Consider a probabilistic polynomial time adversary $\mathcal{A}$ in experiment Expt_j for $j \in \{0, 1\}$ such that

$$|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]| > \epsilon(\lambda),$$

for some non-negligible $\epsilon(\lambda)$. We construct an algorithm $\mathcal{B}$ that given a DLIN challenge $\left(g, g^{\mathbf{A}}\right)$ where $\mathbf{A} \leftarrow \mathbb{Z}_p^{3\times m}$, algorithm $\mathcal{B}$ simulates the distinguisher $\mathcal{A}$ to output 0 if $\mathsf{Rk}(\mathbf{A}) = 2$ and 1 if $\mathsf{Rk}(\mathbf{A}) = 3$ with non-negligible advantage $\epsilon'(\lambda) \geq \epsilon(\lambda) - \mathrm{negl}(\lambda)$.

- **Key generation:** Given two challenge identities $(\mathbf{id}_0^*, \mathbf{id}_1^*)$ from the adversary $\mathcal{A}$, the algorithm $\mathcal{B}$ sets $\mathbf{id}^* = \mathbf{id}_0^*$ parsed as $(\mathrm{id}_1^*, \ldots, \mathrm{id}_\ell^*)$. Next, given the DLIN challenge $(g, g^{\mathbf{A}})$, $\mathcal{B}$ sets up pp as follows. $\mathbf{A}_0$ is the first two rows of $\mathbf{A}$. Algorithm $\mathcal{B}$ samples a full-rank $\mathbf{B} \leftarrow \mathbb{Z}_p^{2\times m}$ and $\mathbf{R}_i^* \leftarrow \mathbb{Z}_p^{m\times m}$ for $i \in [\ell]$. It sets
$$\mathbf{A}_i = \mathbf{A}_0\mathbf{R}_i^* - \mathrm{id}_i^*\mathbf{B}.$$
Observe that $g^{\mathbf{A}_i}$ can be computed from $g^{\mathbf{A}_0}$ given $\mathbf{R}_i^*$, id_i^*, and $\mathbf{B}$. Finally, $\mathcal{B}$ chooses a random $\mathbf{v}^* \leftarrow \mathbb{Z}_p^{2m}$ and sets $\mathbf{u} = \left[\mathbf{A}_0 \,\middle|\, \sum_{i\in[\ell]} \mathbf{A}_0\mathbf{R}_i^*\right] \mathbf{v}^* \in \mathbb{Z}_p^2$. Observe that $g^{\mathbf{u}}$ can be computed from $g^{\mathbf{A}_0}$ and $\mathbf{R}_i^*$.

- **Secret key queries:** On query $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_\ell)$, it samples random $s_1, \ldots, s_\ell \leftarrow \mathbb{Z}_p$. Let
$$\delta = \sum_{i\in[\ell]} s_i(\mathrm{id}_i - \mathrm{id}_i^*).$$
If $\delta = 0$, $\mathcal{B}$ aborts and outputs a uniform bit. Otherwise, it chooses random $\mathbf{w} \leftarrow \mathbb{Z}_p^m$ and a random $\mathbf{x}$ in $\mathbb{Z}_p^m$ such that
$$\delta\,\mathbf{B}\mathbf{x} = -\mathbf{A}_0\mathbf{w} + \mathbf{u}.$$
It is easy to compute $g^{\mathbf{x}}$ given $g^{\mathbf{A}}$, $g^{\mathbf{u}}$, and $\mathbf{B}$.

  Let
$$\mathbf{v} = \begin{bmatrix} \mathbf{w} - \left(\sum_{i\in[\ell]} s_i\mathbf{R}_i^*\right)\mathbf{x} \\ \mathbf{x} \end{bmatrix}. \tag{5.1}$$
It is easy to compute $g^{\mathbf{v}}$ given $\{s_i\}_{i\in[\ell]}$, $\mathbf{R}_i^*$, $g^{\mathbf{w}}$, and $g^{\mathbf{x}}$. Observe that:
$$\begin{aligned}
\mathbf{F}_{\mathbf{id},(s_1,\ldots,s_\ell)} \cdot \mathbf{v} &= \left[\mathbf{A}_0 \,\middle|\, \left(\sum_{i\in[\ell]} s_i\mathbf{A}_i\right) + \left(\sum_{i\in[\ell]} s_i\mathrm{id}_i\right)\mathbf{B}\right]\mathbf{v} \\
&= \left[\mathbf{A}_0 \,\middle|\, \left(\sum_{i\in[\ell]} s_i\mathbf{A}_0\mathbf{R}_i^*\right) - \left(\sum_{i\in[\ell]} s_i\mathrm{id}_i^*\right)\mathbf{B} + \left(\sum_{i\in[\ell]} s_i\mathrm{id}_i\right)\mathbf{B}\right]\mathbf{v} \\
&= \left[\mathbf{A}_0 \,\middle|\, \mathbf{A}_0\left(\sum_{i\in[\ell]} s_i\mathbf{R}_i^*\right) + \delta\,\mathbf{B}\right]\begin{bmatrix} \mathbf{w} - \left(\sum_{i\in[\ell]} s_i\mathbf{R}_i^*\right)\mathbf{x} \\ \mathbf{x} \end{bmatrix} \\
&= \mathbf{A}_0\mathbf{w} - \mathbf{A}_0\left(\sum_{i\in[\ell]} s_i\mathbf{R}_i^*\right)\mathbf{x} + \mathbf{A}_0\left(\sum_{i\in[\ell]} s_i\mathbf{R}_i^*\right)\mathbf{x} + \delta\,\mathbf{B}\mathbf{x} = \mathbf{u}.
\end{aligned}$$
To answer the secret key query, $\mathcal{B}$ outputs $(s_1, \ldots, s_\ell, \mathbf{z} = g^{\mathbf{v}})$.

- **Challenge query:** On query $\mathbf{id}^* = (\mathrm{id}_1^*, \ldots, \mathrm{id}_\ell^*)$, given the message $\mathfrak{m}_0^*$, the algorithm $\mathcal{B}$ proceeds as follows. Let $[-\mathbf{y}^\intercal-] \in \mathbb{Z}_p^{1\times m}$ denote the third row of $\mathbf{A}$. The challenge encryption is constructed as follows:
$$\left((\mathbf{c}_0^*)^\intercal, (\mathbf{c}_1^*)^\intercal, \ldots, (\mathbf{c}_\ell^*)^\intercal, c_{\ell+1}^*\right) = \left(g^{\mathbf{y}^\intercal}, g^{\mathbf{y}^\intercal\mathbf{R}_1^*}, \ldots, g^{\mathbf{y}^\intercal\mathbf{R}_\ell^*}, \hat{e}(g,g)^{\left[\mathbf{y}^\intercal \,\middle|\, \sum_{i\in[\ell]}\mathbf{y}^\intercal\mathbf{R}_i^*\right]\mathbf{v}^*} \cdot \mathfrak{m}_0^*\right).$$

We argue that the public parameters are distributed statistically close to the real distribution. We note that the matrices $\mathbf{R}_i^*$ for $i \in [\ell]$ are used to construct the public parameters, answer secret key queries, and construct the challenge ciphertext. Below we show how the secret key queries are distributed *identically* to the real scheme, and therefore independent of $\mathbf{R}_i^*$. Next, form the extended leftover hash lemma (cf. Lemma 2.5) by setting $k = \ell m$ we observe that the two distributions

$$(\mathbf{A}_0, \mathbf{A}_0 \cdot [\mathbf{R}_1^* | \cdots | \mathbf{R}_\ell^*], [\mathbf{R}_1^* | \cdots | \mathbf{R}_\ell^*]^\intercal \mathbf{y}) \quad \text{and} \quad \left(\mathbf{A}_0, \left[\widetilde{\mathbf{A}}_1 \middle| \cdots \middle| \widetilde{\mathbf{A}}_\ell\right], [\mathbf{R}_1^* | \cdots | \mathbf{R}_\ell^*]^\intercal \mathbf{y}\right)$$

are statistically close, where $\widetilde{\mathbf{A}}_i$ for $i \in [\ell]$ are matrices chosen independently and uniformly from $\mathbb{Z}_p^{2\times m}$. Observe that the third component is the challenge ciphertext. Thus, even given the (specially constructed) challenge ciphertext, the second component is statistically close to uniform matrices over $\mathbb{Z}_p^{2\times m}$. Subtracting $[\mathrm{id}_1^* \mathbf{B} | \cdots | \mathrm{id}_\ell^* \mathbf{B}]$ still keeps it uniform. Thus, the parameters $\left(\mathbf{A}, \{\mathbf{A}_i\}_{i\in[\ell]}, \mathbf{B}\right)$ are distributed statistically close to the correpsonding parameters in the real distribution.

Next, we argue that the answers to secret key queries are distributed correctly. If the simluation doesn't abort, observe that $s_1, \ldots, s_\ell$ are distributed as in the real scheme. We show that $\mathbf{v}$ (and hence $\mathbf{z}$) is distributed identically to the real scheme. Observe that $\mathbf{v}$ in the real scheme satisfies $\mathbf{F}_{\mathbf{id},(s_1,\ldots,s_\ell)} \mathbf{v} = \mathbf{u} \pmod q$. Therefore $\mathbf{v}$ is chosen from a subspace of dimension $2m - 2$ from the constraints of the above equation. In the simulation, $\boldsymbol{ID}$ is chosen uniformly from $\mathbb{Z}_p^m$ and $\mathbf{x}$ comes from a subspace of dimension $m-2$ from the constraints in equation (5.1). Therefore, $\mathbf{v}$ comes from a subspace of dimension $m + (m-2) = 2m - 2$ as required.

And finally, we argue that if $\mathsf{Rk}(\mathbf{A}) = 2$, then the challenge ciphertext is well-formed and if $\mathsf{Rk}(\mathbf{A}) = 3$, then the challenge ciphertext is distributed uniformly over $\mathbb{G}^{(\ell+1)m} \times \mathbb{G}_{\mathrm{T}}$ and independently of $\mathcal{A}$'s view.

- **Case 1: $\mathsf{Rk}(\mathbf{A}) = 2$.** We have that $\mathbf{y}^\intercal = \mathbf{r}^\intercal \mathbf{A}_0$ for some $\mathbf{r} \in \mathbb{Z}_p^2$. Therefore, we have the following:

  $$g^{\mathbf{y}^\intercal} = g^{\mathbf{r}^\intercal \mathbf{A}_0}$$
  $$g^{\mathbf{y}^\intercal \mathbf{R}_i^*} = g^{\mathbf{r}^\intercal \mathbf{A}_0 \mathbf{R}_i^*} = g^{\mathbf{r}^\intercal [\mathbf{A}_i + \mathrm{id}_i^* \mathbf{B}]} \text{ for } i \in [\ell]$$
  $$\hat{e}(g,g)^{\left[\mathbf{y}^\intercal \,|\, \sum_{i\in[\ell]} \mathbf{y}^\intercal \mathbf{R}_i^*\right]\mathbf{v}^*} = \hat{e}(g,g)^{\mathbf{r}^\intercal \left[\mathbf{A}_0 \,|\, \sum_{i\in[\ell]} \mathbf{A}_0 \mathbf{R}_i^*\right]\mathbf{v}^*} = \hat{e}(g,g)^{\mathbf{r}^\intercal \mathbf{u}}.$$

  Note that $\mathbf{r}$ is distributed uniformly in $\mathbb{Z}_p^2$ by definition. Thus, the ciphertext is well-formed.

- **Case 2: $\mathsf{Rk}(\mathbf{A}) = 3$.** We have that $\mathbf{y}$ is uniform in $\mathbb{Z}_p^m$ and independent of $\mathbf{A}_0$. We consider $\mathcal{A}$'s view and argue that the challenge ciphertext is distributed uniformly over $(\mathbb{G}^m)^{\ell+1} \times \mathbb{G}_{\mathrm{T}}$ and independent of $\mathcal{A}$'s view. It suffices to argue the distribution of the ciphertext in an information-theoretic sense (against a computationally unbounded adversary). $\mathcal{A}$'s view in the simulation comprises the public parameters $(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{B}, \mathbf{u})$ and the challenge ciphertext $\left(\mathbf{c}_0^*, \mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*, c_{\ell+1}^*\right)$. As $\mathcal{A}$ is unbounded, the secret key queries do not reveal any extra information and can be simulated by an unbounded adversary itself. Let $\mathbf{U}_i^* = \mathbf{A}_0 \mathbf{R}_i^*$. First note that as $\mathbf{y}$ is uniform over $\mathbb{Z}_p$, so is $\mathbf{c}_0^*$. Observe that for every $i \in [\ell]$, and for every possible $\mathbf{c}_i^* = g^{\mathbf{d}_i^*}$ where $\mathbf{d}_i^* \in \mathbb{Z}_p^m$ the number of solutions $\mathbf{R}_i^*$ such that

  $$\begin{bmatrix} \mathbf{A}_0 \\ \mathbf{y}^\intercal \end{bmatrix} \cdot \mathbf{R}_i^* = \begin{bmatrix} \mathbf{A}_0 \mathbf{R}_i^* \\ \mathbf{y}^\intercal \mathbf{R}_i^* \end{bmatrix} = \begin{bmatrix} \mathbf{U}_i^* \\ \mathbf{d}_i^{*\intercal} \end{bmatrix}$$

  is the same. Thus, even given $\mathbf{U}_i^*$ (which can be computed from $\mathbf{A}_i$, $\mathbf{B}$, and $\mathbf{id}^*$) as $\mathbf{R}_i^*$ is chosen uniformly from $\mathbb{Z}_p^{m\times m}$ each $\mathbf{c}_i^*$ is distributed uniformly over $\mathbb{G}^m$ for every $i \in [\ell]$.

Next, observe that $\mathbf{v}^*$ has min-entropy $2m \log p$ and given $\mathbf{u}$, from Lemma 2.1 with probability at least $1 - \epsilon$ over choices of $\mathbf{u}$, $\mathbf{v}^*$ still has min-entropy $(2m-2)\log p - \log(1/\epsilon)$ for any negligible $\epsilon = \epsilon(\lambda)$. Next, we consider $d_{\ell+1} = \left[\mathbf{y}^\intercal \,\middle|\, \sum_{i\in[\ell]} \mathbf{d}_i^{*\intercal}\right] \mathbf{v}^*$ which can be written as $\mathbf{f}^\intercal \mathbf{v}^*$ for a uniformly distributed vector $\mathbf{f}$ in $\mathbb{Z}_p^m$. As $d_{\ell+1}$ is of length $\log p$ bits, the vector $\mathbf{v}^*$ has sufficient min-entropy (more precisely, at least $\log p + \omega(\log \lambda)$ bits) so that $\mathbf{f}$ when applied extracts from it. Therefore, we have $(\mathbf{f}^\intercal, \mathbf{f}^\intercal \mathbf{v}^*) \approx (\mathbf{f}^\intercal, r)$ where $\mathbf{f}$ is uniform in $\mathbb{Z}_p^{2m}$ and r is uniform in $\mathbb{Z}_p$. This implies, in particular, that the last component of the ciphertext, $\hat{e}(g,g)^{d_{\ell+1}} \cdot \mathfrak{m}_0^*$, is distributed uniformly over $\mathbb{G}_{\mathrm{T}}$.

To complete the proof of Claim 5.3, it suffices to bound the probability of $\mathcal{B}$ aborting during the simulation. The probability that $\mathcal{B}$ aborts during the simulation is the probability of the following event: given fixed values $\Delta\mathrm{id}_1, \ldots, \Delta\mathrm{id}_\ell \in \mathbb{Z}_p$, over random choices of $s_i \leftarrow \mathbb{Z}_p$, $\delta \stackrel{\mathsf{def}}{=} \sum_{i\in[\ell]} s_i \cdot \Delta\mathrm{id}_i = 0 \pmod{p}$. This is exactly $1/p$ as we can fix all s_i except s_ℓ and observe that there is exactly one choice of s_ℓ such that $\delta = 0$.

Therefore, the advantage of $\mathcal{B}$ is:

$$\begin{aligned}
\epsilon' &= \left|\Pr\left[\mathcal{B}\left((g, g^{\mathbf{A}})|_{\mathsf{Rk}(\mathbf{A})=2}\right) = 1\right] - \Pr\left[\mathcal{B}\left((g, g^{\mathbf{A}})|_{\mathsf{Rk}(\mathbf{A})=3}\right) = 1\right]\right| \\
&= \Bigg|\Pr\left[\mathcal{B}\left((g, g^{\mathbf{A}})|_{\mathsf{Rk}(\mathbf{A})=2}\right) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\,\overline{\mathsf{Abort}}\,\right] + \frac{1}{2} \cdot \Pr[\mathsf{Abort}] \\
&\qquad\quad - \left(\Pr\left[\mathcal{B}\left((g, g^{\mathbf{A}})|_{\mathsf{Rk}(\mathbf{A})=3}\right) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\,\overline{\mathsf{Abort}}\,\right] + \frac{1}{2} \cdot \Pr[\mathsf{Abort}]\right)\Bigg| \\
&= \left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right] - \Pr\left[\mathsf{Expt}_1(\lambda) = 1 \,\middle|\, \overline{\mathsf{Abort}}\right] \cdot \Pr\left[\overline{\mathsf{Abort}}\right]\right| \\
&= |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_0 = 1 \,|\, \mathsf{Abort}] \cdot \Pr[\mathsf{Abort}] \\
&\qquad\quad - (\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1 \,|\, \mathsf{Abort}] \cdot \Pr[\mathsf{Abort}])| \\
&= |(\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]) - (\Pr[\mathsf{Expt}_0(\lambda) = 1 \,|\, \mathsf{Abort}] \cdot \Pr[\mathsf{Abort}] \\
&\qquad\qquad\qquad\qquad\qquad\qquad - \Pr[\mathsf{Expt}_1(\lambda) = 1 \,|\, \mathsf{Abort}] \cdot \Pr[\mathsf{Abort}])| \\
&\geq |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]| \\
&\qquad\quad - \Pr[\mathsf{Abort}] \cdot |\Pr[\mathsf{Expt}_0(\lambda) = 1 \,|\, \mathsf{Abort}] - \Pr[\mathsf{Expt}_1(\lambda) = 1 \,|\, \mathsf{Abort}]| \\
&\geq \epsilon - 1/p.
\end{aligned}$$

Under the DLIN assumption, ϵ' is negligible, which implies that ϵ is negligible completing the proof of Claim 5.3. ∎

Claim 5.4. *Based on the DLIN assumption,* $|\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \leq \mathrm{negl}(\lambda)$.

The proof of Claim 5.4 is identical to the proof of Claim 5.3. With the above two claims, we now proceed to prove Lemma 5.2.

