An extended abstract of this paper appears in the Proceedings of Eurocrypt 2012, LNCS 7237, Springer. This is the full version.

Identity-Based (Lossy) Trapdoor Functions and Applications

Mihir Bellare[1] Eike Kiltz[2] Chris Peikert[3] Brent Waters[4]

May 2012

Abstract

We provide the first constructions of identity-based (injective) trapdoor functions. Furthermore, they are lossy. Constructions are given both with pairings (DLIN) and lattices (LWE). Our lossy identity-based trapdoor functions provide an automatic way to realize, in the identity-based setting, many functionalities previously known only in the public-key setting. In particular we obtain the first deterministic and efficiently searchable IBE schemes and the first hedged IBE schemes, which achieve best possible security in the face of bad randomness. Underlying our constructs is a new definition, namely *partial* lossiness, that may be of broader interest.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. Email: mihir@cs.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-0627779 and CCF-0915675.

[2] Horst Görtz Institut für IT-Sicherheit, Ruhr-Universität Bochum, D-44780 Bochum. Email: eike.kiltz@rub.de. URL: http://www.cits.rub.de/personen/kiltz.html.

[3] School of Computer Science, College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332-0765. Email: cpeikert@cc.gatech.edu. URL: http://www.cc.gatech.edu/~cpeikert/.

[4] Department of Computer Science, University of Texas at Austin, 1616 Guadalupe, Suite 2.408, Austin, TX 78701. Email: bwaters@cs.utexas.edu. URL: http://userweb.cs.utexas.edu/~bwaters/.

1 Introduction

A trapdoor function F specifies, for each public key pk, an injective, *deterministic* map F_{pk} that can be inverted given an associated secret key (trapdoor). The most basic measure of security is one-wayness. The canonical example is RSA [55].

Suppose there is an algorithm that generates a "fake" public key pk^* such that F_{pk^*} is no longer injective but has image much smaller than its domain and, moreover, given a public key, you can't tell whether it is real or fake. Peikert and Waters [52] call such a TDF lossy. Intuitively, F_{pk} is close to a function F_{pk^*} that provides information-theoretic security. Lossiness implies one-wayness [52].

Lossy TDFs have quickly proven to be a powerful tool. Applications include IND-CCA [52], deterministic [16], hedged [8] and selective-opening secure public-key encryption [10]. Lossy TDFs can be constructed from DDH [52], QR [35], DLIN [35], DBDH [24], LWE [52] and HPS (hash proof systems) [40]. RSA was shown in [44] to be lossy under the Φ-hiding assumption of [26], leading to the first proof of security of RSA-OAEP [13] without random oracles.

Lossy TDFs and their benefits belong, so far, to the realm of public-key cryptography. The purpose of this paper is to bring them to identity-based cryptography, defining and constructing identity-based TDFs (IB-TDFs), both one-way and lossy. We see this as having two motivations, one more theoretical, the other more applied, yet admittedly both foundational, as we discuss before moving further.

Theoretical angle. Trapdoor functions are the primitive that began public key cryptography [31, 55]. Public-key encryption was built from TDFs. (Via hardcore bits.) Lossy TDFs enabled the first DDH and lattice (LWE) based TDFs [52].

It is striking that identity-based cryptography developed entirely differently. The first realizations of IBE [21, 30, 58] directly used randomization and were neither underlain by, nor gave rise to, any IB-TDFs.

We ask whether this asymmetry between the public-key and identity-based worlds (TDFs in one but not the other) is inherent. This seems to us a basic question about the nature of identity-based cryptography that is worth asking and answering.

Application angle. Is there anything here but idle curiosity? IBE has already been achieved *without* IB-TDFs, so why go backwards to define and construct the latter? The answer is that *losssy* IB-TDFs enable new applications that we do not know how to get in other ways.

Stepping back, identity-based cryptography [59] offers several advantages over its public-key counterpart. Key management is simplified because an entity's identity functions as their public key. Key revocation issues that plague PKI can be handled in alternative ways, for example by using $\mathsf{identity}$+date as the key under which to encrypt to $\mathsf{identity}$ [21]. There is thus good motivation to go beyond basics like IBE [21, 30, 58, 17, 18, 62, 36] and identity-based signatures [11, 32] to provide identity-based counterparts of other public-key primitives.

Furthermore we would like to do this in a systematic rather than ad hoc way, leading us to seek tools that enable the transfer of multiple functionalities in relatively blackbox ways. The applications of lossiness in the public-key realm suggest that lossy IBTDFs will be such a tool also in the identity-based realm. As evidence we apply them to achieve identity-based deterministic encryption and identity-based hedged encryption. The first, the counterpart of deterministic public-key encryption [7, 16], allows efficiently searchable identity-based encryption of database entries while maintaining the maximal possible privacy, bringing the key-management benefits of the identity-based setting to this application. The second, counterpart of hedged symmetric and public-key encryption [56, 8], makes IBE as resistant as possible in the face of low-quality randomness, which is important given the widespread deployment of IBE and the real danger of bad-randomness based attacks evidenced by the ones on the Sony Playstation and Debian Linux. We hope that our framework will facilitate further such transfers.

We clarify that the solutions we obtain are not practical but they show that the security goals can be achieved in principle, which was not at all clear prior to our work. Allowed random oracles, we can give solutions that are much more efficient and even practical.

CONTRIBUTIONS IN BRIEF. We define IB-TDFs and two associated security notions, one-wayness and lossiness, showing that the second implies the first.

The first wave of IBE schemes was from pairings [21, 58, 17, 18, 62, 61] but another is now emerging from lattices [36, 29, 2, 3]. We aim accordingly to reach our ends with either route and do so successfully. We provide lossy IB-TDFs from a standard pairings assumption, namely the Decision Linear (DLIN) assumption of [19]. We also provide IB-TDFs based on Learning with Errors (LWE) [53], whose hardness follows from the worst-case hardness of certain lattice-related problems [53, 50]. (The same assumption underlies lattice-based IBE [36, 29, 2, 3] and public-key lossy TDFs [52].) None of these results relies on random oracles.

Existing work brought us closer to the door with lattices, where one-way IB-TDFs can be built by combining ideas from [36, 29, 2]. Based on techniques from [50, 45] we show how to make them lossy. With pairings, however it was unclear how to even get a one-way IB-TDF, let alone one that is lossy. We adapt the matrix-based framework of [52] so that by populating matrix entries with ciphertexts of a very special kind of *anonymous* IBE scheme it becomes possible to implicitly specify per-identity matrices defining the function. No existing anonymous IBE has the properties we need but we build one that does based on methods of [23]. Our results with pairings are stronger because the lossy branches are universal hash functions which is important for applications.

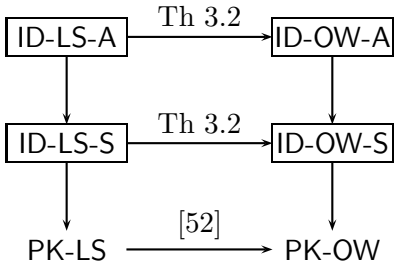
Public-key lossy TDFs exist aplenty and IBE schemes do as well. It is natural to think one could easily combine them to get IB-TDFs. We have found no simple way to do this. Ultimately we do draw from both sources for techniques but our approaches are intrusive. Let us now look at our contributions in more detail.

NEW PRIMITIVES AND DEFINITIONS. Public parameters $pars$ and an associated master secret key having been chosen, an IB-TDF F associates to any identity a map $F_{pars,id}$, again injective and deterministic, inversion being possible given a secret key derivable from id via the master secret key. One-wayness means F_{pars,id^*} is hard to invert on random inputs for an adversary-specified challenge identity id^*. Importantly, as in IBE, this must hold even when the adversary may obtain, via a key-derivation oracle, a decryption key for any non-challenge identity of its choice [21]. This key-derivation capability contributes significantly to the difficulty of realizing the primitive. As with IBE, security may be selective (the adversary must specify id^* before seeing $pars$) [28] or adaptive (no such restriction) [21].

The most direct analog of the definition of lossiness from the public-key setting would ask that there be a way to generate "fake" parameters $pars^*$, indistinguishable from the real ones, such that F_{pars^*,id^*} is lossy (has image smaller than domain). In the selective setting, the fake parameter generation algorithm Pg^* can take id^* as input, making the goal achievable at least in principle, but in the adaptive setting it is impossible to achieve, since, with id^* not known in advance, Pg^* is forced to make $F_{pars^*,id}$ lossy for all id, something the adversary can immediately detect using its key-derivation oracle.

We ask whether there is an adaptation of the definition of lossiness that is achievable in the adaptive case while sufficing for applications. Our answer is a definition of *δ-lossiness*, a metric of partial lossiness parameterized by the probability δ that F_{pars^*,id^*} is lossy. The definition is unusual, involving an adversary advantage that is the difference, not of two probabilities as is common in cryptographic metrics, but of two differently weighted ones. We will achieve selective lossiness with degree $\delta = 1$, but in the adaptive case the best possible is degree $1/\operatorname{poly}$ with the polynomial depending on the number of key-derivation queries of the adversary, and this what we will achieve. We show that lossiness with degree δ implies one-wayness, in both the selective and adaptive settings, as long as δ is at least $1/\operatorname{poly}$.

In summary, in the identity-based setting (ID) there are two notions of security, one-wayness (OW) and lossiness (LS), each of which could be selective (S) or adaptive (A), giving rise to four kinds of IB-TDFs. The left side of Figure 1 shows how they relate to each other and to the two kinds of TDFs —OW and LS— in the public-key setting (PK). The un-annotated implications are trivial, ID-LS-A $\rightarrow$ ID-LS-S meaning that δ-lossiness of the first type implies δ-lossiness of the other for all δ. It is not however via this implication that we achieve ID-LS-S, for, as the table shows, we achieve it with degree higher than



Primitive	δ	Achieved under
ID-LS-A	$1/\text{poly}$	DLIN, LWE
ID-LS-S	1	DLIN, LWE

Figure 1: Types of TDFs based on setting (PK=Public-key, ID=identity-based), security (OW=one-way, LS=loss) and whether the latter is selective (S) or adaptive (A). An arrow A $\rightarrow$ B in the diagram on the left means that TDF of type B is implied by (can be constructed from) TDF of type A. Boxed TDFs are the ones we define and construct. The table on the right shows the δ for which we prove δ-lossiness and the assumptions used. In both the S and A settings the δ we achieve is best possible and suffices for applications.

ID-LS-A.

Closer Look. One's first attempt may be to build an IB-TDF from an IBE scheme. In the random oracle (RO) model, this can be done by a method of [9], namely specify the coins for the IBE scheme by hashing the message with the RO. It is entirely unclear how to turn this into a standard model construct and it is also unclear how to make it lossy.

To build ID-TDFs from lattices we consider starting from the public-key TDF of [52] (which is already lossy) and trying to make it identity-based, but it is unclear how to do this. However, Gentry, Peikert and Vaikuntanathan (GPV) [36] showed that the function $g_{\mathbf{A}} \colon B_{\alpha}^{n+m} \rightarrow \mathbb{Z}_q^n$ defined by $g_{\mathbf{A}}(\mathbf{x}, \mathbf{e}) = \mathbf{A}^T \cdot \mathbf{x} + \mathbf{e}$ is a TDF for appropriate choices of the domain and parameters, where matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is a uniformly random public key which is constructed together with a trapdoor as for example in [4, 5, 46]. We make this function identity-based using the trapdoor extension and delegation methods introduced by Cash, Hofheinz, Kiltz and Peikert [29], and improved in efficiency by Agrawal, Boneh and Boyen [2] and Micciancio and Peikert [46]. Finally, we obtain a lossy IB-TDF by showing that this construction is already lossy.

With pairings there is no immediate way to get an IB-TDF that is even one-way, let alone lossy. We aim for the latter, there being no obviously simpler way to get the former. In the selective case we need to ensure that the function is lossy on the challenge identity id^* yet injective on others, this setup being indistinguishable from the one where the function is always injective. Whereas the matrix diagonals in the construction of [52] consisted of ElGamal ciphertexts, in ours they are ciphertexts for identity id^* under an anonymous IBE scheme, the salient property being that the "anonymity" property should hide whether the underlying ciphertext is to id^* or is a random group element. Existing anonymous IBE schemes, in particular that of Boyen and Waters (BW) [23], are not conducive and we create a new one. A side benefit is a new anonymous IBE scheme with ciphertexts and private keys having one less group element than BW but still proven secure under DLIN.

A method of Boneh and Boyen [17] can be applied to turn selective into adaptive security but the reduction incurs a factor that is equal to the size of the identity space and thus ultimately exponential in the security parameter, so that adaptive security according to the standard asymptotic convention would not have been achieved. To achieve it, we want to be able to "program" the public parameters so that they will be lossy on about a $1/Q$ fraction of "random-ish" identities, where Q is the number of key-derivation queries made by the attacker. Ideally, with probability around $1/Q$ all of (a successful) attacker's queries will land outside the lossy identity-space, but the challenge identity will land inside it so that we achieve δ-lossiness with δ around $1/Q$.

This sounds similar to the approach of Waters [62] for achieving adaptively secure IBE but there are some important distinctions, most notably that the technique of Waters is information-theoretic while ours is of necessity computational, relying on the DLIN assumption. In the reduction used by Waters the

partitioning of the identities into two classes was based solely on the reduction algorithm's internal view of the public parameters; the parameters themselves were distributed independently of this partitioning and thus the adversary view was the same as in a normal setup. In contrast, the partitioning in our scheme will actually directly affect the parameters and how the system behaves. This is why we must rely on a computational assumption to show that the partitioning in undetectable. A key novel feature of our construction is the introduction of a system that will produce lossy public parameters for about a $1/Q$ fraction of the identities.

Applications. Deterministic PKE is a TDF providing the best possible privacy subject to being deterministic, a notion called PRIV that is much stronger than one-wayness [7]. An application is encryption of database records in a way that permits logarithmic-time search, improving upon the linear-time search of PEKS [20]. Boldyreva, Fehr and O'Neill [16] show that lossy TDFs whose lossy branch is a universal hash (called universal lossy TDFs) achieve (via the LHL [15, 39]) PRIV-security for message sequences which are blocksources, meaning each message has some min-entropy even given the previous ones, which remains the best result without ROs. Deterministic IBE and the resulting efficiently-searchable IBE are attractive due to the key-management benefits. We can achieve them because our DLIN-based lossy IB-TDFs are also universal lossy. (This is not true, so far, for our LWE based IB-TDFs.)

To provide IND-CPA security in practice, IBE relies crucially on the availability of fresh, high-quality randomness. This is fine in theory but in practice RNGs (random number generators) fail due to poor entropy gathering or bugs, leading to prominent security breaches [37, 38, 25, 49, 48, 1, 63, 33]. Expecting systems to do a better job is unrealistic. Hedged encryption [8] takes poor randomness as a fact of life and aims to deliver best possible security in the face of it, providing privacy as long as the message together with the "randomness" have some min-entropy. Hedged PKE was achieved in [8] by combining IND-CPA PKE with universal lossy TDFs. We can adapt this to IBE and combine existing (randomized) IBE schemes with our DLIN-based universal lossy IB-TDFs to achieved hedged IBE. This is attractive given the widespread use of IBE in practice and the real danger of randomness failures.

Both applications are for the case of selective security. We do not achieve them in the adaptive case.

Related Work. A number of papers have studied security notions of trapdoor functions beyond traditional one-wayness. Besides lossiness [52] there is Rosen and Segev's notion of correlated-product security [57], and Canetti and Dakdouk's extractable trapdoor functions [27]. The notion of adaptive one-wayness for tag-based trapdoor functions from Kiltz, Mohassel and O'Neill [43] can be seen as the special case of our selective IB-TDF in which the adversary is denied key-derivation queries. Security in the face of these queries was one of the main difficulties we faced in realizing IB-TDFs.

Organization. We define IB-TDFs, one-wayness and δ-lossiness in Section 2. We also define extended IB-TDFs, an abstraction that will allow us to unify and shorten the analyses for the selective and adaptive security cases. In Section 3 we show that δ-lossiness implies one-wayness as long as δ is at least $1/\operatorname{poly}$. This allows us to focus on achieving δ-lossiness. In Section 4 we provide our pairing-based schemes and in Appendix 5 our lattice-based schemes. In Appendix B we sketch how to apply δ-lossy IB-TDFs to achieve deterministic and hedged IBE.

Subsequent work. Escala, Herranz, Libert and Ráfols [34] provide an alternative definition of partial lossiness based on which they achieve deterministic, PRIV-secure IBE for blocksources, and hedged IBE, in the adaptive case, which answers an open question from our work. They also define and construct hierarchical identity-based (lossy) trapdoor functions.

2 Definitions

Notation and conventions. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes the number of its coordiates and $\mathbf{x}[i]$ denotes its i-th coordinate. Coordinates may be numbered $1, \ldots, |\mathbf{x}|$ or $0, \ldots, |\mathbf{x}| - 1$ as convenient. A string x is identified with a vector over $\{0,1\}$ so that $|x|$ denotes its length and $x[i]$ its i-th bit. The

proc Initialize(id) // $\mathrm{OW}_{\mathsf{F}}, \mathrm{Real}_{\mathsf{F}}$
$(pars, msk) \xleftarrow{\$} \mathsf{F.Pg}$; $IS \leftarrow \emptyset$; $id^* \leftarrow id$
Return $pars$

proc GetDK(id) // $\mathrm{OW}_{\mathsf{F}}, \mathrm{Real}_{\mathsf{F}}$
$IS \leftarrow IS \cup \{id\}$
$dk \leftarrow \mathsf{F.Kg}(pars, msk, id)$
Return dk

proc Ch(id) // OW_{F}
$id^* \leftarrow id$; $x \xleftarrow{\$} \mathsf{InSp}$
$y \leftarrow \mathsf{F.Ev}(pars, id^*, x)$
Return y

proc Finalize(x') // OW_{F}
Return $((x' = x)$ and $(id^* \notin IS))$

proc Initialize(id) // $\mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}$
$(pars, msk) \xleftarrow{\$} \mathsf{LF.Pg}(id)$; $IS \leftarrow \emptyset$; $id^* \leftarrow id$
Return $pars$

proc GetDK(id) // $\mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}$
$IS \leftarrow IS \cup \{id\}$
$dk \leftarrow \mathsf{LF.Kg}(pars, msk, id)$
Return dk

proc Ch(id) // $\mathrm{Real}_{\mathsf{F}}, \mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}$
$id^* \leftarrow id$

proc Finalize(d') // $\mathrm{Real}_{\mathsf{F}}$
Return $((d' = 1)$ and $(id^* \notin IS))$

proc Finalize(d') // $\mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}$
Return $((d' = 1)$ and $(id^* \notin IS)$ and $(\lambda(\mathsf{F.Ev}(pars, id^*, \cdot)) \geq \ell))$

Figure 2: Games defining one-wayness and δ-lossiness of IBTDF F with associated sibling LF.

empty string is denoted ε. If S is a set then $|S|$ denotes its size, S^a denotes the set of a-vectors over S, $S^{a\times b}$ denotes the set of a by b matrices with entries in S, and so on. The (i,j)-th entry of a 2 dimensional matrix $\mathbf{M}$ is denoted $\mathbf{M}[i,j]$ and the (i,j,k)-th entry of a 3 dimensional matrix $\mathbf{M}$ is denoted $\mathbf{M}[i,j,k]$. If $\mathbf{M}$ is a n by μ matrix then $\mathbf{M}[j,\cdot]$ denotes the vector $(\mathbf{M}[j,1],\ldots,\mathbf{M}[j,\mu])$. If $a = (a_1,\ldots,a_n)$ then $(a_1,\ldots,a_n) \leftarrow a$ means we parse a as shown. Unless otherwise indicated, an algorithm may be randomized. By $y \xleftarrow{\$} A(x_1,x_2,\ldots)$ we denote the operation of running A on inputs $x_1,x_2,\ldots$ and fresh coins and letting y denote the output. We denote by $[A(x_1,x_2,\ldots)]$ the set of all possible outputs of A on inputs $x_1,x_2,\ldots$. The (Kronecker) delta function Δ is defined by $\Delta(a,b) = 1$ if $a = b$ and 0 otherwise. If a,b are equal-length vectors of reals then $\langle a,b\rangle = a[1]b[1] + \cdots + a[|a|]b[|b|]$ denotes their inner product.

