# Identity-Based (Lossy) Trapdoor Functions and Applications

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

[1] Department of Computer Science & Engineering,
University of California San Diego, USA
http://cseweb.ucsd.edu/~mihir/

[2] Horst Görtz Institut für IT-Sicherheit, Ruhr-Universität Bochum, Germany
http://www.cits.rub.de/personen/kiltz.html

[3] School of Computer Science, College of Computing,
Georgia Institute of Technology, USA
http://www.cc.gatech.edu/~cpeikert/

[4] Department of Computer Science, University of Texas at Austin, USA
http://userweb.cs.utexas.edu/~bwaters/

**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 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 [49].

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 [47] call such a TDF lossy. Intuitively, $F_{pk}$ is close to a function $F_{pk^*}$ that provides information-theoretic security. Lossiness implies one-wayness [47].

Lossy TDFs have quickly proven to be a powerful tool. Applications include IND-CCA [47], deterministic [16], hedged [7] and selective-opening secure public-key encryption [9]. Lossy TDFs can be constructed from DDH [47], QR [33], DLIN [33], DBDH [23], LWE [47] and HPS (hash proof systems) [38]. RSA was shown in [41] to be lossy under the $\Phi$-hiding assumption of [25], leading to the first proof of security of RSA-OAEP [13] without random oracles.

D. Pointcheval and T. Johansson (Eds.): EUROCRYPT 2012, LNCS 7237, pp. 228–245, 2012.

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 [30,49]. Public-key encryption was built from TDFs. (Via hardcore bits.) Lossy TDFs enabled the first DDH and lattice (LWE) based TDFs [47].

It is striking that identity-based cryptography developed entirely differently. The first realizations of IBE [21,29,52] 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 [53] 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 identity+date as the key under which to encrypt to identity [21]. There is thus good motivation to go beyond basics like IBE [21,29,52,17,18,55,34] and identity-based signatures [11,31] 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 [6,16], 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 [50,7], 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.

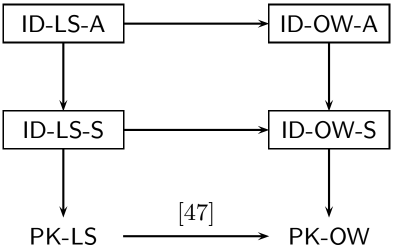
The first wave of IBE schemes was from pairings [21,52,17,18,55,54] but another is now emerging from lattices [34,28,2,3]. We aim accordingly to reach our ends with either route and do so successfully. We provide lossy IB-TDFs from a standard pairings assumption, namely the Decision Linear (DLIN) assumption of [19]. We also provide IB-TDFs based on Learning with Errors (LWE) [48], whose hardness follows from the worst-case hardness of certain lattice-related problems [48,46]. (The same assumption underlies lattice-based IBE [34,28,2,3] and public-key lossy TDFs [47].) 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 [34,28,2]. Based on techniques from [46,42] 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 [47] 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 [22]. 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*) [27] 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 $\mathsf{Pg}^*$ 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, $\mathsf{Pg}^*$ is forced to make $F_{pars^*,id}$ lossy for all $id$, something the adversary can immediately detect using its key-derivation oracle.



| **Primitive** | $\delta$ | **Achieved under** |
|---|---|---|
| ID-LS-A | 1/ poly | DLIN, LWE |
| ID-LS-S | 1 | DLIN, LWE |

**Fig. 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 → 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 $\delta$ for which we prove $\delta$-lossiness and the assumptions used. In both the S and A settings the $\delta$ we achieve is best possible and suffices for applications.

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 *$\delta$-lossiness*, a metric of partial lossiness parameterized by the probability $\delta$ 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 $1/\text{poly}$ 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 $\delta$ implies one-wayness, in both the selective and adaptive settings, as long as $\delta$ is at least $1/\text{poly}$.

