Cumulatively All-Lossy-But-One Trapdoor Functions from Standard Assumptions

Benoît Libert, Ky Nguyen, Alain Passelègue

Cumulatively All-Lossy-But-One Trapdoor Functions from Standard Assumptions

Benoît Libert[1,2], Ky Nguyen[3,4], and Alain Passelègue[2,4]

[1] CNRS, Laboratoire LIP, France
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL), France
benoit.libert@ens-lyon.fr
[3] DIENS, École normale supérieure, CNRS, PSL University, Paris, France
(✉) ky.nguyen@ens.psl.eu
[4] Inria, France
alain.passelegue@inria.fr

Abstract. Chakraborty, Prabhakaran, and Wichs (PKC'20) recently introduced a new tag-based variant of lossy trapdoor functions, termed cumulatively all-lossy-but-one trapdoor functions (CALBO-TDFs). Informally, CALBO-TDFs allow defining a public tag-based function with a (computationally hidden) special tag, such that the function is lossy for all tags except when the special secret tag is used. In the latter case, the function becomes injective and efficiently invertible using a secret trapdoor. This notion has been used to obtain advanced constructions of signatures with strong guarantees against leakage and tampering, and also by Dodis, Vaikunthanathan, and Wichs (EUROCRYPT'20) to obtain constructions of randomness extractors with extractor-dependent sources. While these applications are motivated by practical considerations, the only known instantiation of CALBO-TDFs so far relies on the existence of indistinguishability obfuscation.

In this paper, we propose the first two instantiations of CALBO-TDFs based on standard assumptions. Our constructions are based on the LWE assumption with a sub-exponential approximation factor and on the DCR assumption, respectively, and circumvent the use of indistinguishability obfuscation by relying on lossy modes and trapdoor mechanisms enabled by these assumptions.

Keywords. Lossy trapdoor functions, cumulative lossiness, standard assumptions.

1 Introduction

As introduced by Peikert and Waters [49], *lossy trapdoor functions* (LTDFs) are function families where evaluation keys can be sampled in two modes: In the *injective mode*, a function $F_{\mathsf{ek}}(\cdot)$ is injective and can be inverted using a trapdoor tk that comes with the evaluation key ek; In the *lossy mode*, a function $F_{\mathsf{ek}}(\cdot)$ has a much smaller image size and thus loses a certain amount of information about its input. The standard security of an LTDF requires that the two modes be indistinguishable. That is, no efficient distinguisher can tell apart lossy evaluation keys from injective ones.

Lossy trapdoor functions have been built from a variety of standard cryptographic assumptions, such as the Decisional Diffie-Hellman (DDH) [49,26,30] and Learning with Errors (LWE) assumptions [49,8,2], the Quadratic Residuosity (QR) [38,26,25] and Composite Residuosity (DCR) assumptions [26], the Phi-hiding assumption [42,3] and more [47,54]. They have found numerous applications in cryptography, including chosen-ciphertext security, trapdoor functions with many hard-core bits, collision-resistant hash functions, oblivious transfer [49], deterministic [9,51] and hedged public-key encryption [6,53] in the standard model, instantiability of RSA-OAEP [42], computational extractors [24,28], pseudo-entropy functions [18], selective-opening security [7], and more.

Several generalizations of LTDFs have been considered. Of particular interest are the tag-based variants, where algorithms take an additional tag as input. In all-but-one LTDFs [49] for instance, the evaluation key obtained by running the sampling algorithm with a special tag tag^* is such that the function $F_{\mathsf{ek}}(\cdot, \mathsf{tag})$ is injective for all tags $\mathsf{tag} \neq \mathsf{tag}^*$, but the function $F_{\mathsf{ek}}(\cdot, \mathsf{tag}^*)$ is lossy. All-but-one LTDFs have been generalized to all-but-N LTDFs [37] (which admit $N > 1$ lossy tags) or all-but-many lossy trapdoor functions (where arbitrarily many lossy tags can be adaptively created). The latter notion notably found applications to selective-opening chosen-ciphertext security with compact ciphertexts [40,44,14].

In a setting where multiple lossy evaluations are provided (e.g., for multiple lossy evaluation keys in the context of standard LTDFs or for multiple lossy tags in the context of tag-based LTDFs), one may want to guarantee that multiple lossy evaluations on the same input x do not reveal more information about x than a single evaluation. This additional property was termed *cumulative lossiness* in [19] where it was formalized by requiring the existence of a (possibly inefficient) algorithm that starts with some fixed, partial information about x and recovers the entire information provided by the multiple lossy evaluations. The fact that all these evaluations can be recovered (even inefficiently) from the same amount of partial information on x then guarantees that multiple lossy evaluations on the same input x preserve the entropy of x. In particular, they do not end up leaking x entirely.

In this paper, we investigate the notion of *cumulatively all-lossy-but-one trapdoor functions*, suggested by Chakraborty, Prabharkaran and Wichs [19], which considers the case where *all tags are lossy, except one*. This notion has been used to obtain advanced constructions of randomness extractors [24] and signatures in the leakage and tampering model [19].

Cumulatively All-Lossy-But-One Lossy Trapdoor Functions. A *cumulatively all-lossy-but-one trapdoor functions* (CALBO-TDFs) is a tag-based LTDFs where the function $F_{\mathsf{ek}}(\cdot, \mathsf{tag})$ is lossy for any tag tag except one special injective tag tag^*, for which $F_{\mathsf{ek}}(\cdot, \mathsf{tag}^*)$ is invertible using a trapdoor td associated with ek. In addition, the lossiness is required to be *cumulative* in the sense that multiple evaluations $F_{\mathsf{ek}}(x, \mathsf{tag}_i)$ for lossy tags $\mathsf{tag}_i \neq \mathsf{tag}^*$ always leak the same information about x. Finally, the evaluation key should computationally hide the special injective tag and evaluation keys generated with distinct injective tags are required to be (computationally) indistinguishable.

In [24], the notion of CALBO-TDFs was relaxed by not requiring the existence of a trapdoor for the injective tag tag^*. This relaxed notion, termed CALBO functions (or CALBOs for short), is also implicit in [18,27]. By dropping the trapdoor requirement, these works obtained CALBOs from standard lossy functions (without trapdoor). Therefore it has been possible to construct CALBOs from many standard assumptions such as DDH, LWE, or DCR.

The design of CALBO-TDFs, for which a trapdoor is required in injective mode, is much harder. Indeed, the only known instantiation so far [19] relies on the existence of *indistinguishability obfuscation* [29] (iO) besides the DDH (or LWE) assumption. At a high level, the construction of [19] starts with *cumulative* LTDFs (C-LTDFs), which can be built from LWE or DDH, and combines it with iO and puncturable PRFs [13,41,15]. The idea of [19] is to generate a CALBO-TDF evaluation key as an obfuscated program in which the special injective tag tag^* is hard-wired together with an injective evaluation key for the underlying C-LTDF. This program, on input tag, outputs the hard-wired injective evaluation key if $\mathsf{tag} = \mathsf{tag}^*$; Otherwise, it samples a lossy evaluation key using randomness derived from a puncturable PRF (of which the key is also hard-wired in the program) evaluated on the input tag, and finally returns the resulting evaluation key. When it comes to evaluating a function for an input x and a tag tag, [19] evaluates the underlying C-LTDF on input x using the evaluation key obtained by running the obfuscated program on input tag. The injectivity on the special tag tag^* and the cumulative lossiness property immediately follow from the same properties in the underlying C-LTDF. Indistinguishability of evaluation keys simply follows from the security of iO, the pseudorandomness of the puncturable PRF when puncturing the tags, and the indistinguishability of lossy and injective keys in the underlying C-LTDF.

In [19], CALBO-TDFs served as a building block to construct *leakage and tamper resilient* signature schemes with a *deterministic* signing algorithm, a notion that provides a natural solution to protect signature schemes against leakage, e.g. physical analysis and timing measurements, or tampering attacks, where the adversary deliberately targets the randomness used by the algorithms. The complexity of the CALBO-TDF candidate of [19] motivates the search for simpler, more efficient instantiations of CALBO-TDFs that avoid the use of heavy hammers like obfuscation and rely on more standard assumptions.

1.1 Our Contributions

We present two constructions of CALBO-TDFs based solely on standard assumptions. Our first construction relies on the LWE assumption [52] with *sub-exponential* approximation factor in reducing LWE to a worst-case lattice problem [5], while our second construction relies on Paillier's Composite Residuosity assumption [48] (DCR).

[5] The approximation factor is closely related to the modulus-to-noise ratio q/σ if the LWE problem is defined over the ring of integers modulo q and the errors are sampled from a discrete Gaussian distribution D_σ.

We thus avoid the use of indistinguishability obfuscation (which was used to hide the hard-wired values including the special tag and the injective evaluation key) by relying on lossy modes and trapdoor mechanisms enabled by LWE and DCR. The first construction uses the lossy mode and trapdoor mechanism of LWE in a similar way to [33,2,46]. By exploiting ideas from [45], it achieves a mildly relaxed notion of cumulative lossiness, where cumulative lossiness only holds with overwhelming probability over the choice of (non-injective) tags. The same relaxed notion was used in the LWE+iO-based construction of [19]. This relaxation does not hurt any of the applications, as shown [19]. Our second construction relies on the lossiness and trapdoor mechanism of the Decision Composite Residuosity (DCR) assumption. In particular, it uses the Damgård-Jurik cryptosystem [20] in a similar way to the LTDF of Freeman *et al.* [26].

1.2 Technical Overview

Relaxed CALBO-TDFs from LWE. We start from the observation that CALBOs (without a trapdoor) can be viewed as selectively secure unpredictable functions when the key of the function is the CALBO's input and the input of the function serves as the CALBO's tag. We then upgrade the LWE-based PRF of Libert, Stehlé and Titiu [45] whose security proof precisely relies on the cumulative lossiness of the LWE function (in its derandomized version based on the rounding technique of [4]) for an appropriate choice of parameters. The LWE function (which maps a pair of short integer column-vectors $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}^n \times \mathbb{Z}^m$ to $\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$, for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$) is known [33] to provide a lossy function, and even a lossy trapdoor function for an appropriate choice of parameters [8,2]. The PRF of [45] interprets a variant of the key-homomorphic PRF of [11] as a lossy function in its security proof. More specifically, letting $\lfloor \cdot \rfloor_p : \mathbb{Z}_q \to \mathbb{Z}_p$ denote the rounding function of [4] for moduli $p < q$ defined as $\lfloor \mathbf{x} \rfloor_p = \lfloor (p/q) \cdot \mathbf{x} \rfloor$, the function mapping $\mathbf{x} \in \mathbb{Z}_q^n$ to $\left\lfloor \mathbf{x}^\top \cdot \mathbf{A} \right\rfloor_p$ is injective when $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is uniformly random and lossy (as shown in [2]) when $\mathbf{A}$ is of the form $\mathbf{D}^\top \cdot \mathbf{B} + \mathbf{E}$ for some random $\mathbf{B} \in \mathbb{Z}_q^{\ell \times m}, \mathbf{D} \in \mathbb{Z}_q^{\ell \times n}$, $\ell \ll n$, and some small-norm matrix $\mathbf{E}$. The PRF of [45] maps an input x to $\lfloor \mathbf{s}^\top \mathbf{A}(x) \rfloor_p$, where $\mathbf{s} \in \mathbb{Z}^n$ is the secret key and $\mathbf{A}(x) \in \mathbb{Z}_q^{n \times m}$ is an input-dependent matrix derived from public matrices. The latter matrix is actually obtained using fully homomorphic encryption techniques, by multiplying Gentry-Sahai-Waters (GSW) ciphertexts [32] indexed by the bits of x. The security proof of [45] "programs" $\mathbf{A}(x)$ in such a way that all evaluation queries reveal a lossy function of the secret key $\mathbf{s}$ while the challenge evaluation reveals a non-lossy function $\lfloor \mathbf{s}^\top \mathbf{A}(x^\star) \rfloor_p$ of $\mathbf{s}$. By choosing a large enough ratio q/p, they show that all evaluation queries reveal the same information about the secret key $\mathbf{s}$, which is exactly what we need to prove cumulative lossiness in the CALBO setting. At the same time, [45] shows that $\lfloor \mathbf{s}^\top \mathbf{A}(x^\star) \rfloor_p$ retains a large amount of entropy conditionally on the information revealed by all evaluation queries.