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{sDP}}_{\mathcal{IBE}_{\mathsf{DLIN1}},\mathcal{A}}(\lambda) \\
&\qquad = \left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{sDP},\mathcal{IBE}_{\mathsf{DLIN1}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{sDP},\mathcal{IBE}_{\mathsf{DLIN1}},\mathcal{A}}(\lambda) = 1\right]\right| \\
&\qquad = |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \\
&\qquad = |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]| + |\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \\
&\qquad \leq \mathrm{negl}(\lambda), \qquad\qquad\qquad\qquad\qquad\qquad \text{(from Claims 5.3 and 5.4)}
\end{aligned}$$

as required. ∎

5.1.2 Proof of Function Privacy

Lemma 5.5. *The scheme $\mathcal{IBE}_{\mathsf{DLIN1}}$ is statistically function private for:*

1. *(T,k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq \lambda + \omega(\log \lambda)$.*
2. *$(k_1,\ldots,k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1,\ldots,k_T)$ such that $k_i \geq i \cdot \lambda + \omega(\log \lambda)$ for every $i \in [T]$.*

Proof. Let $X \in \{(T,k)\text{-block}, (k_1,\ldots,k_T)\}$, and let $\mathcal{A}$ be a computationally unbounded X-source function-privacy adversary that makes a polynomial number $Q = Q(\lambda)$ of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{DLIN1}},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{DLIN1}},\mathcal{A}}$ (we refer the reader to Definition 3.3 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$, respectively.

As the adversary $\mathcal{A}$ is computationally unbounded, we assume without loss of generality that $\mathcal{A}$ does not query the $\mathsf{KeyGen}(\mathrm{msk},\cdot)$ oracle. Indeed, as the public parameters in our scheme uniquely determine the secret key, such queries can be internally simulated by $\mathcal{A}$. Moreover, as discussed in Section 3.1, it suffices to focus on adversaries $\mathcal{A}$ that query the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once. From this point on we fix the public parameters pp chosen by the setup algorithm, and show that the two distributions $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$ are statistically close for any such pp.

Denote by $\boldsymbol{ID} = (ID_1,\ldots,ID_T)$ the random variable corresponding to the X-source with which $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. As $\mathcal{A}$ is computationally unbounded, and having fixed the public parameters, we can in fact assume that

$$\mathsf{View}_{\mathsf{mode}} = \left(\left(s_{1,1},\ldots,s_{1,\ell}, \sum_{j=1}^{\ell} s_{1,j} \cdot \mathrm{id}_{1,j} \right), \ldots, \left(s_{T,1},\ldots,s_{T,\ell}, \sum_{j=1}^{\ell} s_{T,j} \cdot \mathrm{id}_{T,j} \right) \right)$$

for $\mathsf{mode} \in \{\mathsf{real},\mathsf{rand}\}$, where $(\mathbf{id}_1,\ldots,\mathbf{id}_T) \leftarrow (ID_1,\ldots,ID_T)$ for $\mathsf{mode} = \mathsf{real}$, $(\mathbf{id}_1,\ldots,\mathbf{id}_T)$ is uniformly distributed over $(\mathcal{ID}_\lambda)^T$ for $\mathsf{mode} = \mathsf{rand}$, $\mathbf{id}_i = (\mathrm{id}_{i,1},\ldots,\mathrm{id}_{i,\ell}) \in \mathbb{Z}_p^\ell$ for every $i \in [T]$, and $s_{i,j} \leftarrow \mathbb{Z}_p$ for every $i \in [T]$ and $j \in [\ell]$. For $\mathsf{mode} \in \{\mathsf{real},\mathsf{rand}\}$ we prove that the distribution $\mathsf{View}_{\mathsf{mode}}$ is statistically-close to uniform.

Note that the collection of functions $\{f_{s_1,\ldots,s_\ell} : \mathbb{Z}_p^\ell \to \mathbb{Z}_p\}_{s_1,\ldots,s_\ell \in \mathbb{Z}_p}$ defined by $f_{s_1,\ldots,s_\ell}(\mathrm{id}_1,\ldots,\mathrm{id}_\ell) = \sum_{j=1}^{\ell} s_j \cdot \mathrm{id}_j$ is universal. This enables us to directly apply Lemma 2.3 (in case $\boldsymbol{ID}$ is a (T,k)-block-source) and Lemma 2.4 (in case $\boldsymbol{ID}$ is a $(k_1,\ldots,k_T)$-source), implying that the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ. The same clearly holds also for $\mathsf{View}_{\mathsf{rand}}$, as the uniform distribution over $(\mathcal{ID}_\lambda)^T$ is, in particular, a (T,k)-block-source and a $(k_1,\ldots,k_T)$-source. ∎

5.2 A Selectively-Secure LWE-Based Scheme

In this section we present an IBE scheme based on the LWE assumption in the standard model. For emphasizing the main ideas underlying our approach, we present here a *selectively* data private scheme (as in Section 5.1), and note that it can be extended to a *fully* data private one using essentially the same approach as in Section 5.3. The scheme is based on the LWE-based IBE of Agrawal, Boneh and Boyen [ABB10] (referred to as the ABB scheme) by applying our "extract-augment-combine" approach, as discussed in Section 1.1.

Specifically, in the ABB scheme, identities are mapped to matrices, and secret keys are short vectors in the corresponding lattice. In our construction, we use a larger identity space (vectors of ABB identities), and we use elements in $\mathbb{Z}_q$ to extract identities. As in the scheme $\mathcal{IBE}_{\mathsf{LWE1}}$ presented in Section 4.2, we use the bit-splitting approach to ensure that the amount of noise that is added in the "combine" step will allow correct decryption.

However, unlike the scheme $\mathcal{IBE}_{\mathsf{DLIN1}}$ described in Section 5.1, this scheme additionally requires parallel repetition. The field size q in lattice-based constructions is allowed to be a small polynomial in the security parameter, which in our case may lead to a non-negligible probability of one secret key being able to decrypt ciphertexts encrypted for other identities. To fix this, one approach is to make q super-polynomial, but this will require a seemingly stronger LWE assumption. Instead, we do a parallel repetition of μ copies of the ciphertext, which are "bound together" using a public lattice. The scheme is described below, and its proofs of data privacy and function privacy are presented in Sections 5.2.1 and 5.2.2, respectively.

The scheme. The scheme $\mathcal{IBE}_{\mathsf{LWE2}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is parameterized by the security parameter $\lambda \in \mathbb{N}$, by lattice parameters m, n and q, a parameter $\ell \in \mathbb{N}$ for randomness extraction, and a parameter $\mu \in \mathbb{N}$ such that q^μ is super-polynomial in λ. We let $\mathcal{ID} = \mathbb{Z}_q^\ell$ denote the identity space, and let $d \in \mathbb{N}$ be an integer such that q is a d-bit prime.

- **Setup:** The algorithm Setup on input 1^λ, samples $\mathbf{A}_0, \{\mathbf{A}_{i,j,k}\}_{(i,j,k)\in[\ell]\times[\mu]\times[d]}, \mathbf{B} \leftarrow \mathbb{Z}_q^{n\times m}$ with a trapdoor $\mathbf{T}_\mathbf{A} \in \mathbb{Z}^{m\times m}$ for the lattice $\Lambda_q^\perp(\mathbf{A}_0)$ using the algorithm $\mathsf{TrapGen}$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. It outputs $\mathrm{pp} = (\mathbf{A}_0, \{\mathbf{A}_{i,j,k}\}_{(i,j,k)\in[\ell]\times[\mu]\times[d]}, \mathbf{B}, \mathbf{u})$ and $\mathrm{msk} = \mathbf{T}_\mathbf{A}$.

- **Key generation:** On input the master secret key msk and a identity $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_\ell) \in \mathbb{Z}_q^\ell$, the algorithm KeyGen chooses a vector $\mathbf{s} \in \mathbb{Z}_q^{\ell\mu}$ of $\ell\mu$ elements $s_{1,1}, \ldots, s_{\ell,\mu} \leftarrow \mathbb{Z}_q$ represented as bits $s_{i,j,k}$ where $s_{i,j} = \sum_{k\in[d]} s_{i,j,k}2^{k-1}$ for all $i \in [\ell], j \in [\mu]$, and computes $\mathbf{F}_{\mathbf{id},\mathbf{s}}$ defined as

$$\mathbf{F}_{\mathbf{id},\mathbf{s}} \overset{\mathsf{def}}{=} \left[\mathbf{A}_0 \left| \sum_{\substack{i\in[\ell]\\k\in[d]}} s_{i,1,k}\mathbf{A}_{i,1,k} + \left(\sum_{i\in[\ell]} s_{i,1}\mathrm{id}_i\right)\mathbf{B} \right| \cdots \right.$$
$$\left. \cdots \left| \sum_{\substack{i\in[\ell]\\k\in[d]}} s_{i,\mu,k}\mathbf{A}_{i,\mu,k} + \left(\sum_{i\in[\ell]} s_{i,\mu}\mathrm{id}_i\right)\mathbf{B} \right.\right] \in \mathbb{Z}_q^{n\times m(\mu+1)}. \tag{5.2}$$

  Using the algorithm $\mathsf{ExtendBasis}$ and the trapdoor $\mathbf{T}_\mathbf{A}$, it constructs a basis $\mathbf{T}_\mathbf{F}$ for the lattice $\Lambda_q^\perp(\mathbf{F}_{\mathbf{id},\mathbf{s}})$ and uses $\mathbf{T}_\mathbf{F}$ in algorithm $\mathsf{SamplePre}$ to sample a vector $\mathbf{e} \in \mathbb{Z}_q^{m(\mu+1)}$ such that $\mathbf{F}_{\mathbf{id},\mathbf{s}} \cdot \mathbf{e} = \mathbf{u} \pmod q$. It publishes $\mathrm{sk}_\mathbf{id} = (\mathbf{s}, \mathbf{e})$.

- **Encryption:** On input the public parameters pp, identity $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_\ell) \in \mathbb{Z}_p^\ell$, and a message $\mathfrak{m} \in \{0,1\}$, the algorithm samples $\mathbf{r} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{\chi}_0 \leftarrow \overline{\Psi}_\alpha^m$, and $\{\mathbf{R}_{i,j,k}\}_{(i,j,k)\in[\ell]\times[\mu]\times[d]} \in \{-1,1\}^{m\times m}$ and computes $\boldsymbol{\chi}_{i,j,k} = \mathbf{R}_{i,j,k}{}^\intercal\boldsymbol{\chi}_0 \in \mathbb{Z}_q^m$. Finally, it samples $\xi \leftarrow \overline{\Psi}_\alpha$ and outputs

$$\left(\mathbf{c}_0, \{\mathbf{c}_{i,j,k}\}_{i\in[\ell],j\in[\mu],k\in[d]}, c_{\ell\mu d+1}\right) =$$
$$\left(\mathbf{A}_0{}^\intercal\mathbf{r} + \boldsymbol{\chi}_0,\ \left\{\left[\mathbf{A}_{i,j,k} + 2^{k-1}\mathrm{id}_i\mathbf{B}\right]^\intercal\mathbf{r} + \boldsymbol{\chi}_{i,j,k}\right\}_{i\in[\ell],j\in[\mu],k\in[d]},\right.$$
$$\left.\mathbf{u}^\intercal\mathbf{r} + \xi + \mathfrak{m}\cdot\frac{q}{2}\right) \in (\mathbb{Z}_q^m)^{(\ell\mu d+1)} \times \mathbb{Z}_q.$$

- **Decryption:** On input the public parameters pp, a ciphertext $(\mathbf{c}_0, \mathbf{c}_{1,1,1}, \ldots, \mathbf{c}_{\ell,\mu,d}, c_{\ell\mu d+1}) \in (\mathbb{Z}_q^m)^{(\ell\mu d+1)} \times \mathbb{Z}_q$ and a secret key $(\mathbf{s}, \mathbf{e})$, the algorithm Dec splits $\mathbf{s} = (s_{1,1}, \ldots, s_{\ell,\mu})$ into bits such that $s_{i,j} = \sum_{k\in[d]} s_{i,j,k} \cdot 2^{k-1}$ for all $(i,j) \in [\ell] \times [\mu]$. It outputs 0 if
$$\left| \mathbf{e}^{\mathsf{T}} \cdot \left[\mathbf{c}_0 \,\middle|\, \sum_{\substack{i\in[\ell]\\ k\in[d]}} s_{i,1,k}\mathbf{c}_{i,1,k} \,\middle|\, \cdots \,\middle|\, \sum_{\substack{i\in[\ell]\\ k\in[d]}} s_{i,\mu,k}\mathbf{c}_{i,\mu,k} \right] - c_{\ell\mu d+1} \pmod q \right| < q/4,$$
and 1 otherwise.

Parameter selection. For the scheme, for n polynomial in the security parameter λ, we let $m = n \cdot \Omega(\log n)$, $q = m^{2.5} \cdot \omega(\sqrt{\log n})$, $\rho = \omega(\log n)$, $\alpha = 1/(m^2 \cdot \omega(\sqrt{\log n}))$, $\mu = \omega(1)$ and $\ell = \omega(\mu)$.

Correctness. We show that if $\mathbf{e}$ is well-formed then by combining the ciphertext components $\mathbf{c}_{i,j,k}$ with $s_{i,j,k}$ as in the test algorithm, we can recover $c_{\ell\mu d+1}$ with error-terms and the message (encoded in the most significant bit) left over. In the second half of the proof of correctness, we show that a simple Lemma suffices to bound the error term away from $q/4$ and therefore show correctness of the test algorithm.

$$\begin{aligned}
&\mathbf{e}^{\mathsf{T}} \cdot \left[\mathbf{c}_0 \,\middle|\, \sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,1,k}\mathbf{c}_{i,1,k} \,\middle|\, \cdots \,\middle|\, \sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,\mu,k}\mathbf{c}_{i,\mu,k} \right] \\
&= \mathbf{e}^{\mathsf{T}} \cdot \left[\mathbf{A}_0{}^{\mathsf{T}}\mathbf{r} + \boldsymbol{\chi}_0 \,\middle|\, \sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,1,k}\left[\mathbf{A}_{i,1,k} + 2^{k-1}\mathrm{id}_i\mathbf{B}\right]^{\mathsf{T}}\mathbf{r} + s_{i,1,k}\boldsymbol{\chi}_{i,1,k} \,\middle|\, \cdots \right. \\
&\qquad\qquad \left.\middle|\, \sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,d,k}\left[\mathbf{A}_{i,d,k} + 2^{k-1}\mathrm{id}_i\mathbf{B}\right]^{\mathsf{T}}\mathbf{r} + s_{i,d,k}\boldsymbol{\chi}_{i,d,k} \right] \\
&= \mathbf{e}^{\mathsf{T}} \cdot \left[\mathbf{A}_0{}^{\mathsf{T}}\mathbf{r} \,\middle|\, \sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,1,k}\mathbf{A}_{i,1,k}{}^{\mathsf{T}}\mathbf{r} + \sum_{i\in[\ell]} s_{i,1}\mathrm{id}_i\mathbf{B}^{\mathsf{T}}\mathbf{r} \,\middle|\, \cdots \right. \\
&\qquad\qquad \left.\middle|\, \sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,d,k}\mathbf{A}_{i,d,k}{}^{\mathsf{T}}\mathbf{r} + \sum_{i\in[\ell]} s_{i,d}\mathrm{id}_i\mathbf{B}^{\mathsf{T}}\mathbf{r} \right] + \mathbf{e}^{\mathsf{T}} \left[\boldsymbol{\chi}_0 \,\middle|\, \left\{ \sum_{i\in[\ell],k\in[d]} s_{i,j,k}\boldsymbol{\chi}_{i,j,k} \right\}_{j\in[\mu]} \right] \\
&= \mathbf{e}^{\mathsf{T}} \cdot \mathbf{F}_{\mathbf{id},\mathbf{s}}{}^{\mathsf{T}}\mathbf{r} + \mathbf{e}^{\mathsf{T}}\boldsymbol{\chi} \\
&= \mathbf{u}^{\mathsf{T}}\mathbf{r} + \mathbf{e}^{\mathsf{T}}\boldsymbol{\chi},
\end{aligned}$$

where $\boldsymbol{\chi} = \left[\boldsymbol{\chi}_0 \,\middle|\, \left\{ \sum_{i\in[\ell],k\in[d]} s_{i,j,k}\boldsymbol{\chi}_{i,j,k} \right\}_{j\in[\mu]} \right]$ and $\mathbf{F}_{\mathbf{id},\mathbf{s}}$ is as defined in Equation (5.2). Observe that excluding the message $\frac{q}{2}\mathfrak{m}$, the ciphertext component $c_{\ell\mu d+1}$ is exactly $\mathbf{u}^{\mathsf{T}}\mathbf{r} + \xi$. The term $\mathbf{u}^{\mathsf{T}}\mathbf{r}$ cancels, and to prove correctness, we need to show that the noise term $(\mathbf{e}^{\mathsf{T}}\boldsymbol{\chi} - \xi)$ is low-norm. Observe that $\boldsymbol{\chi}_{i,j,k} = \mathbf{R}_{i,j,k}{}^{\mathsf{T}}\boldsymbol{\chi}_0$ and let $\mathbf{e} = [\,\mathbf{e}_0 \,|\, \mathbf{e}_1 \,|\, \cdots \,|\, \mathbf{e}_\mu\,]$, then we can re-write the noise term as
$$\left(\mathbf{e}_0 + \sum_{\substack{i\in[\ell],j\in[\mu]\\ k\in[d]}} s_{i,j,k}\mathbf{R}_{i,j,k}\mathbf{e}_j \right)^{\mathsf{T}} \boldsymbol{\chi}_0 - \xi.$$

Observing that $\|\mathbf{e}\|_2 \le m \cdot \omega(\sqrt{\log m})$ (as in Section 4.2), we can now apply the following lemma to bound the size of the error term.

Lemma 5.6 ([ABB10, Lemma 15])**.** *For parameters $m, n \in \mathbb{N}$, let $\mathbf{R}$ be a $k \times m$ matrix chosen uniformly at random from $\{-1,1\}^{k \times m}$. Then, for all $\mathbf{v} \in \mathbb{Z}^m$, $\Pr\big[\|\mathbf{R}\mathbf{v}\|_2 > 12\sqrt{k+m} \cdot \|\mathbf{v}\|_2\big] < e^{-(k+m)}$.*

As the $s_{i,j,k}$'s are binary, we have

$$\left\| \mathbf{e}_0 + \sum_{i \in [\ell]} \sum_{j \in [\mu]} \sum_{k \in [d]} s_{i,j,k} \mathbf{R}_{i,j,k} \mathbf{e}_j \right\|_2 \le \Big(1 + 12\ell\mu d\sqrt{2m}\Big) \cdot m = \tilde{O}(\ell\mu d m^{3/2}).$$

Therefore, applying Lemma 2.10, the noise term is bounded by

$$\Big(\sqrt{m}/2 + q\alpha\omega(\sqrt{\log m})\Big) \cdot \tilde{O}(\ell\mu d m^{3/2}) < q/10,$$

(from our choice of parameters) which completes the proof of correctness.

Extension to multi-bit encryption. We note that as in the lattice-based IBE schemes of [GPV08, ABB10], it is possible to encrypt N bits simultaneously at the expense of $N-1$ additional $\mathbb{Z}_q^n$ vectors in the public parameters, and $N-1$ additional $\mathbb{Z}_q$ elements in the ciphertexts in our scheme. We refer the reader to [ABB10, Section 6.5] for more details.