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 Fig. 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 → ID-LS-S meaning that $\delta$-lossiness of the first type implies $\delta$-lossiness of the other for all $\delta$. 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 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 [8], 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 [47] (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) [34] showed that the function $g_{\mathbf{A}}\colon B_{\alpha}^{n+m} \to \mathbb{Z}_q^n$ 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 [4,5,43]. We make this function identity-based using the trapdoor extension and delegation methods introduced by Cash, Hofheinz, Kiltz and Peikert [28], and improved in efficiency by Agrawal, Boneh and Boyen [2] and Micciancio and Peikert [43]. 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 [47] 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) [22], 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 $\delta$-lossiness with $\delta$ around $1/Q$.

This sounds similar to the approach of Waters [55] 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 [6]. 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 [15,37]) 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 [35,36,24,45,44,1,56,32]. Expecting systems to do a better job is unrealistic. Hedged encryption [7] 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 [7] 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. It remains open to 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 [47] there is Rosen and Segev's notion of correlated-product security [51], and Canetti and Dakdouk's extractable trapdoor functions [26]. The notion of adaptive one-wayness for tag-based trapdoor functions from Kiltz, Mohassel and O'Neill [40] 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 $\delta$-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 [10] we show that $\delta$-lossiness implies one-wayness as long as $\delta$ is at least $1/\text{poly}$. This allows us to focus on achieving $\delta$-lossiness. In Section 3 we provide our pairing-based schemes and in [10] our lattice-based schemes. In [10] we sketch how to apply $\delta$-lossy IB-TDFs to achieve deterministic and hedged IBE.

## 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

```
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($id$)   // $\mathrm{OW}_{\mathsf{F}}$
$id^* \leftarrow id$ ; $x \xleftarrow{\$} \mathsf{InSp}$
$y \leftarrow \mathsf{F.Ev}(pars, id^*, x)$
Return $y$

proc Finalize($x'$)   // $\mathrm{OW}_{\mathsf{F}}$
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($d'$)   // $\mathrm{Lossy}_{\mathsf{F},\mathsf{LF},\ell}$
If $(\lambda(\mathsf{F.Ev}(pars, id^*, \cdot)) < \ell)$ then return false
Return $((d' = 1)$ and $(id^* \notin IS)$
```

**Fig. 2.** Games defining one-wayness and $\delta$-lossiness of IBTDF $\mathsf{F}$ with sibling $\mathsf{LF}$

$1, \ldots, |\mathbf{x}|$ 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 empty string is denoted $\varepsilon$. 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 $\mu$ 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 $\Delta$ 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 Fig. 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 $\mathrm{G}^A$, is called the output of the game, and we let "$\mathrm{G}^A$" 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 $\mathsf{InSp}$ and identity space $\mathsf{IDSp}$. The parameter generation algorithm $\mathsf{F.Pg}$ takes no input and returns common parameters *pars* and a master secret key *msk*. On input $\mathit{pars}, \mathit{msk}, \mathit{id}$, the key generation algorithm $\mathsf{F.Kg}$ produces a decryption key *dk* for identity *id*. For any *pars* and $\mathit{id} \in \mathsf{IDSp}$, the *deterministic* evaluation algorithm $\mathsf{F.Ev}$ defines a function $\mathsf{F.Ev}(\mathit{pars}, \mathit{id}, \cdot)$ with domain $\mathsf{InSp}$. We require *correct inversion*: For any *pars*, any $\mathit{id} \in \mathsf{IDSp}$ and any $\mathit{dk} \in [\mathsf{F.Kg}(\mathit{pars}, \mathit{id})]$, the deterministic inversion algorithm $\mathsf{F.Ev}^{-1}$ defines a function that is the inverse of $\mathsf{F.Ev}(\mathit{pars}, \mathit{id}, \cdot)$, meaning $\mathsf{F.Ev}^{-1}(\mathit{pars}, \mathit{id}, \mathit{dk}, \mathsf{F.Ev}(\mathit{pars}, \mathit{id}, x)) = x$ 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 $\mathsf{AxSp}$. Fixing a particular auxiliary input $\mathit{aux} \in \mathsf{AxSp}$ for $\mathsf{F.Pg}$ results in an IBTDF scheme that we denote $\mathsf{E}(\mathit{aux})$ 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 $\mathrm{OW}_{\mathsf{F}}$ of Fig. 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 $\mathrm{OW}_{\mathsf{F}}$ of Fig. 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 Fig. 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 $\lambda(f) = \lg(|\mathrm{Dom}(f)|/|\mathrm{Im}(f)|)$ or equivalently $|\mathrm{Im}(f)| = |\mathrm{Dom}(f)| \cdot 2^{-\lambda(f)}$. We say that $f$ is $\ell$-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 $\mathsf{InSp}$ and identity space $\mathsf{IDSp}$. A *sibling* for $\mathsf{F}$ 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 $\mathsf{F}$ and whose auxiliary input space is $\mathsf{IDSp}$. 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 Fig. 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 $\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) = \delta \cdot \Pr[\mathrm{Real}^A_{\mathsf{F}}] - \Pr[\mathrm{Lossy}^A_{\mathsf{F},\mathsf{LF},\ell}]$ of the adversary is *not*, as usual, the difference in the probabilities that the games return true, but is instead parameterized by a probability $\delta \in [0, 1]$.