We introduce two modifications in the function of [45]. First, we only need a selectively secure version of their PRF since the injective tag tag^* is known

ahead of time in the security experiment whereas [45] has to prove adaptive security using an admissible hash function [10]. We thus remove the admissible hash function and directly compute $\mathbf{A}(x)$ as a product of public GSW ciphertexts indexed by the tag bits without encoding them first. As a second modification w.r.t [45], we need to extend the tag-dependent matrix $\mathbf{A}(x)$ so as to ensure invertibility in injective mode.

Our CALBO construction can be outlined as follows. Given the injective tag $\mathsf{tag}^* \in \{0,1\}^t$, the setup algorithm first generates $\mathbf{A} = \mathbf{D}^\top \cdot \mathbf{B} + \mathbf{E} \in \mathbb{Z}_q^{n \times m}$ as a lossy matrix, where $\mathbf{B} \in \mathbb{Z}_q^{\ell \times m}$, $\mathbf{D} \in \mathbb{Z}_q^{\ell \times n}$ and $\mathbf{E} \in \mathbb{Z}^{n \times m}$, with $\ell \ll n < m$. Then, the setup algorithm embeds $(\mathbf{A}, \mathbf{B})$ in the evaluation key ek via a set of GSW ciphertexts [32]

$$\mathbf{A}_{i,b} = \mathbf{A} \cdot \mathbf{R}_{i,b} + \delta_{b,\mathsf{tag}_i^*} \cdot \mathbf{G} \qquad \forall i \in [t],\, b \in \{0,1\} \tag{1}$$

where tag_i^* denotes the i-th bit of tag^*, $\delta_{b,\mathsf{tag}_i^*} = (b \stackrel{?}{=} \mathsf{tag}_i^*)$, $\mathbf{G} \in \mathbb{Z}_q^{n \times \lceil n \cdot \log q \rceil}$ is the gadget matrix of Micciancio and Peikert [46], and $\mathbf{R}_{i,b} \in \{0,1\}^{m \times \lceil n \cdot \log q \rceil}$ for each $i \in [t]$. The trapdoor tk (which allows inverting in injective mode) contains $\{\mathbf{R}_{i,b}\}_{i \in [t], b \in \{0,1\}}$. The computational indistinguishability of keys for different injective tags follows from the LWE assumption. The latter implies that the lossy matrix $\mathbf{A} = \mathbf{D}^\top \cdot \mathbf{B} + \mathbf{E}$ is indistinguishable from a uniform matrix in $\mathbb{Z}_q^{n \times m}$. When $\mathbf{A}$ is uniform, the Leftover Hash Lemma implies that each product $\mathbf{A} \cdot \mathbf{R}_{i,b}$ is statistically close to the uniform distribution $U(\mathbb{Z}_q^{n \times m})$. This ensures that matrices (1) statistically hide tag^* as they are statistically indistinguishable from i.i.d. random matrices over $\mathbb{Z}_q$.

In order to evaluate the function on an input $\mathbf{x}$ for a tag tag, the evaluation algorithm computes a product of GSW ciphertexts $\{\mathbf{A}_{i,\mathsf{tag}_i}\}_{i=1}^t$ chosen among $\{(\mathbf{A}_{i,0}, \mathbf{A}_{i,1})\}_{i=1}^t$ and then obtains a ciphertext $\mathbf{A}(\mathsf{tag})$ encrypting the logical AND $C_{\mathsf{tag}}(\mathsf{tag}^*) \triangleq \bigwedge_{i=1}^t (\mathsf{tag}_i \stackrel{?}{=} \mathsf{tag}_i^*)$, where $\{\mathsf{tag}_i\}_{i=1}^t$ are the bits of tag. Said otherwise, the tag-dependent matrix $\mathbf{A}(\mathsf{tag}) = \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}} + C_{\mathsf{tag}^*}(\mathsf{tag}) \cdot \mathbf{G}$ is an encryption of $C_{\mathsf{tag}}(\mathsf{tag}^*) = \prod_{i=1}^t \delta_{\mathsf{tag}_i,\mathsf{tag}_i^*}$, where the circuit $C_{\mathsf{tag}}(\cdot)$ is homomorphically evaluated by computing a subset product of GSW ciphertexts in the most sequential way (according to the terminology in [5]) so as to minimize the noise growth. This is done by making sure that each multiplication always involves a fresh GSW ciphertext.

Finally, the output of the evaluation is $\left\lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A}(\mathsf{tag})] \right\rceil_p$. Here, we slightly modify [45] where the challenge evaluation is of the form $\left\lfloor \mathbf{x}^\top \mathbf{A}(\mathsf{tag}) \right\rceil_p$. The reason is that, in order to ensure invertibility for the injective tag tag^*, we need to exploit the fact that $\mathbf{A}(\mathsf{tag}^*)$ depends on $\mathbf{G}$. To this end, we need an injective evaluation of $\mathbf{x}$ to be of the form

$$\left\lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A}(\mathsf{tag}^*)] \right\rceil_p = \left\lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}^*} + \mathbf{G}] \right\rceil_p$$

for some small-norm matrix $\mathbf{R}_{\mathsf{tag}^*} \in \mathbb{Z}^{n \times \lceil n \cdot \log q \rceil}$. In this case, the binary matrices $\mathbf{R}_{i,b}$ contained in tk can be used to compute $\mathbf{R}_{\mathsf{tag}^*}$, which is a Micciancio-Peikert trapdoor [46] for the matrix $[\mathbf{A} \,|\, \mathbf{A}(\mathsf{tag}^*)]$ and allows inverting the function $\mathbf{x} \to \left\lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A}(\mathsf{tag}^*)] \right\rceil_p$ in the same way as in the LTDF of [2].

In lossy mode (when tag differs from tag^* in at least one bit), we can achieve cumulative lossiness only for a fixed input, due to the error introduced by the rounding operation. The argument is essentially the same as that in [45]: We exploit the lossy form of $\mathbf{A}$ and the fact that, for any lossy tag $\mathsf{tag} \neq \mathsf{tag}^*$, the matrix $[\mathbf{A} \,|\, \mathbf{A}(\mathsf{tag})] = [\mathbf{A} \,|\, \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}}]$ does not depend on $\mathbf{G}$. Then, with overwhelming probability, multiple evaluations $\lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}}] \rceil_p$ always reveal the same information about $\mathbf{x} \in \mathbb{Z}^n$ since w.h.p. we have

$$\lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}}] \rceil_p = \lfloor \mathbf{x}^\top \cdot \mathbf{D}^\top \cdot \mathbf{B} \,|\, (\mathbf{x}^\top \cdot \mathbf{D}^\top \cdot \mathbf{B}) \cdot \mathbf{R}_{\mathsf{tag}} \rceil_p$$

when q/p is sufficiently large. Hence, the evaluations $\left\{ \lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A}(\mathsf{tag})] \rceil_p \right\}_{\mathsf{tag} \neq \mathsf{tag}^*}$ do not reveal any more information than $\mathbf{D} \cdot \mathbf{x} \in \mathbb{Z}_q^\ell$. Concerning the relaxation of cumulative lossiness, Chakraborty *et al.* [19] have the same restriction in their use of the LWE assumption. However, as discussed in [19, Apppendix A], this relaxed notion is not a problem in their applications of CALBO-TDFs.

CALBO-TDFs FROM DCR. We give a construction of CALBO-TDFs based on the Damgård-Jurik homomorphic encryption scheme [20] with additional insights from [21,22]. The construction is obtained by composing together multiple instances of the DCR-based lossy trapdoor *permutation* of Freeman *et al.* [26], which is index-dependent as its domain depends on the evaluation key. Recall that the Damgård-Jurik cryptosystem uses the group $\mathbb{Z}^*_{N^{\zeta+1}}$, where $N = pq$ is an RSA modulus and $\zeta \geq 1$ is some natural number. Given an injective tag $\mathsf{tag}^* \in \{0,1\}^t$, the evaluation key ek of our CALBO-TDFs includes (N, ζ) and the following Damgård-Jurik ciphertexts

$$g_{i,b} = (1+N)^{\delta_{b,\mathsf{tag}^*_i}} \cdot \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1} \qquad \forall (i,b) \in [t] \times \{0,1\} \ ,$$

where $\alpha_{i,b} \hookleftarrow U(\mathbb{Z}^*_N)$ for each $i \in [t]$, $b \in \{0,1\}$, $\delta_{b,\mathsf{tag}^*_i} = (b \stackrel{?}{=} \mathsf{tag}^*_i)$ and tag^*_i denotes the i-th bit of tag^*. The trapdoor tk consists of the Damgård-Jurik decryption key.

For an evaluation of an input $x \in \mathbb{Z}_{N^{\zeta+1}}$ given a tag tag, we first write $x_0 \coloneqq x = y_0 \cdot N + z_0$ for $(y_0, z_0) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_N$. Then, we iterate for $i \in [t]$ and, at each iteration, we compute a Damgård-Jurik ciphertext x_i of y_{i-1}:

$$x_i = g_{i,\mathsf{tag}_i}^{y_{i-1}} \cdot z_{i-1}^{N^\zeta} \bmod N^{\zeta+1} \ .$$

The output of the function consists of x_t.

In the injective mode (where $\mathsf{tag} = \mathsf{tag}^*$), we have that g_{i,tag^*_i} is an encryption of 1 for each $i \in [t]$. Then, each x_i is an encryption of y_{i-1}. Using tk, the inverter can thus recover (y_{i-1}, z_{i-1}) from x_i and eventually recover (y_0, z_0) and $x = x_0$ as long as $z_{i-1} \in \mathbb{Z}^*_N$ at each iteration. For any input x such that $z_{i-1} \notin \mathbb{Z}^*_N$ at some iteration, the evaluation algorithm outputs 0 (analogously to an index-dependent DCR-based LTDF proposed by Auerbach *et al.* [3, Section 6.1]). We note that our DCR-based construction is not perfectly invertible in injective mode, the fraction of inputs for which the function is not invertible is negligible.

Moreover, finding such inputs is as hard as factoring N and thus contradicts the DCR assumption.

In the lossy mode (where $\mathsf{tag} \neq \mathsf{tag}^*$), let the smallest index $i \in [t]$ such that $\mathsf{tag}_i \neq \mathsf{tag}^*$. For this index i, g_{i,tag_i} is a Damgård-Jurik encryption of 0, and so is x_i at the i-th evaluation step. This implies that x_i loses information about y_{i-1} as it can take at most $\varphi(N)$ values.

We then observe that injectivity and indistinguishability follow from the correctness and semantic security of Damgård-Jurik. Cumulative lossiness can be argued using the same arguments as in the CALBO function of [24, Section 5.3.1]. At each evaluation step, the information $(y_{i-1}, z_{i-1}) \in \mathbb{Z}_{N^\varsigma} \times \mathbb{Z}_N$ about x is fully carried over to the next step of the evaluation if $\mathsf{tag}_i = \mathsf{tag}_i^*$ and $z_{i-1} \in \mathbb{Z}_N^*$. As soon as tag_i differs from tag_i^*, the information about y_{i-1} is lost and subsequent evaluation steps (and therefore the final output of the evaluation) only depend on at most $\log \varphi(N) < \log N$ bits of x. Since there are t positions where a lossy tag can differ from tag^* for the first time, the function $\{F_{\mathsf{ek}}(\cdot, \mathsf{tag})\}_{\mathsf{tag} \neq \mathsf{tag}^*}$ has image size at most $\varphi(N)^t$. So, the union of all lossy evaluations $\{F_{\mathsf{ek}}(x, \mathsf{tag})\}_{\mathsf{tag} \neq \mathsf{tag}^*}$ on some input x reveals at most $\log(\varphi(N)^t) < t \cdot \log N$ bits about x.

1.3 Related Work

Dodis, Vaikuntanathan and Wichs [24, Section 5.3.1] considered a notion of cumulatively all-lossy-but-one (CALBO) functions without trapdoor, which they used to extract randomness from extractor-dependent sources. They showed that CALBOs can be generically realized from standard lossy functions by relaxing the injectivity property. Due to their relaxed notion of injectivity, their construction is not invertible in injective mode. Our DCR-based CALBO-TDF is inspired by their construction (which is itself similar to the pseudo-entropy function of Braverman *et al.* [18]) with the difference that we do not need to compose a standard lossy function with a compressing d-wise independent function at each iterative evaluation step. This is the reason why our injective mode is invertible.