Security. In Sections 5.2.1 and 5.2.2 we prove the following theorem:

Theorem 5.7. *In the standard model the scheme $\mathcal{IBE}_{\mathsf{LWE2}}$ is selectively-secure data private based on the LWE assumption, and is statistically functional private for:*

1. *(T,k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \ge \mu \cdot \Omega(\log\lambda) + \omega(\log\lambda)$.*
2. *$(k_1,\dots,k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1,\dots,k_T)$ such that $k_i \ge i\mu \cdot \Omega(\log\lambda) + \omega(\log\lambda)$ for every $i \in [T]$.*

5.2.1 Proof of (Selective) Data Privacy

Lemma 5.8. *The scheme $\mathcal{IBE}_{\mathsf{LWE2}}$ is selectively-secure data private based on the LWE assumption in the standard model.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial time adversary. We consider a series of experiments that interact with the adversary. Experiment Expt_0 is identical to $\mathsf{Expt}^{(0)}_{\mathsf{sDP},\mathcal{IBE}_{\mathsf{LWE2}},\mathcal{A}}$ in Definition 2.7. Experiment Expt_1 is identical to Expt_0 except in step (3). The challenger replaces a well-constructed challenge ciphertext with independently and uniformly sampled $\big(\mathbf{u}_0, \{\mathbf{u}_{i,j,k}\}_{i\in[\ell],j\in[\mu],k\in[d]}, u_{\ell\mu d+1}\big) \leftarrow (\mathbb{Z}_q^m)^{(\ell\mu d+1)} \times \mathbb{Z}_q$. Experiment Expt_2 is identical to $\mathsf{Expt}^{(1)}_{\mathsf{sDP},\mathcal{IBE}_{\mathrm{DLIN1}},\mathcal{A}}$ in Definition 2.7. Now we can state the following claim.

Claim 5.9. *Based on the LWE assumption, it holds that $|\Pr[\mathsf{Expt}_0(\lambda)=1] - \Pr[\mathsf{Expt}_1(\lambda)=1]| \le \mathrm{negl}(\lambda)$.*

Proof. Denote by Q_T the number of secret key queries of the probabilistic polynomial-time adversary $\mathcal{A}$ in experiment Expt_0 and Expt_1 such that

$$|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]| > \epsilon(\lambda),$$

for some non-negligible $\epsilon(\lambda)$. We construct an algorithm $\mathcal{B}$ that solves the LWE problem with advantage $\epsilon'(\lambda) \geq \epsilon(\lambda) - \mathrm{negl}(\lambda)$. The algorithm $\mathcal{B}$ is given $(\mathbf{E}, \mathbf{f}) \in \mathbb{Z}_q^{n\times(m+1)} \times \mathbb{Z}_q^{m+1}$ and interacts with $\mathcal{A}$ to decide whether $\mathbf{f}$ comes from the uniform distribution or from the distribution $\mathbf{E}^\intercal\mathbf{s} + \boldsymbol{\chi}$ for $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\boldsymbol{\chi} \leftarrow \overline{\Psi}_\alpha^{m+1}$ as follows.

- **Setup:** The algorithm $\mathcal{B}$ parses the matrix $\mathbf{E}$ as $(\mathbf{A}_0 \,|\, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$. It samples a random matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times m}$ with trapdoor $\mathbf{T_B}$ using algorithm $\mathsf{TrapGen}$. It receives the challenge identities $\mathbf{id}_0^*$ and $\mathbf{id}_1^*$ from the selective-security adversary $\mathcal{A}$ and sets $\mathbf{id}^* = \mathbf{id}_0^*$ which is encoded as $(\mathrm{id}_1^*, \ldots, \mathrm{id}_\ell^*) \in \mathbb{Z}_q^\ell$. It chooses random matrices $\{\mathbf{R}_{i,j,k}\}_{i\in[\ell],j\in[\mu],k\in[d]} \in \{-1,1\}^{m\times m}$ and computes $\mathbf{A}_{i,j,k} = \mathbf{A}_0\mathbf{R}_{i,j,k} - 2^{k-1} \cdot \mathrm{id}_i^*\mathbf{B}$. It publishes $\mathrm{pp} = (\mathbf{A}_0, \{\mathbf{A}_{i,j,k}\}_{i\in[\ell],Jj\in[\mu],k\in[d]}, \mathbf{B}, \mathbf{u})$.

- **Key generation:** On input $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_\ell) \in \mathbb{Z}_q^\ell$ the algorithm $\mathcal{B}$ first samples a random vector $\mathbf{s} \in \mathbb{Z}_q^{\ell\mu}$ of $\ell\mu$ elements $s_{1,1}, \ldots, s_{\ell,\mu} \leftarrow \mathbb{Z}_q$ and computes

  $$\delta_1 = \sum_{i\in[\ell]} s_{i,1}(\mathrm{id}_i - \mathrm{id}_i^*),\ \ldots,\ \delta_\mu = \sum_{i\in[\ell]} s_{i,\mu}(\mathrm{id}_i - \mathrm{id}_i^*).$$

  If for all $j \in [\mu]$, $\delta_j = 0$, *abort* the simulation and output a uniform bit $b' \leftarrow \{0,1\}$. Otherwise, let j^* be an index such that $\delta_{j^*} \neq 0$. Consider the matrix

  $$\begin{aligned}\mathbf{F}' &= \left[\mathbf{A}_0 \,\middle|\, \sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,j^*,k}\mathbf{A}_{i,j^*,k} + \left(\sum_{i\in[\ell]} s_{i,j^*}\mathrm{id}_i\right)\mathbf{B}\right] \\ &= \left[\mathbf{A}_0 \,\middle|\, \mathbf{A}\left(\sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,j^*,k}\mathbf{R}_{i,j^*,k}\right) - \left(\sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,j^*,k}2^{k-1} \cdot \mathrm{id}_i^*\right)\mathbf{B} + \left(\sum_{i\in[\ell]} s_{i,j^*}\mathrm{id}_i\right)\mathbf{B}\right] \\ &= \left[\mathbf{A}_0 \,\middle|\, \mathbf{A}\underbrace{\left(\sum_{i\in[\ell]}\sum_{k\in[d]} s_{i,j^*,k}\mathbf{R}_{i,j^*,k}\right)}_{\mathbf{R}^*} - \underbrace{\left(\sum_{i\in[\ell]} s_{i,j^*}\left(\mathrm{id}_i - \mathrm{id}_i^*\right)\right)}_{\delta_{j^*}}\mathbf{B}\right].\end{aligned}$$

  We use $\mathsf{ExtendBasis}$ to compute a trapdoor $\mathbf{T}_{\mathbf{F}'}$ for the lattice $\Lambda_q^\perp(\mathbf{F}')$ given trapdoor $\mathbf{T_B}$ for lattice $\Lambda_q^\perp(\mathbf{B})$. This requires $\delta_{j^*} \neq 0$ and low-norm $\mathbf{R}^*$ (which follows from the fact that $\mathbf{R}_{i,j^*,k}$ are $\{-1,1\}$ matrices, and $s_{i,j^*,k} \in \{0,1\}$). Given a trapdoor for $\Lambda_q^\perp(\mathbf{F}')$, we can use $\mathsf{ExtendBasis}$ once again, in a straightforward manner to sample a short vector $\mathbf{e} \in \mathbb{Z}_q^{m(\mu+1)}$ such that $\mathbf{F}_{\mathbf{id},\mathbf{s}} \cdot \mathbf{e} = \mathbf{u} \pmod q$ (where $\mathbf{F}_{\mathbf{id},\mathbf{s}}$ is as defined in Eq (5.2)). It outputs the secret key $(\mathbf{s}, \mathbf{e})$.

- **Challenge:** Eventually $\mathcal{A}$ requests the challenge ciphertext corresponding to $(\mathbf{id}_0^*, \mathfrak{m}_0^*)$ or $(\mathbf{id}_1^*, \mathfrak{m}_1^*)$ for the adversary's choice of $\mathfrak{m}_0^*$ and $\mathfrak{m}_1^*$ upon which $\mathcal{B}$ does the following. First, it sets $\mathfrak{m} = \mathfrak{m}_0^*$ and parses $\mathbf{f} = [\mathbf{f}_0^\intercal \,|\, f_1]^\intercal \in \mathbb{Z}_q^m \times \mathbb{Z}_q$ from the LWE challenge. Next, for all $i \in [\ell]$, $j \in [\mu]$ and $k \in [d]$, $\mathcal{B}$ computes $\mathbf{c}_{i,j,k}^* = \mathbf{R}_{i,j,k}{}^\intercal\mathbf{f}_0$, $\mathbf{c}_0^* = \mathbf{f}_0$, and $c_{\ell\mu d+1}^* = f_1 + \frac{q}{2}\mathfrak{m}$. It outputs the challenge ciphertext $\left(\mathbf{c}_0^*, \{\mathbf{c}_{i,j,k}^*\}_{(i,j,k)\in[\ell]\times[\mu]\times[d]}, c_{\ell\mu d+1}^*\right)$.

- **Output:** If $\mathcal{A}$ at the end of the simulation outputs a bit b guessing Expt_b, $\mathcal{B}$ outputs the same bit b.

It is easy to see from the construction that in the challenge phase, if $\mathcal{B}$ is given an LWE instance, i.e., $\mathbf{f} = \mathbf{E}^\intercal \mathbf{r} + \boldsymbol{\chi}$, for some random $\mathbf{r} \in \mathbb{Z}_q^n$ and low-norm error term $\boldsymbol{\chi} \leftarrow \overline{\Psi}_\alpha^{m+1}$, then

$$
\begin{aligned}
\left(\mathbf{c}_0^*, \{\mathbf{c}_{i,j,k}^*\}, c_{\ell\mu d+1}^*\right) &= \left(\mathbf{A}_0^\intercal \mathbf{r} + \boldsymbol{\chi}_0, \{[\mathbf{A}_0 \mathbf{R}_{i,j,k}]^\intercal \mathbf{r} + \mathbf{R}_{i,j,k}{}^\intercal \boldsymbol{\chi}_0\}_{i\in[\ell], j\in[\mu], k\in[d]}, \mathbf{u}^\intercal \mathbf{r} + \xi + \frac{q}{2}\mathfrak{m}\right) \\
&= \left(\mathbf{A}_0^\intercal \mathbf{r} + \boldsymbol{\chi}_0, \left\{\left[\mathbf{A}_{i,j,k} + 2^{k-1}\mathrm{id}_i^* \mathbf{B}_i\right]^\intercal \mathbf{r} + \boldsymbol{\chi}_{i,j,k}\right\}_{i\in[\ell], j\in[\mu], k\in[d]}, \mathbf{u}^\intercal \mathbf{r} + \xi + \frac{q}{2}\mathfrak{m}\right)
\end{aligned}
$$

(where $\boldsymbol{\chi} = [\boldsymbol{\chi}_0{}^\intercal \,|\, \xi]^\intercal \in \mathbb{Z}_q^m \times \mathbb{Z}_q$) is a well-formed ciphertext corresponding to $\mathbf{id}_0^*$ and $\mathfrak{m}_0^*$ and therefore, the challenge is distributed as in Expt_0.

Next, we need to argue that if $\mathbf{f}$ is random in $\mathbb{Z}_q^{m+1}$, then the challenge ciphertext is distributed as in Expt_1. This requires the use of the leftover hash lemma (cf. [DOR+08]) as in [ABB10].

The challenge ciphertext is distributed as $\left(\mathbf{f}_0, \widetilde{\mathbf{R}}^\intercal \mathbf{f}_0, f_1 + \frac{q}{2}\mathfrak{m}\right)$ for $\widetilde{\mathbf{R}} = \left[\{\mathbf{R}_{i,j,k}\}_{i\in[\ell], j\in[\mu], k\in[d]}\right] \in \{\pm 1\}^{m \times m(\ell\mu d)}$. Note that f_1 is uniform over $\mathbb{Z}_q$ independent of the rest of the components, and can therefore be ignored as $f_1 + \frac{q}{2}\mathfrak{m}$ is distributed correctly. A direct application of the leftover hash lemma (cf. Lemma 2.6) with $(\mathbf{A}_0{}^\intercal \,|\, \mathbf{f}_0)$ as the hash function implies that $\mathbf{A}_0 \widetilde{\mathbf{R}}$ (from the public parameters) and $\widetilde{\mathbf{R}}^\intercal \mathbf{f}_0$ (from the ciphertext) are statistically close to uniform and independent quantities (given $\mathbf{A}_0$ and $\mathbf{f}_0$). Therefore, the simulation simulates a ciphertext that is statistically close to the real distribution.

Additionally, in both simulations, it follows once again from the application of the leftover hash lemma that pp generated by $\mathcal{B}$ is statistically close to the real distribution.

Thus, to complete the proof of Claim 5.9, it suffices to bound the probability of $\mathcal{B}$ aborting the simulation. Recollect that $\mathcal{B}$ aborts depending on the values of δ_i's defind earlier. As calculated in a similar case in Section 5.1, the probability that any particular $\delta_i = 0$ is $1/q$. As $s_{i,j}$'s are chosen uniformly and independently at random, for all $i \neq j$, δ_i and δ_j are independent events. Therefore, the probability that $\mathcal{B}$ aborts is $1/q^\mu$.

As argued in the proof of Lemma 5.2, we can calculate that ϵ' is at least $\epsilon - 1/q^\mu$. Based on the LWE assumption, ϵ' is negligible, and with our choice of parameters for μ, ϵ is also negligible thereby completing the proof. ∎

Claim 5.10. *Based on the LWE assumption, it holds that* $|\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \leq \mathrm{negl}(\lambda)$.

The proof of the above claim is identical to the proof of Claim 5.9 except in the simulation, we use $\mathbf{id}_1^*$ when simulating Expt_2. To complete the proof of the theorem,

$$
\begin{aligned}
&\mathbf{Adv}_{\mathcal{IBE},\mathcal{A}}^{\mathsf{DP}}(\lambda) \\
&\qquad = \left|\Pr\left[\mathsf{Expt}_{\mathsf{DP},\mathcal{IBE},\mathcal{A}}^{(0)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_{\mathsf{DP},\mathcal{IBE},\mathcal{A}}^{(1)}(\lambda) = 1\right]\right| \\
&\qquad = |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \\
&\qquad = |\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]| + |\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]| \\
&\qquad \leq \mathrm{negl}(\lambda), \qquad\qquad \text{(from Claims 5.9 and 5.10)}
\end{aligned}
$$

as required. ∎

5.2.2 Proof of Function Privacy

Lemma 5.11. *The scheme $\mathcal{IBE}_{\mathsf{LWE2}}$ is statistically function private for:*

1. *(T, k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq \mu \cdot \Omega(\log \lambda) + \omega(\log \lambda)$.*
2. *$(k_1, \ldots, k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1, \ldots, k_T)$ such that $k_i \geq i\mu \cdot \Omega(\log \lambda) + \omega(\log \lambda)$ for every $i \in [T]$.*

Proof. Let $X \in \{(T, k)\text{-block}, (k_1, \ldots, k_T)\}$, and let $\mathcal{A}$ be a computationally unbounded X-source function-privacy adversary that makes a polynomial number $Q_{\mathsf{RoR}} = Q_{\mathsf{RoR}}(\lambda)$ of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{LWE2}},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\mathcal{IBE}_{\mathsf{LWE2}},\mathcal{A}}$ (we refer the reader to Definition 3.3 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$, respectively.

The collection of functions $\{g_{s_1,\ldots,s_\ell} : \mathbb{Z}_q^\ell \to \mathbb{Z}_q\}_{s_1,\ldots,s_\ell \in \mathbb{Z}_q}$ defined by $g_{s_1,\ldots,s_\ell}(\mathrm{id}_1, \ldots, \mathrm{id}_\ell) = \sum_{i \in [\ell]} s_i \mathrm{id}_i$ is universal. Observe that trapdoor generation uses μ *independent* universal functions $g_1, \ldots, g_\mu$ defined as above. Thus, we define a collection of functions

$$\mathcal{F} \overset{\mathsf{def}}{=} \{f_{s_{1,1},\ldots,s_{\ell,\mu}} : \mathbb{Z}_q^{\ell\mu} \to \mathbb{Z}_q^\mu\} \text{ as } f_{s_{1,1},\ldots,s_{\ell,\mu}}(\mathrm{id}_1, \ldots, \mathrm{id}_\ell) = \left(\sum_{i \in [\ell]} s_{i,1}\mathrm{id}_i, \ldots, \sum_{i \in [\ell]} s_{i,\mu} s_{i,\mu}\mathrm{id}_i \right) \tag{5.3}$$

which is also universal.

We fix the public parameters, pp, and master secret key, msk, of the scheme, and show that the two distributions $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$ are statistically close for any such pp and msk. As the adversary $\mathcal{A}$ is computationally unbounded, we assume without loss of generality that $\mathcal{A}$ does not query the $\mathsf{KeyGen}(\mathrm{msk}, \cdot)$ oracle. In addition, as discussed in Section 3.1, we can assume that $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once.

Denote by $\boldsymbol{ID} = \left(ID^{(1)}, \ldots, ID^{(T)}\right)$ the random variable corresponding to the X-source with which $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. Having already fixed pp and msk, observing that $\mathbf{B}$ and $\mathbf{A}_{i,j,k}$'s are fixed for any keyword $\mathbf{id}^{(i)} \leftarrow ID^{(i)}$, it suffices to consider the view of the adversary

$$\mathsf{View}_{\mathsf{mode}} = \left(\begin{matrix} \left(s^{(1)}_{1,1}, \ldots, s^{(1)}_{\ell,1}, \sum_{i \in [\ell]} s^{(1)}_{i,1}\mathrm{id}^{(1)}_i \right), & \ldots, & \left(s^{(1)}_{1,\mu}, \ldots, s^{(1)}_{\ell,\mu}, \sum_{i \in [\ell]} s^{(1)}_{i,\mu}\mathrm{id}^{(1)}_i \right) \\ \vdots & \ddots & \vdots \\ \left(s^{(T)}_{1,1}, \ldots, s^{(T)}_{\ell,1}, \sum_{i \in [\ell]} s^{(T)}_{i,1}\mathrm{id}^{(T)}_i \right), & \ldots, & \left(s^{(T)}_{1,\mu}, \ldots, s^{(T)}_{\ell,\mu}, \sum_{i \in [\ell]} s^{(T)}_{i,\mu}\mathrm{id}^{(T)}_i \right) \end{matrix} \right)$$

for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, where $\left(\mathbf{id}^{(1)}, \ldots, \mathbf{id}^{(T)}\right) \leftarrow \left(ID^{(1)}, \ldots, ID^{(T)}\right)$ for $\mathsf{mode} = \mathsf{real}$, $\left(\mathbf{id}^{(1)}, \ldots, \mathbf{id}^{(T)}\right)$ is uniformly distributed over $(\mathcal{ID}_\lambda)^T$ for $\mathsf{mode} = \mathsf{rand}$, $s^{(i)}_{j,k} \leftarrow \mathbb{Z}_q$ for every $i \in [T]$ and $(j, k) \in [\ell] \times [\mu]$. For $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ we prove that the distribution $\mathsf{View}_{\mathsf{mode}}$ is statistically-close to uniform.