Discussion. The PW [47] 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 $\mathsf{F.Ev}(pars, id^*, \cdot)$ 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 $\delta$. Roughly, it means that a $\delta$ fraction of identities are lossy. The advantage represents the computational loss while $\delta$ 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 $\mathsf{InSp}$ and identity space $\mathsf{IDSp}$. 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 $\epsilon$ 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 $\mathsf{IBE}$ is deterministic if $\mathsf{IBE.Enc}$ is deterministic. A deterministic IBE scheme is identical to an IBTDF.

## 3 IB-TDFs from Pairings

In [10] we show that $\delta$-lossiness implies one-wayness in both the selective and adaptive cases. We now show how to achieve $\delta$-lossiness using pairings.

Setup. Throughout we fix a bilinear map $\mathbf{e}\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 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 $\mathsf{true}$ if $b = b'$ and $\mathsf{false}$ 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 $\overline{id} = (1, id[1], \ldots, id[\mu]) \in \mathbb{Z}_p^{\mu+1}$ and $\mathcal{H}(\mathbf{U}, id) = \prod_{k=0}^{\mu} \mathbf{U}[k]^{\overline{id}[k]}$. $\mathcal{H}$ is the BB hash function [17] when $\mu = 1$, and the Waters' one [22] when $\mathsf{IDSp} = \{0,1\}^{\mu}$ and an $id \in \mathsf{IDSp}$ is viewed as a $\mu$-vector over $\mathbb{Z}_p$. We also let $f(\mathbf{y}, id) = \sum_{k=0}^{\mu} \mathbf{y}[k]\overline{id}[k]$ and $\overline{f}(\mathbf{y}, id) = f(\mathbf{y}, id) \bmod p$.

Overview. In the Peikert-Waters [47] 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 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 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.

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 [10] 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.

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 $1 \leq j \leq n$

$$C_1 \leftarrow \textstyle\prod_{i=1}^{n} \mathbf{G}[i]^{x[i]} \;;\; C_2 \leftarrow \prod_{i=1}^{n} \hat{\mathbf{G}}[i]^{x[i]}$$
$$\mathbf{C}_3[j] \leftarrow \textstyle\prod_{i=1}^{n} \prod_{k=0}^{\mu} \mathbf{W}[i,j,k]^{x[i]\overline{id}[k]} \;;\; \mathbf{C}_4[j] \leftarrow \prod_{i=1}^{n} \mathbf{J}[i,j]^{x[i]}$$