In a recent work, Quach, Waters, and Wichs [50] introduced a new notion of *targeted lossy functions* (TLFs), where lossy evaluations only lose information on some targeted inputs and no trapdoor allows efficiently inverting in the injective mode. Quach *et al.* [50] also extended TLFs to *targeted all-lossy-but-one* (T-ALBOs) and *targeted all-injective-but-one* (T-AIBOs) variants. Interestingly, it was shown in [50] that, in contrast to lossy *trapdoor* functions, TLFs, T-ALBOs, and T-AIBOs can be realized in Minicrypt. We can also consider the relaxation of targeted lossiness alone, while still asking for a trapdoor in the injective mode. This notion was discussed in [30] where a construction based on the Computational Diffie-Hellman assumption was given.

Lossy algebraic filters (LAFs) [39,43] are tag-based lossy functions that were used to construct public-key encryption schemes with circular chosen-ciphertext security [39]. They provide similar functionalities to CALBO in that they explicitly require multiple evaluations $\{F_{\mathsf{ek}}(x, \mathsf{tag}_i)\}_i$ on distinct lossy tags to always leak the same information about x. One difference is that LAFs admit arbitrarily many injective tags and arbitrarily many lossy tags. The requirement is

that lossy tags should be hard to find without a trapdoor key. In contrast to CALBO-TDFs, LAFs do not support efficient inversion on injective tags.

2 Background

We write $[n]$ to denote the set $\{1, 2, \ldots, n\}$ for an integer n. For any $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers with addition and multiplication modulo q, containing the representatives in the interval $(-q/2, q/2)$. We always set q as a prime integer. For $2 \leq p < q$ and $x \in \mathbb{Z}_q$, we define $\lfloor x \rceil_p \coloneqq \lfloor (p/q) \cdot x \rceil \in \mathbb{Z}_p$ where the operator $\lfloor y \rceil$ means taking the largest integer less than or equal to y. This notation is readily extended to vectors over $\mathbb{Z}_q$. Given a distribution D, we write $x \sim D$ to denote a random variable x distributed according to D. For a finite set S, we let $U(S)$ denote the uniform distribution over S. If X and Y are distributions over the same domain $\mathcal{D}$, then $\Delta(X, Y)$ denotes their statistical distance. We write ppt as a shorthand for "probabilistic polynomial-time" when considering the complexity of algorithms. We use a generalized version of the *Leftover Hash Lemma* [36].

Lemma 1 ([1], Lemma 14). *Let $\mathcal{H} = \{h : X \to Y\}_{h \in \mathcal{H}}$ be a family of universal hash functions. Let $f : X \to Z$ be some function. Let $T_1, \ldots, T_k$ be k independent random variables over X and we define $\gamma \coloneqq \max_k \gamma(T_i)$ where $\gamma(T_i) \coloneqq \max_{t \in X} \Pr[T_i = t]$. Then, we have*

$$\Delta\Big(\; (h, h(T_1), f(T_1), \ldots, h(T_k), f(T_k)) \, ; \Big(h, U_Y^{(1)}, f(T_1), \ldots, U_Y^{(k)}, f(T_k)\Big) \; \Big) \leq \frac{k}{2}\sqrt{\gamma \cdot |Y| \cdot |Z|}$$

where $U_Y^{(1)}, \ldots, U_Y^{(k)}$ denote k uniformly random variables over Y.

2.1 Cumulatively All-Lossy-But-One Trapdoor Functions

We now recall the definition of *cumulatively all-lossy-but-one trapdoor functions (CALBO-TDFs)*, a notion recently introduced in [19,24] as an extension of lossy trapdoor functions. We also recall its variant with relaxed cumulative lossiness that we achieve assuming LWE. We refer the reader to the introduction for an overview of these notions in the general context of lossy trapdoor functions.

Definition 1 (CALBO-TDF). *Let $\lambda \in \mathbb{N}$ be a security parameter and $\ell, \alpha : \mathbb{N} \to \mathbb{N}$ be functions. Let $\mathcal{T} = \{\mathcal{T}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of sets of tags. An (ℓ, α)-*cumulatively-all-lossy-but-one trapdoor function family (CALBO-TDF) *with respect to the tag family $\mathcal{T}$ is a triple of algorithms* $(\mathsf{Sample}, \mathsf{Eval}, \mathsf{Invert})$*, where the first is probabilistic and the latter two are deterministic:*

- $\mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$*: on inputs 1^λ and $\mathsf{tag}^* \in \mathcal{T}_\lambda$, sample and output* $(\mathsf{ek}, \mathsf{tk})$.
- $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x)$*: on inputs $x \in \{0,1\}^{\ell(\lambda)}$, an evaluation key* ek *and* tag*, output an element y in some set $\mathcal{R}$ of images.*

– $\mathsf{Invert}(\mathsf{tk}, \mathsf{tag}, y)$*: on inputs* $y \in \mathcal{R}$*, a trapdoor key* tk*, and* tag*, output* $x' \in \{0,1\}^{\ell(\lambda)}$*.*

We require the following properties:

– *(Injectivity) There exists a negligible function* $\mathsf{negl} : \mathbb{N} \to \mathbb{N}$ *such that for all* $\lambda \in \mathbb{N}$*,* $\mathsf{tag}^* \in \mathcal{T}_\lambda$*,* $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$ *we have*

$$\frac{|\{x \in \{0,1\}^{\ell(\lambda)} \ : \ \mathsf{Invert}(\mathsf{tk}, \mathsf{tag}^*, \mathsf{Eval}(\mathsf{ek}, \mathsf{tag}^*, x)) = x\}|}{2^{\ell(\lambda)}} \geq 1 - \mathsf{negl}(\lambda) \ .$$

– *(α-cumulative lossiness) For all* $\lambda \in \mathbb{N}$*, all tags* $\mathsf{tag}^* \in \mathcal{T}_\lambda$*, and all* $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$*, there exists a (possibly inefficient) function* $\mathsf{compress}_{\mathsf{ek}} : \{0,1\}^{\ell(\lambda)} \to \mathcal{R}_{\mathsf{ek}}$ *where* $|\mathcal{R}_{\mathsf{ek}}| \leq 2^{\ell(\lambda)-\alpha(\lambda)}$ *such that for all* $\mathsf{tag} \neq \mathsf{tag}^*$ *and* $x \in \{0,1\}^{\ell(\lambda)}$*, there exists a (possibly inefficient) function* $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}} : \mathcal{R}_{\mathsf{ek}} \to \mathcal{R}$ *satisfying*

$$\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x) = \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(x)) \ . \tag{2}$$

– *(Indistinguishability) For all* $\mathsf{tag}_0^*, \mathsf{tag}_1^* \in \mathcal{T}_\lambda$*, the two ensembles*

$$\{\mathsf{ek}_0 : (\mathsf{ek}_0, \mathsf{tk}_0) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}_0^*)\}_{\lambda \in \mathbb{N}}$$
$$\{\mathsf{ek}_1 : (\mathsf{ek}_1, \mathsf{tk}_1) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}_1^*)\}_{\lambda \in \mathbb{N}}$$

are computationally indistinguishable.

An alternative, relaxed notion of CALBO-TDFs was also proposed in [19,24]. In this relaxed variant, cumulative lossiness is slightly simplified by requiring Equation (2) to only hold with overwhelming probability over the choice of tags. This minor relaxation does not impact applications, as the relaxed notion was proven sufficient for all known applications of CALBO-TDFs in [19, Appendix A]. We use this relaxation in our LWE-based construction in Section 3.1, and recall it below. We refer to this notion as relaxed CALBO-TDFs.

(relaxed α-cumulative lossiness) There exists a negligible function $\mathsf{negl} : \mathbb{N} \to (0,1)$ and for sufficiently large $\lambda \in \mathbb{N}$, for any $\mathsf{tag}^* \in \mathcal{T}_\lambda$, for all $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$, there exists a (possibly inefficient) function $\mathsf{compress}_{\mathsf{ek}} : \{0,1\}^{\ell(\lambda)} \to \mathcal{R}_{\mathsf{ek}}$ where $|\mathcal{R}_{\mathsf{ek}}| \leq 2^{\ell(\lambda)-\alpha(\lambda)}$ such that for any fixed randomly chosen $x \in \{0,1\}^{\ell(\lambda)}$, there exists a (possibly inefficient) function $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}} : \mathcal{R}_{\mathsf{ek}} \to \mathcal{R}$ satisfying

$$\Pr[\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x) = \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(x))] \geq 1 - \mathsf{negl}(\lambda) \ ,$$

where the probability is taken over the choices of $\mathsf{tag} \neq \mathsf{tag}^*$. We call $\mathsf{negl}(\lambda)$ the *lossiness error* of the CALBO-TDF.

Lossiness rate. We define the *lossiness rate* of an (ℓ, α)-CALBO-TDF by the rate of bits lost on lossy tags, namely $1 - (\ell - \alpha)/\ell = \alpha/\ell$. This is similar to the notion of lossiness rate used in [49,30]. Ideally, we want this rate to be as close to 1 as possible, for example $1 - o(1)$.

2.2 Lattices

Unless stated otherwise, we write vectors as column vectors. For a full-row rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define the lattice $\Lambda(\mathbf{A})$ admitting $\mathbf{A}$ as a basis by $\Lambda(\mathbf{A}) = \{\mathbf{s}^\top \cdot \mathbf{A} : \mathbf{s} \in \mathbb{Z}_q^n\}$. We also define the lattice $\Lambda^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}$. Given a vector $\mathbf{x} \in \mathbb{Z}_q^n$, we define its ℓ_∞-norm as $\|\mathbf{x}\|_\infty = \max_{i \in [n]} |\mathbf{x}[i]|$ where $\mathbf{x}[i]$ denotes the i-th coordinate of $\mathbf{x}$. We let $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}[1]^2 + \cdots + \mathbf{x}[n]^2}$ denote the Euclidean norm of $\mathbf{x}$. The *minimum distance* measured in ℓ_∞-norm of a lattice Λ is given by $\lambda_1^\infty(\Lambda) \coloneqq \min_{\mathbf{x} \neq \mathbf{0}} \|\mathbf{x}\|_\infty$. For a basis $\mathbf{B}$ of $\mathbb{R}^n$, the origin-centered parallelepiped is defined as $\mathcal{P}_{1/2}(\mathbf{B}) \coloneqq \mathbf{B} \cdot [-1/2, 1/2)^n$. We also use the following infinity norm for matrices $\mathbf{B} \in \mathbb{Z}^{n \times m}$:

$$\|\mathbf{B}\|_\infty = \max_{i \in [n]} \left(\sum_{j=1}^{m} |\mathbf{B}_{i,j}| \right) \quad .$$

Let $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix and $\mathbf{c} \in \mathbb{R}^n$ be a vector. We define the *Gaussian function* over $\mathbb{R}^n$ by $\rho_{\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) = \exp(-\pi(\mathbf{x}-\mathbf{c})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{c}))$ and if $\mathbf{\Sigma} = \sigma^2 \cdot \mathbf{I}_n$ and $\mathbf{c} = \mathbf{0}$, we write ρ_σ for $\rho_{\mathbf{\Sigma},\mathbf{c}}$. For any discrete set $\Lambda \subset \mathbb{R}^n$, the *discrete Gaussian distribution* $D_{\Lambda,\mathbf{\Sigma},\mathbf{c}}$ has probability mass $\Pr_{X \sim D_{\Lambda,\mathbf{\Sigma},\mathbf{c}}}[X = \mathbf{x}] = \frac{\rho_{\mathbf{\Sigma},\mathbf{c}}(\mathbf{x})}{\rho_{\mathbf{\Sigma},\mathbf{c}}(\Lambda)}$, for any $\mathbf{x} \in \Lambda$. When $\mathbf{c} = \mathbf{0}$ and $\mathbf{\Sigma} = \sigma^2 \cdot \mathbf{I}_n$ we denote $D_{\Lambda,\mathbf{\Sigma},\mathbf{c}}$ by $D_{\Lambda,\sigma}$.

Learning-With-Errors Assumption. Our first CALBO-TDFs relies on the LWE assumption.