We know that $\left(ID^{(1)}, \ldots, ID^{(T)}\right)$ is an X-source, and the collection of functions $\mathcal{F}$ (defined in Equation (5.3) is universal. This enables us to directly apply Lemma 2.3 (in case $\boldsymbol{ID}$ is a (T, k)-block-source) and Lemma 2.4 (in case $\boldsymbol{ID}$ is a $(k_1, \ldots, k_T)$-source), implying that the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ. The same clearly holds also for $\mathsf{View}_{\mathsf{rand}}$, as the uniform distribution over $(\mathcal{ID}_\lambda)^T$ is, in particular, a (T, k)-block-source and a $(k_1, \ldots, k_T)$-source. ▮

5.3 A Fully-Secure DLIN-Based Scheme

In this section we present an IBE scheme based on the DLIN assumption in the standard model. The scheme is a *fully secure* variant of the one described in Section 5.1. The scheme is described below, and its proofs of data privacy and function privacy are presented in Sections 5.3.1 and 5.3.2, respectively.

The scheme. Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^λ, and outputs $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e})$ where $\mathbb{G}$ and $\mathbb{G}_\mathrm{T}$ are groups of prime order p, $\mathbb{G}$ is generated by g, p is a λ-bit prime number, and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathrm{T}$ is a non-degenerate efficiently computable bilinear map. The scheme $\mathcal{IBE}_{\mathsf{DLIN2}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is parameterized by the security parameter $\lambda \in \mathbb{N}$. For any such $\lambda \in \mathbb{N}$, the scheme has parameters $m > 3$, $n = \omega(\log \lambda)$, identity space $\mathcal{ID}_\lambda = \{0,1\}^n$, and message space $\mathcal{M} = \mathbb{G}_\mathrm{T}$.

- **Setup:** On input 1^λ the setup algorithm Setup samples $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e}) \leftarrow \mathsf{GroupGen}(1^\lambda)$. Next, the algorithm samples $\mathbf{A}_0, \mathbf{B}, \{\mathbf{A}_j\}_{j\in[n]} \leftarrow \mathbb{Z}_p^{2\times m}$ and $\mathbf{u} \leftarrow \mathbb{Z}_p^2$. It outputs the master secret key $\mathrm{msk} = \big(\mathbf{A}_0, \mathbf{B}, \{\mathbf{A}_j\}_{j\in[n]}, \mathbf{u}\big)$ and the public parameters $\mathrm{pp} = \big(g^{\mathbf{A}_0}, \mathbf{B}, \{g^{\mathbf{A}_j}\}_{j\in[n]}, g^{\mathbf{u}}\big)$.

- **Key generation:** On input a master secret key msk and identity $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_n) \in \{0,1\}^n$. Next, it samples $\mathbf{S} \leftarrow \mathbb{Z}_p^{m\times 2}$ and computes

$$\mathbf{F}_{\mathbf{id},\mathbf{S}} = \left[\mathbf{A}_0 \,\middle|\, \mathbf{BS} + \left(\sum_{j\in[n]} \mathrm{id}_j \mathbf{A}_j \right) \mathbf{S} \right] \in \mathbb{Z}_p^{2\times(m+2)}$$

  It samples uniformly at random a vector $\mathbf{v} \in \mathbb{Z}_p^{m+2}$ such that $\mathbf{F}_{\mathbf{id},\mathbf{S}} \cdot \mathbf{v} = \mathbf{u} \pmod{p}$ and sets $\mathbf{z} = g^{\mathbf{v}} \in \mathbb{G}^{m+2}$. It outputs $\mathrm{sk}_{\mathbf{id}} = (\mathbf{S}, \mathbf{z})$.

- **Encryption:** On input the public parameters pp, an identity $\mathbf{id} = (\mathrm{id}_1, \ldots, \mathrm{id}_n) \in \{0,1\}^n$, and a message $\mathfrak{m} \in \mathbb{G}_\mathrm{T}$, the algorithm samples $\mathbf{r} \leftarrow \mathbb{Z}_p^2$. It computes $\mathbf{D}(\mathbf{id}) \overset{\mathsf{def}}{=} \sum_{j\in[n]} \mathrm{id}_j \mathbf{A}_j$. It sets $\mathbf{c}_0^\intercal = g^{\mathbf{r}^\intercal \mathbf{A}_0} \in \mathbb{G}^{1\times m}$, $\mathbf{c}_1^\intercal = g^{\mathbf{r}^\intercal[\mathbf{B}+\mathbf{D}(\mathbf{id})]} \in \mathbb{G}^{1\times m}$, and $c_2 = \hat{e}(g,g)^{\mathbf{r}^\intercal \mathbf{u}} \cdot \mathfrak{m} \in \mathbb{G}_\mathrm{T}$ and outputs $(\mathbf{c}_0, \mathbf{c}_1, c_2) \in \mathbb{G}^{2m} \times \mathbb{G}_\mathrm{T}$.

- **Decryption:** On input a ciphertext $(\mathbf{c}_0, \mathbf{c}_1, c_2) \in \mathbb{G}^{2m} \times \mathbb{G}_\mathrm{T}$ and a secret key $\mathrm{sk}_{\mathbf{id}} = (\mathbf{S}, \mathbf{z}) \in \mathbb{Z}_p^{m\times 2} \times \mathbb{G}^{m+2}$, the algorithm outputs

$$c_2 \cdot \hat{e}\left(\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1^{\mathbf{S}} \end{bmatrix}, \begin{matrix} | \\ \mathbf{z} \\ | \end{matrix} \right)^{-1}.$$

Correctness. Consider the vector

$$\mathbf{d}^\intercal = \left[\mathbf{c}_0^\intercal \middle| (\mathbf{c}_i^\intercal)^{\mathbf{S}_i}\right] = g^{\mathbf{r}^\intercal\left[\mathbf{A}_0 \middle| \mathbf{BS} + (\sum_{j\in[n]} \mathrm{id}_j \mathbf{A}_j)\mathbf{S}\right]} = g^{\mathbf{r}^\intercal \mathbf{F}_{\mathbf{id},\mathbf{S}}}.$$

We have $\hat{e}(\mathbf{d}, \mathbf{z}) = \hat{e}(g,g)^{\mathbf{r}^\intercal \mathbf{F}_{\mathbf{id},\mathbf{S}} \cdot \mathbf{v}} = \hat{e}(g,g)^{\mathbf{r}^\intercal \mathbf{u}}$. Therefore, dividing c_2 by $\hat{e}(\mathbf{d}, \mathbf{z})$ eliminates the term $\hat{e}(g,g)^{\mathbf{r}^\intercal \mathbf{u}}$ which recovers $\mathfrak{m}$ correctly.

Security. In Sections 5.3.1 and 5.3.2 we prove the following theorem:

Theorem 5.12. *The scheme $\mathcal{IBE}_{\mathsf{DLIN2}}$ is data private based on the DLIN assumption, and is statistically function private for:*

1. *(T,k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq 4\log p + \omega(\log \lambda)$.*
2. *$(k_1,\dots,k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1,\dots,k_T)$ such that $k_i \geq 4i\log p + \omega(\log \lambda)$ for every $i \in [T]$.*

5.3.1 Proof of Data Privacy

Lemma 5.13. *The scheme $\mathcal{IBE}_{\mathsf{DLIN2}}$ is data private based on the DLIN assumption in the standard model.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial time adversary for the scheme $\mathcal{IBE}_{\mathsf{DLIN2}}$. We denote by $\mathbf{id}^{(1)},\dots,\mathbf{id}^{(Q)}$ (bits of which are denoted $\mathrm{id}_j^{(i)}$ for $j \in [n]$) the Q secret key queries generated by the adversary $\mathcal{A}$. The challenge identities are denoted $\left(\mathbf{id}^{*(0)}, \mathbf{id}^{*(1)}\right)$. We define a (non-negligible) function of the security parameter $\alpha = \alpha(\lambda) \in [0,1]$ to denote a *lower bound* of the probability of a particular event relating to the simulation (see the description of Expt_2 and Lemma 5.14 below). We consider the following experiments for each $b \in \{0,1\}$.

- Experiment $\mathsf{Expt}_0^{(b)}$ is identical to $\mathsf{Expt}^{(b)}_{\mathsf{DP},\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}$ as in Definition 2.7.
- Experiment $\mathsf{Expt}_1^{(b)}$ is obtained from $\mathsf{Expt}_0^{(b)}$ by outputting the output of $\mathsf{Expt}_0^{(b)}$ with probability α and a random bit with probability $1-\alpha$ (denoted by Abort).
- Experiment $\mathsf{Expt}_2^{(b)}$ is obtained from $\mathsf{Expt}_0^{(b)}$ by introducing an "artificial" abort event independent of the adversary's view. We use the programmable family of hash functions introduced by Hofheinz and Kiltz [HK12] denoted $\mathcal{H}_{\mathsf{HK},Q}$ (see Section 2.5). At the end of $\mathsf{Expt}_2^{(b)}$, we sample a hash function $H \leftarrow \mathcal{H}_{\mathsf{HK},Q}$. When $\mathsf{Expt}_2^{(b)}$ receives the guess b' from $\mathcal{A}$, it does the following:
  1. *Abort check:* For each query $\mathbf{id}^{(i)}$ for $i \in [Q]$, let $\mathbf{S}^{(i)} \in \mathbb{Z}_p^{m\times 2}$ denote the uniform matrix chosen during secret key generation. The challenger checks the following conditions:
     (a) For each $i \in [Q]$, if $H\left(\mathbf{id}^{(i)}\right) \cdot \mathbf{B}\mathbf{S}^{(i)} \in \mathbb{Z}_p^{2\times 2}$ is full-rank.
     (b) For bit $b \in \{0,1\}$, $H\left(\mathbf{id}^{*(b)}\right) = 0$.
     If either (or both) these conditions are not satisfied, the experiment outputs a random bit instead of b'. Let α denote the probability over choices of the hash function H (for any particular set of distinct queries $\left(\mathbf{id}^{*(b)}, \mathbf{id}^{(1)},\dots,\mathbf{id}^{(Q)}\right)$) that both conditions above are *true*. Lemma 5.14 derives a bound for α.
  2. *Artificial abort:* Following the approach of Cash et al. [CHK+10] (generalizing that of Waters [Wat05]) approximate $\varrho^{(b)} = \Pr\left[\overline{\mathsf{Abort}} \middle| \left(\mathbf{id}^{*(b)}, \mathbf{id}^{(1)},\dots,\mathbf{id}^{(Q)}\right)\right]$ by sampling sufficiently many independent hash functions. For any polynomial $S = S(\lambda)$, Hoeffding's inequality yields that with $\lceil \lambda S/\alpha \rceil$ samples, we can obtain an approximation $\tilde{\varrho}^{(b)} \geq \alpha$ of $\varrho^{(b)}$ such that:
  $$\Pr\left[\left|\varrho^{(b)} - \tilde{\varrho}^{(b)}\right| \geq \frac{\alpha}{S}\right] \leq \frac{1}{2^{\lambda}}, \tag{5.4}$$

for security parameter λ. The challenger samples a random bit $\tilde{b} \in \{0,1\}$ such that $\Pr\left[\tilde{b} = 1\right] = 1 - \alpha/\tilde{\varrho}^{(b)} \in [0,1]$. If $\tilde{b} = 1$ then the adversary outputs a random bit (artificial abort). Else, it outputs the bit b' from the challenger.

- Experiment $\mathsf{Expt}_3^{(b)}$ is obtained from $\mathsf{Expt}_2^{(b)}$ by replacing the challenge ciphertext with uniform $(\mathbf{c}_0, \mathbf{c}_1, c_2) \leftarrow \mathbb{G}^{2m} \times \mathbb{G}_\mathrm{T}$ that is sampled independently of the view of $\mathcal{A}$.

Observe that the bit b is only used in the challenge phase and in experiments $\mathsf{Expt}_3^{(0)}$ and $\mathsf{Expt}_3^{(1)}$ the challenge phase is independent of the bit b. Additionally, whenever experiments $\mathsf{Expt}_3^{(0)}$ and $\mathsf{Expt}_3^{(1)}$ abort, they output a uniform bit. From this, we conclude that $\mathsf{Expt}_3^{(0)} = \mathsf{Expt}_3^{(1)}$. We will argue through a series of claims that $\left|\Pr\left[\mathsf{Expt}_0^{(0)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_0^{(1)}(\lambda) = 1\right]\right|$ is negligible, thus completing the proof. First we derive a bound for α.

Lemma 5.14. *For distinct $(Q+1)$-tuple of queries $\mathbf{id}^*, \mathbf{id}^{(1)}, \ldots, \mathbf{id}^{(Q)} \in \{0,1\}^n$ define the following events:*

- $\mathsf{Event}_\mathsf{T}$ *(secret key queries) is the event in which for each $i \in [Q]$, $H\left(\mathbf{id}^{(i)}\right) \cdot \mathbf{BS}^{(i)}$ is a full-rank matrix in $\mathbb{Z}_p^{2\times 2}$.*
- $\mathsf{Event}_\mathsf{C}$ *(challenge query) is the event in which $H\left(\mathbf{id}^*\right) = 0$.*

Then, for every distinct $(Q+1)$-tuple of queries $\mathbf{id}^*, \mathbf{id}^{(1)}, \ldots, \mathbf{id}^{(Q)}$, and any set of full rank $\mathbf{B}_1, \ldots, \mathbf{B}_\ell$, we have:

$$\Pr[\mathsf{Event}_\mathsf{T} \wedge \mathsf{Event}_\mathsf{C}] \geq \alpha = \left(1 - \frac{2Q}{p}\right) \cdot \Theta\left(\frac{1}{Q\sqrt{n}}\right),$$

where the probability is taken over choices of $H \in \mathcal{H}_\mathsf{HK}$ and uniformly distributed matrices $\mathbf{S}^{(i)} \in \mathbb{Z}_p^{m\times 2}$ for $i \in [Q]$.

Proof. We defer the proof to the end of the section for readability. ∎

Next, we derive a series of claims relating the experiments described above.

Claim 5.15. *It holds that*

$$\left|\Pr\left[\mathsf{Expt}_1^{(0)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1^{(1)}(\lambda) = 1\right]\right| = \alpha \cdot \left|\Pr\left[\mathsf{Expt}_0^{(0)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_0^{(1)}(\lambda) = 1\right]\right|$$

Proof. For each $b \in \{0,1\}$, $\Pr\left[\mathsf{Expt}_2^{(b)}(\lambda) = 1\right] = \alpha \cdot \Pr\left[\mathsf{Expt}_1^{(b)}(\lambda) = 1\right] + \frac{1}{2}(1-\alpha)$. ∎

Claim 5.16. *For each $b \in \{0,1\}$ and for any polynomial $S = S(\lambda)$, it holds that*

$$\left|\Pr\left[\mathsf{Expt}_2^{(b)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1^{(b)}(\lambda) = 1\right]\right| \leq \frac{\alpha}{S} + \frac{1}{2^\lambda}.$$

Proof. Let $\mathsf{Abort}_2^{(b)}$ and $\mathsf{Abort}_1^{(b)}$ denote the events in which experiments $\mathsf{Expt}_2^{(b)}$ and $\mathsf{Expt}_1^{(b)}$ abort respectively. Then,

$$\Pr\left[\overline{\mathsf{Abort}_1^{(b)}}\right] = \alpha \text{ and } \Pr\left[\overline{\mathsf{Abort}_2^{(b)}}\right] = \varrho^{(b)} \cdot \frac{\alpha}{\widetilde{\varrho}^{(b)}} = \alpha \cdot \frac{\varrho^{(b)}}{\widetilde{\varrho}^{(b)}}.$$

Equation (5.4) implies that with probability at least $1-2^{-\lambda}$ it holds that

$$\left|\Pr\left[\overline{\mathsf{Abort}_1^{(b)}}\right]-\Pr\left[\overline{\mathsf{Abort}_2^{(b)}}\right]\right|=\alpha\cdot\left|\frac{\widetilde{\varrho}^{(b)}-\varrho^{(b)}}{\widetilde{\varrho}^{(b)}}\right|\le\frac{\alpha^2}{S\widetilde{\varrho}^{(b)}}\le\frac{\alpha}{S}. \tag{5.5}$$

As Equation (5.4) holds for any tuple $\left(\mathbf{id}^{*(b)},\mathbf{id}^{(1)},\ldots,\mathbf{id}^{(Q)}\right)$ with probability at least $1-2^{-\lambda}$, we obtain that the statistical distance between the outputs of the experiments $\mathsf{Expt}_1^{(b)}$ and $\mathsf{Expt}_2^{(b)}$ is at most $\alpha/S+2^{-\lambda}$. ▮

As a corollary, using the triangle inequality, we get

Corollary 5.17. *For any polynomial $S=S(\lambda)$, it holds that*

$$\begin{aligned}&\left|\Pr\left[\mathsf{Expt}_1^{(0)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_1^{(1)}(\lambda)=1\right]\right|\\&\qquad\le 2\cdot\left(\frac{\alpha}{S}+\frac{1}{2^{\lambda}}\right)+\left|\Pr\left[\mathsf{Expt}_2^{(0)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_2^{(1)}(\lambda)=1\right]\right|.\end{aligned}$$

To analyze experiments $\mathsf{Expt}_2^{(b)}$ and $\mathsf{Expt}_3^{(b)}$, we need a computational assumption.

Claim 5.18. *Based on DLIN assumption, for each $b\in\{0,1\}$, it holds that*

$$\left|\Pr\left[\mathsf{Expt}_2^{(b)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_3^{(b)}(\lambda)=1\right]\right|\le\operatorname{negl}(\lambda).$$

Proof. Given a DLIN challenge $\left(g,g^{\mathbf{A}}\right)$ where $\mathbf{A}\leftarrow\mathbb{Z}_p^{3\times m}$, algorithm $\mathcal{B}$ simulates a distinguisher $\mathcal{A}$ between experiments $\mathsf{Expt}_2^{(b)}$ and $\mathsf{Expt}_3^{(b)}$ to output 0 if $\mathsf{Rk}(\mathbf{A})=2$ and 1 if $\mathsf{Rk}(\mathbf{A})=3$.