Games. A game —look at Figure 2 for an example— has an **Initialize** procedure, procedures to respond to adversary oracle queries, and a **Finalize** procedure. To execute a game G is executed with an adversary A means to run the adversary and answer its oracle queries by the corresponding procedures of G. The adversary must make exactly one query to **Initialize**, this being its first oracle query. (This means the adversary can give **Initialize** an input, an extension of the usual convention [14].) It must make exactly one query to **Finalize**, this being its last oracle query. The reply to this query, denoted G^A, is called the output of the game, and we let "G^A" denote the event that this game output takes value true. Boolean flags are assumed initialized to false.

IBTDFs. An *identity-based trapdoor function* (IBTDF) is a tuple $\mathsf{F} = (\mathsf{F.Pg}, \mathsf{F.Kg}, \mathsf{F.Ev}, \mathsf{F.Ev}^{-1})$ of algorithms with associated input space InSp and identity space IDSp. The parameter generation algorithm $\mathsf{F.Pg}$ takes no input and returns common parameters $pars$ and a master secret key msk. On input $pars, msk, id$, the key generation algorithm $\mathsf{F.Kg}$ produces a decryption key dk for identity id. For any $pars$ and $id \in \mathsf{IDSp}$, the *deterministic* evaluation algorithm $\mathsf{F.Ev}$ defines a function $\mathsf{F.Ev}(pars, id, \cdot)$ with domain InSp. We require *correct inversion*: For any $pars$, any $id \in \mathsf{IDSp}$ and any $dk \in [\mathsf{F.Kg}(pars, id)]$, the deterministic inversion algorithm $\mathsf{F.Ev}^{-1}$ defines a function that is the inverse of $\mathsf{F.Ev}(pars, id, \cdot)$, meaning $\mathsf{F.Ev}^{-1}(pars, id, dk, \mathsf{F.Ev}(pars, id, x)) = x$ for all $x \in \mathsf{InSp}$.

E-IBTDF. To unify and shorten the selective and adaptive cases of our analyses it is useful to define and specify a more general primitive. An extended IBTDF (E-IBTDF) $\mathsf{E} = (\mathsf{E.Pg}, \mathsf{E.Kg}, \mathsf{E.Ev}, \mathsf{E.Ev}^{-1})$ consists of four algorithms that are just like the ones for an IBTDF except that $\mathsf{F.Pg}$ takes an additional *auxiliary* input from an auxiliary input space AxSp. Fixing a particular auxiliary input $aux \in \mathsf{AxSp}$ for $\mathsf{F.Pg}$ results in an IBTDF scheme that we denote $\mathsf{E}(aux)$ and call the IBTDF induced by aux. Not all

these induced schemes need, however, satisfy the correct inversion requirement. If the one induced by *aux* does, we say that *aux* grants invertibility. Looking ahead we will build an E-IBTDF and then obtain our IBTDF as the one induced by a particular auxiliary input, the other induced schemes being the basis of the siblings and being used in the proof.

ONE-WAYNESS. One-wayness of IBTDF $\mathsf{F} = (\mathsf{F.Pg}, \mathsf{F.Kg}, \mathsf{F.Ev}, \mathsf{F.Ev}^{-1})$ is defined via game OW_{F} of Figure 2. The adversary is allowed only one query to its challenge oracle **Ch**. The advantage of such an adversary I is $\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{F}}(I) = \Pr\left[\,\mathrm{OW}^{I}_{\mathsf{F}}\,\right]$.

SELECTIVE VERSUS ADAPTIVE ID. We are interested in both these variants for all the notions we consider. To avoid a proliferation of similar definitions, we capture the variants instead via different adversary classes relative to the same game. To exemplify, consider game OW_{F} of Figure 2. Say that an adversary A is *selective-id* if the identity *id* in its queries to **Initialize** and **Ch** is always the same, and say it is *adaptive-id* if this is not necessarily true. Selective-id security for one-wayness is thus captured by restricting attention to selective-id adversaries and full (adaptive-id) security by allowing adaptive-id adversaries. Now, adopt the same definitions of selective and adaptive adversaries relative to *any* game that provides procedures called **Initialize** and **Ch**, regardless of how these procedures operate. In this way, other notions we will introduce, including partial lossiness defined via games also in Figure 2, will automatically have selective-id and adaptive-id security versions.

PARTIAL LOSSINESS. We first provide the formal definitions and later explain them and their relation to standard definitions. If f is a function with domain a (non-empty) set $\mathrm{Dom}(f)$ then its image is $\mathrm{Im}(f) = \{\ f(x)\ :\ x \in \mathrm{Dom}(f)\ \}$. We define the *lossiness* $\lambda(f)$ of f via

$$\lambda(f) = \lg \frac{|\mathrm{Dom}(f)|}{|\mathrm{Im}(f)|} \qquad \text{or equivalently} \qquad |\mathrm{Im}(f)| = |\mathrm{Dom}(f)| \cdot 2^{-\lambda(f)}\ .$$

We say that f is ℓ-lossy if $\lambda(f) \geq \ell$. Let IBTDF $\mathsf{F} = (\mathsf{F.Pg}, \mathsf{F.Kg}, \mathsf{F.Ev}, \mathsf{F.Ev}^{-1})$ be an IBTDF with associated input space InSp and identity space IDSp. A *sibling* for F is an E-IBTDF $\mathsf{LF} = (\mathsf{LF.Pg}, \mathsf{LF.Kg}, \mathsf{F.Ev}, \mathsf{F.Ev}^{-1})$ whose evaluation and inversion algorithms, as the notation indicates, are those of F and whose auxiliary input space is IDSp. Algorithm $\mathsf{LF.Pg}$ will use this input in the selective-id case and ignore it in the adaptive-id case. Consider games $\mathrm{Real}_{\mathsf{F}}$ and $\mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}$ of Figure 2. The first uses the real parameter and key-generation algorithms while the second uses the sibling ones. A los-adversary A is allowed just one **Ch** query, and the games do no more than record the challenge identity id^*. The advantage of the adversary is *not*, as usual, the difference in the probabilities that the games return true, but is instead parameterized by a probability $\delta \in [0, 1]$and defined via

$$\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) = \delta \cdot \Pr\left[\,\mathrm{Real}^{A}_{\mathsf{F}}\,\right] - \Pr\left[\,\mathrm{Lossy}^{A}_{\mathsf{F},\mathsf{LF},\ell}\,\right]\ . \tag{1}$$

DISCUSSION. The PW [52] notion of lossy TDFs in the public-key setting asks for an alternative "sibling" key-generation algorithm, producing a public key but no secret key, such that two conditions hold. The first, which is combinatorial, asks that the functions defined by sibling keys are lossy. The second, which is computational, asks that real and sibling keys are indistinguishable. The first change for the IB setting is that one needs an alternative parameter generation algorithm which produces not only *pars* but a master secret key *msk*, and an alternative key-generation algorithm that, based on *msk*, can issue decryption keys to users. Now we would like to ask that the function $\mathsf{F.Ev}(\mathit{pars}, id^*, \cdot)$ be lossy on the challenge identity id^* when *pars* is generated via $\mathsf{LF.Pg}$, but, in the adaptive-id case, we do not know id^* in advance. Thus the requirement is made via the games.

We would like to define the advantage normally, meaning with $\delta = 1$, but the resulting notion is not achievable in the adaptive-id case. (This can be shown via attack.) With the relaxation, a low (close to zero) advantage means that the probability that the adversary finds a lossy identity id^* and then outputs 1 is less than the probability that it merely outputs 1 by a factor not much less than δ. Roughly, it means that a δ fraction of identities are lossy. The advantage represents the computational loss while δ represents a necessary information-theortic loss.

IBE. Recall that an IBE scheme $\mathsf{IBE} = (\mathsf{IBE.Pg}, \mathsf{IBE.Kg}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ is a tuple of algorithms with associated message space InSp and identity space IDSp. The parameter generation algorithm $\mathsf{IBE.Pg}$ takes no input and returns common parameters *pars* and a master secret key *msk*. On input $pars, msk, id$, the key generation algorithm $\mathsf{IBE.Kg}$ produces a decryption key *dk* for identity *id*. On input $pars$, $id \in \mathsf{IDSp}$ and a message $M \in \mathsf{InSp}$ the encryption algorithm $\mathsf{IBE.Enc}$ returns a ciphertext. The decryption algorithm $\mathsf{IBE.Dec}$ is deterministic. The scheme has decryption error ϵ if $\Pr[\mathsf{IBE.Dec}(pars, id, dk, \mathsf{IBE.Enc}(pars, id, M)) \neq M] \leq \epsilon$ for all *pars*, all $id \in \mathsf{IDSp}$, all $dk \in [\mathsf{F.Kg}(pars, id)]$ and all $M \in \mathsf{InSp}$. We say that IBE is deterministic if $\mathsf{IBE.Enc}$ is deterministic. A deterministic IBE scheme is identical to an IBTDF.

3 Implications of Partial Lossiness

Theorem 3.2 shows that partial lossiness implies one-wayness. We discuss other applications in Appendix B. We first need a simple lemma.

Lemma 3.1 *Let f be a function with non-empty domain $\mathrm{Dom}(f)$. Then for any adversary A*

$$\Pr[A(y) = x \,:\, x \xleftarrow{\$} \mathrm{Dom}(f)\,;\, y \leftarrow f(x)] \;\leq\; 2^{-\lambda(f)}\,. \quad \blacksquare$$

Proof of Lemma 3.1: For $y \in \mathrm{Im}(f)$ let $f^{-1}(y)$ be the set of all $x \in \mathrm{Dom}(f)$ such that $f(x) = y$. The probability in question is

$$\sum_{y \in \mathrm{Im}(f)} \Pr[\, A(y) = x \mid f(x) = y \,] \cdot \Pr[\, f(x) = y \,] \;\leq\; \sum_{y \in \mathrm{Im}(f)} \frac{1}{|f^{-1}(y)|} \cdot \frac{|f^{-1}(y)|}{|\mathrm{Dom}(f)|} \;=\; \frac{|\mathrm{Im}(f)|}{|\mathrm{Dom}(f)|} \;=\; 2^{-\lambda(f)}$$

where the probability is over x chosen at random from $\mathrm{Dom}(f)$ and the coins of A if any. (Since A is unbounded, it can be assumed wlog to be deterministic.) $\blacksquare$

Theorem 3.2 [δ-lossiness implies one-wayness] *Let $\mathsf{F} = (\mathsf{F.Pg}, \mathsf{F.Kg}, \mathsf{F.Ev}, \mathsf{F.Ev}^{-1})$ be a IBTDF with associated input space InSp. Let $\mathsf{LF} = (\mathsf{LF.Pg}, \mathsf{LF.Kg}, \mathsf{F.Ev}, \mathsf{F.Ev}^{-1})$ be a lossy sibling for F. Let $\delta > 0$ and let $\ell \geq 0$. Then for any ow-adversary I there is a los-adversary A such that*

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{F}}(I) \leq \frac{\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) + 2^{-\ell}}{\delta}\,. \tag{2}$$

The running time of A is that if I plus the time for a computation of $\mathsf{F.Ev}$. If I is a selective adversary then so is A. $\blacksquare$

In asymptotic terms, the theorem says that δ-lossiness implies one-wayness as long as δ^{-1} is bounded above by a polynomial in the security parameter and ℓ is super-logarithmic. This means δ need only be non-negligible. The last sentence of the theorem, saying that if I is selective then so is A, is important because it says that the theorem covers both the selective and adaptive security cases, meaning selective δ-lossiness implies selective one-wayness and adaptive δ-lossiness implies adaptive one-wayness.

Proof of Theorem 3.2: Adversary A runs I. When I makes query $\mathbf{Initialize}(id)$, adversary A does the same, obtaining *pars* and returning this to I. Adversary A answers I's queries to its **GetDK** oracle via its own oracle of the same name. When I makes its (single) **Ch** query id^*, adversary A also makes query $\mathbf{Ch}(id^*)$. Additionally, it picks x at random from InSp and returns $y = \mathsf{F.Ev}(pars, id^*, x)$ to I. The latter eventually halts with output x'. Adversary A returns 1 if $x' = x$ and 0 otherwise. By design we clearly have $\Pr\left[\mathrm{Real}^A_{\mathsf{F}}\right] = \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{F}}(I)$. But game $\mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}$ returns true only if $\mathsf{F.Ev}(pars, id^*, \cdot)$ is ℓ-lossy, in which case the probability that $x = x'$ is small by Lemma 3.1. In detail, assuming wlog that I never queries id^* to **GetDK**, we have

$$\begin{aligned}
\Pr\left[\mathrm{Lossy}^A_{\mathsf{F},\mathsf{LF},\ell}\right] &= \Pr\left[\, x = x' \mid \lambda(\mathsf{F.Ev}(pars, id^*, \cdot)) \geq \ell \,\right] \cdot \Pr\left[\, \lambda(\mathsf{F.Ev}(pars, id^*, \cdot)) \geq \ell \,\right] \\
&\leq \Pr\left[\, x = x' \mid \lambda(\mathsf{F.Ev}(pars, id^*, \cdot)) \geq \ell \,\right] \;\leq\; 2^{-\ell}\,,
\end{aligned}$$

the last inequality by Lemma 3.1 applied to the function $f = \mathsf{F.Ev}(pars, id^*, \cdot)$. From Equation (1) we have

$$\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) \;=\; \delta \cdot \Pr\left[\,\mathrm{Real}^A_{\mathsf{F}}\,\right] - \Pr\left[\,\mathrm{Lossy}^A_{\mathsf{F},\mathsf{LF},\ell}\,\right] \;\geq\; \delta \cdot \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{F}}(I) - 2^{-\ell}\;.$$

Equation (2) follows. ▌ In Section B we discuss the application to deterministic and hedged IBE.

4 IB-TDFs from pairings

In Section 3 we show that δ-lossiness implies one-wayness in both the selective and adaptive cases. We now show how to achieve δ-lossiness using pairings.

Setup. Throughout we fix a bilinear map $\mathbf{e}$: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ where $\mathbb{G}, \mathbb{G}_T$ are groups of prime order p. By $\mathbf{1}, \mathbf{1}_T$ we denote the identity elements of $\mathbb{G}, \mathbb{G}_T$, respectively. By $\mathbb{G}^* = \mathbb{G} - \{\mathbf{1}\}$ we denote the set of generators of $\mathbb{G}$. The advantage of a dlin-adversary B is

$$\mathbf{Adv}^{\mathrm{dlin}}(B) \;=\; 2\Pr[\mathrm{DLIN}^B] - 1\;,$$

where game DLIN is as follows. The **Initialize** procedure picks $g, \hat{g}$ at random from $\mathbb{G}^*$, s at random from $\mathbb{Z}_p^*$, $\hat{s}$ at random from $\mathbb{Z}_p$ and X at random from $\mathbb{G}$. It picks a random bit b. If $b = 1$ it lets $T \leftarrow X^{s+\hat{s}}$ and otherwise picks T at random from $\mathbb{G}$. It returns $(g, \hat{g}, g^s, \hat{g}^{\hat{s}}, X, T)$ to the adversary B. The adversary outputs a bit b' and **Finalize**, given b' returns true if $b = b'$ and false otherwise. For integer $\mu \geq 1$, vectors $\mathbf{U} \in \mathbb{G}^{\mu+1}$ and $\mathbf{y} \in \mathbb{Z}_p^{\mu+1}$, and vector $id \in \mathbb{Z}_p^{\mu}$ we let

$$\overline{id} = (1, id[1], \ldots, id[\mu]) \in \mathbb{Z}_p^{\mu+1} \qquad \text{and} \qquad \mathcal{H}(\mathbf{U}, id) = \textstyle\prod_{k=0}^{\mu} \mathbf{U}[k]^{\overline{id}[k]}\;.$$

$\mathcal{H}$ is the BB hash function [17] when $\mu = 1$, and the Waters' one [23] when $\mathsf{IDSp} = \{0,1\}^{\mu}$ and an $id \in \mathsf{IDSp}$ is viewed as a μ-vector over $\mathbb{Z}_p$. We also let

$$f(\mathbf{y}, id) = \textstyle\sum_{k=0}^{\mu} \mathbf{y}[k]\overline{id}[k] \qquad \text{and} \qquad \overline{f}(\mathbf{y}, id) = f(\mathbf{y}, id) \bmod p\;.$$

4.1 Overview

In the Peikert-Waters [52] design, the matrix entries are ciphertexts of an underlying homomorphic encryption scheme, and the function output is a vector of ciphertexts of the same scheme. We begin by presenting an IBE scheme, that we call the basic IBE scheme, such that the function outputs of our eventual IB-TDF will be a vector of ciphertexts of this IBE scheme. Towards building the IB-TDF, the first difficulty we run into in setting up the matrix is that ciphertexts depend on the identity and we cannot have a different matrix for every identity. Thus, our approach is more intrusive. We will have many matrices which contain certain "atoms" from which, given an identity, one can reconstruct ciphertexts of the IBE scheme. The result of this intrusive approach is that security of the IB-TDF relies on more than security of the base IBE scheme. Our ciphertext pseudorandomness lemma (Lemma 4.1) shows something stronger, namely that even the atoms from which the ciphertexts are created look random under DLIN. This will be used to establish Lemma 4.2, which moves from the real to the lossy setup. The heart of the argument is the proofs of the lemmas, which are in the appendices.

We introduce a general framework that allows us to treat both the selective-id and adaptive-id cases in as unified a way as possible. We will first specify an E-IBTDF. The selective-id and adaptive-id IB-TDFs are obtained via different auxiliary inputs. Furthermore, the siblings used to prove lossiness also emanate from this E-IBTDF. With this approach, the main lemmas become usable in both the selective-id and adaptive-id cases with only minor adjustments for the latter due to artifical aborts. This saves us from repeating similar arguments and significantly compacts the proof.

4.2 Our basic IBE scheme

We associate to any integer $\mu \geq 1$ and any identity space $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$ an IBE scheme $\mathsf{IBE}[\mu, \mathsf{IDSp}]$ that has message space $\{0,1\}$ and algorithms as follows:

1. Parameters: Algorithm $\mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Pg}$ lets $g \xleftarrow{\$} \mathbb{G}^*$; $t \xleftarrow{\$} \mathbb{Z}_p^*$; $\hat{g} \leftarrow g^t$. It then lets $H, \hat{H} \xleftarrow{\$} \mathbb{G}$; $\mathbf{U}, \hat{\mathbf{U}} \xleftarrow{\$} \mathbb{G}^{\mu+1}$. It returns $pars = (g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}})$ as the public parameters and $msk = t$ as the master secret key.
2. Key generation: Given parameters $(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}})$, master secret t and identity $id \in \mathsf{IDSp}$, algorithm $\mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Kg}$ returns decryption key (D_1, D_2, D_3, D_4) computed by letting $r, \hat{r} \xleftarrow{\$} \mathbb{Z}_p$ and setting
$$D_1 \leftarrow \mathcal{H}(\mathbf{U}, id)^{tr} \cdot H^{t\hat{r}} \; ; \; D_2 \leftarrow \mathcal{H}(\hat{\mathbf{U}}, id)^{r} \cdot \hat{H}^{\hat{r}} \; ; \; D_3 \leftarrow g^{-tr} \; ; \; D_4 \leftarrow g^{-t\hat{r}} \; .$$
3. Encryption: Given parameters $(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}})$, identity $id \in \mathsf{IDSp}$ and message $M \in \{0,1\}$, algorithm $\mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Enc}$ returns ciphertext (C_1, C_2, C_3, C_4) computed as follows. If $M = 0$ then it lets $s, \hat{s} \xleftarrow{\$} \mathbb{Z}_p$ and $C_1 \leftarrow g^s$; $C_2 \leftarrow \hat{g}^{\hat{s}}$; $C_3 \leftarrow \mathcal{H}(\mathbf{U}, id)^s \cdot \mathcal{H}(\hat{\mathbf{U}}, id)^{\hat{s}}$; $C_4 \leftarrow H^s \hat{H}^{\hat{s}}$. If $M = 1$ it lets $C_1, C_2, C_3, C_4 \xleftarrow{\$} \mathbb{G}$.
4. Decryption: Given parameters $(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}})$, identity $id \in \mathsf{IDSp}$, decryption key (D_1, D_2, D_4, D_4) for id and ciphertext (C_1, C_2, C_3, C_4), algorithm $\mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Dec}$ returns 0 if $\mathbf{e}(C_1, D_1)\mathbf{e}(C_2, D_2)$ $\mathbf{e}(C_3, D_3)\mathbf{e}(C_4, D_4) = \mathbf{1}_T$ and 1 otherwise.