Definition 2. *Let $\alpha : \mathbb{N} \to (0,1)$ and $m \geq n \geq 1$, $q \geq 2$ be functions of a security parameter $\lambda \in \mathbb{N}$. The* **Learning with Errors** (LWE) *problem consists in distinguishing between the distributions $(\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)$ and $U(\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m)$, where $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$, $\mathbf{s} \sim U(\mathbb{Z}_q^n)$ and $\mathbf{e} \sim D_{\mathbb{Z}^m,\alpha q}$. For an algorithm $\mathcal{A} : \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m \to \{0,1\}$, we define*

$$\mathbf{Adv}^{\mathsf{LWE}}_{q,m,n,\alpha}(\mathcal{A}) = \left| \Pr[\mathcal{A}\left(\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top\right) = 1] - \Pr[\mathcal{A}\left(\mathbf{A}, \mathbf{u}\right) = 1 \right| ,$$

where the probabilities are over $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$, $\mathbf{s} \sim U(\mathbb{Z}_q^n)$, $\mathbf{u} \sim U(\mathbb{Z}_q^m)$ and $\mathbf{e} \sim D_{\mathbb{Z}^m,\alpha q}$ and the internal randomness of $\mathcal{A}$. We say that $\mathsf{LWE}_{q,m,n,\alpha}$ is hard if for all ppt algorithm $\mathcal{A}$, the advantage $\mathbf{Adv}^{\mathsf{LWE}}_{q,m,n,\alpha}(\mathcal{A})$ is negligible in λ.

We require that $\alpha \geq 2\sqrt{n}/q$ for the reduction from worst-case lattice problems and refer the readers to, e.g., [17] for more details.

We will need the techniques for *homomorphic encryption (HE)* [32] in order to build CALBO-TDFs from LWE. In this paper, we consider only binary circuits with fan-in-2 gates for homomorphic evaluation. We use the terms *size* and *depth* of a circuit to refer to the number of its gates and the length of its longest input-to-output path, respectively. The syntax of HE schemes is recalled in Appendix A.1. We note that in our construction from LWE, we do not need the general fully homomorphic encryption thanks to the fact that all evaluated circuits have bounded depths, for the sole purpose of comparing tags. Hence, *leveled* homomorphic encryption suffices for our purposes.

Gadget matrix. We recall the "gadget matrix" from [46] and their homomorphic properties. The technique is later developed further in [12,34,35]. For an integer modulus q, the gadget vector over $\mathbb{Z}_q$ is defined as $\mathbf{g} = (1, 2, 4, \ldots, 2^{\lceil \log q \rceil - 1})$. The gadget matrix $\mathbf{G}_n$ is the tensor (or Kronecker) product $\mathbf{I}_n \otimes \mathbf{g} \in \mathbb{Z}_q^{n \times n'}$ where $n' = \lceil n \log q \rceil$. There exists an efficiently computable function $\mathbf{G}_n^{-1} : \mathbb{Z}_q^{n \times n'} \to \{0,1\}^{n' \times n'}$ such that $\mathbf{G}_n \cdot \mathbf{G}_n^{-1}(\mathbf{A}) = \mathbf{A}$ for all $\mathbf{A} \in \mathbb{Z}_q^{n \times n'}$. In particular, we can define $\mathbf{G}_n^{-1}$ to be the entry-wise binary decomposition of matrices in $\mathbb{Z}_q^{n \times n'}$. In the following, we omit the subscript n and write $\mathbf{G}$ when it is clear from context. Lemma 2 helps bound the noise of the output ciphertext after homomorphically evaluating a depth-τ circuit C containing only AND gates. This will affect our parameter choices for the LWE-based CALBO-TDFs as well as our later argument for its relaxed cumulative lossiness.

Lemma 2 (Adapted from [32,12,16]). *Let $\lambda \in \mathbb{N}$ and $m = m(\lambda), n = n(\lambda)$. We define $n' \coloneqq \lceil n \log q \rceil$. Let $C : \{0,1\}^t \to \{0,1\}$ be a AND Boolean circuit of depth τ. Let $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + b_i \cdot \mathbf{G} \in \mathbb{Z}_q^{n \times m}$ with $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{R}_i \in \{-1,1\}^{m \times n'}$ and $b_i \in \{0,1\}$, for $i \leq t$. There exist deterministic algorithms FHEval and $\mathsf{EvalPriv}$ with running time $\mathsf{poly}(4^\tau, t, m, n, \log q)$ that satisfy:*

$$\mathsf{FHEval}(C, (\mathbf{A}_i)_i) = \mathbf{A} \cdot \mathbf{R}_C + C(b_1, \ldots, b_t) \cdot \mathbf{G} = \mathbf{A} \cdot \mathbf{R}_C + \bigwedge_{i=1}^{t} b_i \cdot \mathbf{G},$$

where $\mathbf{R}_C = \mathsf{EvalPriv}\big(C, ((\mathbf{R}_i, b_i))_i\big)$ and $\|\mathbf{R}_C\|_\infty \leq \max_i\{\|\mathbf{R}_i\|_\infty\} \cdot (n'+1)^\tau$.

Lossy mode of LWE. We recall the Lossy sampler for LWE that is introduced by Goldwasser *et al.* in [33] and later developed by Alwen *et al.* in [2].

Definition 3. *Let $\chi = \chi(\lambda)$ be an efficiently sampleable distribution over $\mathbb{Z}$. We define an efficient* lossy sampler $(\mathbf{A}, \mathbf{B}) \leftarrow \mathsf{Lossy}(1^m, 1^n, 1^\ell, q, \chi)$ *via:*

> $\mathsf{Lossy}(1^m, 1^n, 1^\ell, q, \chi)$*: Sample* $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{\ell \times m}), \mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{\ell \times n}), \mathbf{E} \hookleftarrow \chi^{n \times m}$, *where $\ell \ll n$, and output $\mathbf{A} = \mathbf{D}^\top \cdot \mathbf{B} + \mathbf{E} \in \mathbb{Z}_q^{n \times m}$ together with $\mathbf{B}$.*

We remark that the lossy sampler reveals the coefficient matrix $\mathbf{B}$ along with $\mathbf{A}$ but as long as the secret matrix $\mathbf{D}$ is not leaked, this does not compromise the pseudorandomness of $\mathbf{A}$. Indeed, it can be shown that under the $\mathsf{LWE}_{q,m,\ell,\alpha}$ assumption, $\mathbf{A}$ is computationally indistinguishable from a uniformly random matrix. Intuitively, the dimension of the secret is now ℓ and we view each row of $\mathbf{D}^\top$ as a secret vector, $\mathbf{B}$ as the uniform coefficients and each row of $\mathbf{A}$ as the resulting LWE vector. Formally, we have the following lemma:

Lemma 3 ([33]). *Let a random matrix $\tilde{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ and let a pair $(\mathbf{A}, \mathbf{B}) \leftarrow \mathsf{Lossy}(1^m, 1^n, 1^\ell, q, \chi)$, where $\chi = D_{\mathbb{Z}, \alpha q}$ is an error distribution. Then, under the $\mathsf{LWE}_{q,m,\ell,\alpha}$ assumption, the following two distributions are computationally indistinguishable: $\mathbf{A} \overset{\text{comp}}{\approx} \tilde{\mathbf{A}}$.*

Trapdoor mechanisms for LWE. Micciancio and Peikert [46] introduced a trapdoor mechanism for LWE. Their technique makes use of the "gadget matrix" $\mathbf{G} \in \mathbb{Z}_q^{n \times n'}$, where $n' = \lceil n \log q \rceil$, and for $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m+n')}$, they call a short matrix $\mathbf{R} \in \mathbb{Z}^{m \times n'}$ a $\mathbf{G}$-trapdoor of $\mathbf{A}'$ if $\mathbf{A}' \cdot [\mathbf{R}^\top \,|\, \mathbf{I}_m]^\top = \mathbf{H}\mathbf{G}$ for some invertible $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$. Micciancio and Peikert also showed that using a $\mathbf{G}$-trapdoor allows one to invert the LWE function $(\mathbf{s}, \mathbf{e}) \mapsto \mathbf{s}^\top \mathbf{A}' + \mathbf{e}^\top$ for any $\mathbf{s} \in \mathbb{Z}_q^n$ and any error $\mathbf{e} \in \mathbb{Z}^{m+n'}$ such that $\|\mathbf{e}\|_2 \leq q/O(\sqrt{n \log q})$. More specifically, we have the following lemma:

Lemma 4 ([46], Theorem 4.1 and Section 5). *Let $n' = \lceil n \log q \rceil$ and $\delta = \mathsf{negl}(n)$. Assume that $m \geq n \log q + 2 \log \frac{n'}{2\delta}$. Then there exists a* ppt *algorithm* GenTrap *that takes as inputs matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{H} \in \mathbb{Z}_q^{n \times n}$, outputs a short matrix $\mathbf{R} \in \{-1, 0, 1\}^{m \times n'}$ and $\mathbf{A}' = [\mathbf{A} \mid -\mathbf{A} \cdot \mathbf{R} + \mathbf{H} \cdot \mathbf{G}] \in \mathbb{Z}_q^{n \times (m+n')}$ such that if $\mathbf{H}$ is invertible, then $\mathbf{R}$ is a $\mathbf{G}$-trapdoor of $\mathbf{A}'$ and we call $\mathbf{H}$ the* invert tag *of $\mathbf{A}'$.*

In particular, inverting the function $g_{\mathbf{G}}(\mathbf{s}, \mathbf{e}) \coloneqq \mathbf{s}^\top \cdot \mathbf{G} + \mathbf{e}^\top$ can be done in quasi-linear time $\tilde{O}(n)$ for any $\mathbf{s} \in \mathbb{Z}_q^n$ and any $\mathbf{e} \in \mathcal{P}_{1/2}(q \cdot (\mathbf{B}^{-1})^\top)$, where $\mathbf{B}$ is a basis of the lattice $\Lambda^\perp(\mathbf{G}) = \{\mathbf{z} \in \mathbb{Z}^{n'} : \mathbf{G} \cdot \mathbf{z} = 0 \pmod q\}$.

In a follow-up work, Alwen *et al.* [2] used GenTrap to construct trapdoors for inverting *Learning with Rounding* (LWR) instances $\lfloor \mathbf{s}^\top \mathbf{A} \rceil_p$. Their main observation is that one can convert $\lfloor \mathbf{s}^\top \mathbf{A} \rceil_p$ to $\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$ where $\|\mathbf{e}\|_2 \leq \sqrt{m} q/p$, by first multiplying with q/p then taking the ceiling value. Afterwards, using a $\mathbf{G}$-trapdoor of $\mathbf{A}$, e.g. a sample from GenTrap, allows one to compute back $\mathbf{s}$. Formally, we have the following lemma:

Lemma 5 ([2], Lemma 6.3). *Let $n' = \lceil n \log q \rceil$ and $\delta = \mathsf{negl}(n)$. Assume that $m \geq n \log q + 2 \log \frac{n'}{2\delta}$ and $p \geq O(\sqrt{(m+n')n'})$. Then there exists a* ppt *algorithm* LWRInvert *that takes as inputs $(\mathbf{A}', \mathbf{R})$ with $\mathbf{R}$ being a $\mathbf{G}$-trapdoor of $\mathbf{A}'$, together with some $\mathbf{c} \in \mathbb{Z}_p^{m+n'}$ such that $\mathbf{c} = \lfloor \mathbf{s}^\top \mathbf{A}' \rceil_p$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, then outputs $\mathbf{s}$.*

We will also need the following technical lemmas. Lemma 6 comes from a work by Gentry, Peikert, and Vaikuntanathan [31].

Lemma 6 ([31], Lemma 5.3). *Let ℓ and q be positive integers and q be prime. Let $n \geq 2\ell \log q$. Then for all but an at most q^{-n} fraction of $\mathbf{D} \in \mathbb{Z}_q^{\ell \times n}$, we have $\lambda_1^\infty(\Lambda(\mathbf{D})) \geq q/4$, where $\Lambda(\mathbf{D}) = \{\mathbf{s}^\top \mathbf{D} : \mathbf{s} \in \mathbb{Z}_q^\ell\}$ and $\lambda_1^\infty(\Lambda(\mathbf{D}))$ is the* minimum distance *of $\Lambda(\mathbf{D})$ measured in the ℓ_∞-norm.*

Lemma 7 ([2], Lemma 2.7). *Let p, q be positive integers and $p < q$. Let $R > 0$ be an integer. Then, the probability that there exists $e \in [-R, R]$ such that $\lfloor y \rceil_p \neq \lfloor y + e \rceil_p$, where $y \hookleftarrow U(\mathbb{Z}_q)$, is at most $2pR/q$.*

The following lemma is well-known, e.g. a simple proof can be found in [45, Lemma 2.3].