- **Key generation:** Given the DLIN challenge $(g,g^{\mathbf{A}})$, $\mathcal{B}$ sets up pp as follows. $\mathbf{A}_0$ is the first two rows of $\mathbf{A}$. $\mathcal{B}$ chooses random $\mathbf{B}\leftarrow\mathbb{Z}_p^{2\times m}$ and $\mathbf{R}_j^*\leftarrow\mathbb{Z}_p^{m\times m}$ for $j\in[n]$. Next, it chooses a hash function $H\leftarrow\mathcal{H}_{\mathsf{HK},Q}$ which define elements $h_j\in\mathbb{Z}_p$ for $j\in[n]$ in the following manner: $H(\cdot)=H_{(h_1,\ldots,h_n)}(\cdot)$ (see Section 2.5). Using these values, the algorithm sets matrices

$$\mathbf{A}_j=\mathbf{A}_0\mathbf{R}_j^*-h_j\mathbf{B}. \tag{5.6}$$

  Observe that $g^{\mathbf{A}_j}$ can be computed from $g^{\mathbf{A}_0}$ given $\mathbf{R}_j^*$, h_j, and $\mathbf{B}$. Finally, $\mathcal{B}$ chooses a random $\mathbf{v}^*\leftarrow\mathbb{Z}_p^{2m}$ and sets $\mathbf{u}=\left[\mathbf{A}_0\,\middle|\,\sum_{j\in[n]}\mathbf{A}_0\mathbf{R}_j^*\right]\mathbf{v}^*\in\mathbb{Z}_p^2$. Observe that $g^{\mathbf{u}}$ can be computed from $g^{\mathbf{A}_0}$ and $\mathbf{R}_j^*$. It publishes parameters $\mathrm{pp}=\left(g^{\mathbf{A}_0},\mathbf{B},\{g^{\mathbf{A}_j}\}_{j\in[n]},g^{\mathbf{u}}\right)$

- **Secret key queries:** On query $\mathbf{id}=(\mathrm{id}_1,\ldots,\mathrm{id}_n)\in\{0,1\}^n$, the algorithm samples a uniform matrix $\mathbf{S}\leftarrow\mathbb{Z}_p^{m\times 2}$. Let $\mathbf{\Delta}\overset{\mathsf{def}}{=}H\left(\mathbf{id}\right)\cdot\mathbf{BS}\in\mathbb{Z}_p^{2\times 2}$ If $\mathbf{\Delta}$ is not full-rank, $\mathcal{B}$ aborts and outputs a random bit. Otherwise, it chooses random $\mathbf{w}\leftarrow\mathbb{Z}_p^m$ and a random vector $\mathbf{x}$ in $\mathbb{Z}_p^m$ such that

$$\mathbf{\Delta x}=-\mathbf{A}_0\mathbf{w}+\mathbf{u}.$$

  It is easy to compute $g^{\mathbf{x}}$ given $g^{\mathbf{A}}$, $g^{\mathbf{u}}$, and $\mathbf{\Delta}$. Let

$$\mathbf{R}_{\mathbf{id},\mathbf{S}}^*\overset{\mathsf{def}}{=}\left(\sum_{j\in[n]}\mathrm{id}_j\mathbf{R}_j^*\right)\mathbf{S}.$$

The secret key component $\mathbf{v}$ is set as follows.

$$\mathbf{v} = \begin{bmatrix} \mathbf{w} - \mathbf{R}^*_{\mathbf{id},\mathbf{S}} \cdot \mathbf{x} \\ \mathbf{x} \end{bmatrix}. \tag{5.7}$$

It is easy to compute $g^{\mathbf{v}}$ given $\mathbf{S}$, $\{\mathbf{R}^*_j\}_{j\in[n]}$, $g^{\mathbf{w}}$, and $g^{\mathbf{x}}$. Observe that:

$$\begin{aligned}
\mathbf{F}_{\mathbf{id},\mathbf{S}} \cdot \mathbf{v} &= \left[\mathbf{A}_0 \middle| \mathbf{BS} + \left(\sum_{j\in[n]} \mathrm{id}_j \mathbf{A}_j \right) \mathbf{S} \right] \mathbf{v} \\
&= \left[\mathbf{A}_0 \middle| \mathbf{BS} + \left(\sum_{j\in[n]} \mathrm{id}_j \left(\mathbf{A}_0 \mathbf{R}^*_j - h_j \mathbf{B} \right) \right) \mathbf{S} \right] \mathbf{v} \\
&= \left[\mathbf{A}_0 \middle| \mathbf{A}_0 \cdot \left(\sum_{j\in[n]} \mathrm{id}_j \mathbf{R}^*_j \mathbf{S} \right) + H(\mathbf{id}) \cdot \mathbf{BS} \right] \mathbf{v} \\
&= \left[\mathbf{A}_0 \,|\, \mathbf{A}_0 \mathbf{R}^*_{\mathbf{id},\mathbf{S}} + \mathbf{\Delta} \right] \begin{bmatrix} \mathbf{w} - \mathbf{R}^*_{\mathbf{id},\mathbf{S}} \cdot \mathbf{x} \\ \mathbf{x} \end{bmatrix} \\
&= \mathbf{A}_0 \mathbf{w} - \mathbf{A}_0 \mathbf{R}^*_{\mathbf{id},\mathbf{S}} \cdot \mathbf{x} + \mathbf{A}_0 \mathbf{R}^*_{\mathbf{id},\mathbf{S}} \cdot \mathbf{x} + \mathbf{\Delta x} \\
&= \mathbf{u}.
\end{aligned}$$

To answer the secret key query, $\mathcal{B}$ outputs $\mathrm{sk}_{\mathbf{id}} = (\mathbf{S}, g^{\mathbf{v}})$.

- **Challenge query:** On challenge query $\left(\mathbf{id}^{*(0)}, \mathfrak{m}^*_0\right)$ and $\left(\mathbf{id}^{*(1)}, \mathfrak{m}^*_1\right)$ the algorithm $\mathcal{B}$ proceeds as follows. It sets $\mathbf{id}^* = \mathbf{id}^{*(b)}$ and $\mathfrak{m}^* = \mathfrak{m}^*_b$ depending on the bit b. If $H(\mathbf{id}^*) \neq 0$, abort and output a uniform bit. Otherwise, let $[-\mathbf{y}^\intercal-] \in \mathbb{Z}_p^{1\times m}$ denote the third row of $\mathbf{A}$. Let $\mathbf{R}^* \overset{\mathsf{def}}{=} \sum_{j\in[n]} \mathrm{id}^*_j \mathbf{R}^*_j$. The challenge encryption is constructed by $\mathcal{B}$ as follows:

$$((\mathbf{c}^*_0)^\intercal, (\mathbf{c}^*_1)^\intercal, c^*_2) = \left(g^{\mathbf{y}^\intercal}, g^{\mathbf{y}^\intercal \mathbf{R}^*}, \hat{e}(g,g)^{[\mathbf{y}^\intercal \,|\, \mathbf{y}^\intercal \mathbf{R}^*]\mathbf{v}^*} \cdot \mathfrak{m}^* \right).$$

- **Output:** The simulator $\mathcal{B}$ receives b' from $\mathcal{A}$ and proceeds as follows. It first does the abort check and artificial abort as in experiment $\mathsf{Expt}_2^{(b)}$ and outputs either b' or a random bit.

We argue next that the adversary's view in the simulation is statistically close to its view in the real scheme.

(a) **Public parameters:** We argue that the public parameters are distributed statistically close to the real distribution. We note that the matrices $\mathbf{R}^*_j$ for $j \in [n]$ are used to construct the public parameters, answer secret key queries, and construct the challenge ciphertext. Below we show how the secret key queries are distributed *identically* to the real scheme, so they are independent of $\mathbf{R}^*_j$. Next, from the extended leftover hash lemma (cf. Lemma 2.5) setting $k = nm$, we observe that the two distributions

$$(\mathbf{A}_0, \mathbf{A}_0 \cdot [\mathbf{R}^*_1 | \cdots | \mathbf{R}^*_n], [\mathbf{R}^*_1 | \cdots | \mathbf{R}^*_n]^\intercal \mathbf{y}) \quad \text{and} \quad \left(\mathbf{A}_0, \left[\widetilde{\mathbf{A}}_1 \middle| \cdots \middle| \widetilde{\mathbf{A}}_n \right], [\mathbf{R}^*_1 | \cdots | \mathbf{R}^*_n]^\intercal \mathbf{y} \right)$$

are statistically close under our choice of parameters,[15] where $\widetilde{\mathbf{A}}_j$ for $j \in [n]$ are matrices chosen independently and uniformly from $\mathbb{Z}_p^{2\times m}$. Observe that the challenge ciphertext is a

[15] Recollect that we require $m \geq 3 + \frac{\omega(\log \lambda)}{\log p}$ and as p is a λ-bit prime, setting $m > 3$ suffices for sufficiently large λ.

deterministic function of the third component. Thus, even given the (specially constructed) challenge ciphertext, the second component is statistically close to uniform matrices over $\mathbb{Z}_p^{2\times m}$. The public parameters are simply the matrices in the second component with $[h_1\mathbf{B}|\cdots| h_n\mathbf{B}]$ added to them. And finally, consider $\mathbf{u} = \left[\mathbf{A}_0 \,\middle|\, \sum_{j\in[n]} \mathbf{A}_0\mathbf{R}_j^*\right] \mathbf{v}^*$. As $\mathbf{v}$ is sampled uniformly from $\mathbb{Z}_p^{2m}$ and with overwhelming probability, $\left[\mathbf{A}_0 \,\middle|\, \sum_{j\in[n]} \mathbf{A}_0\mathbf{R}_j^*\right]$ is full-rank, in the simulation, $\mathbf{u}$ is distributed identically to its distribution in the real scheme. Thus, we conclude that the distribution of parameters $\big(\mathbf{A}, \{\mathbf{A}_j\}_{j\in[n]}, \mathbf{B}, \mathbf{u}\big)$ is statistically close to the real distribution.

(b) **Secret keys:** Next, we argue that the answers to secret key queries are distributed correctly. If the simluation doesn't abort, observe that $\mathbf{S}$ is distributed as in the real scheme. We show that $\mathbf{v}$ (and hence $\mathbf{z}$) is distributed identically to the real scheme. Observe that $\mathbf{v}$ in the real scheme satisfies $\mathbf{F}_{\mathbf{id},\mathbf{S}} \cdot \mathbf{v} = \mathbf{u} \pmod q$. Therefore $\mathbf{v}$ is chosen from a subspace of dimension m from the constraints of the above equation. In the simulation, $\mathbf{w}$ is chosen uniformly from $\mathbb{Z}_p^m$ and $\mathbf{x}$ is uniquely determined by the constraints in equation (5.7). Therefore, $\mathbf{v}$ comes from a subspace of dimension m as required.

(c) **Challenge ciphertext:** And finally, we argue that if $\mathsf{Rk}(\mathbf{A}) = 2$, then the challenge ciphertext is well-formed and if $\mathsf{Rk}(\mathbf{A}) = 3$, then the challenge ciphertext is distributed statistically close to uniform over $\mathbb{G}^{2m} \times \mathbb{G}_{\mathrm{T}}$ and independently of $\mathcal{A}$'s view.

- **Case 1, $\mathsf{Rk}(\mathbf{A}) = 2$:** We have that $\mathbf{y}^\intercal = \mathbf{r}^\intercal\mathbf{A}_0$ for some $\mathbf{r} \in \mathbb{Z}_p^2$. Therefore, we have the following: $g^{\mathbf{y}^\intercal} = g^{\mathbf{r}^\intercal\mathbf{A}_0}$,

$$\begin{aligned} g^{\mathbf{y}^\intercal\mathbf{R}^*} &= g^{\mathbf{r}^\intercal\mathbf{A}_0\mathbf{R}^*} \\ &= g^{\mathbf{r}^\intercal\left[\sum_{j\in[n]} \mathbf{A}_0 \mathrm{id}_j^* \mathbf{R}_j^*\right]} \\ &= g^{\mathbf{r}^\intercal\left[\sum_{j\in[n]} \mathrm{id}_j^*\mathbf{A}_j + \mathbf{B} - H(\mathbf{id})\mathbf{B}\right]} \\ &= g^{\mathbf{r}^\intercal[\mathbf{B}+\mathbf{D}(\mathbf{id})]} \end{aligned} \qquad \begin{aligned} &\hat{e}(g,g)^{[\mathbf{y}^\intercal \,|\, \mathbf{y}^\intercal\mathbf{R}^*]\mathbf{v}^*} \\ &= \hat{e}(g,g)^{\mathbf{r}^\intercal\left[\mathbf{A}_0 \,\middle|\, \sum_{j\in[n]} \mathbf{A}_0\mathbf{R}_j^*\right]\mathbf{v}^*} \\ &= \hat{e}(g,g)^{\mathbf{r}^\intercal\mathbf{u}}. \end{aligned}$$

  Note that $\mathbf{r}$ is distributed uniformly in $\mathbb{Z}_p^2$ by definition. Thus, the ciphertext is well-formed.

- **Case 2, $\mathsf{Rk}(\mathbf{A}) = 3$:** We have that $\mathbf{y}$ is uniform in $\mathbb{Z}_p^m$ and independent of $\mathbf{A}_0$. We consider $\mathcal{A}$'s view and argue that the challenge ciphertext is distributed uniformly over $(\mathbb{G}^m)^{2m} \times \mathbb{G}_{\mathrm{T}}$ and independent of $\mathcal{A}$'s view. It suffices to argue the distribution of the ciphertext in an information-theoretic sense (against a computationally unbounded adversary). $\mathcal{A}$'s view in the simulation comprises the public parameters $(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_n, \mathbf{B}, \mathbf{u})$ and the challenge ciphertext $((\mathbf{c}_0^*), (\mathbf{c}_1^*), c_2^*)$. As $\mathcal{A}$ is unbounded, the secret key queries do not reveal any extra information and can be simulated by an unbounded adversary itself. Let $\mathbf{U}_j^* = \mathbf{A}_0\mathbf{R}_j^*$. First note that as $\mathbf{y}$ is uniform over $\mathbb{Z}_p$, so is $\mathbf{c}_0^*$. Observe that for every $j \in [n]$ and for every possible $\mathbf{d}_j^* \in \mathbb{Z}_p^m$ the number of solutions $\mathbf{R}_j^*$ such that

$$\begin{bmatrix} \mathbf{A}_0 \\ \mathbf{y}^\intercal \end{bmatrix} \cdot \mathbf{R}_j^* = \begin{bmatrix} \mathbf{A}_0\mathbf{R}_j^* \\ \mathbf{y}^\intercal\mathbf{R}_j^* \end{bmatrix} = \begin{bmatrix} \mathbf{U}_j^* \\ \mathbf{d}_j^{*\intercal} \end{bmatrix}$$

  is the same. Thus, even given $\mathbf{U}_j^*$ (which can be computed from $\mathbf{A}_j$, $\mathbf{B}$) as $\mathbf{R}_j^*$ is chosen uniformly from $\mathbb{Z}_p^{m\times m}$ each $\mathbf{d}_j^*$ is distributed uniformly over $\mathbb{G}^m$ for every $j \in [n]$. As p is prime, for any $\mathbf{id}^*$, $\sum_{j\in[n]} \mathrm{id}_j^*\mathbf{R}_j^*$ and hence $\mathbf{c}_1^*$ is uniform.

Next, observe that $\mathbf{v}^*$ has min-entropy $2m \log p$ and given $\mathbf{u}$, from Lemma 2.1 with probability at least $1 - \epsilon$ over choices of $\mathbf{u}$, $\mathbf{v}^*$ still has min-entropy $(2m-2)\log p - \log(1/\epsilon)$ for every negligible $\epsilon = \epsilon(\lambda)$. Next, we consider $d_2 = \left[\mathbf{y}^\intercal \,\middle|\, \sum_{j\in[n]} \mathbf{d}_j^{*\intercal}\right] \mathbf{v}^*$ which can be written as $\mathbf{f}^\intercal \mathbf{v}^*$ for a uniformly distributed vector $\mathbf{f}$ in $\mathbb{Z}_p^m$. As d_2 is of length $\log p$ bits, the vector $\mathbf{v}^*$ has sufficient min-entropy (more precisely, at least $\log p + \omega(\log \lambda)$ bits) so that $\mathbf{f}$ acts as an 'inner-product' extractor when applied to it. Therefore, we have $(\mathbf{f}^\intercal, \mathbf{f}^\intercal \mathbf{v}^*) \approx (\mathbf{f}^\intercal, r)$ where $\mathbf{f}$ is uniform in $\mathbb{Z}_p^{2m}$ and r is uniform in $\mathbb{Z}_p$. This implies, in particular, that the last component of the ciphertext, $\hat{e}(g,g)^{d_2} \cdot \mathfrak{m}^*$ is distributed uniformly over $\mathbb{G}_\mathrm{T}$.

This concludes the proof that the challenge ciphertext $((\mathbf{c}_0^*)^\intercal, (\mathbf{c}_1^*)^\intercal, c_2^*)$ is distributed uniformly over $\mathbb{G}^{2m} \times \mathbb{G}_\mathrm{T}$.

To complete the proof of Claim 5.18, observe that the hash function H is independent of the view of the adversary as the public parameters are distributed statistically close to the real distribution. Additionally, the challenger $\mathcal{B}$ aborts only in the following cases:

1. If on input a secret key query for $\mathbf{id}$, for matrices $\mathbf{S}$ sampled in the secret key query, the matrix $\boldsymbol{\Delta} = H(\mathbf{id}) \cdot \mathbf{BS}$ is not full rank. In this case, the challenger cannot simulate a secret key.
2. If for the challenge identity $\mathbf{id}^*$, $H(\mathbf{id}^*) \neq 0$. In this case, the challenger ciphertext cannot be constructed from $\mathbf{R}_j^*$'s alone.
3. The artificial abort bit $\tilde{b}$ is set to true.

Each of the three cases above are identical to the abort conditions in $\mathsf{Expt}_2^{(b)}$ (and hence, $\mathsf{Expt}_3^{(b)}$). Thus, $\mathcal{B}$ simulates an experiment statistically close to Expt_2 if the DLIN challenge matrix $\mathbf{A}$ is of rank 2 and an experiment statistically close to Expt_3 if the DLIN challenge matrix $\mathbf{A}$ is of rank 3 which completes the proof of Claim 5.18. ∎

With Claims 5.15 and 5.18, and Corollary 5.17 derived above, we can complete the proof of Lemma 5.13.

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{DP}}_{\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}(\lambda)\\
&= \left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{DP},\mathcal{IBE},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{DP},\mathcal{IBE},\mathcal{A}}(\lambda) = 1\right]\right|\\
&= \left|\Pr\left[\mathsf{Expt}^{(0)}_{0}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{0}(\lambda) = 1\right]\right|\\
&= \frac{1}{\alpha} \cdot \left|\Pr\left[\mathsf{Expt}^{(0)}_{1}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{1}(\lambda) = 1\right]\right| && \text{(Claim 5.15)}\\
&\leq 2 \cdot \left(\frac{1}{S} + \frac{1}{\alpha \cdot 2^\lambda}\right) + \frac{1}{\alpha} \cdot \left|\Pr\left[\mathsf{Expt}^{(0)}_{2}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{2}(\lambda) = 1\right]\right| && \text{(Cor. 5.17)}\\
&\leq 2 \cdot \left(\frac{1}{S} + \frac{1}{\alpha \cdot 2^\lambda}\right) + \frac{1}{\alpha} \cdot \left(\left|\Pr\left[\mathsf{Expt}^{(0)}_{3}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{3}(\lambda) = 1\right]\right| + \mathrm{negl}(\lambda)\right) && \text{(Claim 5.18)}\\
&\leq 2 \cdot \left(\frac{1}{S} + \frac{1}{\alpha \cdot 2^\lambda}\right) + \frac{1}{\alpha} \cdot (0 + \mathrm{negl}(\lambda))
\end{aligned}$$

As α is at least $1/P(\lambda)$ for some *fixed* polynomial $P(\lambda)$ and the above result holds for every polynomial $S = S(\lambda)$, the advantage of $\mathcal{A}$ remains negligible which completes the proof of Lemma 5.13.