This scheme has non-zero decryption error (at most $2/p$) yet our IBTDF will have zero inversion error. This scheme turns out to be IND-CPA+ANON-CPA although we will not need this in what follows. Instead we will have to consider a distinguishing game related to this IBE scheme and our IBTDF. In Appendix A we give a (more natural) variant of $\mathsf{IBE}[\mu, \mathsf{IDSp}]$ that is more efficient and encrypts strings rather than bits. The improved IBE scheme can still be proved IND-CPA+ANON-CPA but it cannot be used for our purpose of building IB-TDFs.

4.3 Our E-IBTDF and IB-TDF

Our E-IBTDF $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}]$ is associated to any integers $n, \mu \geq 1$ and any identity space $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$. It has message space $\{0,1\}^n$ and auxiliary input space $\mathbb{Z}_p^{\mu+1}$, and the algorithms are as follows:

1. Parameters: Given auxiliary input $\mathbf{y}$, algorithm $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Pg}$ lets $g \xleftarrow{\$} \mathbb{G}^*$; $t \xleftarrow{\$} \mathbb{Z}_p^*$; $\hat{g} \leftarrow g^t$; $U \xleftarrow{\$} \mathbb{G}^*$. It then lets $\mathbf{H}, \hat{\mathbf{H}} \xleftarrow{\$} \mathbb{G}^n$; $\mathbf{V}, \hat{\mathbf{V}} \xleftarrow{\$} \mathbb{G}^{n \times (\mu+1)}$ and $\mathbf{s} \xleftarrow{\$} (\mathbb{Z}_p^*)^n$; $\hat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_p^n$. It returns $pars = (g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}, \mathbf{H}, \hat{\mathbf{H}}, \mathbf{V}, \hat{\mathbf{V}}, U)$ as the public parameters and $msk = t$ as the master secret key where for $1 \leq i, j \leq n$ and $0 \leq k \leq \mu$:
$$\mathbf{G}[i] \leftarrow g^{\mathbf{s}[i]} \; ; \; \hat{\mathbf{G}}[i] \leftarrow \hat{g}^{\hat{\mathbf{s}}[i]} \; ; \; \mathbf{J}[i,j] \leftarrow \mathbf{H}[j]^{\mathbf{s}[i]} \hat{\mathbf{H}}[j]^{\hat{\mathbf{s}}[i]} \; ; \; \mathbf{W}[i,j,k] \leftarrow \mathbf{V}[j,k]^{\mathbf{s}[i]} \hat{\mathbf{V}}[j,k]^{\hat{\mathbf{s}}[i]} U^{\mathbf{s}[i]\mathbf{y}[k]\Delta(i,j)} \; ,$$
where we recall that $\Delta(i,j) = 1$ if $i = j$ and 0 otherwise is the Kronecker Delta function.
2. Key generation: Given parameters $(g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}, \mathbf{H}, \hat{\mathbf{H}}, \mathbf{V}, \hat{\mathbf{V}}, U)$, master secret t and identity $id \in \mathsf{IDSp}$, algorithm $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Kg}$ returns decryption key $(\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4)$ where $\mathbf{r} \xleftarrow{\$} (\mathbb{Z}_p^*)^n$; $\hat{\mathbf{r}} \xleftarrow{\$} \mathbb{Z}_p^n$ and for $1 \leq i \leq n$
$$\mathbf{D}_1[i] \leftarrow \mathcal{H}(\mathbf{V}[i,\cdot], id)^{t\mathbf{r}[i]} \cdot \mathbf{H}[i]^{t\hat{\mathbf{r}}[i]} \; ; \; \mathbf{D}_2[i] \leftarrow \mathcal{H}(\hat{\mathbf{V}}[i,\cdot], id)^{\mathbf{r}[i]} \cdot \hat{H}[i]^{\hat{\mathbf{r}}[i]} \; ; \; \mathbf{D}_3[i] \leftarrow g^{-t\mathbf{r}[i]} \; ; \; \mathbf{D}_4[i] \leftarrow g^{-t\hat{\mathbf{r}}[i]} \; .$$
3. Evaluate: Given parameters $(g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}, \mathbf{H}, \hat{\mathbf{H}}, \mathbf{V}, \hat{\mathbf{V}}, U)$, identity $id \in \mathsf{IDSp}$ and input $x \in \{0,1\}^n$, algorithm $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Ev}$ returns $(C_1, C_2, \mathbf{C}_3, \mathbf{C}_4)$ where for $1 \leq j \leq n$
$$C_1 \leftarrow \textstyle\prod_{i=1}^n \mathbf{G}[i]^{x[i]} \; ; \; C_2 \leftarrow \prod_{i=1}^n \hat{\mathbf{G}}[i]^{x[i]} \; ; \; \mathbf{C}_3[j] \leftarrow \prod_{i=1}^n \prod_{k=0}^\mu \mathbf{W}[i,j,k]^{x[i]\overline{id}[k]} \; ; \; \mathbf{C}_4[j] \leftarrow \prod_{i=1}^n \mathbf{J}[i,j]^{x[i]}$$
4. Invert: Given parameters $(g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}, \mathbf{H}, \hat{\mathbf{H}}, \mathbf{V}, \hat{\mathbf{V}}, U)$, identity $id \in \mathsf{IDSp}$, decryption key $(\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4)$ for id and output (ciphertext) $(C_1, C_2, \mathbf{C}_3, \mathbf{C}_4)$, algorithm $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Ev}^{-1}$ returns $x \in$

$\{0,1\}^n$ where for $1 \le j \le n$ it sets $x[j] = 0$ if $\mathbf{e}(C_1, \mathbf{D}_1[j])\mathbf{e}(C_2, \mathbf{D}_2[j])\mathbf{e}(\mathbf{C}_3[j], \mathbf{D}_3[j])\mathbf{e}(\mathbf{C}_4[j], \mathbf{D}_4[j]) = \mathbf{1}_T$ and 1 otherwise.

INVERTIBILITY. We observe that if parameters $(g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}, \mathbf{H}, \hat{\mathbf{H}}, \mathbf{V}, \hat{\mathbf{V}}, U)$ were generated with auxiliary input $\mathbf{y}$ and $(C_1, C_2, \mathbf{C}_3, \mathbf{C}_4) = \overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Ev}((g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}), id, x)$ then for $1 \le j \le n$

$$
\begin{aligned}
C_1 &= \textstyle\prod_{i=1}^{n} g^{\mathbf{s}[i]x[i]} = g^{\langle \mathbf{s}, x\rangle} && (3)\\
C_2 &= \textstyle\prod_{i=1}^{n} \hat{g}^{\hat{\mathbf{s}}[i]x[i]} = \hat{g}^{\langle \hat{\mathbf{s}}, x\rangle} && (4)\\
\mathbf{C}_3[j] &= \textstyle\prod_{i=1}^{n}\prod_{k=0}^{\mu} \mathbf{V}[j,k]^{\mathbf{s}[i]x[i]\overline{id}[k]}\hat{\mathbf{V}}[j,k]^{\hat{\mathbf{s}}[i]x[i]\overline{id}[k]} U^{\mathbf{s}[i]x[i]\mathbf{y}[k]\overline{id}[k]\Delta(i,j)}\\
&= \textstyle\prod_{i=1}^{n} \mathcal{H}(\mathbf{V}[j,\cdot], id)^{\mathbf{s}[i]x[i]}\mathcal{H}(\hat{\mathbf{V}}[j,\cdot], id)^{\hat{\mathbf{s}}[i]x[i]} U^{\mathbf{s}[i]x[i]f(\mathbf{y},id)\Delta(i,j)}\\
&= \mathcal{H}(\mathbf{V}[j,\cdot], id)^{\langle \mathbf{s}, x\rangle}\mathcal{H}(\hat{\mathbf{V}}[j,\cdot], id)^{\langle \hat{\mathbf{s}}, x\rangle} U^{\mathbf{s}[j]x[j]f(\mathbf{y},id)} && (5)\\
\mathbf{C}_4[j] &= \textstyle\prod_{i=1}^{n} \mathbf{H}[j]^{\mathbf{s}[i]x[i]}\hat{\mathbf{H}}[j]^{\hat{\mathbf{s}}[i]x[i]} = \mathbf{H}[j]^{\langle \mathbf{s}, x\rangle}\hat{\mathbf{H}}[j]^{\langle \hat{\mathbf{s}}, x\rangle}\ . && (6)
\end{aligned}
$$

Thus if $x[j] = 0$ then $(C_1, C_2, \mathbf{C}_3[j], \mathbf{C}_4[j])$ is an encryption, under our base IBE scheme, of the message 0, with coins $\langle \mathbf{s}, x\rangle \bmod p, \langle \hat{\mathbf{s}}, x\rangle \bmod p$, parameters $(g, \hat{g}, \mathbf{H}[j], \hat{\mathbf{H}}[j], \mathbf{V}[j,\cdot], \hat{\mathbf{V}}[j,\cdot])$ and identity id. The inversion algorithm will thus correctly recover $x[j] = 0$. On the other hand suppose $x[j] = 1$. Then $\mathbf{e}(C_1, \mathbf{D}_1[j])\mathbf{e}(C_2, \mathbf{D}_2[j])\mathbf{e}(\mathbf{C}_3[j], \mathbf{D}_3[j])\mathbf{e}(\mathbf{C}_4[j], \mathbf{D}_4[j]) = \mathbf{e}(U^{\mathbf{s}[j]x[j]f(\mathbf{y},id)}, \mathbf{D}_3[j])$. Now suppose $f(\mathbf{y}, id) \bmod p \neq 0$. Then $U^{\mathbf{s}[j]x[j]f(\mathbf{y},id)} \neq \mathbf{1}$ because we chose $\mathbf{s}[j]$ to be non-zero modulo p and $\mathbf{D}_3[j] \neq \mathbf{1}$ because we chose $\mathbf{r}[j]$ to be non-zero modulo p. So the result of the pairing is never $\mathbf{1}_T$, meaning the inversion algorithm will again correctly recover $x[j] = 1$. We have established that auxiliary input $\mathbf{y}$ grants invertibility, meaning induced IBTDF $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}](\mathbf{y})$ satisfies the correct inversion condition, if $f(\mathbf{y}, id) \bmod p \neq 0$ for all $id \in \mathsf{IDSp}$.

OUR IBTDF. We associate to any integers $n, \mu \ge 1$ and any identity space $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$ the IBTDF scheme induced by our E-IBTDF $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}]$ via auxiliary input $\mathbf{y} = (1, 0, \ldots, 0) \in \mathbb{Z}_p^{\mu+1}$, and denote this IBTDF scheme by $\overline{\mathsf{F}}[n, \mu, \mathsf{IDSp}]$. This IBTDF satisfies the correct inversion requirement because $f(\mathbf{y}, id) = \overline{id}[0] = 1 \not\equiv 0 \pmod p$ for all id. We will show that this IBTDF is selective-id secure when $\mu = 1$ and $\mathsf{IDSp} = \mathbb{Z}_p$, and adaptive-id secure when $\mathsf{IDSp} = \{0,1\}^\mu$. In the first case, it is fully lossy (i.e. 1-lossy) and in the second it is δ-lossy for appropriate δ. First we prove two technical lemmas that we will use in both cases.

4.4 Ciphertext pseudorandomness lemma

Consider games ReC, RaC of Figure 3 associated to some choice of $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$. The adversary provides the **Initialize** procedure with an auxiliary input $\mathbf{y} \in \mathbb{Z}_p^{\mu+1}$. Parameters are generated as per our base IBE scheme with the addition of U. The decryption key for id is computed as per our base IBE scheme except that the games refuse to provide it when $f(\mathbf{y}, id) = 0$. The challenge oracle, however, does not return ciphertexts of our IBE scheme. In game ReC, it returns group elements that resemble diagonal entries of the matrices in the parameters of our E-IBTDF, and in game RaC it returns random group elements. Notice that the challenge oracle does not take an identity as input. (Indeed, it has no input.) As usual it must be invoked exactly once. The following lemma says the games are indistinguishable under DLIN. The proof is in Section 4.7.

Lemma 4.1 *Let $\mu \ge 1$ be an integer and $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$. Let P be an adversary. Then there is an adversary B such that*

$$\Pr\left[\,\mathrm{ReC}^P\,\right] - \Pr\left[\,\mathrm{RaC}^P\,\right] \;\le\; (\mu + 2) \cdot \mathbf{Adv}^{\mathrm{dlin}}(B)\ . \qquad (7)$$

The running time of B is that of P plus some overhead.

proc Initialize($\mathbf{y}$) // ReC, RaC
$(pars, msk) \xleftarrow{\$} \mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Pg}$
$(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}}) \leftarrow pars$
$U \xleftarrow{\$} \mathbb{G}^*$
Return $(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}}, U)$

proc GetDK(id) // ReC, RaC
If $f(\mathbf{y}, id) = 0$ then $dk \leftarrow \perp$
Else $dk \leftarrow \mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Kg}(pars, msk, id)$
Return dk

proc Ch() // ReC
$s \xleftarrow{\$} \mathbb{Z}_p^*$; $\hat{s} \xleftarrow{\$} \mathbb{Z}_p$; $G \leftarrow g^s$; $\hat{G} \leftarrow \hat{g}^{\hat{s}}$; $S \leftarrow H^s \hat{H}^{\hat{s}}$
For $k = 0, \ldots, \mu$ do $\mathbf{Z}[k] \leftarrow (U^{\mathbf{y}[k]} \mathbf{U}[k])^s \hat{\mathbf{U}}[k]^{\hat{s}}$
Return $(G, \hat{G}, S, \mathbf{Z})$

proc Ch() // RaC
$G, \hat{G}, S \xleftarrow{\$} \mathbb{G}$; $\mathbf{Z} \xleftarrow{\$} \mathbb{G}^{\mu+1}$
Return $(G, \hat{G}, S, \mathbf{Z})$

proc Finalize(d') // ReC, RaC
Return $(d' = 1)$

Figure 3: Games ReC ("Real Ciphertexts") and RaC ("Random Ciphertexts") associated to $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$.

proc Initialize(id)
$\mathbf{y}_0 \xleftarrow{\$} \mathsf{Aux}(id)$; $\mathbf{y}_1 \leftarrow (1, 0, \ldots, 0)$; $\text{WIN} \leftarrow \mathsf{true}$
$g \xleftarrow{\$} \mathbb{G}^*$; $t \xleftarrow{\$} \mathbb{Z}_p^*$; $\hat{g} \leftarrow g^t$; $U \xleftarrow{\$} \mathbb{G}^*$
$\mathbf{H}, \hat{\mathbf{H}} \xleftarrow{\$} \mathbb{G}^n$; $\mathbf{V}, \hat{\mathbf{V}} \xleftarrow{\$} \mathbb{G}^{n \times (\mu+1)}$; $\mathbf{s} \xleftarrow{\$} (\mathbb{Z}_p^*)^n$; $\hat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_p^n$
For $i = 1, \ldots, n$ do
  $\mathbf{G}[i] \leftarrow g^{\mathbf{s}[i]}$; $\hat{\mathbf{G}}[i] \leftarrow \hat{g}^{\hat{\mathbf{s}}[i]}$
  For $j = 1, \ldots, n$ do
    $\mathbf{J}[i, j] \leftarrow \mathbf{H}[j]^{\mathbf{s}[i]} \hat{\mathbf{H}}[j]^{\hat{\mathbf{s}}[i]}$
    For $k = 0, \ldots, \mu$ do
      If ($i = j$ and $i \le l$) then $\mathbf{W}[i, j, k] \xleftarrow{\$} \mathbb{G}$
      Else $\mathbf{W}[i, j, k] \leftarrow \mathbf{V}[j, k]^{\mathbf{s}[i]} \hat{\mathbf{V}}[j, k]^{\hat{\mathbf{s}}[i]} U^{\mathbf{s}[i] \mathbf{y}_b[k] \Delta(i,j)}$
$pars \leftarrow (g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}, \mathbf{H}, \hat{\mathbf{H}}, \mathbf{V}, \hat{\mathbf{V}}, U)$; $msk \leftarrow t$
$IS \leftarrow \emptyset$; $id^* \leftarrow id$
Return $pars$

proc GetDK(id)
$IS \leftarrow IS \cup \{id\}$
If $f(\mathbf{y}_0, id) = 0$ then $\text{WIN} \leftarrow \mathsf{false}$; $dk \leftarrow \perp$
Else $dk \leftarrow \overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Kg}(pars, msk, id)$
Return dk

proc Ch(id)
$id^* \leftarrow id$
If $f(\mathbf{y}_0, id) \neq 0$ then $\text{WIN} \leftarrow \mathsf{false}$

proc Finalize(d')
Return $((d' = 1)$ and $(id^* \notin IS)$ and WIN$)$

Figure 4: Games $\text{RL}_{l,b}$ ($0 \le l \le n$ and $b \in \{0, 1\}$) associated to $n, \mu, \mathsf{IDSp}, \mathsf{Aux}$ for proof of Lemma 4.2.

4.5 Proof of Lemma 4.2

Consider the games of Figure 4. Game $\text{RL}_{l,b}$ makes the diagonal entries of $\mathbf{W}$ (namely all the $\mu + 1$ entries with $i = j$) random for $i \le l$ and otherwise makes them using $\mathbf{y}_b$. Game $\text{RL}_{0,1}$ is the same as game RL_0 and game $\text{RL}_{0,0}$ is the same as game RL_n. Games $\text{RL}_{n,0}, \text{RL}_{n,1}$ are identical: both make all diagonal entries of $\mathbf{W}$ (meaning, $i = j$) random, and when $i \neq j$ we have $\Delta(i, j) = 0$ so $\mathbf{y}_b(k)$ has no impact on $\mathbf{W}[i, j, k]$ in the Else statement. Thus we have

$$\Pr[\text{RL}_0^A] - \Pr[\text{RL}_n^A] \;=\; \big(\Pr[\text{RL}_{0,1}^A] - \Pr[\text{RL}_{n,1}^A]\big) + \big(\Pr[\text{RL}_{n,0}^A] - \Pr[\text{RL}_{0,0}^A]\big) \ .$$

We will design adversaries P_0, P_1 so that

$$\Pr[\text{ReC}^{P_0}] - \Pr[\text{RaC}^{P_0}] \;=\; \frac{1}{n} \cdot \big(\Pr[\text{RL}_{n,0}^A] - \Pr[\text{RL}_{0,0}^A]\big) \tag{8}$$

$$\Pr[\text{ReC}^{P_1}] - \Pr[\text{RaC}^{P_1}] \;=\; \frac{1}{n} \cdot \big(\Pr[\text{RL}_{0,1}^A] - \Pr[\text{RL}_{n,1}^A]\big) \ . \tag{9}$$

Adversary P picks $b \xleftarrow{\$} \{0, 1\}$ and runs P_b. This yields Equation (10). Now we present adversary P_b ($b \in \{0, 1\}$). It runs adversary A, responding to its oracle queries as follows.

When A makes query $\mathbf{Initialize}(id)$, adversary P_b begins with

$l \xleftarrow{\$} \{1,\ldots,n\}$; $\mathbf{y}_0 \xleftarrow{\$} \mathsf{Aux}(id)$; $\mathbf{y}_1 \leftarrow (1,0,\ldots,0)$; $\textsc{Win}_A \leftarrow \mathsf{true}$; $IS_A \leftarrow \emptyset$
$(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}}, U) \xleftarrow{\$} \mathbf{Initialize}(\mathbf{y}_b)$; $(G, \hat{G}, S, \mathbf{Z}) \xleftarrow{\$} \mathbf{Ch}()$.