**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 $1 \leq j \leq n$ 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 $1 \leq j \leq n$

$$C_1 = \textstyle\prod_{i=1}^{n} g^{\mathbf{s}[i]x[i]} = g^{\langle \mathbf{s}, x \rangle} \tag{1}$$

$$C_2 = \textstyle\prod_{i=1}^{n} \hat{g}^{\hat{\mathbf{s}}[i]x[i]} = \hat{g}^{\langle \hat{\mathbf{s}}, x \rangle} \tag{2}$$

$$\begin{aligned} \mathbf{C}_3[j] &= \textstyle\prod_{i=1}^{n} \prod_{k=0}^{\mu} \mathbf{V}[j,k]^{\mathbf{s}[i]x[i]\overline{id}[k]} \hat{\mathbf{V}}[j,k]^{\hat{\mathbf{s}}[i]x[i]\overline{id}[k]} U^{\mathbf{s}[i]x[i]\mathbf{y}[k]\overline{id}[k]\varDelta(i,j)} \\ &= \textstyle\prod_{i=1}^{n} \mathcal{H}(\mathbf{V}[j,\cdot], id)^{\mathbf{s}[i]x[i]} \mathcal{H}(\hat{\mathbf{V}}[j,\cdot], id)^{\hat{\mathbf{s}}[i]x[i]} U^{\mathbf{s}[i]x[i]f(\mathbf{y}, id)\varDelta(i,j)} \\ &= \mathcal{H}(\mathbf{V}[j,\cdot], id)^{\langle \mathbf{s}, x \rangle} \mathcal{H}(\hat{\mathbf{V}}[j,\cdot], id)^{\langle \hat{\mathbf{s}}, x \rangle} U^{\mathbf{s}[j]x[j]f(\mathbf{y}, id)} \end{aligned} \tag{3}$$

$$\mathbf{C}_4[j] = \textstyle\prod_{i=1}^{n} \mathbf{H}[j]^{\mathbf{s}[i]x[i]} \hat{\mathbf{H}}[j]^{\hat{\mathbf{s}}[i]x[i]} = \mathbf{H}[j]^{\langle \mathbf{s}, x \rangle} \hat{\mathbf{H}}[j]^{\langle \hat{\mathbf{s}}, x \rangle} \,. \tag{4}$$

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 $n, \mu \geq 1$ 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 $\delta$-lossy for appropriate $\delta$. First we prove two technical lemmas that we will use in both cases.

```
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 $(d' = 1)$
```

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

Ciphertext pseudorandomness lemma. Consider games ReC, RaC of Fig. 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 [10].

**Lemma 1.** *Let $\mu \geq 1$ be an integer and $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$. Let $P$ be an adversary. Then there is an adversary $B$ such that* $\Pr\left[\mathrm{ReC}^P\right] - \Pr\left[\mathrm{RaC}^P\right] \leq (\mu+2) \cdot \mathbf{Adv}^{\mathrm{dlin}}(B)$. *The running time of $B$ is that of $P$ plus some overhead.*

Real-to-lossy lemma. Consider games $\mathrm{RL}_0, \mathrm{RL}_n$ of Fig. 4 associated to some choice of $n, \mu, \mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$ and auxiliary input generator $\mathsf{Aux}$ for $\overline{\mathsf{E}}[n, \mu, \mathsf{IDSp}]$. The latter is an algorithm that takes input an identity in $\mathsf{IDSp}$ and returns an auxiliary input in $\mathbb{Z}_p^{\mu+1}$. Game $\mathrm{RL}_0$ obtains an auxiliary input $\mathbf{y}_0$ via $\mathsf{Aux}$ 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 $\mathsf{true}$ under the same condition as game Real but additionally requiring that $f(\mathbf{y}_0, id) \neq 0$ for all **GetDK**($id$) queries and $f(\mathbf{y}_0, id) = 0$ for the **Ch**($id$) query. Game $\mathrm{RL}_n$ generates parameters with the auxiliary input provided by $\mathsf{Aux}$ but is otherwise identical to game $\mathrm{RL}_0$. The following lemma says it is hard to distinguish these games. We will apply this by defining $\mathsf{Aux}$ in such a way that its output $\mathbf{y}_0$ results in a lossy setup. The proof of the following is in [10].