∎

Proof of Lemma 5.14. Fix a tuple of queries $\mathbf{id}^*, \mathbf{id}^{(1)}, \ldots, \mathbf{id}^{(Q)} \in \{0,1\}$ and full rank $\mathbf{B} \in \mathbb{Z}_p^{2\times m}$. We let Good denote the event

$$\mathsf{Good} \overset{\mathsf{def}}{=} \left\{H\left(\mathbf{id}^*\right) = 0 \wedge H\left(\mathbf{id}^{(1)}\right) \neq 0 \wedge \cdots \wedge H\left(\mathbf{id}^{(Q)}\right) \neq 0\right\}$$

over the choice of $H \leftarrow \mathcal{H}_{\mathsf{HK},Q}$. For brevity, let $\vec{\mathbf{S}} = \left\{\mathbf{S}^{(j)}\right\}_{j\in[Q]}$ denote all the choices of the matrices. We have,

$$\begin{aligned}\Pr_{H,\vec{\mathbf{S}}}[\mathsf{Event}_\mathsf{T} \wedge \mathsf{Event}_\mathsf{C}] &\geq \Pr_{H,\vec{\mathbf{S}}}[\mathsf{Event}_\mathsf{T} \wedge \mathsf{Event}_\mathsf{C} | \mathsf{Good}] \cdot \Pr[\mathsf{Good}] \\ &\geq \Pr_{H,\vec{\mathbf{S}}}[\mathsf{Event}_\mathsf{T} | \mathsf{Good}] \cdot (\alpha_\mathsf{HK}) \qquad (5.8)\end{aligned}$$

where Equation (5.8) follows from the definition of α_HK (see Section 2.5) and the fact that the event Good implies the event $\mathsf{Event}_\mathsf{C}$. Thus, it suffices to lower bound the probability of the event $\mathsf{Event}_\mathsf{T}|_\mathsf{Good}$.

To do so, fix any H such that Good occurs. This implies, in particular, that $H\left(\mathbf{id}^{(i)}\right) \neq 0$ for all $i \in [Q]$.

Consider a particular i. From Lemma 2.13 we have that if $\mathbf{S}$ is distributed uniformly over $\mathbb{Z}_p^{m\times 2}$ matrices then for any fixed full-rank $\mathbf{B} \in \mathbb{Z}_p^{2\times m}$, $\mathbf{BS}$ is also distributed uniformly over $\mathbb{Z}_p^{2\times 2}$ matrices. As $H\left(\mathbf{id}^{(i)}\right) \neq 0$ and $\mathbf{B}$ is of full rank, then the matrix $H\left(\mathbf{id}^{(i)}\right) \cdot \mathbf{BS}^{(i)}$ is uniformly distributed in $\mathbb{Z}_p^{2\times 2}$ over uniform choices of $\mathbf{S}^{(i)}$. Therefore, the probability that $H\left(\mathbf{id}^{(i)}\right) \cdot \mathbf{BS}^{(i)}$ is of full rank is the probability that a uniform matrix is at least $1 - 2/p$ (from Lemma 2.12). A straightforward union bound implies that $\Pr[\mathsf{Event}_\mathsf{T}|\mathsf{Good}]$ is at least $1 - 2Q/p$. As this is true for every H (conditioned on Good), substituting in Equation (5.8), we get

$$\begin{aligned}\Pr_{H,\vec{\mathbf{S}}}[\mathsf{Event}_\mathsf{T} \wedge \mathsf{Event}_\mathsf{C}] &\geq \left(1 - \frac{2Q}{p}\right) \cdot (\alpha_\mathsf{HK}) \\ &\geq \left(1 - \frac{2Q}{p}\right) \Theta\left(\frac{1}{Q\sqrt{n}}\right) \qquad \text{(from Lemma 2.11)}\end{aligned}$$

as required.

5.3.2 Proof of Function Privacy

Lemma 5.19. *The scheme $\mathcal{IBE}_\mathsf{DLIN2}$ is statistically function private for:*

1. *(T,k)-block-sources for any $T = \mathrm{poly}(\lambda)$ and $k \geq 4\log p + \omega(\log\lambda)$.*
2. *$(k_1,\ldots,k_T)$-sources for any $T = \mathrm{poly}(\lambda)$ and $(k_1,\ldots,k_T)$ such that $k_i \geq 4i\log p + \omega(\log\lambda)$ for every $i \in [T]$.*

Proof. Let $X \in \{(T,k)\text{-block}, (k_1,\ldots,k_T)\}$, and let $\mathcal{A}$ be a computationally unbounded X-source function-privacy adversary that makes a polynomial number $Q = Q(\lambda)$ of queries to the RoR^FP oracle. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^\mathsf{real}_{\mathsf{FP},\mathcal{IBE}_\mathsf{DLIN2},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^\mathsf{rand}_{\mathsf{FP},\mathcal{IBE}_\mathsf{DLIN2},\mathcal{A}}$ (we refer the reader to Definition 3.3 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_\mathsf{real}$ and $\mathsf{View}_\mathsf{rand}$, respectively.

Let $\vec{\mathbf{A}}$ denote all the matrices $\mathbf{A}_j \in \mathbb{Z}_p^{2\times m}$ for $j \in [n]$. We define the function family

$$\mathcal{F} \overset{\mathsf{def}}{=} \left\{ f_{\mathbf{S}}^{(\vec{\mathbf{A}})} : \{0,1\}^n \to \mathbb{Z}_p^{2\times 2} \right\}_{\mathbf{S}\in\mathbb{Z}_p^{2\times 2}} \text{ as } f_{\mathbf{S}}^{(\vec{\mathbf{A}})}(\mathbf{id}) \overset{\mathsf{def}}{=} \left(\sum_{j\in[n]} \mathrm{id}_j \mathbf{A}_j \right) \mathbf{S}. \tag{5.9}$$

We argue that such a function family is universal with an overwhelmingly high probability over the choice of $\mathbf{A}_j$'s. We start off with the following lemma.

Claim 5.20. *Let* $\mathsf{LowRank}$ *denote the event (over the choices of* $\vec{\mathbf{A}}$*) that there is a not-all-zero sum of* $\mathbf{A}_1, \dots, \mathbf{A}_n$ *with coefficients in* $\{-1,0,1\}$ *that is of rank less than 2. Then we have*

$$\Pr_{\mathbf{A}_j \leftarrow \mathbb{Z}_p^{2\times m}}[\mathsf{LowRank}] \le \frac{2\cdot 3^n}{p^{m-1}}. \tag{5.10}$$

Proof. Fix $\alpha_1, \dots, \alpha_n \in \{-1,0,1\}$ such that not all $\alpha_j = 0$. Observe that over choices of $\mathbf{A}_j$, $\sum_{j\in[n]} \alpha_j \mathbf{A}_j$ is uniformly distributed over $\mathbb{Z}_p^{2\times m}$. Therefore, applying Lemma 2.12, we have that $\Pr\left[\mathsf{Rk}\left(\sum_{j\in[n]} \alpha_j \mathbf{A}_j\right) < 2\right] \le 2/p^{m-1}$. A straightforward union bound over all choices of α_j's gives us that

$$\Pr\left[\exists\, \alpha_1, \dots, \alpha_n \in \{-1,0,1\} \text{ such that } \mathsf{Rk}\left(\sum_{j\in[n]} \alpha_j \mathbf{A}_j\right) < 2\right] \le 3^n \cdot \frac{2}{p^{m-1}}.$$

∎

Claim 5.21. *With all but a negligible probability over the choice of* $\vec{\mathbf{A}}$*, the function family* $\mathcal{F}$ *defined in Equation* (5.9) *is universal.*

Proof. In Claim 5.20 we showed that the event $\mathsf{LowRank}$ occurs with only a negligible probability. Thus, it suffices to show that for any fixing of the $\mathbf{A}_j$'s such that $\mathsf{LowRank}$ does not occur, the function family $\mathcal{F}$ is universal. From this point on we fix the $\mathbf{A}_j$'s such that $\mathsf{LowRank}$ does not occur. We need to prove that for any two distinct identities $\mathbf{id}, \mathbf{id}' \in \{0,1\}^n$ it holds that

$$\Pr_{\mathbf{S}\leftarrow\mathbb{Z}_p^{m\times 2}}\left[\left(\sum_{j\in[n]} (\mathrm{id}_j - \mathrm{id}'_j)\mathbf{A}_j\right)\mathbf{S} = \mathbf{0}\right] \le \frac{1}{p^4}. \tag{5.11}$$

As $\mathbf{id} \neq \mathbf{id}'$ there exists an index $j^* \in [n]$ such that $\mathrm{id}_{j^*} \neq \mathrm{id}'_{j^*}$. The fact that the event $\mathsf{LowRank}$ does not occur, guarantees that the matrix $\mathbf{V} \overset{\mathsf{def}}{=} \left(\sum_{j\in[n]}(\mathrm{id}_j - \mathrm{id}'_j)\mathbf{A}_j\right)$ is of rank 2, and therefore the matrix $\mathbf{V}\cdot\mathbf{S}$ is uniformly distributed (according to Lemma 2.13) Therefore,

$$\begin{aligned}
\Pr_{\mathbf{S}\leftarrow\mathbb{Z}_p^{m\times 2}}\left[\left(\sum_{j\in[n]} (\mathrm{id}_j - \mathrm{id}'_j)\mathbf{A}_j\right)\mathbf{S} = \mathbf{0}\right] &\le \Pr_{\mathbf{S}\leftarrow\mathbb{Z}_p^{m\times 2}}[\mathbf{V}\mathbf{S} = \mathbf{0}] \\
&\le \Pr_{\mathbf{U}\leftarrow\mathbb{Z}_p^{2\times 2}}[\mathbf{U} = \mathbf{0}] \\
&\le \frac{1}{p^4},
\end{aligned}$$

as required. ∎

Thus, the function family in Equation (5.9) is universal conditioned on the event $\overline{\mathsf{LowRank}}$. Now, as before, we fix the public parameters pp and the master secret key msk of the scheme to show that the two distributions $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$ are statistically close for any such pp and msk. Next, as the adversary $\mathcal{A}$ is computationally unbounded, we assume without loss of generality that $\mathcal{A}$ does not query the $\mathsf{KeyGen}(\mathrm{msk}, \cdot)$ oracle. In addition, as discussed in Section 3.1, we can assume that $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once.

Denote by $\boldsymbol{ID} = \left(ID^{(1)}, \ldots, ID^{(T)}\right)$ the random variable corresponding to the X-source with which $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. Having already fixed pp and msk, observing that $\mathbf{B}_i$'s are independent of the identity $\mathbf{id}$, we can assume that

$$\begin{aligned}\mathsf{View}_{\mathsf{mode}} &= \left(\mathbf{S}^{(1)}, \left(\sum_{j\in[n]} \mathrm{id}_j^{(1)}\mathbf{A}_j\right)\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(T)}, \left(\sum_{j\in[n]} \mathrm{id}_j^{(T)}\mathbf{A}_j\right)\mathbf{S}^{(T)}\right) \\ &= \left(\mathbf{S}^{(1)}, f_{\mathbf{S}^{(1)}}\left(\mathbf{id}^{(1)}\right), \ldots, \mathbf{S}^{(T)}, f_{\mathbf{S}^{(T)}}\left(\mathbf{id}^{(T)}\right)\right)\end{aligned}$$

for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, where $\left(\mathbf{id}^{(1)}, \ldots, \mathbf{id}^{(T)}\right) \leftarrow \left(ID^{(1)}, \ldots, ID^{(T)}\right)$ for $\mathsf{mode} = \mathsf{real}$, $\left(\mathbf{id}^{(1)}, \ldots, \mathbf{id}^{(T)}\right)$ is uniformly distributed over $(\mathcal{ID}_\lambda)^T$ for $\mathsf{mode} = \mathsf{rand}$, $\mathbf{S}^{(i)} \leftarrow \mathbb{Z}_q^{m\times 2}$ for every $i \in [T]$. For $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ we prove that the distribution $\mathsf{View}_{\mathsf{mode}}$ is statistically-close to uniform.

We know that $\left(ID^{(1)}, \ldots, ID^{(T)}\right)$ is an X-source, and the collection of functions defined in Equation (5.10) is universal. This enables us to directly apply Lemma 2.3 (in case $\boldsymbol{ID}$ is a (T, k)-block-source) and Lemma 2.4 (in case $\boldsymbol{ID}$ is a $(k_1, \ldots, k_T)$-source) with the function family $\mathcal{F}$, implying that the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ. The same clearly holds also for $\mathsf{View}_{\mathsf{rand}}$, as the uniform distribution over $(\mathcal{ID}_\lambda)^T$ is, in particular, a (T, k)-block-source and a $(k_1, \ldots, k_T)$-source. From our choice of parameters, $\mathsf{LowRank}$ occurs with negligible probability which concludes the proof. ∎

5.4 Enhanced Function Privacy of the Fully-Secure DLIN-Based Scheme

In this section we prove the following theorem:

Theorem 5.22. *The scheme $\mathcal{IBE}_{\mathsf{DLIN2}}$ is enhanced function private for:*

1. *(T, k)-block-sources for any constant T and $k \geq 4\log p + \omega(\log\lambda)$.*
2. *$(k_1, \ldots, k_T)$-sources for any constant T and $(k_1, \ldots, k_T)$ such that $k_i \geq 4i\log p + \omega(\log\lambda)$ for every $i \in [T]$.*

Proof outline. To prove the enhanced function privacy of the scheme $\mathcal{IBE}_{\mathsf{DLIN2}}$ we consider the following hybrids. In the first hybrid, the oracles $\mathsf{RoR}^{\mathsf{FP}}$ and $\mathsf{Enc}^{\mathsf{FP}}$ are as in Definition 3.4 with $\mathsf{mode} = \mathsf{real}$. In the second hybrid, the oracle $\mathsf{Enc}^{\mathsf{FP}}$ is modified to output ciphertexts that are generated uniformly at random and independent of id, subject to decrypting correctly for the corresponding key $\mathrm{sk}_{\mathrm{id}}$ generated by $\mathsf{RoR}^{\mathsf{FP}}$. To show that the two hybrids are computationally indistinguishable, we follow the proof of data privacy (see Section 5.3.1) where a DLIN challenge is embedded to produce either well-formed or ill-formed ciphertexts.

A statistical argument nearly identical to the proof of function privacy of the scheme (see Section 5.3.2) shows that the view of the adversary in the second hybrid is statistically close to the view of the adversary in a third hybrid where ciphertexts remain ill-formed, but $\mathsf{RoR}^{\mathsf{FP}}$ outputs secret keys with $\mathsf{mode} = \mathsf{rand}$. Finally, as in moving from the first hybrid to the second hybrid, we consider a fourth hybrid indistinguishable (under the DLIN assumption) from the third one in which the oracles $\mathsf{RoR}^{\mathsf{FP}}$ and $\mathsf{Enc}^{\mathsf{FP}}$ are as in Definition 3.4 with $\mathsf{mode} = \mathsf{rand}$.

Proof of Theorem 5.22. Let $X \in \{(T,k)\text{-block}, (k_1,\dots,k_T)\}$, and let $\mathcal{A}$ be a probabilistic polynomial-time X-source enhanced function-privacy adversary. As discussed in Section 3.2, we can assume that $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once[16]. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{EFP},\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{EFP},\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}$ (we refer the reader to Definition 3.4 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$, respectively.

Denote by $\mathbf{id}^{(1)},\dots,\mathbf{id}^{(T)}$ the T identities sampled from either the adversary's query $\boldsymbol{ID}$ or sampled uniformly from $\mathcal{ID}^T$ by $\mathsf{RoR}^{\mathsf{FP}}$. Additionally, let $\mathbf{id}^{(T+1)},\dots,\mathbf{id}^{(T+Q)}$ denote the Q queries generated by $\mathcal{A}$ to the oracle $\mathsf{KeyGen}(\mathrm{msk},\cdot)$.

Let $\alpha = \alpha(\lambda) \in [0,1]$ be a non-negligible function of the security parameter that will be determined later on (see the description of $\mathsf{Expt}_2^{(j)}$ and Lemma 5.14). We consider the following experiments for each $j \in [T]$.

- Experiment $\mathsf{Expt}_0^{(1)}$ is identical to $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{EFP},\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}$ as in Definition 3.4.

- Experiment $\mathsf{Expt}_1^{(1)}$ is obtained from $\mathsf{Expt}_0^{(1)}$ by outputting the output of $\mathsf{Expt}_0^{(1)}$ with probability α and a random bit with probability $1-\alpha$ (denoted by Abort).

- Experiment $\mathsf{Expt}_2^{(1)}$ is obtained from $\mathsf{Expt}_0^{(1)}$ by introducing an "artificial" abort event independent of the adversary's view. We use the programmable family of hash functions introduced by Hofheinz and Kiltz [HK12] denoted $\mathcal{H}_{\mathsf{HK},Q+T}$ (see Section 2.5). At the end of $\mathsf{Expt}_2^{(1)}$, we sample a hash function $H \leftarrow \mathcal{H}_{\mathsf{HK},Q+T}$. When $\mathsf{Expt}_2^{(1)}$ receives the guess b' from $\mathcal{A}$, it does the following:

  1. *Abort check:* For each query $\mathbf{id}^{(i)}$ for $i \in [Q]$, let $\mathbf{S}^{(i)} \in \mathbb{Z}_p^{m\times 2}$ denote the uniform matrix chosen during secret key generation. The challenger checks the following conditions:
     (a) For each $i \in [Q+T]\backslash\{1\}$, if $H\left(\mathbf{id}^{(i)}\right)\cdot\mathbf{BS}^{(i)} \in \mathbb{Z}_p^{2\times 2}$ is full-rank.
     (b) $H\left(\mathbf{id}^{*(1)}\right) = 0$.
     If either (or both) these conditions are not satisfied, the experiment outputs a random bit instead of b'. Let α denote the probability over choices of the hash function H (for any particular set of distinct queries $\left(\mathbf{id}^{(1)},\mathbf{id}^{(2)},\dots,\mathbf{id}^{(Q+T)}\right)$) that both conditions above are *true*. Recollect that Lemma 5.14 derives a bound for α.
  2. *Artificial abort:* Following the approach of Cash et al. [CHK+10] (generalizing that of Waters [Wat05]) approximate $\varrho^{(1)} = \Pr\left[\overline{\mathsf{Abort}}\middle|\left(\mathbf{id}^{(1)},\mathbf{id}^{(2)},\dots,\mathbf{id}^{(Q+T)}\right)\right]$ by sampling sufficiently many independent hash functions. For any polynomial $S = S(\lambda)$, Hoeffding's inequality yields that with $\lceil \lambda S/\alpha \rceil$ samples, we can obtain an approximation $\widetilde{\varrho}^{(1)} \geq \alpha$ of $\varrho^{(1)}$ such that:
     $$\Pr\left[\left|\varrho^{(1)} - \widetilde{\varrho}^{(1)}\right| \geq \frac{\alpha}{S}\right] \leq \frac{1}{2^\lambda}, \tag{5.12}$$
     for security parameter λ. The challenger samples a random bit $\widetilde{b} \in \{0,1\}$ such that $\Pr\left[\widetilde{b} = 1\right] = 1 - \alpha/\widetilde{\varrho}^{(1)} \in [0,1]$. If $\widetilde{b} = 1$ then the adversary outputs a random bit (artificial abort). Else, it outputs the bit b' from the challenger.