Here P_b has called its own **Initialize** procedure with input $\mathbf{y}_b$ and then called its **Ch** procedure. Now it creates parameters *pars* for A as follows:

$\mathbf{h}, \hat{\mathbf{h}} \xleftarrow{\$} \mathbb{Z}_p^n$; $\mathbf{v}, \hat{\mathbf{v}} \xleftarrow{\$} \mathbb{Z}_p^{n\times(\mu+1)}$; $\mathbf{s} \xleftarrow{\$} (\mathbb{Z}_p^*)^n$; $\hat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_p^n$
For $i = 1,\ldots,n$ do
  If $(i = l)$ then $\mathbf{H}[i] \leftarrow H$; $\hat{\mathbf{H}}[i] \leftarrow \hat{H}$; $\mathbf{G}[i] \leftarrow G$; $\hat{\mathbf{G}}[i] \leftarrow \hat{G}$
  If $(i \neq l)$ then $\mathbf{H}[i] \leftarrow g^{\mathbf{h}[i]}$; $\hat{\mathbf{H}}[i] \leftarrow \hat{g}^{\hat{\mathbf{h}}[i]}$; $\mathbf{G}[i] \leftarrow g^{\mathbf{s}[i]}$; $\hat{\mathbf{G}}[i] \leftarrow \hat{g}^{\hat{\mathbf{s}}[i]}$
  For $k = 0,\ldots,\mu$ do
    If $(i = l)$ then $\mathbf{V}[i,k] \leftarrow \mathbf{U}[k]$; $\hat{\mathbf{V}}[i,k] \leftarrow \hat{\mathbf{U}}[k]$
    If $(i \neq l)$ then $\mathbf{V}[i,k] \leftarrow g^{\mathbf{v}[i,k]}$; $\hat{\mathbf{V}}[i,k] \leftarrow \hat{g}^{\hat{\mathbf{v}}[i,k]}$
For $i = 1,\ldots,n$ do
  For $j = 1,\ldots,n$ do
    If $(i = l$ and $j = i)$ then $\mathbf{J}[i,j] \leftarrow S$
    If $(i = l$ and $j \neq i)$ then $\mathbf{J}[i,j] \leftarrow G^{\mathbf{h}[j]}\hat{G}^{\hat{\mathbf{h}}[j]}$
    If $(i \neq l)$ then $\mathbf{J}[i,j] \leftarrow \mathbf{H}[j]^{\mathbf{s}[i]}\hat{\mathbf{H}}[j]^{\hat{\mathbf{s}}[i]}$
    For $k = 0,\ldots,\mu$ do
      If $(i = j$ and $i \leq l-1)$ then $\mathbf{W}[i,j,k] \xleftarrow{\$} \mathbb{G}$
      If $(i = j$ and $i = l)$ then $\mathbf{W}[i,j,k] \leftarrow \mathbf{Z}[k]$
      Else $\mathbf{W}[i,j,k] \leftarrow \mathbf{V}[j,k]^{\mathbf{s}[i]}\hat{\mathbf{V}}[j,k]^{\hat{\mathbf{s}}[i]}U^{\mathbf{s}[i]\mathbf{y}_b[k]\Delta(i,j)}$
pars $\leftarrow (g, \hat{g}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{J}, \mathbf{W}, \mathbf{H}, \hat{\mathbf{H}}, \mathbf{V}, \hat{\mathbf{V}}, U)$

It returns *pars* to A.

When adversary A makes query $\mathbf{GetDK}(id)$, adversary P_b proceeds as follows. In this code, **GetDK** is P_b's own oracle:

$IS_A \leftarrow IS_A \cup \{id\}$
If $f(\mathbf{y}_0, id) = 0$ then $\textsc{Win}_A \leftarrow \mathsf{false}$; $dk \leftarrow \perp$
Else
  $(D_1, D_2, D_3, D_4) \xleftarrow{\$} \mathbf{GetDK}(id)$
  $\mathbf{r}' \xleftarrow{\$} (\mathbb{Z}_p^*)^n$; $\hat{\mathbf{r}}' \xleftarrow{\$} \mathbb{Z}_p^n$
  For $i = 1,\ldots,n$ do
    If $i = l$ then $(\mathbf{D}_1[i], \mathbf{D}_2[i], \mathbf{D}_3[i], \mathbf{D}_4[i]) \leftarrow (D_1, D_2, D_3, D_4)$
    Else
      $\mathbf{D}_1[i] \leftarrow \mathcal{H}(\mathbf{V}[i,\cdot], id)^{\mathbf{r}'[i]}\mathbf{H}[i]^{\hat{\mathbf{r}}'[i]}$; $\mathbf{D}_2[i] \leftarrow g^{f(\hat{\mathbf{v}}, id)\mathbf{r}'[i]}g^{\hat{\mathbf{h}}[i]\hat{\mathbf{r}}[i]}$
      $\mathbf{D}_3[i] \leftarrow g^{-\mathbf{r}'[i]}$; $\mathbf{D}_4[i] \leftarrow g^{-\hat{\mathbf{r}}'[i]}$
  $dk \leftarrow (\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4)$

It returns dk to A. Notice that P_b's invocation of **GetDK** will never return $\perp$. In the case $b = 1$ this is true because $f(\mathbf{y}_1, \cdot) = 1 \neq 0$. In the case $b = 0$ it is true because the case $f(\mathbf{y}_0, id) = 0$ was excluded by the If statement. To justify the above simulation, define $\mathbf{r}, \hat{\mathbf{r}}$ by $\mathbf{r}[i] = \mathbf{r}'[i]/t$ and $\hat{\mathbf{r}}[i] = \hat{\mathbf{r}}'[i]/t$ for $i \neq l$ and $\mathbf{r}[l], \hat{\mathbf{r}}[l]$ as the randomness underlying (D_1, D_2, D_3, D_4). Then think of $\mathbf{r}, \hat{\mathbf{r}}$ as the randomness used by the real key generation algorithm. Here t is the secret key, so that $\hat{g} = g^t$.

When adversary A makes query $\mathbf{Ch}(id)$, adversary P_b proceeds as follows:

$id^* \leftarrow id$
If $f(\mathbf{y}_0, id) \neq 0$ then $\textsc{Win}_A \leftarrow \mathsf{false}$.

$\underline{\textbf{proc Initialize}(id)}$ // RL_0
$\mathbf{y}_0 \xleftarrow{\$} \mathsf{Aux}(id)$; $\mathbf{y}_1 \leftarrow (1, 0, \ldots, 0)$
$(pars, msk) \xleftarrow{\$} \overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Pg}(\mathbf{y}_1)$
$IS \leftarrow \emptyset$; $id^* \leftarrow id$; $\textsc{Win} \leftarrow \mathsf{true}$
Return $pars$

$\underline{\textbf{proc Initialize}(id)}$ // RL_n
$\mathbf{y}_0 \xleftarrow{\$} \mathsf{Aux}(id)$; $\mathbf{y}_1 \leftarrow (1, 0, \ldots, 0)$
$(pars, msk) \xleftarrow{\$} \overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Pg}(\mathbf{y}_0)$
$IS \leftarrow \emptyset$; $id^* \leftarrow id$; $\textsc{Win} \leftarrow \mathsf{true}$
Return $pars$

$\underline{\textbf{proc GetDK}(id)}$ // $\mathrm{RL}_0, \mathrm{RL}_n$
$IS \leftarrow IS \cup \{id\}$
If $f(\mathbf{y}_0, id) = 0$ then $\textsc{Win} \leftarrow \mathsf{false}$; $dk \leftarrow \perp$
Else $dk \leftarrow \overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Kg}(pars, msk, id)$
Return dk

$\underline{\textbf{proc Ch}(id)}$ // $\mathrm{RL}_0, \mathrm{RL}_n$
$id^* \leftarrow id$
If $f(\mathbf{y}_0, id) \neq 0$ then $\textsc{Win} \leftarrow \mathsf{false}$

$\underline{\textbf{proc Finalize}(d')}$ // $\mathrm{RL}_0, \mathrm{RL}_n$
Return $((d' = 1)$ and $(id^* \notin IS)$ and $\textsc{Win})$

Figure 5: Games $\mathrm{RL}_0, \mathrm{RL}_n$ ("Real-to-Losssy") associated to $n, \mu, \mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$ and auxiliary input generator algorithm Aux.

Finally, A halts with output d'. Adversaries P_0, P_1 compute their output differently. Adversary P_1 returns 1 if

$$(d' = 1) \text{ and } id^* \notin IS_A \text{ and } \textsc{Win}_A$$

and 0 otherwise. Adversary P_0 does the opposite, returning 0 if the above condition is true and 1 otherwise. We obtain Equations (8), (9) as follows:

$$\begin{aligned}
\Pr[\mathrm{ReC}^{P_1}] - \Pr[\mathrm{RaC}^{P_1}] &= \frac{1}{n}\sum_{l=1}^{n} \Pr[\mathrm{RL}^A_{l-1,1}] - \Pr[\mathrm{RL}^A_{l,1}] \\
&= \Pr[\mathrm{RL}^A_{0,1}] - \Pr[\mathrm{RL}^A_{n,1}] \\
\Pr[\mathrm{ReC}^{P_0}] - \Pr[\mathrm{RaC}^{P_0}] &= \frac{1}{n}\sum_{l=1}^{n} (1 - \Pr[\mathrm{RL}^A_{l-1,0}]) - (1 - \Pr[\mathrm{RL}^A_{l,0}]) \\
&= \frac{1}{n}\sum_{l=1}^{n} \Pr[\mathrm{RL}^A_{l,0}] - \Pr[\mathrm{RL}^A_{l-1,0}] \\
&= \Pr[\mathrm{RL}^A_{n,0}] - \Pr[\mathrm{RL}^A_{0,0}] \ .
\end{aligned}$$

4.6 Real-to-lossy lemma

Consider games $\mathrm{RL}_0, \mathrm{RL}_n$ of Figure 5 associated to some choice of $n, \mu, \mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$ and auxiliary input generator Aux for $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}]$. The latter is an algorithm that takes input an identity in IDSp and returns an auxiliary input in $\mathbb{Z}_p^{\mu+1}$. Game RL_0 obtains an auxiliary input $\mathbf{y}_0$ via Aux but generates parameters exactly as $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}].\mathsf{Pg}$ with the real auxiliary input $\mathbf{y}_1$. The game will return true under the same condition as game Real but additionally requiring that $f(\mathbf{y}_0, id) \neq 0$ for all $\mathbf{GetDK}(id)$ queries and $f(\mathbf{y}_0, id) = 0$ for the $\mathbf{Ch}(id)$ query. Game RL_n generates parameters with the auxiliary input provided by Aux but is otherwise identical to game RL_0. The following lemma says it is hard to distinguish these games. We will apply this by defining Aux in such a way that its output $\mathbf{y}_0$ results in a lossy setup. The proof of the following is in Section 4.5.

Lemma 4.2 *Let $n, \mu \geq 1$ be integers and $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$. Let Aux be an auxiliary input generator for $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}]$ and A an adversary. Then there is an adversary P such that*

$$\Pr[\mathrm{RL}^A_0] - \Pr[\mathrm{RL}^A_n] \ \leq \ 2n \cdot \left(\Pr\left[\mathrm{ReC}^P\right] - \Pr\left[\mathrm{RaC}^P\right]\right) \ . \tag{10}$$

proc Initialize($\mathbf{y}$) // $\mathrm{PC}, \mathrm{PC}_l$
$(pars, msk) \xleftarrow{\$} \mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Pg}$
$(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}}) \leftarrow pars$
$U \xleftarrow{\$} \mathbb{G}^*$
Return $(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}}, U)$

proc GetDK(id) // $\mathrm{PC}, \mathrm{PC}_l$
If $f(\mathbf{y}, id) = 0$ then $dk \leftarrow \perp$
Else $dk \leftarrow \mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Kg}(pars, msk, id)$
Return dk

proc Ch() // PC
$s \xleftarrow{\$} \mathbb{Z}_p^*$; $\hat{s} \xleftarrow{\$} \mathbb{Z}_p$; $G \leftarrow g^s$; $\hat{G} \leftarrow \hat{g}^{\hat{s}}$; $S \leftarrow H^s \hat{H}^{\hat{s}}$
For $k = 0, \ldots, \mu$ do $\mathbf{Z}[k] \leftarrow (U^{\mathbf{y}[k]} \mathbf{U}[k])^s \hat{\mathbf{U}}[k]^{\hat{s}}$
Return $(G, \hat{G}, S, \mathbf{Z})$

proc Ch() // PC_l
$s \xleftarrow{\$} \mathbb{Z}_p^*$; $\hat{s} \xleftarrow{\$} \mathbb{Z}_p$; $G \leftarrow g^s$; $\hat{G} \leftarrow \hat{g}^{\hat{s}}$; $S \xleftarrow{\$} \mathbb{G}$
For $k = 0, \ldots, l-1$ do $\mathbf{Z}[k] \xleftarrow{\$} \mathbb{G}$
For $k = l, \ldots, \mu$ do $\mathbf{Z}[k] \leftarrow (U^{\mathbf{y}[k]} \mathbf{U}[k])^s \hat{\mathbf{U}}[k]^{\hat{s}}$
Return $(G, \hat{G}, S, \mathbf{Z})$

proc Finalize(d') // $\mathrm{PC}, \mathrm{PC}_l$
Return $(d' = 1)$

Figure 6: Games $\mathrm{PC}, \mathrm{PC}_l$ $(0 \le l \le \mu + 1)$ associated to $\mathsf{IDSp} \subseteq \mathbb{Z}_p^{\mu+1}$ for the proof of Lemma 4.1.

The running time of P is that of A plus some overhead. If A is selective-id then so is P.

The last statement allows us to use the lemma in both the selective-id and adaptive-id cases.

4.7 Proof of Lemma 4.1

Consider the games of Figure 6. Game PC is the same as game ReC. Game PC_l $(0 \le l \le \mu + 1)$ makes S random and also makes the first $l - 1$ entries of $\mathbf{Z}$ random and the rest real. Thus $\mathrm{PC}_{\mu+1}$ is the same as RaC. We will design adversaries B_1, B_2 so that

$$\mathbf{Adv}^{\mathrm{dlin}}(B_1) = \Pr[\mathrm{PC}^P] - \Pr[\mathrm{PC}_0^P] \quad (11)$$

$$\mathbf{Adv}^{\mathrm{dlin}}(B_2) = \frac{1}{\mu+1}\left(\Pr[\mathrm{PC}_0^P] - \Pr[\mathrm{PC}_{\mu+1}^P]\right) \quad (12)$$

Adversary B will run B_1 with probability $1/(\mu+2)$ and B_2 with probability $(\mu+1)/(\mu+2)$. This yields Equation (7).

On input $(g, \hat{g}, g^s, \hat{g}^{\hat{s}}, H, T)$ where T is either $H^{s+\hat{s}}$ or random, adversary B_1 runs adversary P, responding to its oracle queries as follows. When P makes query **Initialize**($\mathbf{y}$), adversary B_1 lets

$\mathbf{u}, \hat{\mathbf{u}} \xleftarrow{\$} \mathbb{Z}_p^{\mu+1}$; $u, v \xleftarrow{\$} \mathbb{Z}_p$; $\hat{H} \leftarrow H \hat{g}^v$; $U \leftarrow \hat{g}^u$
For $k = 0, \ldots, \mu$ do $\mathbf{U}[k] \leftarrow U^{-\mathbf{y}[k]} g^{\mathbf{u}[k]}$; $\hat{\mathbf{U}}[k] \leftarrow \hat{g}^{\hat{\mathbf{u}}[k]}$

It returns $(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}}, U)$ to P. When P makes its (single) **Ch**() query, adversary B_1 lets

$S \leftarrow T \hat{g}^{v\hat{s}}$
For $k = 0, \ldots, \mu$ do $\mathbf{Z}[k] \leftarrow g^{s\mathbf{u}[k]} \hat{g}^{\hat{s}\hat{\mathbf{u}}[k]}$

It returns $(g^s, \hat{g}^{\hat{s}}, S, \mathbf{Z})$ to P. Notice that for $0 \le k \le \mu$

$$\mathbf{Z}[k] = g^{s\mathbf{u}[k]} \hat{g}^{\hat{s}\hat{\mathbf{u}}[k]} = (U^{\mathbf{y}[k]-\mathbf{y}[k]} g^{\mathbf{u}[k]})^s \hat{g}^{\hat{s}\hat{\mathbf{u}}[k]} = (U^{\mathbf{y}[k]} \mathbf{U}[k])^s \hat{\mathbf{U}}[k]^{\hat{s}} .$$

Also if $T = H^{s+\hat{s}}$ then $S = T \hat{g}^{v\hat{s}} = H^s (H \hat{g}^v)^{\hat{s}} = H^s \hat{H}^{\hat{s}}$ as in PC while if T is random, so is S, as in PC_0. When P makes query **GetDK**(id), adversary B_1 does the following:

If $f(\mathbf{y}, id) = 0$ then $dk \leftarrow \perp$

Else

$r', \hat{r}' \xleftarrow{\$} \mathbb{Z}_p$

$D_1 \leftarrow g^{-f(\mathbf{y},id)ur'} g^{f(\mathbf{u},id)r'} H^{-f(\mathbf{u},id)\hat{r}'/f(\mathbf{y},id)}$; $D_2 \leftarrow g^{f(\hat{u},id)r'} H^{-f(\mathbf{u},id)\hat{r}'/f(\mathbf{y},id)} \hat{H}^{u\hat{r}'}$

$D_3 \leftarrow H^{\hat{r}'/f(\mathbf{y},id)} g^{-r'}$; $D_4 \leftarrow \hat{g}^{-u\hat{r}'}$; $dk \leftarrow (D_1, D_2, D_3, D_4)$

It returns dk to P. We now show this key is properly distributed. Let h be such that $H = g^h$ and let

$$r \;=\; \frac{r'}{t} - \frac{h\hat{r}'}{tf(\mathbf{y},id)} \bmod p \quad \text{and} \quad \hat{r} \;=\; u\hat{r}' \bmod p\,.$$

Since $t, f(\mathbf{y}, uid)$ are non-zero modulo p and $r', \hat{r}'$ are random, $r, \hat{r}$ are random as well. The following computes the correct secret key components with the above randomness and shows that they are the ones of the simulation:

$$\begin{aligned}
\mathcal{H}(\mathbf{U}, id)^{tr} H^{t\hat{r}} &= \mathbf{U}[0]^{tr} \left(\textstyle\prod_{k=1}^{\mu} \mathbf{U}[k]^{id[k]tr}\right) H^{t\hat{r}} \\
&= U^{-\mathbf{y}[0]tr} g^{\mathbf{u}[0]tr} \left(\textstyle\prod_{k=1}^{\mu} U^{-\mathbf{y}[k]id[k]tr} g^{\mathbf{u}[k]id[k]tr}\right) H^{t\hat{r}} \\
&= U^{-f(\mathbf{y},id)tr} g^{f(\mathbf{u},id)tr} H^{t\hat{r}} \\
&= U^{-f(\mathbf{y},id)(r'-h\hat{r}'/f(\mathbf{y},id))} g^{f(\mathbf{u},id)(r'-h\hat{r}'/f(\mathbf{y},id))} H^{tu\hat{r}'} \\
&= \hat{g}^{-hu\hat{r}'} g^{-f(\mathbf{y},id)ur'} g^{f(\mathbf{u},id)r'} g^{-f(\mathbf{u},id)h\hat{r}'/f(\mathbf{y},id)} g^{htu\hat{r}'} \\
&= g^{-f(\mathbf{y},id)ur'} g^{f(\mathbf{u},id)r'} H^{-f(\mathbf{u},id)\hat{r}'/f(\mathbf{y},id)} \;=\; D_1 \\
\mathcal{H}(\hat{\mathbf{U}}, id)^{r} \hat{H}^{\hat{r}} &= \hat{\mathbf{U}}[0]^{r} \left(\textstyle\prod_{k=1}^{\mu} \hat{\mathbf{U}}[k]^{id[k]r}\right) \hat{H}^{\hat{r}} \;=\; \hat{g}^{\hat{u}[0]r} \left(\textstyle\prod_{k=1}^{\mu} \hat{g}^{\hat{u}[k]id[k]r}\right) \hat{H}^{\hat{r}} \\
&= \hat{g}^{f(\hat{u},id)r} \hat{H}^{\hat{r}} \;=\; g^{f(\hat{u},id)tr} \hat{H}^{\hat{r}} \\
&= g^{f(\hat{u},id)(r'-h\hat{r}'/f(\mathbf{y},id))} \hat{H}^{u\hat{r}'} \;=\; g^{f(\hat{u},id)r'} H^{-f(\mathbf{u},id)\hat{r}'/f(\mathbf{y},id)} \hat{H}^{u\hat{r}'} \;=\; D_2 \\
g^{-tr} &= g^{h\hat{r}'/f(\mathbf{y},id) - r'} \;=\; H^{\hat{r}'/f(\mathbf{y},id)} g^{-r'} \;=\; D_3 \\
g^{-t\hat{r}} &= g^{-tu\hat{r}'} \;=\; \hat{g}^{-u\hat{r}'} \;=\; D_4\,.
\end{aligned}$$

Finally adversary P outputs d'. Adversary B_1 also outputs d', so we have Equation (11).