Lemma 8. *Let q be a prime a $D_{m,n,q}$ be a distribution over $\mathbb{Z}_q^{n\times m}$ such that $\Delta(D_{m,n,q}, U(\mathbb{Z}_q^{n\times m})) \leq \epsilon$. Then, let $V_{n,q}$ be any distribution over $\mathbb{Z}_q^n$, we have $\Delta(V_{n,q}^\top \cdot D_{m,n,q},\, U(\mathbb{Z}_q^m)) \leq \epsilon + \alpha \cdot \left(1 - \frac{1}{q^m}\right)$ where $\alpha \coloneqq \Pr[\mathbf{v} \hookleftarrow V_{n,q} : \mathbf{v} = \mathbf{0}]$.*

2.3 Composite Residuosity

Our second CALBO-TDFs relies on Paillier's composite residuosity assumption.

Definition 4 ([48,20]). *Let a composite $N = pq$, for primes p, q, and let an integer $\zeta \geq 1$. The* **ζ-Decision Composite Residuosity** *(ζ-DCR) problem is to distinguish between the distributions $D_0 := \{z = z_0^{N^\zeta} \bmod N^{\zeta+1} \mid z_0 \hookleftarrow U(\mathbb{Z}_N^*)\}$ and $D_1 := \{z \hookleftarrow U(\mathbb{Z}_{N^{\zeta+1}}^*)\}$.*

For each $\zeta > 0$, the ζ-DCR assumption was shown to be equivalent to the original 1-DCR assumption [20]. Damgård and Jurik [20] initially gave their security proof using a recursive argument (rather than a sequence of hybrid experiments) that loses a factor 2 at each step, thus incurring an apparent security loss 2^ζ. However, the semantic security of their scheme under the 1-DCR assumption for any polynomial ζ is a well-known result. The proof of Lemma 9 is perhaps folklore, for instance, a full proof can be found in [23].

Lemma 9. *Let $\zeta = \mathsf{poly}(\lambda)$. Then ζ-DCR is equivalent to 1-DCR with a security loss at most ζ.*

3 Cumulatively All-Lossy-But-One Trapdoor Functions

We now describe two constructions of CALBO-TDFs from standard assumptions. So far, the only known CALBO-TDFs construction was proposed by Chakraborty *et al.* [19] and relies on puncturable PRFs, cumulatively-lossy-trapdoor functions (C-LTDFs) and indistinguishability obfuscation (iO). This construction relies on iO to obfuscate a program, which first compares a given input tag with the hardcoded injective tag and outputs the hardcoded injective evaluation key if the comparison goes through. Otherwise, it generates a fresh lossy key. All auxiliary key generations in the program are realized using the algorithms from the underlying C-LTDF. The obfuscated program is described in the evaluation key for the CALBO-TDF. An evaluation on a pair of tag and input proceeds by first calling the obfuscated program on the given tag to get a C-LTDF key, then use the evaluation of the C-LTDF on the received key and the given input. The obfuscated program uses a puncturable PRF, which receives the given tag as input, to generate randomness needed for producing a fresh lossy key. Our constructions are much simpler and require neither CPRFs nor iO. They thus drastically improve the efficiency compared to [19].

We construct CALBO-TDFs from the LWE and DCR assumptions. Our LWE-based CALBO-TDFs only achieves the relaxed variant of cumulative lossiness while our DCR-based construction achieves the full notion. The fact that we

have to relax the cumulative lossiness in the LWE case seems intrinsic due to the noise that appears in the LWE samples. We remark that Chakraborty *et al.* faced a similar problem when constructing C-LTDFs from LWE as well as when boostrapping C-LTDFs to CALBO-TDFs using iO in [19].

3.1 Relaxed CALBO-TDFs from LWE

In this section, we describe our construction of CALBO-TDFs from LWE. It is inspired from the PRF from [45], which can be seen as a CALBO-TDFs without inversion. We extend ideas from [45] to achieve inversion via trapdoors.

Let λ be a security parameter and let $\ell = \ell(\lambda), n = n(\lambda), m = m(\lambda), q = q(\lambda), p = p(\lambda), t = t(\lambda), \beta = \beta(\lambda)$ be natural numbers and $\chi = \chi(\lambda) = D_{\mathbb{Z},\alpha q}$ be an LWE error distribution. We denote $n' = \lceil n \log q \rceil$. The tag space is $\mathcal{T}_\lambda = \{0,1\}^t$. Our construction now goes as follows:

$\mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$: Sample $(\mathbf{A}, \mathbf{B}) \leftarrow \mathsf{Lossy}(1^m, 1^n, 1^\ell, q, \chi)$, then set the evaluation key $\mathsf{ek} := \left(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{B} \in \mathbb{Z}_q^{\ell \times m}, \{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i=1}^t\right)$ where

$$\mathbf{A}_{i,b} = \mathbf{A} \cdot \mathbf{R}_{i,b} + \delta_{b,\mathsf{tag}_i^*} \cdot \mathbf{G} \in \mathbb{Z}_q^{n \times n'} \quad \forall i \in [t],\, b \in \{0,1\}$$

for $\mathbf{R}_{i,b} \leftarrow U(\{0,1\}^{m \times n'})$, tag_i^* denotes the i-th bit of tag^*, and $\delta_{b,\mathsf{tag}_i^*} = (b \stackrel{?}{=} \mathsf{tag}_i^*)$. Afterwards, set the trapdoor key $\mathsf{tk} := \{\mathbf{R}_{i,b}\}_{i \in [t], b \in \{0,1\}}$ and output $(\mathsf{ek}, \mathsf{tk})$.

$\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, \mathbf{x} \in [0,\beta]^n)$: Let $C_{\mathsf{tag}} : \{0,1\}^t \to \{0,1\}$ be the circuit $C_{\mathsf{tag}}(\mathsf{tag}') = \prod_{i=1}^t \delta_{\mathsf{tag}_i,\mathsf{tag}_i'}$ and $\delta_{\mathsf{tag}_i,\mathsf{tag}_i'} = 1$ if and only if $\mathsf{tag}_i = \mathsf{tag}_i'$. Parse the evaluation key $\mathsf{ek} = (\mathbf{A}, \mathbf{B}, \{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i=1}^t)$ and perform the homomorphic evaluation

$$\begin{aligned}\mathbf{A}(\mathsf{tag}) &:= \mathsf{FHEval}\left(C_{\mathsf{tag}}, \left(\mathbf{A}_{i,\mathsf{tag}_i}\right)_{i=1}^t\right) = \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}} + C_{\mathsf{tag}}(\mathsf{tag}^*) \cdot \mathbf{G} \\ &= \begin{cases} \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}} + \mathbf{G} \text{ if } \mathsf{tag} = \mathsf{tag}^* \\ \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}} \text{ otherwise} \end{cases} \in \mathbb{Z}_q^{n \times n'} \end{aligned} \tag{3}$$

where the procedure FHEval is specified by:

$$\mathsf{FHEval}\left(C_{\mathsf{tag}}, \left(\mathbf{A}_{i,\mathsf{tag}_i}\right)_{i=1}^t\right) := \mathbf{A}_{1,\mathsf{tag}_1} \cdot \mathbf{G}^{-1}\left(\mathbf{A}_{2,\mathsf{tag}_2} \cdot \mathbf{G}^{-1}\left(\cdots \mathbf{G}^{-1}(\mathbf{A}_{t,\mathsf{tag}_t}) \cdots\right)\right)$$

and $\mathbf{R}_{\mathsf{tag}} \in \mathbb{Z}^{m \times n'}$. Finally, compute and output $\left\lfloor \mathbf{x}^\top \cdot [\mathbf{A} \,|\, \mathbf{A}(\mathsf{tag})] \right\rceil_p$.

$\mathsf{Invert}(\mathsf{tk}, \mathsf{tag}^*, \mathbf{y} \in \mathbb{Z}_p^{m+n'})$: Parse the trapdoor key $\mathsf{tk} = \{\mathbf{R}_{i,b}\}_{i \in [t], b \in \{0,1\}}$ then compute

$$\mathsf{FHEval}\left(C_{\mathsf{tag}^*}, \left(\mathbf{A}_{i,\mathsf{tag}_i^*}\right)_{i=1}^t\right) = \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}^*} + \mathbf{G} \ ,$$

and following Lemma 2, obtain $\mathsf{EvalPriv}\left(C_{\mathsf{tag}^*}, ((\mathbf{R}_{i,\mathsf{tag}_i^*}, \mathsf{tag}_i^*))_{i \in [t]}\right) = \mathbf{R}_{\mathsf{tag}^*}$. Afterwards, compute $\mathbf{x} \leftarrow \mathsf{LWRInvert}([\mathbf{A} \,|\, \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}^*} + \mathbf{G}], -\mathbf{R}_{\mathsf{tag}^*}, \mathbf{y})$ as per Lemma 5 and output $\mathbf{x}$.

The way we carry out the homomorphic computation FHEval involved in equation (3) is not unique. Roughly speaking, at each step of the homomorphic evaluation of C_{tag}, we "decompose" the result from the previous step using $\mathbf{G}^{-1}$ (the decomposed entries become binary) before multiplying so as to obtain a ciphertext for the AND gate's output. This gives the smallest possible increase in the error term of the resulting homomorphic ciphertext, following Lemma 2. Different approaches for computing FHEval will lead to different error increases. Indeed, we homomorphically evaluate the circuit C_{tag} in the most possible "sequential" way, which is inspired by [5], and always multiply ciphertexts whose noise terms are not too large. A less sequential computation will work, but at the cost of a larger modulus, which then becomes exponential not only in the security paramter but also in the depth of C_{tag}.

Parameter selection. Let λ be the security parameter. First of all, we set the bound $\beta = 1$ for the entries of inputs, which gives a domain $\{0,1\}^n$. We set the tag length $t = \log \lambda$, which means the circuits to be homomorphically evaluated have depths bounded by $t - 1 \leq \log \lambda$. By Lemma 6, we must choose ℓ such that $n \geq 2\ell \log q$. In addition, for the trapdoor mechanism to work, Lemma 5 requires that $m \geq n \log q + 2 \log \frac{n'}{2\delta}$ and $p \geq O(\sqrt{(m+n')n'})$, where $n' = \lceil n \log q \rceil$ and $\delta = \mathsf{negl}(n)$.

We will need $m \geq n \log q + \omega(\log n)$ in order to apply Lemma 3. Moreover, for the $\mathsf{LWE}_{q,m,n-1,\alpha}$ problem to be hard, it is necessary that $q \leq 2^{n^\epsilon} < 2^n$ and $2\sqrt{n}/q \leq \alpha \leq n \cdot 2^{-n^\epsilon}$, for some $0 < \epsilon < 1$. We refer to [17, Corollary 3.2] for more details on these bounds for q and α. Similarly, we also need to ensure that the $\mathsf{LWE}_{q,m,\ell,\alpha}$ problem is hard. Last but not least, we need $q/p > 2^\lambda$ for the rounding operation to anihilate the noise term, following Lemma 7. Concretely, let $0 < \epsilon < 1$ be a constant and $d \geq 1$, we set up the parameters as follows:

$$
\begin{aligned}
&n = \Theta(\lambda^d); \quad n' = n \log q = \Theta\left(\lambda^{d+d\epsilon}\right); \quad \beta = 1; \quad t = \log \lambda; \quad q = 2^{n^\epsilon} = \Theta\left(2^{\lambda^{d\epsilon}}\right); \\
&\alpha = n \cdot 2^{-n^\epsilon} = \Theta\left(\lambda^d \cdot 2^{-\lambda^{d\epsilon}}\right); \qquad m = 2\lambda + \lceil n \log q \rceil = \Theta\left(\lambda^{d+d\epsilon}\right); \\
&\ell = \frac{n}{2 \log q} = \Theta\left(\lambda^{d-d\epsilon}\right); \qquad p = \Theta\left(\sqrt{(m+n')n'}\right) = \Theta\left(\lambda^{d+d\epsilon}\right) \quad .
\end{aligned}
$$

Theorem 1. *Let $\lambda \in \mathbb{N}$ be a security parameter. Under the $\mathsf{LWE}_{q,m,\ell,\alpha}$ and $\mathsf{LWE}_{q,m,n-1,\alpha}$ assumptions, the above construction $(\mathsf{Sample}, \mathsf{Eval}, \mathsf{Invert})$ is a relaxed $(n, n - \ell \log q)$-cumulatively-all-lossy-but-one trapdoor function family with tag space $\mathcal{T}_t = \{0,1\}^t$.*

Proof. We now prove each of the required properties.