[16] Given that $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once, recall that $\mathsf{Enc}^{\mathsf{FP}}$ now takes as input queries of the form $(j,\mathfrak{m})$ for $j \in [T]$, and outputs an encryption of $\mathfrak{m}$ under the identity id_j, where $(\mathrm{id}_1,\dots,\mathrm{id}_T)$ is the vector of identities that was sampled by the real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$.

- Experiment $\mathsf{Expt}_3^{(1)}$ is obtained from $\mathsf{Expt}_2^{(1)}$ as follows. Let $\left(\mathbf{S}^{(1)}, \mathbf{z}_1\right)$ denote $\mathrm{sk}_{\mathbf{id}^{(1)}}$, and replace the outputs of the oracle $\mathsf{Enc}^{\mathsf{FP}}$ on query $(1, \mathfrak{m})$ with uniform $(\mathbf{c}_0, \mathbf{c}_1, c_2) \leftarrow \mathbb{G}^{2m} \times \mathbb{G}_{\mathrm{T}}$ sampled independent of the view of $\mathcal{A}$ subject to

$$\mathfrak{m} = c_2 \cdot \hat{e}\left(\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1^{\mathbf{S}^{(1)}} \end{bmatrix}, \begin{matrix} | \\ \mathbf{z}_1 \\ | \end{matrix}\right)^{-1}.$$

  We refer to these ciphertexts as *ill-formed* ciphertexts as they are generated independently of $\mathbf{id}^{(1)}$ and depend only on $\mathrm{sk}_{\mathbf{id}^{(1)}}$

- For $2 \leq j \leq T$, experiment $\mathsf{Expt}_0^{(j)}$ is identical to $\mathsf{Expt}_3^{(j-1)}$.

- For $2 \leq j \leq T$, experiments $\mathsf{Expt}_1^{(j)}$ through $\mathsf{Expt}_3^{(j)}$ are derived starting from $\mathsf{Expt}_0^{(j)}$ in a manner identical to how experiments $\mathsf{Expt}_1^{(1)}$ through $\mathsf{Expt}_3^{(1)}$ are derived above, starting from $\mathsf{Expt}_0^{(1)}$. The abort check and artificial aborts concentrate on $\mathbf{id}^{(j)}$ in place of $\mathbf{id}^{(1)}$.

Additionally, we define the corresponding experiments $\widetilde{\mathsf{Expt}}_0^{(j)}, \ldots, \widetilde{\mathsf{Expt}}_3^{(j)}$ that are derived starting from $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{EFP},\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}$ (see Definition 3.4). Also, let $P_i^{(j)}$ and $\widetilde{P}_i^{(j)}$ denote respectively the probabilities $\Pr\left[\mathsf{Expt}_i^{(j)} = 1\right]$ and $\Pr\left[\widetilde{\mathsf{Expt}}_i^{(j)} = 1\right]$. It immediately follows that for all $1 \leq j \leq T-1$,

$$P_0^{(i+1)} = P_3^{(i)} \text{ and } \widetilde{P}_0^{(i+1)} = \widetilde{P}_3^{(i)} \tag{5.13}$$

Observe that in $\mathsf{Expt}_3^{(T)}$ and $\widetilde{\mathsf{Expt}}_3^{(T)}$, the adversary's view comprises secret keys $\mathrm{sk}_{\mathbf{id}_1}, \ldots, \mathrm{sk}_{\mathbf{id}_T}$ and ill-formed ciphertexts that are independent of the identities. Following the proof of Lemma 5.19, it holds that the distributions

$$(\mathrm{pp}, \mathrm{msk}, \mathrm{sk}_{\mathbf{id}^{(1)}}, \ldots, \mathrm{sk}_{\mathbf{id}^{(T)}}) \tag{5.14}$$

are statistically close where the identities are sampled as in mode real and rand respectively. The experiments $\mathsf{Expt}_3^{(T)}$ and $\widetilde{\mathsf{Expt}}_3^{(T)}$ have identical abort conditions and the rest of the adversary's view in each of these experiment is a function of the distribution in Equation (5.14). Thus, it holds that

$$\left|P_3^{(T)} - \widetilde{P}_3^{(T)}\right| \leq \mathrm{negl}(\lambda). \tag{5.15}$$

In what follows, we show that $\left|P_0^{(1)} - \widetilde{P}_0^{(1)}\right|$ is negligible (following the lines of the proof of Lemma 5.13). Additionally, we require a lower bound for α for which we can apply Lemma 5.14 (as in the proof of Lemma 5.13 with $Q + T - 1$ instead of Q).

Observe that experiments $\mathsf{Expt}_0^{(j)}$, $\mathsf{Expt}_1^{(j)}$, and $\mathsf{Expt}_2^{(j)}$ only involve the artificial abort and the programmable hash function family. Therefore, following the proofs of Claims 5.15 and 5.16 and Corollary 5.17, we can state the following corresponding claim and corollary.

Claim 5.23. *It holds that*

$$\left|P_1^{(0)} - \widetilde{P}_1^{(1)}\right| = \alpha \cdot \left|P_0^{(0)} - \widetilde{P}_0^{(1)}\right|.$$

Claim 5.24. *For any polynomial $S = S(\lambda)$, for every $j \in [T]$, it holds that*

$$\left|P_1^{(j)} - \widetilde{P}_1^{(j)}\right| \leq 2 \cdot \left(\frac{\alpha}{S} + \frac{1}{2^\lambda}\right) + \left|P_2^{(j)} - \widetilde{P}_2^{(j)}\right|.$$

Next, we state the following claim that analyzes the experiments $\mathsf{Expt}_2^{(j)}$ and $\mathsf{Expt}_3^{(j)}$. The structure of the DLIN assumption allows us to use the simulation in the proof of Claim 5.18 with some small modifications to simulate the adversary's view in the *enhanced* function privacy experiment. An identical argument holds for the experiments $\widetilde{\mathsf{Expt}_2}^{(j)}$ and $\widetilde{\mathsf{Expt}_3}^{(j)}$.

Claim 5.25. *Based on the DLIN assumption, it holds that*

$$\left|P_2^{(j)} - P_3^{(j)}\right| \leq \text{negl}(\lambda) \quad \textit{and} \quad \left|\widetilde{P}_2^{(j)} - \widetilde{P}_3^{(j)}\right| \leq \text{negl}(\lambda).$$

Proof. We fix j for the rest of the proof. As stated above, we only consider experiments $\mathsf{Expt}_2^{(j)}$ and $\mathsf{Expt}_3^{(j)}$ and note that an identical proof works for $\widetilde{\mathsf{Expt}_2}^{(j)}$ and $\widetilde{\mathsf{Expt}_3}^{(j)}$.

For simplicity, in the proof we focus on adversaries $\mathcal{A}$ that query the $\mathsf{Enc}^{\mathsf{FP}}$ oracle only once.[17] Let $\mathbf{id}_1, \ldots, \mathbf{id}_T$ be the T identities sampled from $\boldsymbol{ID}$. Given a DLIN challenge $\left(g, g^{\mathbf{A}}\right)$ where $\mathbf{A} \leftarrow \mathbb{Z}_p^{3\times m}$, we construct an algorithm $\mathcal{B}$ that simulates a distinguisher $\mathcal{A}$ between experiments $\mathsf{Expt}_2^{(j)}$ and $\mathsf{Expt}_3^{(j)}$ to output 0 if $\mathsf{Rk}(\mathbf{A}) = 2$ and 1 if $\mathsf{Rk}(\mathbf{A}) = 3$. Let $\mathbf{A}_0$ denote the first two rows of $\mathbf{A}$.

- **Key generation:** The key generation algorithm sets up matrices $\mathbf{A}_0$, $\mathbf{B}$, and $\mathbf{A}_i$ for $i \in [n]$ as in the proof of Claim 5.18 (see Equation (5.6)). Additionally, the algorithm samples $\mathbf{S}^{(j)} \leftarrow \mathbb{Z}_p^{m\times 2}$ and a random $\mathbf{v}_j \leftarrow \mathbb{Z}_p^{m+2}$ and sets (implicitly) $\mathbf{u} = \left[\mathbf{A}_0 \,\middle|\, \sum_{i\in[n]} \mathbf{A}_0 \mathbf{R}_i^* \mathbf{S}^{(j)}\right] \cdot \mathbf{v}_j \in \mathbb{Z}_p^2$. The public parameters are setup such that if $H(\mathbf{id}_j) = 0$, then

$$\text{sk} = \left(\mathbf{S}^{(j)}, g^{\mathbf{v}_j}\right) \tag{5.16}$$

  is a valid secret key for the identity $\mathbf{id}_j$. Observe that $g^{\mathbf{u}}$ can be computed given $g^{\mathbf{A}_0}$ and matrices $\mathbf{R}_i^*$.

- **Secret key queries:** Secret key queries on identities $\{\text{id}_{T+1}, \ldots, \text{id}_{T+Q}\}$ are answered identically (including the abort condition) to secret key queries in the proof of Claim 5.18. Additionally, $\mathcal{B}$ runs the secret key algorithm on queries $\{\mathbf{id}_1, \ldots, \mathbf{id}_T\} \backslash \{\mathbf{id}_j\}$. The secret key $\text{sk}_{\mathbf{id}_j}$ is constructed during key generation (see Equation (5.16)). In the rest of the proof, for all $i \in [Q+T]$ we let $\left(\mathbf{S}^{(i)}, \mathbf{z}_i\right)$ denote secret keys $\text{sk}_{\mathbf{id}_i}$.

- **Encryption oracle query:** On input $(i, \mathfrak{m})$, the algorithm considers the following cases:

  1. $i < j$. The algorithm outputs ill-formed ciphertexts as follows: it samples uniform $(\mathbf{c}_0, \mathbf{c}_1, c_2) \leftarrow \mathbb{G}^{2m} \times \mathbb{G}_{\mathrm{T}}$ independently of the view of $\mathcal{A}$ subject to

$$\mathfrak{m} = c_2 \cdot \hat{e}\left(\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1^{\mathbf{S}^{(i)}} \end{bmatrix}, \begin{matrix} | \\ \mathbf{z}_j \\ | \end{matrix}\right)^{-1}.$$

  2. $i > j$. The algorithm outputs well-formed ciphertexts by running $\mathsf{Enc}(\text{pp}, \mathbf{id}_i, \mathfrak{m})$.

[17] In fact, it is easily observed that the DLIN challenge can be embedded as the output of any particular $\mathsf{Enc}^{\mathsf{FP}}$ query, and therefore a straightforward hybrid argument across the $\mathsf{Enc}^{\mathsf{FP}}$ queries can be applied to the proof to extend it to multiple $\mathsf{Enc}^{\mathsf{FP}}$ queries.

3. $i = j$. Recollect that it suffices to consider a single $\mathsf{Enc}^{\mathsf{FP}}$ oracle query. In this case, the algorithm $\mathcal{B}$ embeds the DLIN challenge. If $H(\mathbf{id}_j) \neq 0$, the algorithm aborts and output a uniform bit. Otherwise, let $[-\mathbf{y}^\mathsf{T}-] \in \mathbb{Z}_p^{1\times m}$ denote the third row of $\mathbf{A}$. Let $\mathbf{R}^* \overset{\mathsf{def}}{=} \sum_{i\in[n]} \mathrm{id}_i \mathbf{R}_i^*$. The encryption is constructed by $\mathcal{B}$ as follows:

$$((\mathbf{c}_0^*)^\mathsf{T}, (\mathbf{c}_1^*)^\mathsf{T}, c_2^*) = \left(g^{\mathbf{y}^\mathsf{T}}, g^{\mathbf{y}^\mathsf{T}\mathbf{R}^*}, \hat{e}(g,g)^{\left[\mathbf{y}^\mathsf{T} \,\middle|\, \mathbf{y}^\mathsf{T}\mathbf{R}^*\mathbf{S}^{(j)}\right]\cdot\mathbf{v}_j} \cdot \mathfrak{m}^*\right).$$

- **Output:** The simulator $\mathcal{B}$ receives b' from $\mathcal{A}$ and proceeds as follows. It first does the abort check and artificial abort as in experiment $\mathsf{Expt}_2^{(j)}$ and outputs either b' or a random bit.

To complete the proof of Claim 5.25 it suffices to show the following:

(a) The public parameters pp are distributed as in the real scheme.

(b) The secret key queries on identities $\left(\mathrm{sk}_{\mathbf{id}_1}, \ldots, \mathrm{sk}_{\mathbf{id}_{T+Q}}\right)$ are distributed as in the real scheme.

(c) As in the experiments $\mathsf{Expt}_2^{(j)}$ and $\mathsf{Expt}_3^{(j)}$, the ciphertexts output by $\mathsf{Enc}^{\mathsf{FP}}$ are ill-formed for identities $1, \ldots j-1$ and well-formed for identities $j+1, \ldots, T$.

(d) The DLIN challenge: If $\mathsf{Rk}(\mathbf{A}) = 2$, then the output of $\mathsf{Enc}^{\mathsf{FP}}(\mathrm{pp}, j, \mathfrak{m})$ is a well-formed ciphertext as in $\mathsf{Expt}_2^{(j)}$. If $\mathsf{Rk}(\mathbf{A}) = 3$, then the output of $\mathsf{Enc}^{\mathsf{FP}}(\mathrm{pp}, j, \mathfrak{m})$ is an ill-formed ciphertext as in $\mathsf{Expt}_3^{(j)}$.

To show item (a) consider the adversary's view that depends on the matrices $\mathbf{R}_i^*$ for $i \in [n]$. For simplicity, we consider the following components.[18]

$$\left(\mathbf{A}_0, \quad \mathbf{A}_0 \cdot [\mathbf{R}_1^* | \cdots | \mathbf{R}_n^*], \quad [\mathbf{R}_1^* | \cdots | \mathbf{R}_n^*]^\mathsf{T}\mathbf{y}, \quad \left[\mathbf{A}_0 \,\middle|\, \sum_{i\in[n]} \mathbf{A}_0\mathbf{R}_i^*\mathbf{S}^{(j)}\right] \cdot \mathbf{v}_j, \quad \mathbf{S}^{(j)}, \quad \mathbf{v}_j\right). \quad (5.17)$$

The last three components correspond to $\mathbf{u}$ in the public key and the secret key $\mathrm{sk}_{\mathbf{id}_j} = \left(\mathbf{S}^{(j)}, \mathbf{v}_j\right)$. In an argument identical to the one that secret keys are distributed correctly in the proof of Claim 5.18 (see item (b) in the corresponding part of the proof) the distribution of $(\mathbf{u}, \mathrm{sk}_{\mathbf{id}_j})$ in the simulation above is *identical* to the distribution of $(\mathbf{u}, \mathrm{sk}_{\mathbf{id}_j})$ in the real scheme. Therefore, the distributions of $\mathbf{u}$ and $\mathrm{sk}_{\mathbf{id}_j}$ are independent of the matrices $\mathbf{R}_i^*$'s used as the simulation trapdoor by $\mathcal{B}$. To show (a), it suffices to show that the following components out of Equation (5.17) are distributed appropriately:

$$(\mathbf{A}_0, \mathbf{A}_0 \cdot [\mathbf{R}_1^* | \cdots | \mathbf{R}_n^*], [\mathbf{R}_1^* | \cdots | \mathbf{R}_n^*]^\mathsf{T}\mathbf{y}).$$

This follows, applying the extended Leftover Hash Lemma (cf. Lemma 2.5) along the lines of the proof of the corresponding item (a) in Claim 5.18.

Items (b) and (d) follow from arguments identical to those used in the proof of Claim 5.18 (see items (b) and (c) in the corresponding part of the proof). Note that for showing item (c), whenever $i \neq j$, the ciphertexts output by $\mathsf{Enc}^{\mathsf{FP}}$ are generated *honestly* as in experiments $\mathsf{Expt}_2^{(j)}$ and $\mathsf{Expt}_3^{(j)}$. The simulation of these ciphertexts, even given pp, do not depend on the DLIN challenge and are always honest. Therefore (c) follows immediately.

Finally, as in the proof of Claim 5.18, we can complete the rest of the proof observing that the abort condition is identical to the abort conditions in $\mathsf{Expt}_2^{(j)}$ (and hence in $\mathsf{Expt}_3^{(j)}$). Thus, $\mathcal{B}$ simulates an experiment that is distributed statistically close to the experiment $\mathsf{Expt}_2^{(j)}$ if the DLIN

[18] The argument showing that the public parameters are distributed correctly is *statistical*; therefore it suffices to discard the exponentation.

challenge matrix $\mathbf{A}$ is of rank 2 and $\mathsf{Expt}_3^{(j)}$ if the DLIN challenge matrix $\mathbf{A}$ is of rank 3 which completes the proof of Claim 5.25. ∎

We complete the proof of Theorem 5.22 as follows.