On input $(g, \hat{g}, g^s, \hat{g}^{\hat{s}}, \hat{U}, T)$ where T is either $\hat{U}^{s+\hat{s}}$ or random, adversary B_2 runs adversary P, responding to its oracle queries as follows. When P makes query $\mathbf{Initialize}(\mathbf{y})$, adversary B_1 lets

$l \xleftarrow{\$} \{0, \ldots, \mu\}$; $\mathbf{u}, \hat{\mathbf{u}} \xleftarrow{\$} \mathbb{Z}_p^{\mu+1}$; $u, h, \hat{h} \xleftarrow{\$} \mathbb{Z}_p$; $H \leftarrow \hat{g}^h$; $\hat{H} \leftarrow \hat{g}^{\hat{h}}$; $U \leftarrow g^u$

For $k = 0, \ldots, \mu$ do $\mathbf{U}[k] \leftarrow \hat{U}^{\Delta(l,k)} g^{\mathbf{u}[k]}$; $\hat{\mathbf{U}}[k] \leftarrow \hat{U}^{\Delta(l,k)} \hat{g}^{\hat{\mathbf{u}}[k]}$

It returns $(g, \hat{g}, H, \hat{H}, \mathbf{U}, \hat{\mathbf{U}}, U)$ to P. When P makes its (single) $\mathbf{Ch}()$ query, adversary B_2 lets

$S \xleftarrow{\$} \mathbb{G}$

For $k = 0, \ldots, l-1$ do $\mathbf{Z}[k] \xleftarrow{\$} \mathbb{G}$

For $k = l, \ldots, \mu$ do $\mathbf{Z}[k] \leftarrow (g^s)^{u\mathbf{y}[k] + \mathbf{u}[k]} (\hat{g}^{\hat{s}})^{\hat{\mathbf{u}}[k]} T^{\Delta(l,k)}$

It returns $(g^s, \hat{g}^{\hat{s}}, S, \mathbf{Z})$ to P. Notice that for $l+1 \le k \le \mu$

$$\mathbf{Z}[k] = (g^s)^{u\mathbf{y}[k] + \mathbf{u}[k]} (\hat{g}^{\hat{s}})^{\hat{\mathbf{u}}[k]} = U^{s\mathbf{y}[k]} \mathbf{U}[k]^s \hat{\mathbf{U}}[k]^{\hat{s}} = (U^{\mathbf{y}[k]} \mathbf{U}[k])^s \hat{\mathbf{U}}[k]^{\hat{s}}\,.$$

If $T = \hat{U}^{s+\hat{s}}$ then

$$\mathbf{Z}[l] = (g^s)^{u\mathbf{y}[l] + \mathbf{u}[l]} (\hat{g}^{\hat{s}})^{\hat{\mathbf{u}}[l]} T = U^{s\mathbf{y}[l]} (\hat{U}^{-1} \mathbf{U}[l])^s (\hat{U}^{-1} \hat{\mathbf{U}}[l])^{\hat{s}} \hat{U}^s \hat{U}^{\hat{s}} = (U^{\mathbf{y}[l]} \mathbf{U}[l])^s \hat{\mathbf{U}}[l]^{\hat{s}}$$

as in game PC_l. On the other hand if T is random then so is $\mathbf{Z}[l]$, as in game PC_{l+1}. When P makes query $\mathbf{GetDK}(id)$, adversary B_2 does the following:

If $f(\mathbf{y}, id) = 0$ then $dk \leftarrow \perp$
Else
$r, \hat{r}' \xleftarrow{\$} \mathbb{Z}_p$
$D_1 \leftarrow \hat{g}^{f(\mathbf{u},id)r}\hat{g}^{h\hat{r}'}$; $D_2 \leftarrow g^{f(\mathbf{u},id)r}\hat{U}^{id[l]r}g^{\hat{h}\hat{r}'}\hat{U}^{-\hat{h}id[l]r/h}$
$D_3 \leftarrow \hat{g}^{-r}$; $D_4 \leftarrow \hat{U}^{id[l]r/h}g^{-\hat{r}'}$; $dk \leftarrow (D_1, D_2, D_3, D_4)$

It returns dk to P. We now show this key is properly distributed. Let $\hat{u}$ be such that $\hat{U} = g^{\hat{u}}$ and let

$$\hat{r} \;=\; \frac{\hat{r}'}{t} - \frac{id[l]\hat{u}r}{th} \bmod p\ .$$

Since t is non-zero modulo p and $\hat{r}'$ is random, $\hat{r}$ is random as well. The following computes the correct secret key components with the above randomness and shows that they are the ones of the simulation:

$$\begin{aligned}
\mathcal{H}(\mathbf{U}, id)^{tr}H^{t\hat{r}} &= \mathbf{U}[0]^{tr}\left(\textstyle\prod_{k=1}^{\mu}\mathbf{U}[k]^{id[k]tr}\right)H^{t\hat{r}}\\
&= g^{\mathbf{u}[0]tr}\left(\textstyle\prod_{k=1}^{\mu}\hat{U}^{id[k]tr\Delta(l,k)}g^{\mathbf{u}[k]id[k]tr}\right)\hat{g}^{ht\hat{r}}\\
&= g^{f(\mathbf{u},id)tr}\hat{U}^{id[l]tr}\hat{g}^{ht\hat{r}} \;=\; g^{f(\mathbf{u},id)tr}\hat{U}^{id[l]tr}\hat{g}^{h(\hat{r}'-id[l]\hat{u}r/h)}\\
&= \hat{g}^{f(\mathbf{u},id)r}\hat{U}^{id[l]tr}\hat{g}^{h\hat{r}'}\hat{g}^{-id[l]\hat{u}r} \;=\; \hat{g}^{f(\mathbf{u},id)r}g^{id[l]\hat{u}rt}\hat{g}^{h\hat{r}'}\hat{g}^{-id[l]\hat{u}r}\\
&= \hat{g}^{f(\mathbf{u},id)r}\hat{g}^{h\hat{r}'} \;=\; D_1\\
\mathcal{H}(\hat{\mathbf{U}}, id)^{r}\hat{H}^{\hat{r}} &= \hat{\mathbf{U}}[0]^{r}\left(\textstyle\prod_{k=1}^{\mu}\hat{\mathbf{U}}[k]^{id[k]r}\right)\hat{H}^{\hat{r}} \;=\; g^{\hat{\mathbf{u}}[0]r}\left(\textstyle\prod_{k=1}^{\mu}\hat{U}^{id[k]r\Delta(l,k)}g^{\hat{\mathbf{u}}[k]id[k]r}\right)\hat{g}^{\hat{h}\hat{r}}\\
&= g^{f(\hat{\mathbf{u}},id)r}\hat{U}^{id[l]r}g^{t\hat{h}\hat{r}} \;=\; g^{f(\hat{\mathbf{u}},id)r}\hat{U}^{id[l]r}g^{\hat{h}(\hat{r}'-id[l]\hat{u}r/h)}\\
&= g^{f(\hat{\mathbf{u}},id)r}\hat{U}^{id[l]r}g^{\hat{h}\hat{r}'}g^{-\hat{h}id[l]\hat{u}r/h} \;=\; g^{f(\mathbf{u},id)r}\hat{U}^{id[l]r}g^{\hat{h}\hat{r}'}\hat{U}^{-\hat{h}id[l]r/h} \;=\; D_2\\
g^{-tr} &= \hat{g}^{-r} \;=\; D_3\\
g^{-t\hat{r}} &= g^{\hat{u}rid[l]/h-\hat{r}'} \;=\; \hat{U}^{id[l]r/h}g^{-\hat{r}'} \;=\; D_4\ .
\end{aligned}$$

Finally adversary P outputs d'. Adversary B_2 also outputs d'. So

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{dlin}}(B_2) &= \frac{1}{\mu+1}\sum_{l=0}^{\mu}\Pr[\mathrm{PC}_l^P] - \Pr[\mathrm{PC}_{l+1}^P]\\
&= \frac{1}{\mu+1}\Pr[\mathrm{PC}_0^P] - \Pr[\mathrm{PC}_{\mu+1}^P]
\end{aligned}$$

and we have Equation (12).

4.8 Selective-id security

We consider IBTDF $\overline{\mathsf{F}}[n, 1, \mathbb{Z}_p]$, the instance of our construction with $\mu = 1$ and $\mathsf{IDSp} = \mathbb{Z}_p$. We show that this IBTDF is selective-id δ-lossy for $\delta = 1$, meaning fully selective-id lossy, and hence selective-id one-way. To do this we define a sibling $\overline{\mathsf{LF}}[n, 1, \mathbb{Z}_p]$. It preserves the key-generation, evaluation and inversion algorithms of $\overline{\mathsf{F}}[n, 1, \mathbb{Z}_p]$ and alters parameter generation to

Algorithm $\overline{\mathsf{LF}}[n, 1, \mathbb{Z}_p].\mathsf{Pg}(id)$
$\mathbf{y} \leftarrow (-id, 1)$; $(pars, msk) \xleftarrow{\$} \overline{\mathsf{E}}[n, 1, \mathbb{Z}_p].\mathsf{Pg}(\mathbf{y})$; Return $(pars, msk)$

The following says that our IBTDF is 1-lossy under the DLIN assumption with lossiness $\ell = n - 2\lg(p)$.

Theorem 4.3 *Let $n > 2\lg(p)$ and let $\ell = n - 2\lg(p)$. Let $\mathsf{F} = \overline{\mathsf{F}}[n, 1, \mathbb{Z}_p]$ be the IBTDF associated by our construction to parameters n, $\mu = 1$ and $\mathsf{IDSp} = \mathbb{Z}_p$. Let $\mathsf{LF} = \overline{\mathsf{LF}}[n, 1, \mathbb{Z}_p]$ be the sibling associated to it as above. Let $\delta = 1$ and let be A a selective-id adversary. Then there is an adversary B such that*

$$\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) \;\le\; 2n(\mu+2) \cdot \mathbf{Adv}^{\mathrm{dlin}}(B)\;. \tag{13}$$

The running time of B is that of A plus overhead.

Proof of Theorem 4.3: On input id, let algorithm Aux return $(-id, 1)$. Let $\mathrm{RL}_0, \mathrm{RL}_n$ be the games of Figure 5 with $\mu = 1$, $\mathsf{IDSp} = \mathbb{Z}_p$ and this Aux. Then we claim

$$\Pr\left[\,\mathrm{Real}^A_{\mathsf{F}}\,\right] = \Pr\left[\,\mathrm{RL}^A_0\,\right] \quad \text{and} \quad \Pr\left[\,\mathrm{Lossy}^A_{\mathsf{F},\mathsf{LF},\ell}\,\right] = \Pr\left[\,\mathrm{RL}^A_n\,\right]\;. \tag{14}$$

To justify this let id^* be the identity queried by A to both **Initialize** and **Ch**. (These queries are the same because A is selective-id.) Then $\mathbf{y}_0 = (-id^*, 1)$ so $f(\mathbf{y}_0, id) = id - id^*$. This is 0 iff $id = id^*$. This means that the conjunct $(id^* \notin IS) \wedge \textsc{Win}$ is always true. The claim of Equation (14) is now true because game RL_0 generates parameters with the real auxiliary input $\mathbf{y}_1 = (1, 0) \in \mathbb{Z}_p^2$ that, via $\overline{\mathsf{E}}[n, 1, \mathbb{Z}_p]$, defines F. However game RL_n generates parameters with auxiliary input $\mathbf{y}_0$. Since $f(\mathbf{y}_0, id^*) = 0$, the dependency of $\mathbf{C}_3[j]$ on $x[j]$ in Equation (5) vanishes when $id = id^*$. Examing equations (3), (4), (5), (6), we now see that with *pars* fixed, the values $\langle \mathbf{s}, x\rangle, \langle \hat{\mathbf{s}}, x\rangle$ determine the ciphertext $(C_1, C_2, \mathbf{C}_3, \mathbf{C}_4)$. Thus there are at most p^2 possible ciphertexts when $id = id^*$, and 2^n possible inputs. This means that $\lambda(\mathsf{F}.\mathsf{Ev}(pars, id^*, \cdot)) \ge n - \lg(p^2) = \ell$, which justifies the second claim of Equation (14). Recalling that $\delta = 1$, Equation (13) follows from Equation (1), Equation (14), Lemma 4.2 and Lemma 4.1. ▌

4.9 Adaptive-id Security

We consider IBTDF $\overline{\mathsf{F}}[n, \mu, \{0,1\}^\mu]$, the instance of our construction with $\mathsf{IDSp} = \{0,1\}^\mu \subset \mathbb{Z}_p^\mu$. We show that this IBTDF is adaptive-id δ-lossy for $\delta = (4(\mu+1)Q)^{-1}$ where Q is the number of key-derivation queries of the adversary. By Theorem 3.2 this means $\overline{\mathsf{F}}[n, \mu, \{0,1\}^\mu]$ is adaptive-id one-way. To do this we define a sibling $\overline{\mathsf{LF}}_Q[n, \mu, \{0,1\}^\mu]$. It preserves the key-generation, evaluation and inversion algorithms of $\overline{\mathsf{F}}[n, \mu, \{0,1\}^\mu]$ and alters parameter generation to $\overline{\mathsf{LF}}[n, \mu, \{0,1\}^\mu].\mathsf{Pg}(id)$ defined via

$$\mathbf{y} \leftarrow \mathsf{Aux}\,;\; (pars, msk) \xleftarrow{\$} \overline{\mathsf{E}}[n, \mu, \{0,1\}^\mu].\mathsf{Pg}(\mathbf{y})\,;\;\; \text{Return } (pars, msk)\;.$$

where algorithm Aux is defined via

$$\begin{array}{l}
\mathbf{y}'[0] \xleftarrow{\$} \{0, \ldots, 2Q-1\}\,;\; \ell \xleftarrow{\$} \{0, \ldots, \mu+1\}\,;\; \mathbf{y}[0] \leftarrow \mathbf{y}'[0] - 2\ell Q \\
\text{For } i = 1 \text{ to } \mu \text{ do } \mathbf{y}[i] \xleftarrow{\$} \{0, \ldots, 2Q-1\} \\
\text{Return } \mathbf{y} \in \mathbb{Z}_p^{\mu+1}
\end{array}$$

The following says that our IBTDF is δ-lossy under the DLIN assumption with lossiness $\ell = n - 2\lg(p)$.

Theorem 4.4 *Let $n > 2\lg(p)$ and let $\ell = n - 2\lg(p)$. Let $\mathsf{F} = \overline{\mathsf{F}}[n, \mu, \{0,1\}^\mu]$ be the IBTDF associated by our construction to parameters n, μ and $\mathsf{IDSp} = \{0,1\}^\mu$. Let A be an adaptive-id adversary that makes a maximal number of $Q < p/(3m)$ queries and let $\delta = (4(\mu+1)Q)^{-1}$. Let $\mathsf{LF} = \overline{\mathsf{LF}}_Q[n, \mu, \{0,1\}^\mu]$ be the sibling associated to F, A as above. Then there is an adversary B such that*

$$\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) \;\le\; 2n(\mu+2) \cdot \mathbf{Adv}^{\mathrm{dlin}}(B)\;. \tag{15}$$

The running time of B is that of A plus $O(\mu^2\rho^{-1}((\mu Q\rho)^{-1}))$ overhead, where $\rho = \frac{1}{2} \cdot \mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A)$.

Proof of Theorem 4.4: Our proof uses a simulation technique due to Waters [62]. We used a slightly improved analysis from [42]. Let Q be the number of queries made by A and let algorithm Aux be defined as above. Let $\mathrm{RL}_0, \mathrm{RL}_n$ be the games of Figure 5 with $\mathsf{IDSp} = \{0,1\}^\mu$ and this Aux. Let $\mathrm{E}(IS, id^*)$ denote the event that when $\underline{\mathbf{procFinalize}(d')}$ is called in RL^A_0 the flag $\textsc{Win} \leftarrow \mathsf{false}$ is set and $id^* \notin IS$. (Note that $\eta(IS, id^*)$ only depends on IS, id^* since $\mathbf{y}_0$ is exclusively used to set $\textsc{Win} \leftarrow \mathsf{false}$.) Let $\eta(IS, id^*)$ be

the probability that $\mathrm{E}(IS, id^*)$ happens. In [42, Lemma 6.2], it was shown (using purely combinatorial arguments) that $\lambda_{\mathrm{low}} := \frac{1}{4(\mu+1)Q} \leq \eta(IS, id^*) \leq \frac{1}{2Q} := \lambda_{\mathrm{up}}$. Since RL_0^A and $\mathrm{Real}_{\mathsf{F}}^A$ are only different when $\mathrm{E}(IS, id^*)$ happens, one would like to argue that $\lambda_{\mathrm{low}} \cdot \Pr\left[\mathrm{Real}_{\mathsf{F}}^A\right] = \Pr\left[\mathrm{RL}_0^A\right]$ but this is not true since $\mathrm{E}(IS, id^*)$ and $\mathrm{Real}_{\mathsf{F}}^A$ may not be independent. To get rid of this unwanted dependence we consider a modification of RL_0 and RL_n which adds some artificial abort such that in total it always sets $\textsc{Win} \leftarrow \mathsf{false}$ with probability around $1 - \lambda_{\mathrm{low}}$, independent of the view of the adversary. (Since, given IS, id^*, the exact value of $\eta(IS, id^*)$ cannot be computed efficiently, it needs to be approximated using sampling.) Concretely, games $\hat{\mathrm{RL}}_0$ and $\hat{\mathrm{RL}}_n$ are defined as RL_0 and RL_n, respectively, the only difference being **Finalize** which is defined as follows.

proc Finalize(d') // $\hat{\mathrm{RL}}_0, \hat{\mathrm{RL}}_n$
Compute an approximation $\eta'(IS, id^*)$ of $\eta(IS, id^*)$
If $\eta'(IS, id^*) > \lambda_{\mathrm{low}}$ then set $\textsc{Win} \leftarrow \mathsf{false}$ with probability $1 - \lambda_{\mathrm{low}}/\eta'(IS, id^*)$
Return $((d' = 1)$ and $(id^* \notin IS)$ and $\textsc{Win})$

We refer to [42] on details how to compute the approximation $\eta'(IS, id^*)$. Using [42, Lemma 6.3], one can show that if we use $O(\mu^2 \rho^{-1}((\mu Q \rho)^{-1}))$ samples to compute approximation $\eta'(IS, id^*)$, then

$$\Pr\left[\mathrm{Real}_{\mathsf{F}}^A\right] - \lambda_{\mathrm{low}}^{-1} \cdot \Pr\left[\hat{\mathrm{RL}}_0^A\right] = \rho. \tag{16}$$

Setting $\rho = \frac{1}{2} \cdot \Pr\left[\mathrm{Real}_{\mathsf{F}}^A\right]$ we obtain

$$\delta \cdot \Pr\left[\mathrm{Real}_{\mathsf{F}}^A\right] = \Pr\left[\hat{\mathrm{RL}}_0^A\right], \tag{17}$$

where $\delta = \lambda_{\mathrm{low}}/2$ is as in the theorem statement. As in the proof of Theorem 4.3, we can show that

$$\Pr\left[\mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}^A\right] = \Pr\left[\hat{\mathrm{RL}}_n^A\right]. \tag{18}$$

Now Equation (15) follows from Equations (1), (17), (18), Lemma 4.2 and (a version incorporating the artificial abort of) Lemma 4.1. ∎

We remark that we could use the proof technique of [12] which avoids the artificial abort but this increases the value of δ, making it dependent on the adversary advantage. The proof technique of [41] could be used to strengthen δ in Theorem 4.4 to $O(\sqrt{m}Q)^{-1}$ which is close to the optimal value Q^{-1}.

5 IB-TDFs from Lattices

Here we give a construction of a lossy IB-TDF from lattices, specifically, the LWE assumption. We note that a one-way IB-TDF can already be derived by applying methods from [29, 2] to the LWE-based injective (not identity-based) trapdoor function from [36].

LWE is a particular type of average-case BDD/GapSVP problem. It has been recognized since [50] that GapSVP (and BDD [45]) induces a form of lossiness. So there is folklore that the GPV LWE-based TDF can be made to satisfy some meaningful notion of lossiness (specifically, for an appropriate input distribution, the output does not reveal the entire input statistically) by replacing its normally uniformly random key with an LWE (BDD/GapSVP) instance. However, a full construction and proof according to the standard notion of lossiness (which compares the domain and images sizes of the function) have not yet appeared in the literature, and there are many quantitative issues to address.

In this section we construct an (ID-based) TDF that is lossy for a natural (uniform) input distribution. We favor simplicity of analysis at the expense of tight bounds, so our construction is highly unoptimized and should be seen mainly as a proof of feasibility. Much tighter constructions and bounds can be achieved using more sophisticated machinery from the literature.