Injectivity. The correctness of FHEval and $\mathsf{EvalPriv}$ in Invert follows Lemma 2. It is straightforward to see that $-\mathbf{R}_{\mathsf{tag}^*}$ is a $\mathbf{G}$-trapdoor for the matrix $\mathbf{A}' := [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}^*} + \mathbf{G}]$. Hence, given as inputs $\mathbf{y} = \mathsf{Eval}(\mathsf{ek}, \mathsf{tag}^*, \mathbf{x}) = \left\lfloor \mathbf{x}^\top \cdot \mathbf{A}' \right\rceil_p$ and the pair $(\mathbf{A}', -\mathbf{R}_{\mathsf{tag}^*})$, the algorithm $\mathsf{LWRInvert}$ will be able to compute back $\mathbf{x}$ as per Lemma 5.

Indistinguishability. Let $\mathsf{tag}_0^*, \mathsf{tag}_1^* \in \{0,1\}^t$ and $(\mathsf{ek}_b, \mathsf{tk}_b) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}_b^*)$ for $b \in \{0,1\}$. We want to prove that ek_0 and ek_1 are indistinguishable. Let $b \in \{0,1\}$. The evaluation key ek_b is parsed as

$$\mathsf{ek}_b = \left(\mathbf{A}^{(b)} \in \mathbb{Z}_q^{n \times m},\ \mathbf{B}^{(b)} \in \mathbb{Z}_q^{\ell \times m},\ \{\mathbf{A}_{i,0}^{(b)}, \mathbf{A}_{i,1}^{(b)}\}_{i=1}^t\right)$$

where $(\mathbf{A}^{(b)}, \mathbf{B}^{(b)}) \leftarrow \mathsf{Lossy}(1^m, 1^n, 1^\ell, q, \chi)$ and $\mathbf{B}^{(b)} \sim U(\mathbb{Z}_q^{\ell \times m})$, $\mathbf{A}_{i,b'}^{(b)}$ are encryptions of $\delta_{b', \mathsf{tag}_{b,i}^*} \in \{0,1\}$ for $i \in [t]$ and $\mathsf{tag}_{b,i}^*$ is the i-th bit of tag_b^*.

Similarly to the proof of semantic security for the GSW encryption scheme [32], we first notice that $\mathbf{A}^{(b)}$ is indistinguishable from a uniformly random matrix $\tilde{\mathbf{A}}^{(b)}$ in $\mathbb{Z}_q^{n \times m}$ thanks to Lemma 3 and the parameter choice $m \geq n \log q + 2\lambda$. Hence, changing $\mathbf{A}^{(b)}$ to $\tilde{\mathbf{A}}^{(b)}$ is computationally indistinguishable under LWE.

We then apply Lemma 1 for the family of universal hash functions $\mathcal{H} = \{h_{\mathbf{A}} : \mathbb{Z}_q^n \to \mathbb{Z}_q^m\}$ where $h_{\mathbf{A}}(\mathbf{x}) \coloneqq \mathbf{x}^\top \cdot \mathbf{A}$ is indexed by $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and q is prime. Therefore, it holds that $\left(\tilde{\mathbf{A}}^{(b)} \mathbf{R}_{i, \mathsf{tag}_{b,i}^*}^{(b)}\right)_{i \in [t]}$ is statistically close to a t-tuple of independent uniformly random matrices. As a result, for all i, the pair $(\tilde{\mathbf{A}}_{i,0}^{(b)}, \tilde{\mathbf{A}}_{i,1}^{(b)})$, where $\tilde{\mathbf{A}}_{i,b'}^{(b)} \coloneqq \tilde{\mathbf{A}}^{(b)} \mathbf{R}_{i, \mathsf{tag}_{b,i}^*}^{(b)} + \delta_{b', \mathsf{tag}_{b,i}^*} \cdot \mathbf{G}$ for $b' \in \{0,1\}$, is statistically close to a pair of uniformly random matrices. In the end, for $b \in \{0,1\}$, ek_b is computationally indistinguishable from $\tilde{\mathsf{ek}}_b$ whose components are sampled uniformly at random in the corresponding domain and the indistinguishability is concluded.

Relaxed cumulative lossiness. Let $\mathsf{tag}^* \in \mathcal{T}_t$, $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$, and fix an input $\mathbf{x} \in [0, \beta]^n = \{0,1\}^n$ by the parameter choice $\beta = 1$. For every $\mathsf{tag} \in \mathcal{T}_t$ such that $\mathsf{tag} \neq \mathsf{tag}^*$, we need to describe two functions $\mathsf{compress}_{\mathsf{ek}}$ and $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}$ such that

$$\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, \mathbf{x}) = \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(\mathbf{x}))$$

except for a negligible probability over the choices of $\mathsf{tag} \neq \mathsf{tag}^*$.

The function $\mathsf{compress}_{\mathsf{ek}}(\mathbf{x} \in \{0,1\}^n)$ is described as follows:

1. Parse ek as $\mathsf{ek} \coloneqq (\mathbf{A}, \mathbf{B}, \{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i=1}^t)$ then use $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}_q^{\ell \times m}$ to recover (inefficiently) $\mathbf{D} \in \mathbb{Z}_q^{\ell \times n}$ and $\mathbf{E} \in \mathbb{Z}^{n \times m}$. This is essentially inverting an LWE function $(\mathbf{D}, \mathbf{E}) \to \mathbf{D}^\top \mathbf{B} + \mathbf{E}$ for the matrix $\mathbf{B}$.
2. Compute and output $\mathbf{D} \cdot \mathbf{x} \in \mathbb{Z}_q^\ell$.

Let $\mathbf{y} \in \mathbb{Z}_q^\ell$ and $\mathsf{tag} \in \mathcal{T}_t$ such that $\mathsf{tag} \neq \mathsf{tag}^*$. The function $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathbf{y})$ is described as follows:

1. Parse the ek as $\mathsf{ek} \coloneqq (\mathbf{A}, \mathbf{B}, \{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i=1}^t)$ then use $(\mathbf{A}, \mathbf{B})$ to (inefficiently) recover $\mathbf{D} \in \mathbb{Z}_q^{\ell \times n}$ and $\mathbf{E} \in \mathbb{Z}_q^{n \times m}$. Using $\mathbf{A}$ and $\{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i=1}^t$, compute $\mathbf{A}(\mathsf{tag})$ as in the Eval algorithm, i.e.

$$\begin{aligned}\mathbf{A}(\mathsf{tag}) \coloneqq \mathsf{FHEval}\left(C_{\mathsf{tag}}, \left(\mathbf{A}_{i,\mathsf{tag}_i}\right)_{i=1}^t\right) &= \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}} + C_{\mathsf{tag}}(\mathsf{tag}^*) \cdot \mathbf{G} \\ &\overset{(*)}{=} \mathbf{A} \cdot \mathbf{R}_{\mathsf{tag}} \in \mathbb{Z}_q^{n \times n'}\end{aligned}$$

where the $(*)$ equality comes from the fact that $\mathsf{tag} \neq \mathsf{tag}^*$. We will denote $\mathbf{A}' \coloneqq [\mathbf{A} \mid \mathbf{A}(\mathsf{tag})] = [\mathbf{A} \mid \left(\mathbf{D}^\top \cdot \mathbf{B} + \mathbf{E}\right) \cdot \mathbf{R}_{\mathsf{tag}}] \in \mathbb{Z}_q^{n \times (m+n')}$.
2. Compute (inefficiently) a matrix $\mathbf{F} \in \mathbb{Z}_q^{\ell \times n'}$ such that $\mathbf{F}$ is an LWE secret for $(\mathbf{D}, \mathbf{A}(\mathsf{tag}))$. Specifically, the matrix $\mathbf{F}$ statisfies that $\mathbf{A}(\mathsf{tag}) = \mathbf{D}^\top \cdot \mathbf{F} + \mathbf{E}_{\mathsf{tag}}$ where $\mathbf{E}_{\mathsf{tag}} \in \mathbb{Z}^{n \times n'}$ has bounded entries. The bound will be analyzed below.
3. Compute (inefficiently) an arbitrary but small matrix $\mathbf{R}' \in \mathbb{Z}^{m \times n'}$ such that $\mathbf{B} \cdot \mathbf{R}' = \mathbf{F}$.
4. Compute and return $\left\lfloor [\mathbf{y}^\top \cdot \mathbf{B} \,|\, \mathbf{y}^\top \cdot \mathbf{F}] \right\rceil_p \in \mathbb{Z}_p^{m+n'}$.

Given a fixed input $\mathbf{x} \in \{0,1\}^n$, for $\mathsf{tag} \in \mathcal{T}_t$ and $\mathsf{tag} \neq \mathsf{tag}^*$, we consider

$$\begin{aligned}
\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(\mathbf{x})) &= \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathbf{D} \cdot \mathbf{x}) \\
&= \left\lfloor [(\mathbf{D} \cdot \mathbf{x})^\top \cdot \mathbf{B} \,|\, (\mathbf{D} \cdot \mathbf{x})^\top \cdot \mathbf{F}] \right\rceil_p \\
&= \left\lfloor [(\mathbf{D} \cdot \mathbf{x})^\top \cdot \mathbf{B} \,|\, (\mathbf{D} \cdot \mathbf{x})^\top \cdot \mathbf{B} \cdot \mathbf{R}'] \right\rceil_p
\end{aligned}$$

where $\mathbf{B}, \mathbf{D}, \mathbf{R}', \mathbf{F}$ are computed as specified in $\mathsf{compress}_{\mathsf{ek}}$ and $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}$.

To begin with, we analyze the bound of the entries in the error matrix $\mathbf{E}_{\mathsf{tag}}$ so that the matrix $\mathbf{F}$ computed in step 2 of $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}$ is uniquely determined. It suffices to bound the infinity norm of $\mathbf{E} \cdot \mathbf{R}_{\mathsf{tag}}$.

We evaluate homomorphically the ciphertexts $\mathbf{A}_{i,b}$ on a circuit C_{tag} defined as a sequential AND-ing of t bits in tag and has depth $t-1$. Moreover, the matrices $\mathbf{A}_{i,b}$ are obtained using binary $\mathbf{R}_i \in \{0,1\}^{m \times n'}$, for all $i \in [t]$ and $b \in \{0,1\}$. As a corollary of Lemma 2, we have $\|\mathbf{R}_{\mathsf{tag}}\|_\infty \leq n'(n'+1)^t$. With $\mathbf{E} \in \mathbb{Z}_q^{n \times m}$, we also have

$$\|\mathbf{E}\|_\infty = \max_{i \in [n]} \left(\sum_{j=1}^{m} |\mathbf{E}_{i,j}| \right) \leq m\alpha q \ .$$

This implies that $\|\mathbf{E} \cdot \mathbf{R}_{\mathsf{tag}}\|_\infty \leq \|\mathbf{E}\|_\infty \cdot \|\mathbf{R}_{\mathsf{tag}}\|_\infty \leq n'(n'+1)^t \cdot m \cdot \alpha q$. We choose the parameters for $n'(n'+1)^t \cdot m \cdot \alpha q$ to be small enough, for example smaller than $q/4$ given a sufficiently large λ. Thus $\left(\mathbf{D}^\top \cdot \mathbf{B} + \mathbf{E}\right) \cdot \mathbf{R}_{\mathsf{tag}}$ uniquely determines $\mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}}$ as a corollary of Lemma 6. Consequently, the (inefficient) step 2 of $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}$ will be able to find the unique $\mathbf{F} = \mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}}$. Then, we have $\mathbf{B} \cdot \mathbf{R}' = \mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}}$ and

$$\left\lfloor [(\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} \,|\, (\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} \cdot \mathbf{R}'] \right\rceil_p = \left\lfloor [(\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} \,|\, (\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}}] \right\rceil_p \ .$$

Let us define an event BAD as

$$\begin{aligned}
&\left\lfloor [(\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} \,|\, (\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}}] \right\rceil_p \\
&\quad \neq \left\lfloor [(\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} + \mathbf{x}^\top \mathbf{E} \,|\, (\mathbf{D}\mathbf{x})^\top \cdot \mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}} + \mathbf{x}^\top \cdot \mathbf{E} \cdot \mathbf{R}_{\mathsf{tag}}] \right\rceil_p
\end{aligned}$$

and we observe that the right-hand side is actually $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, \mathbf{x})$. A simple computation gives us $\Pr[\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, \mathbf{x}) = \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(\mathbf{x}))] \geq 1 -$

$\Pr[\mathsf{BAD}]$ where the probabilities are taken over the choices of $\mathsf{tag} \in \mathcal{T}_t$ such that $\mathsf{tag} \neq \mathsf{tag}^*$, for the fixed input $\mathbf{x} \in \{0,1\}^n$. Lemma 10 proves that $\Pr[\mathsf{BAD}]$ is negligible in λ under our parameter selection and the proof is completed. □

Lemma 10. *We have the following bound:*

$$\Pr[\mathsf{BAD}] \leq 2^{t+1} \cdot p \cdot m\alpha \cdot \left(1 + n'(n'+1)^t\right) \ .$$

A proof for Lemma 10 can be found in Appendix B.1.