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

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

**Fig. 4.** 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 $\mathsf{Aux}$

**Lemma 2.** *Let $n, \mu \geq 1$ be integers and $\mathsf{IDSp} \subseteq \mathbb{Z}_p^\mu$. Let $\mathsf{Aux}$ 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 $\Pr[\mathrm{RL}_0^A] - \Pr[\mathrm{RL}_n^A] \leq 2n \cdot \left(\Pr\left[\,\mathrm{ReC}^P\,\right] - \Pr\left[\,\mathrm{RaC}^P\,\right]\right)$. 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.

Selective-id security. We show that IBTDF $\overline{\mathsf{F}}[n, 1, \mathbb{Z}_p]$ is selective-id $\delta$-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 $\ell = n - 2\lg(p)$. The proof is in [10].

**Theorem 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 $\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) \leq 2n(\mu + 2) \cdot \mathbf{Adv}^{\mathrm{dlin}}(B)$. The running time of $B$ is that of $A$ plus overhead.*

Adaptive-id Security. We show that IBTDF $\overline{\mathsf{F}}[n, \mu, \{0,1\}^\mu]$ is adaptive-id $\delta$-lossy for $\delta = (4(\mu+1)Q)^{-1}$ where $Q$ is the number of key-derivation queries of the adversary. By [10] 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 $\mathsf{Aux}$ is defined via

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

The following says that our IBTDF is $\delta$-lossy under the DLIN assumption with lossiness $\ell = n - 2\lg(p)$. The proof is in [10].

**Theorem 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 $n$, $\mu$ 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 $\mathsf{F}, A$ as above. Then there is an adversary $B$ such that $\mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A) \leq 2n(\mu+2) \cdot \mathbf{Adv}^{\mathrm{dlin}}(B)$. The running time of $B$ is that of $A$ plus $O(\mu^2\rho^{-1}((\mu Q\rho)^{-1}))$ overhead, where $\rho = \frac{1}{2} \cdot \mathbf{Adv}^{\delta\text{-los}}_{\mathsf{F},\mathsf{LF},\ell}(A)$.*

We remark that we could use the proof technique of [12] which avoids the artificial abort but this increases the value of $\delta$, making it dependent on the adversary advantage. The proof technique of [39] could be used to strengthen $\delta$ in Theorem 4 to $O(\sqrt{m}Q)^{-1}$ which is close to the optimal value $Q^{-1}$.

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

Bellare was supported in part by NSF CNS-0627779 and CCF-0915675. Kiltz was supported by a Sofja Kovalevskaja Award of the Alexander von Humboldt Foundation, funded by the German Federal Ministry for Education and Research. Waters was supported in part by NSF CNS-0915361 and CNS-0952692, AFOSR FA9550-08-1-0352, DARPA PROCEED, DARPA N11AP20006, a Google Faculty Research award, the Alfred P. Sloan Fellowship, a Microsoft Faculty Fellowship, and a Packard Foundation Fellowship.

## References

1. Abeni, P., Bello, L., Bertacchini, M.: Exploiting DSA-1571: How to break PFS in SSL with EDH (July 2008), `http://www.lucianobello.com.ar/exploiting_DSA-1571/index.html`
2. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
3. Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
4. Ajtai, M.: Generating Hard Instances of the Short Basis Problem. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999)
5. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory of Computing Systems 48(3), 535–553 (2009); Preliminary version in STACS 2009

6. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and Efficiently Searchable Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
7. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged Public-Key Encryption: How to Protect against Bad Randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (2009)
8. Bellare, M., Halevi, S., Sahai, A., Vadhan, S.P.: Many-to-One Trapdoor Functions and Their Relation to Public-Key Cryptosystems. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 283–298. Springer, Heidelberg (1998)
9. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
10. Bellare, M., Kiltz, E., Peikert, C., Waters, B.: Identity-based (lossy) trapdoor functions and applications. IACR ePrint Archive, Report 2011/479, Full version of this abstract (2011), `http://eprint.iacr.org/`
11. Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. Journal of Cryptology 22(1), 1–61 (2009)
12. Bellare, M., Ristenpart, T.: Simulation without the Artificial Abort: Simplified Proof and Improved Concrete Security for Waters' IBE Scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (2009)
13. Bellare, M., Rogaway, P.: Optimal Asymmetric Encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
14. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
15. Bennet, C., Brassard, G., Crépeau, C., Maurer, U.: Generalized privacy amplification. IEEE Transactions on Information Theory 41(6) (1995)
16. Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
17. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
18. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
19. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
20. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
21. Boneh, D., Franklin, M.K.: Identity based encryption from the Weil pairing. SIAM Journal on Computing 32(3), 586–615 (2003)
22. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
23. Boyen, X., Waters, B.: Shrinking the Keys of Discrete-Log-Type Lossy Trapdoor Functions. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 35–52. Springer, Heidelberg (2010)

24. Brown, D.R.: A weak randomizer attack on RSA-OAEP with e=3. IACR ePrint Archive, Report 2005/189 (2005), http://eprint.iacr.org/
25. Cachin, C., Micali, S., Stadler, M.A.: Computationally Private Information Retrieval with Polylogarithmic Communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
26. Canetti, R., Dakdouk, R.R.: Towards a Theory of Extractable Functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 595–613. Springer, Heidelberg (2009)
27. Canetti, R., Halevi, S., Katz, J.: A Forward-Secure Public-Key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
28. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
29. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
30. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
31. Dodis, Y., Katz, J., Xu, S., Yung, M.: Strong Key-Insulated Signature Schemes. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 130–144. Springer, Heidelberg (2002)
32. Dorrendorf, L., Gutterman, Z., Pinkas, B.: Cryptanalysis of the windows random number generator. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM CCS 2007, pp. 476–485. ACM Press (October 2007)
33. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)
34. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC, pp. 197–206. ACM Press (May 2008)
35. Goldberg, I., Wagner, D.: Randomness in the Netscape browser. Dr. Dobb's Journal (January 1996)
36. Gutterman, Z., Malkhi, D.: Hold Your Sessions: An Attack on Java Session-Id Generation. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 44–57. Springer, Heidelberg (2005)
37. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999)
38. Hemenway, B., Ostrovsky, R.: Lossy trapdoor functions from smooth homomorphic hash proof systems. Electronic Colloquium on Computational Complexity TR09-127 (2009)
39. Hofheinz, D., Kiltz, E.: Programmable Hash Functions and Their Applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
40. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive Trapdoor Functions and Chosen-Ciphertext Security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
41. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under Chosen-Plaintext Attack. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 295–313. Springer, Heidelberg (2010)

42. Lyubashevsky, V., Micciancio, D.: On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 577–594. Springer, Heidelberg (2009)
43. Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
44. Mueller, M.: Debian OpenSSL predictable PRNG bruteforce SSH exploit (May 2008), `http://milw0rm.com/exploits/5622`
45. Ouafi, K., Vaudenay, S.: Smashing SQUASH-0. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 300–312. Springer, Heidelberg (2009)
46. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 333–342. ACM Press (May/June 2009)
47. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC, pp. 187–196. ACM Press (May 2008)
48. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press (May 2005)
49. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signature and public-key cryptosystems. Communications of the Association for Computing Machinery 21(2), 120–126 (1978)
50. Rogaway, P., Shrimpton, T.: A Provable-Security Treatment of the Key-Wrap Problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006)
51. Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
52. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: SCIS 2000, Okinawa, Japan (January 2000)
53. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
54. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
55. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
56. Yilek, S., Rescorla, E., Shacham, H., Enright, B., Savage, S.: When private keys are public: Results from the 2008 Debian OpenSSL vulnerability. In: IMC 2009. ACM (2009)