5.1 Background

For a real matrix $\mathbf{X}$, we let $s_1(\mathbf{X})$ denote its largest singular value (also known as spectral norm), i.e., $s_1(\mathbf{X}) = \max_{\mathbf{y}\neq\mathbf{0}} \|\mathbf{X}\mathbf{y}\|/\|\mathbf{y}\|$. It is easy to verify that the spectral norm satisfies the triangle inequality $s_1(\mathbf{X}+\mathbf{Y}) \leq s_1(\mathbf{X}) + s_1(\mathbf{Y})$ and $s_1(\mathbf{X}\mathbf{Y}) \leq s_1(\mathbf{X})s_1(\mathbf{Y})$. Throughout this section we let n be the main security parameter, and let $\omega(\sqrt{\log n})$ denote a *fixed* function that grows asymptotically faster than $\sqrt{\log n}$.

Probability distributions. The *discrete Gaussian* distribution with parameter $s > 0$ over the integers $\mathbb{Z}$, written $D_{\mathbb{Z},s}$, assigns probability proportional to $\exp(-\pi x^2/s^2)$ to each $x \in \mathbb{Z}$ (and probability zero elsewhere). It is extended to a product distribution over $\mathbb{Z}^n$ in the natural way, i.e., $D_{\mathbb{Z}^n,s} = D^n_{\mathbb{Z},s}$.

We say that a random variable X over $\mathbb{R}$ is *subgaussian* with parameter s if for all $t \geq 0$, we have $\Pr[|X| \geq t] \leq 2\exp(-\pi t^2/s^2)$. More generally, we say that a random vector $\mathbf{x}$ (respectively, a random matrix $\mathbf{X}$) or its distribution is subgaussian of parameter s if all its one-dimensional marginals $\langle \mathbf{x}, \mathbf{u}\rangle$ (respectively, $\mathbf{u}^t\mathbf{X}\mathbf{v}$) for unit vectors $\mathbf{u}, \mathbf{v}$ are subgaussian of parameter s. The concatenation of n independent subgaussian variables with common parameter s, interpreted as either a vector or matrix, is also subgaussian with parameter s. It is also known that $D_{\mathbb{Z},s}$ is subgaussian with parameter s (see [46, Lemma 2.8]). We need the following standard fact from random matrix theory (see, e.g., [60]).

Lemma 5.1 *For a random matrix $\mathbf{X} \in \mathbb{R}^{h\times w}$ that is subgaussian with parameter s, we have $s_1(\mathbf{X}) = s \cdot O(\sqrt{h} + \sqrt{w})$ except with probability $2^{-\Omega(h+w)}$.*

Lattices and LWE. Throughout the remainder of this section we let $q = q(n)$ denote a prime, and $\mathbb{Z}_q$ denote the ring of integers modulo q. It is possible to generalize our constructions to moduli of other forms (e.g., prime powers) using known facts from the literature (see, e.g., [46]), but this somewhat complicates the constructions and the statements of the bounds we use, so we stick with prime moduli for simplicity.

As in many recent papers, we work with a family of "q-ary" lattices (and their cosets), represented by parity-check matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. The precise definition of these lattices will not be needed in this work, so we omit it and refer the interested reader to, e.g., [36] for details. The following lemma is special case of [36, Lemma 5.3] and [46, Lemma 2.4], and the properties of the "smoothing parameter" (see [47, 36]).

Lemma 5.2 *For prime q and integer $b \geq 2$, let $\bar{m} \geq n\log_b q + \omega(\log n)$. With overwhelming probability over the uniformly random choice of $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$, the following holds: for $\mathbf{r} \leftarrow D^{\bar{m}}_{\mathbb{Z},b\cdot\omega(\sqrt{\log n})}$, the distribution of $\mathbf{A}\mathbf{r} \in \mathbb{Z}_q^n$ is $negl(n)$-far from uniform.*

Note that by the triangle inequality for statistical distance, the above statement also holds where $\mathbf{r}$ is replaced by $\mathbf{R} \leftarrow D^{\bar{m}\times w}_{\mathbb{Z},b\cdot\omega(\sqrt{\log n})}$, and $\mathbf{A}\mathbf{r} \in \mathbb{Z}_q^n$ with $\mathbf{A}\mathbf{R} \in \mathbb{Z}_q^{n\times w}$, for any $w = \mathrm{poly}(n)$.

The (decisional) *learning with errors* (LWE) problem [54] in dimension n with error rate $\alpha \in (0,1)$, stated in matrix form, is: given an input $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$ (for any $m = \mathrm{poly}(n)$) where $\mathbf{A}$ is uniformly random, and $\mathbf{b}$ is either of the form $\mathbf{b}^t = \mathbf{x}^t \left[\begin{smallmatrix}\mathbf{I}_m \\ \mathbf{A}\end{smallmatrix}\right] \bmod q$ for $\mathbf{x} \leftarrow D^{m+n}_{\mathbb{Z},\alpha q}$, or is uniformly random and independent of $\mathbf{A}$, distinguish which is the case with non-negligible advantage.[1] By a routine hybrid argument, replacing $\mathbf{x}$ with a matrix $\mathbf{X}$ having any number $w = \mathrm{poly}(n)$ of independent columns (each drawn from $D^{m+n}_{\mathbb{Z},\alpha q}$), and replacing $\mathbf{b}^t$ with either $\mathbf{B}^t = \mathbf{X}^t \left[\begin{smallmatrix}\mathbf{I} \\ \mathbf{A}\end{smallmatrix}\right] \bmod q$ or a uniformly random $\mathbf{B}$ of the same dimension, yields an equivalent problem (up to a w factor in the adversary's advantage). When $\alpha q > 2\sqrt{n}$, this decision problem is at least as hard as approximating several problems on n-dimensional lattices in the *worst case* to within $\tilde{O}(n/\alpha)$ factors with a *quantum* algorithm [54], or via a *classical* algorithm for a subset of these problems [50].

[1] This is actually the "normal form" of the LWE problem, which is equivalent to the one from [54] in which the portion of $\mathbf{x}$ that is multiplied by $\mathbf{A}^t$ is uniformly random in $\mathbb{Z}_q^n$; see, e.g., [6]. In addition, for simplicity of analysis we use a true discrete Gaussian error distribution $D_{\mathbb{Z},\alpha q}$ instead of a "rounded" continuous Gaussian as in [54]; hardness for this error distribution is implied by the results of [51].

Trapdoors for lattices. We recall the notion and efficient construction of a (strong) trapdoor for q-ary lattices, due recently to Micciancio and Peikert [46]. This construction uses a public "gadget" matrix $\mathbf{G}$ over $\mathbb{Z}_q$, defined as

$$\mathbf{G} = \mathbf{I}_n \otimes [1, b, b^2, \ldots, b^{w-1}] \in \mathbb{Z}_q^{n \times nw} \tag{19}$$

for some integer base $b \geq 2$ and $w = \lceil \log_b q \rceil$. (Note that [46] mainly focuses on the case $b = 2$; in our constructions we will need to take b to be larger, but still constant.)

Following [46], we say that an integer matrix $\mathbf{R} \in \mathbb{Z}^{(m-nw) \times nw}$ is a *trapdoor* with tag $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$ for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ if $\mathbf{A} \left[\begin{smallmatrix} \mathbf{R} \\ \mathbf{I} \end{smallmatrix} \right] = \mathbf{H} \cdot \mathbf{G}$. In our constructions, $\mathbf{H}$ will always be either an invertible matrix, or the zero matrix. The trapdoor generation algorithm of [46] works for any $m \geq n(\log_b q + w) + \omega(\log n)$ and generates a nearly uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, together with a trapdoor $\mathbf{R}$ (with a desired tag $\mathbf{H}$) for $\mathbf{A}$. Letting $\bar{m} = m - nw \geq n \log_b q + \omega(\log n)$, it chooses $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times \bar{m}}$ uniformly at random, chooses $\mathbf{R} \leftarrow D_{\mathbb{Z}, b \cdot \omega(\sqrt{\log n})}^{\bar{m} \times nw}$, and lets $\mathbf{A} = [\bar{\mathbf{A}} \mid \mathbf{H} \cdot \mathbf{G} - \bar{\mathbf{A}}\mathbf{R}]$. It is clear by inspection that $\mathbf{R}$ is a trapdoor for $\mathbf{A}$, and by Lemma 5.2 the distribution of $\mathbf{A}$ is $negl(n)$-far from uniform.

We recall two of the main operations enabled by a trapdoor: inversion of the (injective) LWE function $g_{\mathbf{A}}(\mathbf{x}) := \mathbf{x}^t \left[\begin{smallmatrix} \mathbf{I} \\ \mathbf{A} \end{smallmatrix} \right] \bmod q$ for "short" integer vectors $\mathbf{x}$, and delegation of a trapdoor for an extended parity-check matrix.

Lemma 5.3 ([46]) *Let $\mathbf{R}$ be a trapdoor with any invertible tag $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$ for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, using a gadget matrix $\mathbf{G}$ with base $b \geq 2$. There are efficient algorithms* Invert *and* $\mathsf{DelTrap}$ *that do the following:*

1. *For $\mathbf{b}^t = g_{\mathbf{A}}(\mathbf{x}) := \mathbf{x}^t \left[\begin{smallmatrix} \mathbf{I}_m \\ \mathbf{A} \end{smallmatrix} \right] \bmod q$ where $\mathbf{x} \in \mathbb{Z}^{m+n}$ is such that $\|\mathbf{x}\| \leq q/\Theta(b \cdot s_1(\mathbf{R}))$, the algorithm $\mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{b})$ outputs $\mathbf{x}$.*
2. *For any invertible tag $\mathbf{H}'$, matrix $\mathbf{A}' \in \mathbb{Z}_q^{n \times nw}$, and any sufficiently large $s = \Omega(b \cdot s_1(\mathbf{R})) \cdot \omega(\sqrt{\log n})$, the algorithm $\mathsf{DelTrap}(\mathbf{R}, [\mathbf{A} \mid \mathbf{A}'], \mathbf{H}', s)$ outputs a trapdoor $\mathbf{R}'$ with tag $\mathbf{H}'$ for $[\mathbf{A} \mid \mathbf{A}']$, where $\mathbf{R}'$ has the same distribution (up to $negl(n)$ statistical distance) for* any *trapdoor $\mathbf{R}$ satisfying the above bound on $s_1(\mathbf{R})$, and $s_1(\mathbf{R}') = O(\sqrt{m})$ with overwhelming probability.*

5.2 Our basic trapdoor function

Let $c > 1$ and integer base $b \geq 2$ be constants to be determined later in the analysis, and let $\hat{n} = cn$, $m \geq \hat{n} \log_b q = cn \log_b q$ be integers. Define $I_\beta = \{0, 1, \ldots, \beta - 1\}$ and I_γ similarly for some positive integers $\beta \geq \gamma$ to be determined later. (The analysis also goes through unchanged for $I_\beta = [-\beta, \ldots, \beta - 1)$ and I_γ defined similarly.)

1. Parameters: The public parameter *pars* is a matrix $\mathbf{A} \in \mathbb{Z}_q^{\hat{n} \times m}$ (which will be close to uniform, either statistically or computationally), and the trapdoor *msk* is a trapdoor $\mathbf{R}$ (for any invertible tag $\mathbf{H}$) for $\mathbf{A}$ with bounded $s_1(\mathbf{R})$. For a sufficiently large $m = \Omega(\hat{n} \log_b q)$, these can be created using the trapdoor generation algorithm described above, or via the $\mathsf{DelTrap}$ algorithm from Item 2 of Lemma 5.3.
2. Evaluate: Given parameter $\mathbf{A}$ and input $\mathbf{x} \in I_\beta^{m+n} \times I_\gamma^{\hat{n}-n}$, algorithm $\mathsf{LWE.Ev}$ outputs

$$b^t = g_{\mathbf{A}}(\mathbf{x}) := \mathbf{x}^t \begin{bmatrix} \mathbf{I}_m \\ \mathbf{A}^t \end{bmatrix} \bmod q.$$

3. Invert: Given parameter $\mathbf{A}$, trapdoor $\mathbf{R}$ and output $\mathbf{b}$, algorithm $\mathsf{LWE.Ev}^{-1}$ returns $\mathbf{x}$ using the inversion algorithm from Item 1 of Lemma 5.3.

The next lemma shows that when $\mathbf{A}$ has a particular non-uniform structure (*without* a trapdoor $\mathbf{R}$), the function $g_{\mathbf{A}}$ is lossy when the parameters are set appropriately; we show how to do so after the proof.

Lemma 5.4 *Suppose that $\mathbf{A} \in \mathbb{Z}_q^{\hat{n} \times m}$ is such that*

$$\begin{bmatrix} \mathbf{I}_m \\ \mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{m+n} \\ \mathbf{E}^t \end{bmatrix} \begin{bmatrix} \mathbf{I}_m \\ \bar{\mathbf{A}} \end{bmatrix}$$

for some $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$ *and* $\mathbf{E}^t \in \mathbb{Z}^{(\hat{n}-n)\times(m+n)}$*. Then for* $\mathbf{x} \in I_\beta^{m+n} \times I_\gamma^{\hat{n}-n}$*, the number of distinct output values* $g_{\mathbf{A}}(\mathbf{x})$ *is at most* $O(\beta + \gamma \cdot s_1(\mathbf{E}))^{m+n}$*.*

In particular, for large enough $\gamma^{c-1} \geq 2^{\Omega(m/n)}$ *and* $\beta \geq \gamma \cdot s_1(\mathbf{E})$*, the function* $g_{\mathbf{A}}$ *is* $\Omega(m)$*-lossy.*

Proof: Notice that

$$g_{\mathbf{A}}(\mathbf{x}) = \mathbf{x}^t \begin{bmatrix} \mathbf{I}_m \\ \mathbf{A} \end{bmatrix} = (\mathbf{x}^t \begin{bmatrix} \mathbf{I}_{m+n} \\ \mathbf{E}^t \end{bmatrix}) \begin{bmatrix} \mathbf{I}_m \\ \bar{\mathbf{A}} \end{bmatrix} \bmod q.$$

It therefore suffices to bound the number of possible values of the form $\mathbf{x}^t \left[\begin{smallmatrix} \mathbf{I} \\ \mathbf{E}^t \end{smallmatrix}\right] \in \mathbb{Z}^{m+n}$. By the triangle inequality, we have

$$\left\| \mathbf{x}^t \left[\begin{smallmatrix} \mathbf{I} \\ \mathbf{E}^t \end{smallmatrix}\right] \right\| \leq \beta\sqrt{m+n} + s_1(\mathbf{E}) \cdot \gamma\sqrt{\hat{n}-n} \leq \sqrt{m+n} \cdot (\beta + \gamma \cdot s_1(\mathbf{E})).$$

Define $N_d(r)$ to be the number of integer points in a d-dimensional Euclidean ball of radius r. For $r \geq \sqrt{d}$, from the volume of the ball and Stirling's approximation, we have $N_d(r) = O(r/\sqrt{d})^d$. Therefore, the number of possible values of the form $\mathbf{x}^t \left[\begin{smallmatrix} \mathbf{I} \\ \mathbf{E}^t \end{smallmatrix}\right] \in \mathbb{Z}^{m+n}$ is $O(\beta + \gamma \cdot s_1(\mathbf{E}))^{m+n}$, as claimed.

For lossiness, observe that for our choice of γ, the base-2 logarithm of the domain size of $g_{\mathbf{A}}$ is

$$(m+n)\lg\beta + n\lg\gamma^{c-1} \geq (m+n)\lg\beta + \Omega(m).$$

Whereas by the above, for $\beta \geq \gamma \cdot s_1(\mathbf{E})$ the base-2 logarithm of the image size of $g_{\mathbf{A}}$ is at most

$$(m+n)\lg O(\beta + \gamma \cdot s_1(\mathbf{E})) = (m+n)\lg\beta + O(m).$$

By choosing a sufficiently large universal constant in the above $\Omega(\cdot)$ expression, we have that the two quantities above differ by $\Omega(m)$, as desired. ∎

We now discuss the constraints on the parameters and show how they can be instantiated. The constant c, base b, and integer γ are chosen based on the relationship between m and n. First, we need $\gamma^{c-1} \geq 2^{\Omega(m/n)}$ as required by Lemma 5.4. In order to generate $\mathbf{A}$ with a trapdoor, we will have $m = \Theta(\hat{n}\log_b q) = \Theta(cn\log_b q)$, so we need $\gamma \geq q^{\Theta(1/\log b)\cdot c/(c-1)}$. For any desired constant $C > 1$, we can choose constants $c > 1$ and $b \geq 2$ so that $\gamma \leq q^{1/C}$. Next, we choose β: to accommodate both the upper bound that suffices for invertibility (Item 1 of Lemma 5.3), and the lower bound on β that suffices for $\Omega(m)$-lossiness (Lemma 5.4), it suffices to take

$$q^{1/C} \cdot s_1(\mathbf{E}) \quad \leq \quad \beta \quad \leq \quad q/\Theta(s_1(\mathbf{R}) \cdot \sqrt{m}). \tag{20}$$

These constraints can be satisfied for sufficiently large

$$q^{1-1/C} \quad \geq \quad \Omega(s_1(\mathbf{R}) \cdot s_1(\mathbf{E}) \cdot \sqrt{m}). \tag{21}$$

In all our instantiations, we will have (with $1 - negl(n)$ probability) $s_1(\mathbf{R}) = \text{poly}(n)$ by the use of the trapdoor generation or delegation algorithms, and $s_1(\mathbf{E}) = \text{poly}(n)$ by the use of LWE with error distribution $D_{\mathbb{Z},\alpha q}$ for $\alpha q = \Theta(\sqrt{n})$ to generate a pseudorandom matrix $\mathbf{A}$. Because $1 - 1/C > 0$ is a constant (which may even be chosen arbitrarily close to 1), we can choose a sufficiently large $q = \text{poly}(n)$ so as to satisfy Equation (21), and can use an error rate of $\alpha = \Theta(\sqrt{n})/q = 1/\text{poly}(n)$.

Remark 5.5 As a concrete (but non-identity-based) instantiation, consider a matrix $\mathbf{A}$ having the form described in Lemma 5.4, where $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$ is uniformly random and the entries of $\mathbf{E}$ are chosen independently from $D_{\mathbb{Z},\alpha q}$, where $\alpha q = \Theta(\sqrt{n})$ so that we can invoke known worst-case hardness results for LWE. Then we have $s_1(\mathbf{R}) = O(\sqrt{m}) \cdot \omega(\sqrt{\log n}) = \tilde{O}(\sqrt{n})$ and $s_1(\mathbf{E}) = O(\sqrt{mn}) = \tilde{O}(n)$ with overwhelming probability, by subgaussianity of $D_{\mathbb{Z},\alpha q}$ and Lemma 5.1. Moreover, under the LWE assumption (in dimension n) with noise rate α, such an $\mathbf{A}$ is indistinguishable from uniform, which makes the lossy function $g_{\mathbf{A}}$ indistinguishable from an invertible one.

Remark 5.6 Our constructions of ID-based lossy TDFs below involve two small variations on the above example. First, the trapdoor $\mathbf{D}(id)$ for an identity will be delegated (using the $\mathsf{DelTrap}$ algorithm) from a trapdoor $\mathbf{R}(id)$, derived from the master trapdoor $\mathbf{R}$, for which $s_1(\mathbf{R}(id)) \leq \text{poly}(n)$. So we will still have $s_1(\mathbf{D}(id)) \leq s_1(\mathbf{R}(id)) \cdot \text{poly}(n) = \text{poly}(n)$. Second, in the lossy case, the hidden matrix $\mathbf{E}$ in the structured matrix $\mathbf{A}$ will no longer be Gaussian itself, but will be the product of some Gaussian $\mathbf{E}'$ (of parameter αq) and another matrix $\mathbf{X}$ with $s_1(\mathbf{X}) = \text{poly}(n)$, so we will still have $s_1(\mathbf{E}) = \text{poly}(n)$ and can still instantiate all the parameters so that $q, 1/\alpha = \text{poly}(n)$.

5.3 Our id-based lossy trapdoor function

SETUP. As above, let $c > 1$ and integer base $b \geq 2$ be constants to be determined later, and let $\hat{n} = cn$, $\bar{m} = \hat{n}\log_b q + \omega(\log n)$, and $m = \bar{m} + 2\hat{n}w$ where $w = \lceil \log_b q \rceil$. For integer $\mu \geq 1$, let $\mathsf{C} : \mathsf{IDSp} \to \mathbb{Z}_q^{\hat{n}\times\hat{n}} \times \{0,1\}^\mu$ denote an injective encoding of identities that will be instantiated for a specific scheme.

OUR E-IBTDF. Our E-IBTDF $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}]$ is associated with an integer $\mu \geq 1$, an identity space IDSp and an injective encoding C. It has domain $\mathsf{InSp} = I_\beta^{m+n} \times I_\gamma^{\hat{n}-n}$ and auxiliary input space $(\mathbb{Z}_q^{\hat{n}\times\hat{n}})^\mu$, and is given by the following algorithms.