3.2 CALBO-TDFs from DCR

In this section we give a construction of CALBO-TDF achieving non-relaxed cumulative lossiness from the DCR assumption. We start by recalling the Damgård-Jurik encryption scheme, whose decryption algorithm along with other useful properties are used in our CALBO-TDFs.

Damgård-Jurik encryption. Damgård and Jurik introduced in [20] a generalization of Paillier's cryptosystem based on the ζ-DCR assumption. Given a modulus $N = pq$ such that $\gcd(N, \varphi(N)) = 1$, where p and q are primes, Damgård and Jurik proved that the multiplicative group $\mathbb{Z}^*_{N^{\zeta+1}}$ is isomorphic to the direct product of $\mathbb{Z}_{N^\zeta}$ and $\mathbb{Z}^*_N$:

Theorem 2 ([20], Theorem 1). *For any N satisfying $\gcd(N, \varphi(N)) = 1$ and for $\zeta < \min(p, q)$, the map $\psi_\zeta : \mathbb{Z}_{N^\zeta} \times \mathbb{Z}^*_N \to \mathbb{Z}^*_{N^{\zeta+1}}$ given by $(m, r) \mapsto (1+N)^m r^{N^\zeta} \pmod{N^{\zeta+1}}$ is invertible in polynomial time using $\mathrm{lcm}(p-1, q-1)$.*

The Damgård-Jurik encryption exploits this isomorphic property: a public key is a pair (N, ζ) associated with secret key (p, q) and ψ_ζ is the encryption function (where r plays the role of randomness), that can be inverted (decryption) given (p, q). Semantic security is easily proven under the ζ-DCR assumption [20, Theorem 2]. We recall the details of their scheme in Appendix A.2.

We are now ready to describe our construction of CALBO-TDFs from the ζ-DCR assumption. We remark that the domain is currently index-dependent, i.e. inputs are taken in $\mathbb{Z}^*_{N^{\zeta+1}}$ where N and ζ are specified in the evaluation key. The domain can be made index-independent by using $\{0,1\}^n$ for some bitlength n in the same way Freeman *et al.* have done in [26], e.g. we can choose any $n \in \mathbb{N}$ such that $n < \min(\log p, \log q)$.

Sample$(1^\lambda, \mathsf{tag}^*)$**:** Given $\mathsf{tag}^* \in \mathcal{T}_t = \{0,1\}^t$, generate an evaluation key

$$\mathsf{ek} := \left(N,\ \zeta,\ \{g_{i,0}, g_{i,1}\ \in \mathbb{Z}^*_{N^{\zeta+1}}\ \}_{i=1}^t\right) \ ,$$

consisting of the following components:

- A modulus $N = pq$ such that $p, q > 2^{l(\lambda)}$ and $\gcd(N, \varphi(N)) = 1$, where $l : \mathbb{N} \to \mathbb{N}$ is a polynomial dictating the bitlength of p and q as a function of λ, and an integer $\zeta > t$.

– Elements $g_{i,0}, g_{i,1} \in \mathbb{Z}^*_{N^{\zeta+1}}$ which are generated as

$$g_{i,b} = (1+N)^{\delta_{b,\mathsf{tag}^*_i}} \cdot \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1} \qquad \forall (i,b) \in [t] \times \{0,1\} \ ,$$

where $\alpha_{i,b} \hookleftarrow U(\mathbb{Z}^*_N)$ for each $i \in [t]$, $b \in \{0,1\}$, tag^*_i denotes the i-th bit of tag^*, and $\delta_{b,\mathsf{tag}^*_i} = (b \stackrel{?}{=} \mathsf{tag}^*_i)$. We note that $g_{i,b}$ is a Damgård-Jurik ciphertext of $\delta_{b,\mathsf{tag}^*_i}$.

Output ek and $\mathsf{tk} = (p,q)$.

Eval$(\mathsf{ek}, \mathsf{tag}, x)$**:** Given an input $x \in \mathbb{Z}_{N^{\zeta+1}}$ and $\mathsf{tag} \in \mathcal{T}_t = \{0,1\}^t$, let $x_0 = x$. Find $(y_0, z_0) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_N$ such that $x_0 = y_0 \cdot N + z_0$. If $\gcd(z_0, N) > 1$, output 0. Otherwise, for $i = 1$ to t, do the following:

1. Parse $x_{i-1} \in \mathbb{Z}_{N^{\zeta+1}}$ as a pair of integers $(y_{i-1}, z_{i-1}) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}^*_N$ such that $x_{i-1} = y_{i-1} \cdot N + z_{i-1}$.
2. Compute $x_i = g_{i,\mathsf{tag}_i}^{y_{i-1}} \cdot z_{i-1}^{N^\zeta} \bmod N^{\zeta+1}$.

In the end, output $z = x_t \in \mathbb{Z}^*_{N^{\zeta+1}}$.

Invert$(\mathsf{tk}, \mathsf{tag}, z)$**:** Set $x_t = z$ and find $(y_t, z_t) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_N$ such that $x_t = y_t \cdot N + z_t$. If $\gcd(z_t, N) > 1$, output 0. Otherwise, for $i = t$ down to $i = 1$, conduct the following steps:

1. Using $\mathsf{tk} = (p,q)$, compute the unique pair $(y_{i-1}, z_{i-1}) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}^*_N$ such that
$$x_i = g_{i,\mathsf{tag}_i}^{y_{i-1}} \cdot z_{i-1}^{N^\zeta} \bmod N^{\zeta+1} \ .$$
This is done by first recovering $y_{i-1} = \mathsf{Dec}((p,q), x_i) \in \mathbb{Z}_{N^\zeta}$ using the Damgård-Jurik decryption algorithm for obtaining $z_{i-1} = \left(x_i \cdot g_{i,\mathsf{tag}_i}^{-y_{i-1}} \bmod N^{\zeta+1}\right)^{N^{-\zeta}} \bmod N$. Note that $z_{i-1} \in \mathbb{Z}^*_N$ is well-defined thanks to the isomorphism ψ_ζ^{-1} used in Damgård-Jurik decryption.
2. Let $x_{i-1} = y_{i-1} \cdot N + z_{i-1}$. Output x_0 when $i = 1$.

The check $\gcd(z_0, N) = 1$ in **Eval** implies that, as long as factoring is hard, it is infeasible to find non-invertible inputs, i.e. $x = y_0 \cdot N + z_0 \in \mathbb{Z}_{N^\zeta+1}$ such that $\gcd(z_0, N) > 1$ for $(y_0, z_0) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_N$. Moreover, the fraction of non-invertible inputs is bounded by $N^\zeta \cdot (p+q)/N^{\zeta+1} = (p+q)/N$, which is negligible. We now prove that the above construction is a CALBO-TDF assuming ζ-DCR holds.

Theorem 3. *Let $\lambda \in \mathbb{N}$ is a security parameter. Let $\zeta = \zeta(\lambda), l = l(\lambda), t = t(\lambda)$ be functions in λ such that $\zeta > t$. Assuming the ζ-DCR assumption, the triplet $(\mathsf{Sample}, \mathsf{Eval}, \mathsf{Invert})$ is a $((\zeta+1)\log N, (\zeta+1)\log N - t\log N - 1)$-cumulatively-all-lossy-but-one trapdoor function family with tag space $\mathcal{T}_t = \{0,1\}^t$.*

Proof. We prove injectivity, indistinguishability and cumulative lossiness properties as defined in Section 2.1. Let $\lambda \in \mathbb{N}$ be a security parameter and $\zeta = \zeta(\lambda), l = l(\lambda), t = t(\lambda)$ be polynomials in λ such that $\zeta > t$. Let $\mathsf{tag}^* \in \mathcal{T}_t$ be the injective tag and $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$.

We first justify why we only need to check $\gcd(z_0, N) = 1$ and can be sure that if it holds, $\gcd(z_i, N) = 1$ for all $i \geq 1$. Indeed, let $i \in [t]$. By construction

$x_i = y_i \cdot N + z_i$ for $(y_i, z_i) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_N$. Suppose $z_0 \in \mathbb{Z}_N^*$, we verify the claim by induction. Indeed $x_1 = \psi_\zeta(y_0, z_0) \in \mathbb{Z}_{N^{\zeta+1}}^*$. Hence $\gcd(z_1, N) = \gcd(z_1 + y_1 \cdot N, N) = \gcd(x_1, N) = 1$. For the inductive step, suppose $z_{i-1} \in \mathbb{Z}_N^*$, then $x_i = \psi_\zeta(y_{i-1}, z_{i-1}) \in \mathbb{Z}_{N^{\zeta+1}}^*$. By the same argument, we have $\gcd(z_i, N) = \gcd(z_i + y_i \cdot N, N) = \gcd(x_i, N) = 1$.

Injectivity. Let $\mathsf{tag}^* \in \{0,1\}^t$ be an injective tag. We consider two cases for invertibility of $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}^*, x)$ given the trapdoor tk of tag^*. If $x \in \mathbb{Z}_{N^{\zeta+1}} \setminus \mathbb{Z}_{N^{\zeta+1}}^*$, equivalently by Theorem 2 it holds that $x = y_0 \cdot N + z_0$ and $\gcd(z_0, N) > 1$, then $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}^*, x) = 0$ by construction and cannot be inverted using tk. The fraction of such inputs in $\mathbb{Z}_{N^{\zeta+1}}$ is

$$\frac{N^\zeta \cdot (N - \varphi(N))}{N^{\zeta+1}} = \frac{p + q - 1}{N} \quad .$$

which is negligible in λ.

Otherwise, suppose that $x \in \mathbb{Z}_{N^{\zeta+1}}^*$. By the correctness of Damgård-Jurik decryption algorithm and Theorem 2, for each $i = t$ down to 1, step 1 in Invert correctly recovers $y_{i-1} \in \mathbb{Z}_{N^\zeta}$ and $z_{i-1} \in \mathbb{Z}_N^*$ such that $x_{i-1} = y_{i-1} \cdot N + z_{i-1}$, where x_{i-1} is used at step $i - 1$ in $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}^*, x)$. Inductively, $x_0 = y_0 \cdot N + z_0$ is recovered correctly. In summary, $\mathsf{Invert}(\mathsf{tk}, \mathsf{tag}^*, \mathsf{Eval}(\mathsf{ek}, \mathsf{tag}^*, x)) = x$ for an overwhelming fraction of the domain $\mathbb{Z}_{N^{\zeta+1}}$ and the injectivity is concluded.

Indistinguishability. Let $\mathsf{tag}_0^*, \mathsf{tag}_1^* \in \{0,1\}^t$ and $(\mathsf{ek}_b, \mathsf{tk}_b) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}_b^*)$ for $b \in \{0,1\}$. We want to prove that ek_0 and ek_1 are indistinguishable. Let $b \in \{0,1\}$. The evaluation key ek_b is parsed as

$$\mathsf{ek}_b = \left(N,\ \zeta,\ \{g_{i,0}^{(b)}, g_{i,1}^{(b)}\ \in \mathbb{Z}_{N^{\zeta+1}}^*\ \}_{i=1}^t\right)$$

where $g_{i,b'}^{(b)}$ is a Damgård-Jurik encryption of $\delta_{b', \mathsf{tag}_{b,i}^*}$ for $i \in [t]$ and $b' \in \{0,1\}$, respectively and $\mathsf{tag}_{b,i}^*$ is the i-th bit of tag_b^*. The indistinguishability readily follows the semantic security of the Damgård-Jurik encryption scheme under a standard hybrid argument.