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{EFP}}_{\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}(\lambda) \\
&= \left|\Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{EFP},\mathcal{IBE},\mathcal{A}}(\lambda)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{EFP},\mathcal{IBE},\mathcal{A}}(\lambda)=1\right]\right| \\
&= \left|\Pr\left[\mathsf{Expt}_0^{(1)}(\lambda)=1\right] - \Pr\left[\widetilde{\mathsf{Expt}}_0^{(1)}(\lambda)=1\right]\right| \\
&= \left|P_0^{(1)} - \widetilde{P}_0^{(1)}\right| \\
&\le 2\cdot\left(\frac{1}{S}+\frac{1}{\alpha\cdot 2^{\lambda}}\right) + \frac{1}{\alpha}\cdot\left|P_2^{(1)} - \widetilde{P}_2^{(1)}\right| && \text{(Claims 5.23 and 5.24)} \\
&\le 2\cdot\left(\frac{1}{S}+\frac{1}{\alpha\cdot 2^{\lambda}}\right) + \frac{1}{\alpha}\cdot\left(\left|P_3^{(1)} - \widetilde{P}_3^{(1)}\right| + 2\cdot\mathrm{negl}(\lambda)\right) && \text{(Claim 5.25)} \\
&\le 2\cdot\left(\frac{1}{S}+\frac{1}{\alpha\cdot 2^{\lambda}}\right) + \frac{1}{\alpha}\cdot\left(\left|P_0^{(2)} - \widetilde{P}_0^{(2)}\right| + \mathrm{negl}(\lambda)\right). && \text{(Equation (5.13))}
\end{aligned}$$

Applying the same argument to $|P_0^{(2)} - \widetilde{P}_0^{(2)}|$ implies

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{EFP}}_{\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}(\lambda) \\
&\le 2\cdot\left(\frac{1}{S}+\frac{1}{\alpha\cdot 2^{\lambda}}\right) + \frac{1}{\alpha}\cdot\left(2\cdot\left(\frac{1}{S}+\frac{1}{\alpha\cdot 2^{\lambda}}\right) + \frac{1}{\alpha}\cdot\left(\left|P_0^{(3)} - \widetilde{P}_0^{(3)}\right| + \mathrm{negl}(\lambda)\right) + \mathrm{negl}(\lambda)\right).
\end{aligned}$$

If we let Γ denote the sum $\left(1/\alpha + 1/\alpha^2 + \cdots + 1/\alpha^T\right)$, recursively applying the argument and collecting terms implies

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{EFP}}_{\mathcal{IBE}_{\mathsf{DLIN2}},\mathcal{A}}(\lambda) &\le 2\Gamma\cdot\left(\frac{1}{S}+\frac{1}{\alpha\cdot 2^{\lambda}}\right) + \Gamma\cdot\mathrm{negl}(\lambda) + \frac{1}{\alpha^T}\cdot\left|P_3^{(T)} - \widetilde{P}_3^{(T)}\right| \\
&\le 2\Gamma\cdot\left(\frac{1}{S}+\frac{1}{\alpha\cdot 2^{\lambda}}\right) + \Gamma\cdot\mathrm{negl}(\lambda) + \frac{1}{\alpha^T}\cdot\mathrm{negl}(\lambda). && \text{(from Eq. (5.15))}
\end{aligned}$$

As α is at least $1/P(\lambda)$ for some *fixed* polynomial $P(\lambda)$ (applying Lemma 5.14 with $Q+T-1$ instead of Q), the above result holds for every polynomial $S = S(\lambda)$, and T is a constant, the advantage of $\mathcal{A}$ is therefore negligible. This completes the proof of Theorem 5.22. ∎

6 Non-Adaptive Enhanced Function Privacy via Collision Resistance

In this section we present a generic method for transforming any IBE scheme into a *non-adaptive* enhanced function-private IBE scheme. Given an IBE scheme with an identity space $\mathcal{ID}$, the new scheme uses a slightly larger identity space $\mathcal{ID}'$, and a mapping from $\mathcal{ID}'$ to $\mathcal{ID}$ which enables to use the key-generation, encryption, and decryption algorithms of the underlying scheme. The mapping uses a pairwise independent permutation π over $\mathcal{ID}'$, and a collision-resistant function $h : \mathcal{ID}' \to \mathcal{ID}$, and maps any $\mathrm{id}' \in \mathcal{ID}'$ to $h(\pi(\mathrm{id}')) \in \mathcal{ID}$. The descriptions of π and h are provided as part of the public parameters of the new scheme.

Such a transformation clearly preserves the data privacy of the underlying scheme due to the fact that the mapping $h \circ \pi : \mathcal{ID}' \to \mathcal{ID}$ is collision resistant. In addition, in terms of function privacy, the crooked leftover hash lemma [DS05, BFO08b] guarantees that when sampling $(\mathrm{id}'_1, \ldots, \mathrm{id}'_T)$ from any (T,k)-block-source $\boldsymbol{ID'}$, for $k \ge \log|\mathcal{ID}| + \omega(\log\lambda)$, the distribution of $(h(\pi(\mathrm{id}'_1)), \ldots, h(\pi(\mathrm{id}'_T)))$ is statistically-close to being independent of $\boldsymbol{ID'}$.

The scheme. Let $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a $\mathsf{anon\text{-}IND\text{-}ID\text{-}CPA}$ secure identity-based encryption scheme with an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$. Given an identity space $\mathcal{ID}' = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$, let $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda\in\mathbb{N}}$ be family of collision-resistant functions $h : \mathcal{ID}'_\lambda \to \mathcal{ID}_\lambda$, and let $\Pi = \{\Pi_\lambda\}_{\lambda\in\mathbb{N}}$ be a pairwise-independent collection of permutations π over $\mathcal{ID}'_\lambda$. We construct an IBE scheme $\mathcal{IBE}^{\mathsf{CRH}} = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ with an identity space $\mathcal{ID}'$ and message space $\mathcal{M}$ as follows.

- **Setup:** On input 1^λ the setup algorithm Setup' first samples $(\mathrm{pp}, \mathrm{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Next, it samples a permutation $\pi \leftarrow \Pi_\lambda$ and a collision-resistant function $h \leftarrow \mathcal{H}_\lambda$. It outputs $\mathrm{pp}' = (\mathrm{pp}, \pi, h)$ and sets $\mathrm{msk}' = \mathrm{msk}$.

- **Key generation:** On input the master secret key msk' and an identity $\mathrm{id}' \in \mathcal{ID}'_\lambda$, the key-generation algorithm KeyGen' computes $\mathrm{id} = h(\pi(\mathrm{id}')) \in \mathcal{ID}_\lambda$, and outputs a secret key $\mathrm{sk}'_{\mathrm{id}'} \leftarrow \mathsf{KeyGen}\left(\mathrm{msk}, \mathrm{id}\right))$.

- **Encryption:** On input the public parameters $\mathrm{pp}' = (\mathrm{pp}, h, \pi)$, an identity $\mathrm{id}' \in \mathcal{ID}'_\lambda$, and a message $m \in \mathcal{M}_\lambda$, the encryption algorithm computes $\mathrm{id} = h(\pi(\mathrm{id}')) \in \mathcal{ID}_\lambda$ and outputs $c \leftarrow \mathsf{Enc}\left(\mathrm{pp}, \mathrm{id}, m\right)$.

- **Decryption:** On input the public parameters $\mathrm{pp}' = (\mathrm{pp}, h, \pi)$, a ciphertext c, and a secret key sk, the decryption algorithm outputs $\mathsf{Dec}\left(\mathrm{pp}, c, \mathrm{sk}\right)$.

Theorem 6.1. *The scheme $\mathcal{IBE}^{\mathsf{CRH}}$ is non-adaptive statistical enhanced function private for (T, k)-block-sources, for any $T = \mathrm{poly}(\lambda)$ and $k \geq \log |\mathcal{ID}_\lambda| + \omega(\log \lambda)$. In addition, assuming that $\mathcal{H}$ is a family of collision-resistant functions, the scheme $\mathcal{IBE}^{\mathsf{CRH}}$ preserves the data privacy of the underlying scheme $\mathcal{IBE}$.*

Proof. We begin by proving the function privacy of the scheme, and then prove its data privacy.

Non-adaptive enhanced function privacy. Let $\mathcal{A}$ be a computationally unbounded (T, k)-block-source function-privacy adversary. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{NA\text{-}EFP},\mathcal{IBE}^{\mathsf{CRH}},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{NA\text{-}EFP},\mathcal{IBE}^{\mathsf{CRH}},\mathcal{A}}$ (we refer the reader to Definition 3.5 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$, respectively.

As the adversary $\mathcal{A}$ is computationally unbounded, we assume without loss of generality that $\mathcal{A}$ does not query the $\mathsf{KeyGen}'(\mathrm{msk}', \cdot)$ oracle. Additionally, we include in the adversary's view not only $\mathrm{sk}'_{\mathrm{id}'_1}, \ldots, \mathrm{sk}'_{\mathrm{id}'_T}$ from the $\mathsf{RoR}^{\mathsf{FP}}$ oracle but even $h(\pi(\mathrm{id}'_1)), \ldots, h(\pi(\mathrm{id}'_T))$. Therefore, given pp' and msk', $\mathcal{A}$ can simulate the output of the $\mathsf{Enc}^{\mathsf{FP}}$ oracle on messages of his choice. Thus, it suffices to show that the distributions $(\mathrm{pp}', \mathrm{msk}', h(\pi(\mathrm{id}'_1)), \ldots, h(\pi(\mathrm{id}'_T))$ where the identities are sampled as in mode real and mode rand, respectively, are statistically close in the two experiments (all other components in the adversary's view are randomized functions of the distribution above and therefore cannot increase the statistical distance). This follows directly from the crooked leftover hash lemma [DS05, BFO08b].

Data privacy. The proof of data privacy of $\mathcal{IBE}^{\mathsf{CRH}}$ from the data privacy of the underlying scheme $\mathcal{IBE}$ is rather straightforward. We only give a brief outline of the proof here and note that the details are fairly straightforward.

Given a challenger for the data privacy security game (see Definitions 2.7 and 2.8) we can easily simulate a challenger for the data privacy security game with $\mathcal{IBE}^{\mathsf{CRH}}$ as follows: We first sample h

and π as in the scheme and use pp generated by the $\mathcal{IBE}$ challenger to construct $\mathrm{pp}' = (\mathrm{pp}, h, \pi)$. Upon KeyGen' query id', the simulator computes $\mathrm{id} = h(\pi(\mathrm{id}'))$ and forwards it to the $\mathcal{IBE}$ key-generation oracle to receive $\mathrm{sk}'_{\mathrm{id}'} = \mathrm{sk}_{\mathrm{id}}$. When $\mathcal{A}$ issues a challenge query, the simulator forwards the challenge query after applying $h(\pi(\cdot))$ to identities in the challenge. If $h(\pi(\mathrm{id}^*))$ collides with a previous id' query, the simulator aborts. Finally, the simulator outputs the bit b that the $\mathcal{IBE}^{\mathsf{CRH}}$ challenger outputs.

We claim that if the simulator does not abort, then it faithfully simulates a $\mathcal{IBE}^{\mathsf{CRH}}$ challenger for the corresponding data privacy security game. Thus, an adversary breaking the data privacy of the scheme breaks the data privacy of the underlying scheme with the same advantage. The probability of the simulator aborting against computationally bounded adversaries is negligible from the collision resistance of the hash function family. ∎

7 Extensions and Open Problems

Our framework for function privacy yields a variety of extensions and open problems, both conceptual ones regarding our new notions, and technical ones regarding our specific approach and its resulting constructions. We now discuss several such extensions and open problems.

Chosen-ciphertext security. In terms of data privacy, in this paper we considered the standard notion of anonymity and message indistinguishability under an adaptive chosen-identity chosen-plaintext attack (known as anon-IND-ID-CPA). A natural extension of our results is to guarantee data privacy even against chosen-ciphertext attacks (known as anon-IND-ID-CCA). We note that our IBE schemes can be extended, using standard techniques, into two-level hierarchical IBE schemes that are anon-IND-ID-CPA-secure and their first level is function private. Then, by applying the generic transformation of Boneh, Canetti, Halevi and Katz [BCH$^+$07], any such scheme can be used to construct an IBE scheme that is anon-IND-ID-CCA-secure and function private.

Applying our approach to other IBE schemes. In Section 3 we presented simple attacks exemplifying that the anonymous IBE schemes presented in [BF03, GPV08, ABB10, KP11] are not function private. Nevertheless, we were able to rely on these schemes for designing new ones that are function private using our "extract-augment-combine" approach. For other anonymous IBE schemes, such as [Gen06, BW06, BKP$^+$12], we were not able to find attacks against their function privacy. An interesting open problem is to explore whether these schemes can be modified (possibly by applying our "extract-augment-combine" approach) to be function private based on standard assumptions. More generally, a natural open problem is to identify a specific property of identity-based encryption schemes that make them amenable to our "extract-augment-combine" approach.

Extension to other classes of functions. As discussed in Section 1, in the general setting of functional encryption our schemes provide function privacy for the class of functions f_{id^*} defined as $f_{id^*}(id, m) = m$ if $id = id^*$, and $f_{id^*}(id, m) = \perp$ otherwise. A fascinating open problem is to construct schemes that are function private for other classes of functions. A possible starting point is to consider function privacy for other, rather simple, functionalities, such as inner-product testing [KSW08].

Robustness of our schemes. As pointed out by Abdalla, Bellare, and Neven [ABN10], when using an anonymous IBE scheme as a public-key searchable encryption scheme [BCO$^+$04, ABC$^+$08], it is often desirable to use a "robust" IBE scheme: It should be difficult to produce a ciphertext that is valid for more than one identity. We note that our schemes do not satisfy such a notion of robustness. However, Abdalla et al. showed two generic transformations that transform any given IBE scheme into a robust one. In particular, these transformations can be applied to each of our schemes to make them robust (these transformations do not change the decryption keys, and thus function privacy is preserved). We leave it as an open problem to directly design function-private IBE schemes that are robust.

Acknowledgements

We thank the anonymous CRYPTO '13 reviewers for many useful comments. This work was supported by NSF, the DARPA PROCEED program, an AFO SR MURI award, a grant from ONR, an IARPA project provided via DoI/NBC, and by Samsung. Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or IARPA. Distrib. Statement "A:" Approved for Public Release, Distribution Unlimited.

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology – EUROCRYPT '10*, pages 553–572, 2010.

[ABC$^+$08] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *Journal of Cryptology*, 21(3):350–391, 2008.

[ABN10] M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *Proceedings of the 7th Theory of Cryptography Conference*, pages 480–497, 2010.

[AFV11] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *Advances in Cryptology – ASIACRYPT '11*, pages 21–40, 2011.

[AGV$^+$13] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption: New perspectives and lower bounds. To appear in *Advances in Cryptology – CRYPTO '13*, 2013.

[BBN$^+$09] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *Advances in Cryptology – ASIACRYPT '09*, pages 232–249, 2009.

[BBO07] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology – CRYPTO '07*, pages 535–552, 2007.

[BCH$^+$07] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

[BCO$^+$04] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology – EUROCRYPT '04*, pages 506–522, 2004.

[BF03] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. Preliminary version in *Advances in Cryptology – CRYPTO '01*, pages 213–229, 2001.

[BFO$^+$08a] M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *Advances in Cryptology – CRYPTO '08*, pages 360–378, 2008.

[BFO08b] A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology – CRYPTO '08*, pages 335–359, 2008.

[BGI$^+$12] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6, 2012.

[BHH$^+$08] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *Advances in Cryptology – CRYPTO '08*, pages 108–125, 2008.

[BKP$^+$12] M. Bellare, E. Kiltz, C. Peikert, and B. Waters. Identity-based (lossy) trapdoor functions and applications. In *Advances in Cryptology – EUROCRYPT '12*, pages 228–245, 2012.

[BO12] M. Bellare and A. O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. Cryptology ePrint Archive, Report 2012/515, 2012.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BS11] Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *Advances in Cryptology – CRYPTO '11*, pages 543–560, 2011.

[BSNS08] J. Baek, R. Safavi-Naini, and W. Susilo. Public key encryption with keyword search revisited. In *Proceedings on the International Conference Computational Science and Its Applications*, pages 1249–1259, 2008.

[BSW09] J. Bethencourt, D. Song, and B. Waters. New techniques for private stream searching. *ACM Transactions on Information and System Security*, 12(3), 2009.

[BSW11] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *Proceedings of the 8th Theory of Cryptography Conference*, pages 253–273, 2011.

[BW06] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Advances in Cryptology – CRYPTO '06*, pages 290–307, 2006.

[BW07] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Proceedings of the 4th Theory of Cryptography Conference*, pages 535–554, 2007.

[Can97] R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology – CRYPTO '97*, pages 455–469, 1997.

[CGK$^+$11] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934, 2011.

[CHK$^+$10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Advances in Cryptology – EUROCRYPT '10*, pages 523–552, 2010.

[CK10] M. Chase and S. Kamara. Structured encryption and controlled disclosure. In *Advances in Cryptology – ASIACRYPT '10*, pages 577–594, 2010.

[CKR$^+$09] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography*, pages 196–214, 2009.

[CKV$^+$10] R. Canetti, Y. T. Kalai, M. Varia, and D. Wichs. On symmetric encryption and point obfuscation. In *Proceedings of the 7th Theory of Cryptography Conference*, pages 52–71, 2010.

[CM05] Y.-C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *Proceedings of the 3rd International Conference on Applied Cryptography and Network Security*, pages 442–455, 2005.

[DOR$^+$08] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[DS05] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 654–663, 2005.

[FOR12] B. Fuller, A. O'Neill, and L. Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In *Proceedings of the 9th Theory of Cryptography Conference*, pages 582–599, 2012.

[Gen06] C. Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT '06*, pages 445–464, 2006.

[GK05] S. Goldwasser and Y. T. Kalai. On the impossibility of obfuscation with auxiliary input. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 553–562, 2005.

[GKP$^+$13] S. Goldwasser, Y. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. Reusable garbled circuits and succinct functional encryption. To appear in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, 2013.

[GO96] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious rams. *Journal of the ACM*, 43(3):431–473, 1996.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of computing*, pages 197–206, 2008.

[GSW04] P. Golle, J. Staddon, and B. R. Waters. Secure conjunctive keyword search over encrypted data. In *Proceedings of the 2nd International Conference on Applied Cryptography and Network Security*, pages 31–45, 2004.

[GVW12] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology – CRYPTO '12*, pages 162–179, 2012.

[HIL$^+$99] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HK12] D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. *Journal of Cryptology*, 25(3):484–527, 2012.

[KP11] K. Kurosawa and L. T. Phong. Maximum leakage resilient IBE and IPE. Cryptology ePrint Archive, Report 2011/628, 2011.

[KPR12] S. Kamara, C. Papamanthou, and T. Roeder. Dynamic searchable symmetric encryption. In *ACM Conference on Computer and Communications Security*, pages 965–976, 2012.

[KSW08] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Advances in Cryptology – EUROCRYPT '08*, pages 146–162, 2008.

[LPS04] B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In *Advances in Cryptology – EUROCRYPT '04*, pages 20–39, 2004.

[MPR$^+$12] I. Mironov, O. Pandey, O. Reingold, and G. Segev. Incremental deterministic public-key encryption. In *Advances in Cryptology – EUROCRYPT '12*, pages 628–644, 2012.

[NS12] M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. *SIAM Journal on Computing*, 41(4):772–814, 2012.

[O'N10] A. O'Neill. Definitional issues in functional encryption. IACR Cryptology ePrint Archive, Report 2010/556, 2010.

[OS07] R. Ostrovsky and W. E. Skeith III. Private searching on streaming data. *Journal of Cryptology*, 20(4):397–430, 2007.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 84–93, 2005.

[RSV13] A. Raghunathan, G. Segev, and S. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In *Advances in Crytology – EUROCRYPT '13*, pages 93–110, 2013.

[SBC$^+$07] E. Shi, J. Bethencourt, H. T.-H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy*, pages 350–364, 2007.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – CRYPTO '84*, pages 47–53, 1984.

[SSW09] E. Shen, E. Shi, and B. Waters. Predicate privacy in encryption systems. In *Proceedings of the 6th Theory of Cryptography Conference*, pages 457–473, 2009.

[SWP00] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

[Vad12] S. Vadhan. Pseudorandomness (draft survey/monograph). Available online at: `http://people.seas.harvard.edu/~salil/pseudorandomness`, 2012.

[vLSD+10] P. van Liesdonk, S. Sedghi, J. Doumen, P. H. Hartel, and W. Jonker. Computationally efficient searchable symmetric encryption. In *Secure Data Management*, pages 87–100, 2010.

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT '05*, pages 114–127, 2005.

[Wee05] H. Wee. On obfuscating point functions. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 523–532, 2005.

[Wee12] H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *Advances in Cryptology – EUROCRYPT '12*, pages 246–262, 2012.