1. Parameters: Given input $\mathbf{A} \in \mathbb{Z}_q^{\hat{n}\times\bar{m}}$ and auxiliary input $\mathbf{H} = (\mathbf{H}[1], \ldots, \mathbf{H}[\mu]) \in (\mathbb{Z}_q^{\hat{n}\times\hat{n}})^\mu$, algorithm $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Pg}$ chooses $\mathbf{R} = (\mathbf{R}[1], \ldots, \mathbf{R}[\mu]) \leftarrow (D^{\bar{m}\times\hat{n}w}_{\mathbb{Z}, b\cdot\omega(\sqrt{\log n})})^\mu$, and lets $\mathbf{U} = (\mathbf{U}[1], \ldots, \mathbf{U}[\mu]) \in (\mathbb{Z}_q^{\hat{n}\times\hat{n}w})^\mu$, where
$$\mathbf{U}[i] := \mathbf{H}[i] \cdot \mathbf{G} - \mathbf{A}\mathbf{R}[i].$$
It also chooses $\mathbf{R}' \leftarrow D^{\bar{m}\times\hat{n}w}_{\mathbb{Z}, b\cdot\omega(\sqrt{\log n})}$ and lets $\mathbf{A}' = \mathbf{A}\mathbf{R}'$. It returns $pars = (\mathbf{A}, \mathbf{A}', \mathbf{U})$ as the public parameters and $msk = (\mathbf{R}, \mathbf{H})$ as the master secret key.

   Note that $\mathbf{R}[i]$ is a trapdoor with tag $\mathbf{H}[i]$ for $[\mathbf{A} \mid \mathbf{U}[i]]$. Moreover, since each $\mathbf{R}[i]$ is subgaussian with parameter $b \cdot \omega(\sqrt{\log n})$, we have (by Lemma 5.1) $s_1(\mathbf{R}[i]) = O(b\sqrt{m}) \cdot \omega(\sqrt{\log n})$ for all i, with overwhelming probability.

   For $pars = (\mathbf{A}, \mathbf{A}', \mathbf{U})$ and a user identity id with $\mathsf{C}(id) = (\mathbf{H}[0], \mathbf{c} \in \{0,1\}^\mu)$, define
$$\mathbf{A}(id) := \Big[\mathbf{A} \mid \mathbf{H}[0] \cdot \mathbf{G} + \sum_{i=1}^{\mu} \mathbf{c}[i]\mathbf{U}[i]\Big].$$
   For $\mathbf{U}$ as constructed by $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Pg}$, we have
$$\mathbf{A}(id) = \Big[\mathbf{A} \mid (\mathbf{H}[0] + \sum_{i=1}^{\mu} \mathbf{c}[i]\mathbf{H}[i]) \cdot \mathbf{G} - \mathbf{A} \cdot \sum_{i=1}^{\mu} \mathbf{c}[i]\mathbf{R}[i]\Big]. \tag{22}$$
   Define
$$\mathbf{R}(id) := \sum_i \mathbf{c}[i]\mathbf{R}[i] \quad \text{and} \quad \mathbf{H}(id) := \mathbf{H}[0] + \sum_i \mathbf{c}[i]\mathbf{H}[i],$$
   and note that $\mathbf{R}(id)$ is a trapdoor with tag $\mathbf{H}(id)$ for $\mathbf{A}(id)$. Moreover, by the above bound on $s_1(\mathbf{R}[i])$ and the triangle inequality, we have $s_1(\mathbf{R}(id)) = O(\mu b\sqrt{m}) \cdot \omega(\sqrt{\log n}) = \text{poly}(n)$ for all id, with overwhelming probability. In what follows we assume that this bound holds.

2. Key generation: Given public parameters $pars = (\mathbf{A}, \mathbf{A}', \mathbf{U})$, master secret $(\mathbf{R}, \mathbf{H})$ and identity $id \in \mathsf{IDSp}$ with $\mathsf{C}(id) = (\mathbf{H}[0], \mathbf{c} \in \{0,1\}^\mu)$, algorithm $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Kg}$ proceeds as follows. It computes $\mathbf{A}(id)$, $\mathbf{R}(id)$, and $\mathbf{H}(id)$ as defined above. Define
$$\mathbf{A}'(id) := [\mathbf{A}(id) \mid \mathbf{A}'].$$
   If $\mathbf{H}(id)$ is invertible, it runs $\mathsf{DelTrap}(\mathbf{R}(id), \mathbf{A}'(id), \mathbf{H}' = \mathbf{I}, s)$ from Item 2 of Lemma 5.3 to generate a trapdoor $\mathbf{D}(id)$ with tag $\mathbf{I}$ for $\mathbf{A}'(id)$, for a sufficiently large $s = \Theta(\mu b^2\sqrt{m}) \cdot \omega(\sqrt{\log n})^2$.

proc Initialize(id) // RL_1

$\mathbf{H}_1 \xleftarrow{\$} \mathsf{Aux}_1(id)$; $\textsc{Win} \leftarrow \mathsf{true}$

$\bar{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{n\times\bar{m}}$; $\mathbf{E}^t \xleftarrow{\$} D_{\mathbb{Z},\alpha q}^{(\hat{n}-n)\times(\bar{m}+n)}$; $\left[\begin{smallmatrix}\mathbf{I}\\ \bar{\mathbf{A}}\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{E}^t\end{smallmatrix}\right]\left[\begin{smallmatrix}\mathbf{I}\\ \bar{\mathbf{A}}\end{smallmatrix}\right]$

$(pars, msk) \xleftarrow{\$} \overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Pg}(\mathbf{A}, \mathbf{H}_1)$

$IS \leftarrow \emptyset$; $id^* \leftarrow id$

Return $pars$

proc Initialize(id) // R_i $(i \in \{0,1\})$

$\mathbf{H}_0 \xleftarrow{\$} \mathsf{Aux}_0(id)$; $\mathbf{H}_1 \xleftarrow{\$} \mathsf{Aux}_1(id)$; $\textsc{Win} \leftarrow \mathsf{true}$

$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{\hat{n}\times\bar{m}}$

$(pars, msk) \xleftarrow{\$} \overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Pg}(\mathbf{A}, \mathbf{H}_i)$

$IS \leftarrow \emptyset$; $id^* \leftarrow id$

Return $pars$

proc GetDK(id) // $\mathrm{RL}_1, \mathrm{R}_0, \mathrm{R}_1$

$IS \leftarrow IS \cup \{id\}$

If either $\mathbf{H}_0(id)$ or $\mathbf{H}_1(id)$ is not invertible
then $\textsc{Win} \leftarrow \mathsf{false}$; $dk \leftarrow \perp$

Else $dk \leftarrow \overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Kg}(pars, msk, id)$

Return dk

proc Ch(id) // $\mathrm{RL}_1, \mathrm{R}_0, \mathrm{R}_1$

$id^* \leftarrow id$

If $\mathbf{H}_0(id) \neq \mathbf{0}_{\hat{n}\times\hat{n}}$ or $\mathbf{H}_1(id) \neq \mathbf{0}_{\hat{n}\times\hat{n}}$
then $\textsc{Win} \leftarrow \mathsf{false}$

proc Finalize(d') // $\mathrm{RL}_1, \mathrm{R}_0, \mathrm{R}_1$

Return $((d' = 1)$ and $(id^* \notin IS)$ and $\textsc{Win})$

Figure 7: Games RL_1 ("Real-to-Losssy") and $\mathrm{R}_0, \mathrm{R}_1$ associated to n, μ, IDSp and auxiliary input generator algorithms Aux_0 and Aux_1.

Note that $s = \Omega(b \cdot s_1(\mathbf{R}(id))) \cdot \omega(\sqrt{\log n})$ as required by Lemma 5.3, and that with overwhelming probability,

$$s_1(\mathbf{D}(id)) = s \cdot O(\sqrt{m}) = O(\mu b^2 m) \cdot \omega(\sqrt{\log n})^2 = \mathrm{poly}(n).$$

3. Evaluate: Given public parameters $pars = (\mathbf{A}, \mathbf{A}', \mathbf{U})$, identity $id \in \mathsf{IDSp}$ and input $\mathbf{x} \in I_\beta^{m+n} \times I_\gamma^{\hat{n}-n}$, algorithm $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Ev}$ computes $\mathbf{A}'(id) = [\mathbf{A}(id) \mid \mathbf{A}']$ as above, and outputs $\mathbf{y} = g_{\mathbf{A}'(id)}(\mathbf{x})$.
4. Invert: Given parameters $(\mathbf{A}, \mathbf{A}', \mathbf{U})$ and identity $id \in \mathsf{IDSp}$ determining $\mathbf{A}'(id)$ as above, trapdoor $\mathbf{D}_{id}$ (with tag $\mathbf{I}$) for $\mathbf{A}'(id)$, and value $\mathbf{y} = g_{\mathbf{A}'(id)}(\mathbf{x})$ as above, algorithm $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Ev}^{-1}$ returns $\mathbf{x}$ using the inversion algorithm from Item 1 of Lemma 5.3.

Key generation, invertibility, and lossiness. The choice of auxiliary input $\mathbf{H}$ determines the ability to generate keys for identities, i.e., the induced IBTDF $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}](\mathbf{H})$ can generate a key $\mathbf{D}_{id}$ for any id such that $\mathbf{H}(id)$ is invertible. By the upper bound on β from Equation (20), inversion is correct as long as $\beta \leq q/\Theta(s_1(\mathbf{D}_{id}) \cdot \sqrt{m})$.

By contrast, suppose that the $\mathbf{A} \in \mathbb{Z}_q^{\hat{n}\times\bar{m}}$ given to $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}].\mathsf{Pg}$ is such that $\left[\begin{smallmatrix}\mathbf{I}\\ \bar{\mathbf{A}}\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{E}^t\end{smallmatrix}\right]\left[\begin{smallmatrix}\mathbf{I}\\ \bar{\mathbf{A}}\end{smallmatrix}\right]$ for some $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$ and $\mathbf{E}^t = [\mathbf{E}_1^t \mid \mathbf{E}_2^t] \in \mathbb{Z}^{(\hat{n}-n)\times\bar{m}} \times \mathbb{Z}^{(\hat{n}-n)\times n}$. (I.e., $\mathbf{A}$ is a structured matrix that satisfies the hypothesis of Lemma 5.4.) Then if $\mathbf{H}(id) = \mathbf{0}$, it can be verified that $\mathbf{A}(id)$ is such that

$$\begin{bmatrix}\mathbf{I}_{\bar{m}+\hat{n}w}\\ \mathbf{A}(id)\end{bmatrix} = \begin{bmatrix}\mathbf{I}_{\bar{m}} & & \\ & \mathbf{I}_{\hat{n}w} & \\ & & \mathbf{I}_n \\ \mathbf{E}_1^t & -\mathbf{E}_1^t \cdot \mathbf{R}(id) & \mathbf{E}_2^t\end{bmatrix}\begin{bmatrix}\mathbf{I}_{\bar{m}} & \\ & \mathbf{I}_{\hat{n}w} \\ \bar{\mathbf{A}} & -\bar{\mathbf{A}} \cdot \mathbf{R}(id)\end{bmatrix},$$

which satisfies the hypothesis of Lemma 5.4 with $[\bar{\mathbf{A}} \mid -\mathbf{A}\cdot\mathbf{R}(id)]$ in place of $\bar{\mathbf{A}}$ and $\tilde{\mathbf{E}}^t = [\mathbf{E}_1^t \mid -\mathbf{E}_1^t\cdot\mathbf{R}(id) \mid \mathbf{E}_2^t]$ in place of $\mathbf{E}^t$. Observe that by the triangle inequality, $s_1(\tilde{\mathbf{E}}) \leq s_1(\mathbf{E})(1+s_1(\mathbf{R}(id))) \leq s_1(\mathbf{E})\,\mathrm{poly}(n)$. In particular, if we have a known $\mathrm{poly}(n)$ upper bound on $s_1(\mathbf{E})$, then as described in the analysis following the proof of Lemma 5.4, we can instantiate the parameters to have correct inversion when $\mathbf{H}(id)$ is invertible, and $\Omega(m)$-lossiness when $\mathbf{H}(id) = \mathbf{0}$.

In what follows we show security of the scheme in the selective-id and adaptive models, under the LWE assumption.

5.4 Real-to-lossy lemma

Consider game RL_1 which is defined as in Figure 7, where $\mathbf{A}$ is such that $\left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{A}\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{E}^t\end{smallmatrix}\right]\left[\begin{smallmatrix}\mathbf{I}\\ \bar{\mathbf{A}}\end{smallmatrix}\right]$ for uniformly random $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$ and $\mathbf{E}^t \leftarrow D_{\mathbb{Z},\alpha q}^{(\hat{n}-n)\times(\bar{m}+n)}$. Games R_0 and R_1 are defined similarly, where the distribution of $\mathbf{A}$ is uniformly random.

The following lemma says it is hard to distinguish game R_0 from RL_1. We will apply this by defining Aux_0 and Aux_1 in such a way that the output of Aux_0 results in the real scheme and the output of Aux_1 results in a lossy setup.

Lemma 5.7 *Let $n, \mu \geq 1$ be integers and IDSp. Let Aux_0 and Aux_1 be auxiliary input generators for $\overline{\mathsf{L}}[\mu, \mathsf{IDSp}, \mathsf{C}]$ and A an adversary. Then there is an adversary B such that*

$$\Pr[\mathrm{R}_0^A] - \Pr[\mathrm{RL}_1^A] \;\leq\; \mathbf{Adv}^{\mathrm{lwe}}_{n,\alpha}(B) + \mathit{negl}(n)\ . \tag{23}$$

The running time of B is that of A plus some overhead. If A is selective-id then so is B.

The last statement allows us to use the lemma in both the selective-id and adaptive-id cases.

Proof: By Remark 5.5 we have that

$$\Pr[\mathrm{R}_1^A] - \Pr[\mathrm{RL}_1^A] \;\leq\; \mathbf{Adv}^{\mathrm{lwe}}_{n,\alpha}(B)\ . \tag{24}$$

We claim that in R_0 and R_1 (where $\mathbf{A}$ is uniformly random) the values $\mathbf{H}_0$ and $\mathbf{H}_1$ are statistically hidden from A's view. By Lemma 5.2, the tuple $(\mathbf{A}, \mathbf{AR}[1], \ldots, \mathbf{AR}[\mu])$ is $\mathit{negl}(n)$-far from uniformly random. Hence the public parameters $(\bar{\mathbf{A}}, \mathbf{A}', \mathbf{U})$ are $\mathit{negl}(n)$-far from uniform for any fixed choice of the auxiliary input $\mathbf{H}$. Since the execution of the remaining game is independent of whether $\mathbf{H}$ comes from Aux_0 or Aux_1, we obtain

$$\Pr[\mathrm{R}_0^A] - \Pr[\mathrm{R}_1^A] \;\leq\; \mathit{negl}(n)\ . \tag{25}$$

which concludes the proof. ∎

5.5 Selective-id Security

We consider IBTDF $\overline{\mathsf{L}}[\mu = 1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}]$, the instance of our construction with identity space $\mathsf{IDSp} = \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}$, uniformly random input $\mathbf{A} \in \mathbb{Z}_q^{\hat{n}\times\bar{m}}$, auxiliary input $\mathbf{H}_0 = \mathbf{H}_0[1] = -\mathsf{C}_{\mathrm{FRD}}(\mathbf{0}) \in \mathbb{Z}_q^{\hat{n}\times\hat{n}}$, and identity encoding $\mathsf{C}'_{\mathrm{FRD}}(id) = (\mathsf{C}_{\mathrm{FRD}}(id), 1) \in \mathbb{Z}_q^{\hat{n}\times\hat{n}} \times \{0,1\}$, where $\mathsf{C}_{\mathrm{FRD}} : \mathbb{Z}_q^{\hat{n}} \to \mathbb{Z}_q^{\hat{n}\times\hat{n}}$ is an "invertible differences" encoding as constructed in [2]. (I.e., for each $\mathbf{x} \neq \mathbf{x}'$, the matrix $\mathsf{C}_{\mathrm{FRD}}(\mathbf{x}) - \mathsf{C}_{\mathrm{FRD}}(\mathbf{x}')$ is invertible over $\mathbb{Z}_q$.)

Note that our scheme satisfies the correct inversion requirement because $\mathbf{H}_0(id) = \mathsf{C}_{\mathrm{FRD}}(id) - \mathsf{C}_{\mathrm{FRD}}(\mathbf{0})$ is invertible for all $id \in \mathsf{IDSp} = \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}$. We show that this IBTDF is selective-id δ-lossy for $\delta = 1$, meaning fully selective-id lossy, and hence selective-id one-way. To do this we define a sibling $\overline{\mathsf{LF}}[\mu = 1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}]$. It preserves the key-generation, evaluation and inversion algorithms of $\overline{\mathsf{L}}[1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}]$ and alters parameter generation to

Algorithm $\overline{\mathsf{LF}}[1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}].\mathsf{Pg}(id)$:
$\bar{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{n\times\bar{m}}$; $\mathbf{E}^t \xleftarrow{\$} D_{\mathbb{Z},\alpha q}^{(\hat{n}-n)\times(\bar{m}+n)}$; $\left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{A}\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{E}^t\end{smallmatrix}\right]\left[\begin{smallmatrix}\mathbf{I}\\ \bar{\mathbf{A}}\end{smallmatrix}\right]$
$\mathbf{H}_1[1] = -\mathsf{C}_{\mathrm{FRD}}(id)$; $(pars, msk) \xleftarrow{\$} \overline{\mathsf{L}}[1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}].\mathsf{Pg}(\mathbf{A}, \mathbf{H_1})$; Return $(pars, msk)$.

The following says that our IBTDF is 1-lossy with lossiness $\Omega(m)$, under the LWE assumption.

Theorem 5.8 *Let $m = c_2 n > c_1 n = \hat{n}$ and $\ell = 2m$. Let $\mathsf{L} = \overline{\mathsf{L}}[1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}]$ be the IBTDF associated by our construction to parameters $\mu = 1$ and $\mathsf{IDSp} = \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}$. Let $\mathsf{LF} = \overline{\mathsf{LF}}[1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}]$ be the sibling associated to it as above. Let $\delta = 1$ and let be A a selective-id adversary. Then there is an adversary B such that*

$$\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{L},\mathsf{LF},\ell}(A) \;\leq\; \mathbf{Adv}^{\mathrm{lwe}}_{n,\alpha}(B) + \mathit{negl}\ . \tag{26}$$

The running time of B is that of A plus overhead.

Proof: On input id, let algorithm Aux_0 return $-\mathsf{C}_{\mathrm{FRD}}(\mathbf{0})$ and algorithm Aux_1 return $-\mathsf{C}_{\mathrm{FRD}}(id)$. Let $\mathrm{R}_0, \mathrm{RL}_1$ be the games of Figure 7 with $\mu = 1$, $\mathsf{IDSp} = \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}$ and auxiliary input generators Aux_0 and Aux_1, respectively. Then we claim

$$\Pr\left[\,\mathrm{Real}_{\mathsf{L}}^{A}\,\right] = \Pr\left[\,\mathrm{R}_0^{A}\,\right] \quad \text{and} \quad \Pr\left[\,\mathrm{Lossy}_{\mathsf{L},\mathsf{LF},\ell}^{A}\,\right] = \Pr\left[\,\mathrm{RL}_1^{A}\,\right]. \tag{27}$$

To justify this let id^* be the identity queried by A to both **Initialize** and **Ch**. (These queries are the same because A is selective-id.) Then $\mathbf{H}_1 = -\mathsf{C}_{\mathrm{FRD}}(id^*)$ so $\mathbf{H}_1(id) = \mathsf{C}_{\mathrm{FRD}}(id) - \mathsf{C}_{\mathrm{FRD}}(id^*)$. Since $\mathsf{C}_{\mathrm{FRD}}$ is an encoding with invertible differences, this is invertible iff $id \neq id^*$. This means that the conjunct $(id^* \notin IS) \wedge \textsc{Win}$ is always true. The claim of Equation (27) is now true because game R_0 generates parameters with uniform $\mathbf{A}$ and auxiliary input $\mathbf{H}_0 = -\mathsf{C}_{\mathrm{FRD}}(\mathbf{0}) \in \mathbb{Z}_q^{\hat{n}\times\hat{n}}$ that, via $\overline{\mathsf{L}}[1, \mathbb{Z}_q^{\hat{n}} \setminus \{\mathbf{0}\}, \mathsf{C}'_{\mathrm{FRD}}]$, defines L. However game RL_1 generates parameters with auxiliary input $\mathbf{H}_1$. Since $\mathbf{H}_1(id^*) = \mathbf{0}$, the function $g_{\mathbf{A}'(id)}$ is $\Omega(m)$-lossy, as argued immediately following the description of the scheme. ∎

5.6 Full Security

We consider IBTDF $\overline{\mathsf{L}}[\mu, \{0,1\}^\mu, \mathsf{C}']$, the instance of our construction with $\mathsf{IDSp} = \{0,1\}^\mu$, uniformly random input $\mathbf{A} \in \mathbb{Z}_q^{\hat{n}\times\bar{m}}$, auxiliary input $\mathbf{H}_0 = (\mathbf{H}_0[1], \ldots, \mathbf{H}_0[\mu]) := (\mathbf{0}_{\hat{n}\times\hat{n}}, \ldots, \mathbf{0}_{\hat{n}\times\hat{n}})$ and $\mathsf{C}'(id) = (\mathbf{1}_{\hat{n}\times\hat{n}}, \mathsf{C}_f(id))$, where $\mathsf{C}_f : \{0,1\}^\mu \to \mathbb{Z}_q^{\hat{n}\times\hat{n}}$ maps $\mathbf{x} \in \{0,1\}^\mu$ into a vector $\mathbf{X}$ of matrices such that $\mathbf{X}[i] = (-1)^{\mathbf{x}[i]} \cdot \mathbf{1}_{\hat{n}\times\hat{n}} \in \mathbb{Z}_q^{\hat{n}\times\hat{n}}$.

Note that our scheme satisfies the correct inversion requirement because $\mathbf{H}_0(id) = \mathbf{1}_{\hat{n}\times\hat{n}}$ is invertible for all $id \in \mathsf{IDSp}$. We show that this IBTDF is adaptive-id δ-lossy for $\delta = (8Q)^{-1}$ where Q is the number of key-derivation queries of the adversary. By Theorem 3.2 this means $\overline{\mathsf{L}}[\mu, \{0,1\}^\mu, \mathsf{C}']$ is adaptive-id one-way. To do this we define a sibling $\overline{\mathsf{LF}}_Q[\mu, \{0,1\}^\mu, \mathsf{C}']$. It preserves the key-generation, evaluation and inversion algorithms of $\overline{\mathsf{L}}[\mu, \{0,1\}^\mu, \mathsf{C}_f]$ and alters parameter generation to

$$\begin{array}{l}\text{Algorithm } \overline{\mathsf{LF}}_Q[\mu, \{0,1\}^\mu, \mathsf{C}'].\mathsf{Pg}(id) : \\ \bar{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{n\times\bar{m}} \,;\, \mathbf{E}^t \xleftarrow{\$} D_{\mathbb{Z},\alpha q}^{(\hat{n}-n)\times(\bar{m}+n)} \,;\, \left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{A}\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{E}^t\end{smallmatrix}\right]\left[\begin{smallmatrix}\mathbf{I}\\ \bar{\mathbf{A}}\end{smallmatrix}\right] \\ \mathbf{H}_1 \xleftarrow{\$} \mathsf{Aux}_1 \,;\, (pars, msk) \xleftarrow{\$} \overline{\mathsf{L}}[\mu, \{0,1\}^\mu, \mathsf{C}'].\mathsf{Pg}(\mathbf{A}, \mathbf{H}_1) \,;\, \text{ Return } (pars, msk)\,.\end{array}$$

where Aux_1 is a randomized algorithm from [2, 22] that generates $\mathbf{H}_1 \in (\mathbb{Z}_q^{\hat{n}\times\hat{n}})^\mu$ such that the image of $\mathbf{H}_1(\cdot)$ is either $\mathbf{0}_{\hat{n}\times\hat{n}}$ or invertible and $\mathbf{H}_1(\cdot)$ is "pairwise independent", i.e, for all $id \neq id'$, $\Pr_{\mathsf{Aux}_1}[\mathbf{H}_1(id) = \mathbf{0}_{\hat{n}\times\hat{n}} \mid \mathbf{H}_1(id') = \mathbf{0}_{\hat{n}\times\hat{n}}] = 1/(2Q)$. The following says that our IBTDF is δ-lossy under the LWE assumption with lossiness $\ell = 2m$.

Theorem 5.9 *Let $m = c_2 n > c_1 n = \hat{n}$ and $\ell = 2m$. Let $\mathsf{L} = \overline{\mathsf{L}}[\mu, \{0,1\}^\mu, \mathsf{C}']$ be the IBTDF associated by our construction to parameters μ and $\mathsf{IDSp} = \{0,1\}^\mu$. Let A be an adaptive-id adversary that makes a maximal number of Q queries and let $\delta = (8Q)^{-1}$. Let $\mathsf{LF} = \overline{\mathsf{LF}}_Q[\mu, \{0,1\}^\mu, \mathsf{C}']$ be the sibling associated to L as above. Then there is an adversary B such that*

$$\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{L},\mathsf{LF},\ell}(A) \;\leq\; \mathbf{Adv}^{\mathrm{lwe}}_{n,\alpha}(B) + negl(n)\,. \tag{28}$$

The running time of B is that of A plus polynomial overhead.

Proof: (Sketch) Let Q be the number of queries made by A and let algorithm Aux be defined as above. Let $\mathrm{R}_0, \mathrm{RL}_1$ be the games of Figure 7 with $\mathsf{IDSp} = \{0,1\}^\mu$ and this Aux_0 and Aux_1. Let $\mathrm{E}(IS, id^*)$ denote the event that when $\mathbf{Finalize}(d')$ is called in R_0^A the flag $\textsc{Win} \leftarrow \mathsf{false}$ is set and $id^* \notin IS$. (Note that $\eta(IS, id^*)$ only depends on IS, id^*.) Let $\eta(IS, id^*)$ be the probability that $\mathrm{E}(IS, id^*)$ happens. In [2], it was shown that $\lambda_{\mathrm{low}} := \frac{1}{4Q} \leq \eta(IS, id^*) \leq \frac{1}{2Q} := \lambda_{\mathrm{up}}$. Since R_0^A and $\mathrm{Real}_{\mathsf{L}}^A$ are only different when $\mathrm{E}(IS, id^*)$ happens, one would like to argue that $\lambda_{\mathrm{low}} \cdot \Pr\left[\,\mathrm{Real}_{\mathsf{L}}^{A}\,\right] = \Pr\left[\,\mathrm{R}_0^{A}\,\right]$ but this is not true since $\mathrm{E}(IS, id^*)$ and $\mathrm{Real}_{\mathsf{L}}^A$ may not be independent. To get rid of this unwanted dependence we consider a modification of R_0 and RL_1 which adds some artificial abort such that in total it always sets $\textsc{Win} \leftarrow \mathsf{false}$

with probability around $1 - \lambda_{\text{low}}$, independent of the view of the adversary. (Since, given IS, id^*, the exact value of $\eta(IS, id^*)$ cannot be computed efficiently, it needs to be approximated using sampling.) Concretely, games $\hat{\text{R}}_0$ and $\hat{\text{RL}}_1$ are defined as R_0 and RL_1, respectively, the only difference being **Finalize** which is defined as follows.

proc Finalize(d') // $\hat{\text{R}}_0, \hat{\text{RL}}_1$
Compute an approximation $\eta'(IS, id^*)$ of $\eta(IS, id^*)$
If $\eta'(IS, id^*) > \lambda_{\text{low}}$ then set $\text{WIN} \leftarrow \mathsf{false}$ with probability $1 - \lambda_{\text{low}}/\eta'(IS, id^*)$
Return $((d' = 1)$ and $(id^* \notin IS)$ and $\text{WIN})$

One can again show that with a polynomial number of samples to compute approximation $\eta'(IS, id^*)$,

$$\delta \cdot \Pr\left[\,\text{Real}_{\mathsf{L}}^{A}\,\right] = \Pr\left[\,\hat{\text{R}}_0^{A}\,\right], \tag{29}$$

where $\delta = \lambda_{\text{low}}/2$ is as in the theorem statement. Similar to the proof of Theorem 5.8, we can show that

$$\Pr\left[\,\text{Lossy}_{\mathsf{L},\mathsf{LF},\ell}^{A}\,\right] = \Pr\left[\,\hat{\text{RL}}_1^{A}\,\right]. \tag{30}$$

Now Equation (28) follows from Equation (1), Equation (29), Equation (30) and Lemma 5.7. ∎

Acknowledgments

We thank Xiang Xie and the (anonymous) reviewers of Eurocrypt 2012 for their careful reading and valuable comments.

References

[1] P. Abeni, L. Bello, and M. Bertacchini. Exploiting DSA-1571: How to break PFS in SSL with EDH, July 2008. `http://www.lucianobello.com.ar/exploiting_DSA-1571/index.html`. 4

[2] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, May 2010. 2, 3, 18, 24, 25

[3] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, Aug. 2010. 2

[4] M. Ajtai. Generating hard instances of the short basis problem. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *ICALP 99*, volume 1644 of *LNCS*, pages 1–9. Springer, July 2099. 3

[5] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theory of Computing Systems*, 48(3):535–553, April 2011. Preliminary version in STACS 2009. 3

[6] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Aug. 2009. 19

[7] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, Aug. 2007. 1, 4, 31

[8] 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 M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, Dec. 2009. 1, 4, 31

[9] M. Bellare, S. Halevi, A. Sahai, and S. P. Vadhan. Many-to-one trapdoor functions and their relation to public-key cryptosystems. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 283–298. Springer, Aug. 1998. 3, 31

[10] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Apr. 2009. 1

[11] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. *Journal of Cryptology*, 22(1):1–61, Jan. 2009. 1

[12] M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, Apr. 2009. 18

[13] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, May 1994. 1

[14] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. 5

[15] C. Bennet, G. Brassard, C. Crépeau, and U. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6), 1995. 4, 31

[16] A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer, Aug. 2008. 1, 4, 31

[17] D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, May 2004. 1, 2, 3, 8

[18] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Aug. 2004. 1, 2

[19] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Aug. 2004. 2

[20] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, May 2004. 4, 31

[21] D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. 1, 2

[22] X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 499–517. Springer, May 2010. 25

[23] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 290–307. Springer, Aug. 2006. 2, 3, 8

[24] X. Boyen and B. Waters. Shrinking the keys of discrete-log-type lossy trapdoor functions. In J. Zhou and M. Yung, editors, *ACNS 10*, volume 6123 of *LNCS*, pages 35–52. Springer, June 2010. 1

[25] D. R. Brown. A weak randomizer attack on RSA-OAEP with e=3. IACR ePrint Archive, Report 2005/189, 2005. `http://eprint.iacr.org/`. 4

[26] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 402–414. Springer, May 1999. 1

[27] R. Canetti and R. R. Dakdouk. Towards a theory of extractable functions. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 595–613. Springer, Mar. 2009. 4

[28] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, May 2003. 2

[29] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, May 2010. 2, 3, 18

[30] C. Cocks. An identity based encryption scheme based on quadratic residues. In B. Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, Dec. 2001. 1

[31] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. 1

[32] Y. Dodis, J. Katz, S. Xu, and M. Yung. Strong key-insulated signature schemes. In Y. Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 130–144. Springer, Jan. 2003. 1

[33] L. Dorrendorf, Z. Gutterman, and B. Pinkas. Cryptanalysis of the windows random number generator. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 476–485. ACM Press, Oct. 2007. 4

[34] A. Escala, J. Herranz, B. Libert, and C. Ráfols. Hierarchical identity-based (lossy) trapdoor functions, May 2012. Manuscript. 4

[35] D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 279–295. Springer, May 2010. 1

[36] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 1, 2, 3, 18, 19

[37] I. Goldberg and D. Wagner. Randomness in the Netscape browser. Dr. Dobb's Journal, January 1996. 4

[38] Z. Gutterman and D. Malkhi. Hold your sessions: An attack on Java session-id generation. In A. Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 44–57. Springer, Feb. 2005. 4

[39] 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. 4, 31

[40] B. Hemenway and R. Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. Electronic Colloquium on Computational Complexity TR09-127, 2009. 1

[41] D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 21–38. Springer, Aug. 2008. 18

[42] E. Kiltz and D. Galindo. Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. *Theor. Comput. Sci.*, 410(47-49):5093–5111, 2009. 17, 18

[43] E. Kiltz, P. Mohassel, and A. O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, May 2010. 4

[44] E. Kiltz, A. O'Neill, and A. Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 295–313. Springer, Aug. 2010. 1

[45] V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 577–594. Springer, Aug. 2009. 2, 18

[46] D. Micciancio and C. Peikert. Trapdoors for lattices: simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, LNCS. Springer, 2012. 3, 19, 20

[47] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, Oct. 2004. 19

[48] M. Mueller. Debian OpenSSL predictable PRNG bruteforce SSH exploit, May 2008. `http://milw0rm.com/exploits/5622`. 4

[49] K. Ouafi and S. Vaudenay. Smashing SQUASH-0. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 300–312. Springer, Apr. 2009. 4

[50] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009. 2, 18, 19

[51] C. Peikert. An efficient and parallel gaussian sampler for lattices. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, Aug. 2010. 19

[52] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008. 1, 2, 3, 4, 6, 8

[53] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. 2

[54] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005. 19

[55] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978. 1

[56] P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, May / June 2006. 1

[57] A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, Mar. 2009. 4

[58] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *SCIS 2000*, Okinawa, Japan, Jan. 2000. 1, 2

[59] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Aug. 1985. 1

[60] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices, January 2011. Available at `http://www-personal.umich.edu/~romanv/papers/non-asymptotic-rmt-plain.pdf`, last accessed 4 Feb 2011. 19

[61] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Aug. 2009. 2

[62] B. R. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, May 2005. 1, 2, 3, 17

[63] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage. When private keys are public: Results from the 2008 Debian OpenSSL vulnerability. In *IMC 2009*. ACM, 2009. 4

A Anonymous IBE

In this section we describe an IBE scheme that is similar to IBE from Section 4 with the difference that it encrypts group elements (rather than bits) and it is slightly more efficient. We associate to any integer $\mu \geq 1$ and any identity space $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$ an IBE scheme $\mathsf{IBE}'[\mu, \mathsf{IDSp}]$ that has message space $\mathbb{G}_T^*$ and algorithms as follows:

1. Parameters: Algorithm $\mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Pg}$ lets $g \xleftarrow{\$} \mathbb{G}^*$; $t, z \xleftarrow{\$} \mathbb{Z}_p^*$; $\hat{g} \leftarrow g^t$; $Z \leftarrow \mathbf{e}(g,g)^z$. It then lets $H, \hat{H}, U \xleftarrow{\$} \mathbb{G}$; $\mathbf{U} \xleftarrow{\$} \mathbb{G}^{\mu+1}$. It returns $pars = (g, \hat{g}, H, U, \hat{H}, \mathbf{U}, Z)$ as the public parameters and $msk = (t, z)$ as the master secret key.

2. Key generation: Given parameters $(g, \hat{g}, H, U, \mathbf{U}, Z)$, master secret (t, z) and identity $id \in \mathsf{IDSp}$, algorithm $\mathsf{IBE}'[\mu, \mathsf{IDSp}].\mathsf{Kg}$ returns decryption key (D_1, D_2, D_3, D_4) computed by letting $r, \hat{r} \xleftarrow{\$} \mathbb{Z}_p$ and setting

$$D_1 \leftarrow g^z \cdot \mathcal{H}(\mathbf{U}, id)^{tr} \cdot H^{t\hat{r}} \; ; \; D_2 \leftarrow U^r \cdot H^{\hat{r}} \; ; \; D_3 \leftarrow g^{-tr} \; ; \; D_4 \leftarrow g^{-t\hat{r}} \; .$$

3. Encryption: Given parameters $(g, \hat{g}, H, U, \mathbf{U}, Z)$, identity $id \in \mathsf{IDSp}$ and message $M \in \mathbb{G}_T^*$, algorithm $\mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Enc}$ returns ciphertext $(C_1, C_2, C_3, C_4, C_5)$ computed as follows. It lets $s, \hat{s} \xleftarrow{\$} \mathbb{Z}_p$ and

$$C_1 \leftarrow g^s \; ; \; C_2 \leftarrow \hat{g}^{\hat{s}} \; ; \; C_3 \leftarrow \mathcal{H}(\mathbf{U}, id)^s \cdot U^{\hat{s}} \; ; \; C_4 \leftarrow H^{s+\hat{s}} \; ; \; C_5 \leftarrow Z^{-s} \cdot M \; .$$

4. Decryption: Given parameters $(g, \hat{g}, H, U, \mathbf{U}, Z)$, identity $id \in \mathsf{IDSp}$, decryption key (D_1, D_2, D_4, D_4) for id and ciphertext $(C_1, C_2, C_3, C_4, C_5)$, algorithm $\mathsf{IBE}[\mu, \mathsf{IDSp}].\mathsf{Dec}$ returns

$$M = \mathbf{e}(C_1, D_1)\mathbf{e}(C_2, D_2)\mathbf{e}(C_3, D_3)\mathbf{e}(C_4, D_4)C_5 \; .$$

Compared to $\mathsf{IBE}[\mu, \mathsf{IDSp}]$ from Section 4 , the efficiency improvement consists of replacing $\mathcal{H}(\hat{\mathbf{U}}, id)$ by U in the computation of D_2 and C_3 and of setting $\hat{H} := H$. Using the techniques of the ciphertext pseudorandomness lemma (Lemma 4.1) one can show that the elements (C_1, C_2, C_3, C_4) of the ciphertext are pseudorandom. (Here the reduction knows the secret z.) In a final similar hybrid step one can also show that, under the Bilinear Diffie-Hellman assumption (which is implied by the DLIN assumption), the element C_5 is also pseudorandom. (Here is reduction knows the secret t.) As our main ID-based TDF result uses anonymous IBE techniques, the main ideas of this systems security is implicit in our main proof. A formal proof of the above stand alone system is deferred to the full version.

B Applications

We expand first on the application to achieving deterministic IBE and then on achieving hedged IBE.

D-PKE. Deterministic PKE (D-PKE) cannot achieve IND-CPA security. Bellare, Boldyreva and O'Neill [7] defined a target notion PRIV for it that captures the best possible security under the condition that encryption is deterministic. D-PKE provides a way to do fast (logarithmic time) search on encrypted data. PEKS [20] offers higher security but takes linear time, and trading some security for a significant increase in searching speed is attractive for large databases.

Achieving PRIV for D-PKE has been (and remains) a challenge. It is possible in the RO model [7]. The best results without ROs are due to Boldyreva, Fehr and O'Neill [16], who show how to achieve PRIV without random oracles for message sequences which are blocksources, meaning each message has some min-entropy even given the previous ones. Using the Leftover Hash Lemma (LHL) [15, 39], they show that any LTDF is a D-PKE scheme that is PRIV-secure for blocksources as long as the lossy branch is a universal hash function.

D-IBE. We introduce deterministic IBE (D-IBE). The PRIV definition is easily extended to this setting. D-IBE offers, over D-PKE, the same advantages that IBE offers over PKE, for example that there are no certificates and encryption depends only on the identity of the receiver. Again, D-IBE can be achieved in the RO model by setting the coins of an IBE scheme to the RO-hash of the message. (This is how PKE is turned into D-PKE in the RO model in [9, 7].) We ask what can be done without ROs.

We show that our constructions of DLIN-based lossy IB-TDFs have the properties necessary to obtain PRIV-secure D-IBE schemes for blocksources under the paradigm of [16] in the selective case. We start by observing that the lossy branches are universal hash functions. This can be seen from Equations (3), (4), (5) and (6). In the lossy case, $f(\mathbf{y}, id) = 0$, and the function has a range R of size p^2. Now if x_1, x_2 are distinct inputs, then the outputs of the function on them collide exactly when $(\langle \mathbf{s}, x_1 \rangle, \langle \hat{\mathbf{s}}, x_1 \rangle) = (\langle \mathbf{s}, x_2 \rangle, \langle \hat{\mathbf{s}}, x_2 \rangle)$. The probability that this happens when $\mathbf{s}, \hat{\mathbf{s}}$ are chosen at random from $\mathbb{Z}_p^n$ is $1/p^2 = 1/|R|$.

Hedged IBE. The definitions and methods of [8] can be extended to the identity-based setting in a straightforward way in the selective setting once we have universal lossy IB-TDFs. There are two approaches. One is generic composition of an IBE scheme with a IB-TDF. The other is to first pad the message with randomness and then apply the IB-TDF.

Adaptive setting. It remains open to achieve deterministic or hedged IBE in the adaptive security setting.