Cumulative lossiness. For $(\mathsf{ek}, \mathsf{tk}) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{tag}^*)$ and $\mathsf{tag} \in \{0,1\}^t$ such that $\mathsf{tag} \neq \mathsf{tag}^*$, we want to describe two (possibly inefficient) functions $\mathsf{compress}_{\mathsf{ek}}$ and $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}$ satisfying $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x) = \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(x))$ for all $x \in \mathbb{Z}_{N^{\zeta+1}}$. The function $\mathsf{compress}_{\mathsf{ek}}(x)$ for $x \in \mathbb{Z}_{N^{\zeta+1}}$ is as follows:

1. Parse the evaluation key as

   $$\mathsf{ek}_b = \left(N,\ \zeta,\ \{g_{i,0}, g_{i,1}\ \in \mathbb{Z}_{N^{\zeta+1}}^*\ \}_{i=1}^t\right)$$

   and (inefficiently) factor $N = pq$.
2. Initialize a list List to empty. Compute $(y, z) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_N$ such that $x = y \cdot N + z$. If $\gcd(z, N) > 1$ then add 0 to List and output List.

3. Otherwise, having p, q, for all $(i, b) \in [t] \times \{0,1\}$, use the Damgård-Jurik decryption $\mathsf{Dec}((p,q), g_{i,b}) = \delta_{b,\mathsf{tag}^*_i}$ and in the end obtain $\mathsf{tag}^* \in \{0,1\}^t$. Moreover, use the isomorphism ψ_ζ^{-1} from Theorem 2 to also recover all the $\alpha_{i,b} \in \mathbb{Z}_N^*$ while knowing $g_{i,b} \in \mathbb{Z}_{N^{\zeta+1}}^*$ and $\delta_{b,\mathsf{tag}^*_i} \in \mathbb{Z}_{N^\zeta}$.
4. For $i = 1$ to t, define

$$\mathsf{sibling}_i \coloneqq \mathsf{tag}^*_{[1..(i-1)]} \,\|\, (1 - \mathsf{tag}^*_i)$$

where $\mathsf{tag}^*_{[1..(i-1)]}$ denotes the first $i-1$ bits of tag^*.
5. For $j = 1$ to t, perform the following:
   - Let $x_0 = x$ and find (y_0, z_0) such that $x_0 = y_0 \cdot N + z_0$.
   - For $k = 1$ to $j-1$, compute

$$x_k = g_{k,\mathsf{sibling}_j[k]}^{y_{k-1}} \cdot z_{k-1}^{N^\zeta} \pmod{N^{\zeta+1}}$$

   where $\mathsf{sibling}_j[k]$ is the k-th bit of $\mathsf{sibling}_j$.
   - Let $b = \mathsf{sibling}_j[j]$. Compute (y_{j-1}, z_{j-1}) such that $x_{j-1} = y_{j-1} \cdot N + z_{j-1}$ and add

$$(\alpha_{j-1,b}^{y_{j-1}} \cdot z_{j-1})^{N^\zeta} \pmod{N^{\zeta+1}} \in \mathbb{Z}_N$$

   to List.
6. Output $\mathsf{List} \in \mathbb{Z}_N^t$.

Given $\mathsf{tag} \neq \mathsf{tag}^*$ and a $\mathsf{List} \in \mathbb{Z}_N^t$, the function $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{List})$ is given below:

1. Parse the evaluation key as $\mathsf{ek}_b = \big(N,\ \zeta,\ \{g_{i,0}, g_{i,1}\ \in \mathbb{Z}_{N^{\zeta+1}}^*\ \}_{i=1}^t\big)$ and (inefficiently) factor $N = pq$.
2. If List contains only one element 0, output 0.
3. Otherwise, having p, q, for all $(i, b) \in [t] \times \{0,1\}$, use the Damgård-Jurik decryption $\mathsf{Dec}((p,q), g_{i,b}) = \delta_{b,\mathsf{tag}^*_i}$ and in the end obtain $\mathsf{tag}^* \in \{0,1\}^t$.
4. Compute $i = \min_{j \in [t]}(\mathsf{tag}_j \neq \mathsf{tag}^*_j)$. It holds that $1 \leq i \leq t$ is well-defined because $\mathsf{tag} \neq \mathsf{tag}^*$.
5. Let $x_i = \mathsf{List}[i]$. For $k = i+1$ to t, conduct the following:
   - Compute (y_{k-1}, z_{k-1}) satisfying $x_{k-1} = y_{k-1} \cdot N + z_{k-1}$.
   - Compute

$$x_k = g_{k,\mathsf{tag}_k}^{y_{k-1}} \cdot z_{k-1}^{N^\zeta} \pmod{N^{\zeta+1}} \ .$$

6. Output $x_t \in \mathbb{Z}_{N^{\zeta+1}}^*$.

Relating to cumulative lossiness, we evaluate $|\{\mathsf{compress}_{\mathsf{ek}}(x) : x \in \mathbb{Z}_{N^{\zeta+1}}\}|$. By construction, for $x \in \mathbb{Z}_{N^{\zeta+1}}^*$, the output of $\mathsf{compress}_{\mathsf{ek}}(x)$ is a list of t elements in $\mathbb{Z}_N$. If $x \in \mathbb{Z}_{N^{\zeta+1}} \setminus \mathbb{Z}_{N^{\zeta+1}}^*$, $\mathsf{compress}_{\mathsf{ek}}(x)$ outputs a list of one single element, namely 0. We then have the bound

$$|\{\mathsf{compress}_{\mathsf{ek}}(x) : x \in \mathbb{Z}_{N^{\zeta+1}}\}| = N^t + 1 \leq 2 \cdot N^t \ .$$

We want to prove that $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x) = \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(x))$ for all $x \in \mathbb{Z}_{N^{\zeta+1}}$ and $\mathsf{tag} \neq \mathsf{tag}^*$. If $x \in \mathbb{Z}_{N^{\zeta+1}} \setminus \mathbb{Z}^*_{N^{\zeta+1}}$, then $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x) = 0$ by construction. Moreover, we have $x = y \cdot N + z$ for $(y, z) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_N$ such that $\gcd(z, N) > 1$. Thus, $\mathsf{compress}_{\mathsf{ek}}(x)$ outputs List containing only 0 and step 2 in $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{List})$ recovers exactly 0. Otherwise, suppose $x \in \mathbb{Z}^*_{N^{\zeta+1}}$. Our main observation is that for $i = \min_{j\in[t]}(\mathsf{tag}_j \neq \mathsf{tag}^*_j)$, the value x_i will uniquely determine x_t, by the fact that ψ_ζ is an isomorphism from Theorem 2. Moreover, because $\mathsf{tag}_i \neq \mathsf{tag}^*_i$ and $\mathsf{tag}_k = \mathsf{tag}^*_k$ for all $k < i$, we have

$$x_i = (\alpha_{i-1,b}^{y_{i-1}} \cdot z_{i-1})^{N^\zeta} \pmod{N^{\zeta+1}}$$

and the sequence $(x_0, \ldots, x_{i-1} = y_{i-1} \cdot N + z_{i-1})$ stays the same as if the input tag is tag^*. By definition of $\mathsf{sibling}_i$, it is easily verified that the loop 5 in $\mathsf{compress}_{\mathsf{ek}}$ constructs List such that $\mathsf{List}[i] = x_i$ and $i = \min_{j\in[t]}(\mathsf{tag}_j \neq \mathsf{tag}^*_j)$. Finally, the loop 5 in $\mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{List})$ performs exactly the same computation as $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x)$ would do, starting from i. Hence, the equality $\mathsf{Eval}(\mathsf{ek}, \mathsf{tag}, x) = \mathsf{expand}_{\mathsf{ek},\mathsf{tag}}(\mathsf{compress}_{\mathsf{ek}}(x))$ is justified. □

Remark 1 The domain is $\mathbb{Z}_{N^{\zeta+1}}$ and its size is $\log(N^{\zeta+1}) = (\zeta+1)\log N$. Moreover, by setting the tag length $t = O(\lambda)$ and the exponent $\zeta = \omega(\lambda)$ so that our CALBO-TDFs can be used for the applications to randomness extractors in [24, Corollary 5.12], the lossiness rate of the above construction becomes

$$\frac{(\zeta+1)\log N - \log(2 \cdot N^t)}{(\zeta+1)\log N} = 1 - \frac{t}{\zeta+1} - \frac{1}{(\zeta+1)\log N} = 1 - o(1)$$

and is indeed better than what the LWE-based CALBO-TDF achieves, which is $1 - \Theta(1)$ by the parameter choices.

Acknowledgements This work was supported in part by the French ANR Project ANR-19-CE39-0011 PRESTO and in part by the French ANR ALAMBIC project (ANR-16-CE39-0006).

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: EUROCRYPT 2010. https://doi.org/10.1007/978-3-642-13190-5_28
2. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited - new reduction, properties and applications. In: CRYPTO 2013, Part I. https://doi.org/10.1007/978-3-642-40041-4_4
3. Auerbach, B., Kiltz, E., Poettering, B., Schoenen, S.: Lossy trapdoor permutations with improved lossiness. In: CT-RSA (2019)
4. Banerjee, A., Peikert, C., Rosen, A.: Pseudo-random functions and lattices. In: Eurocrypt (2012)
5. Banerjee, A., Peikert, C.: New and improved key-homomorphic pseudorandom functions. In: CRYPTO 2014, Part I. https://doi.org/10.1007/978-3-662-44371-2_20

6. 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: Asiacrypt (2009)
7. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: EUROCRYPT 2009. https://doi.org/10.1007/978-3-642-01001-9_1
8. Bellare, M., Kiltz, E., Peikert, C., Waters, B.: Identity-based (lossy) trapdoor functions and applications. In: EUROCRYPT 2012. https://doi.org/10.1007/978-3-642-29011-4_15
9. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: CRYPTO 2008. https://doi.org/10.1007/978-3-540-85174-5_19
10. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Crypto (2004)
11. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key-homomorphic PRFs and their applications. In: Crypto (2013)
12. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: EUROCRYPT 2014. https://doi.org/10.1007/978-3-642-55220-5_30
13. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: ASIACRYPT 2013, Part II. https://doi.org/10.1007/978-3-642-42045-0_15
14. Boyen, X., Li, Q.: All-but-many lossy trapdoor functions from lattices and applications. In: CRYPTO 2017, Part III. https://doi.org/10.1007/978-3-319-63697-9_11
15. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: PKC 2014. https://doi.org/10.1007/978-3-642-54631-0_29
16. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: ITCS (2014)
17. Brakerski, Z., Vaikuntanathan, V.: Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In: CRYPTO 2016, Part III. https://doi.org/10.1007/978-3-662-53015-3_13
18. Braverman, M., Hassidim, A., Kalai, Y.T.: Leaky pseudo-entropy functions. In: ICS 2011
19. Chakraborty, S., Prabhakaran, M., Wichs, D.: Witness maps and applications. In: PKC 2020, Part I. https://doi.org/10.1007/978-3-030-45374-9_8
20. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: PKC 2001. https://doi.org/10.1007/3-540-44586-2_9
21. Damgård, I., Nielsen, J.B.: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: CRYPTO 2002. https://doi.org/10.1007/3-540-45708-9_37
22. Damgård, I., Nielsen, J.B.: Universally composable efficient multiparty computation from threshold homomorphic encryption. In: CRYPTO 2003. https://doi.org/10.1007/978-3-540-45146-4_15
23. Devevey, J., Libert, B., Peters, T.: Rational Modular Encoding in the DCR Setting: Non-Interactive Range Proofs and Paillier-Based Naor-Yung in the Standard Model. In: PKC 2022. Yokohama (virtual event), Japan (Mar 2022), https://hal.inria.fr/hal-03807457

24. Dodis, Y., Vaikuntanathan, V., Wichs, D.: Extracting randomness from extractor-dependent sources. In: EUROCRYPT 2020, Part I. https://doi.org/10.1007/978-3-030-45721-1_12
25. Döttling, N., Garg, S., Ishai, Y., Malavolta, G., Mour, T., Ostrovsky, R.: Trapdoor hash functions and their applications. In: Crypto (2019)
26. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. In: PKC 2010. https://doi.org/10.1007/978-3-642-13013-7_17
27. Garg, A., Kalai, Y.T., Khurana, D.: Computational extractors with negligible error in the CRS model. Cryptology ePrint Archive, Report 2019/1116, https://eprint.iacr.org/2019/1116
28. Garg, A., Kalai, Y.T., Khurana, D.: Low error efficient computational extractors in the CRS model. In: EUROCRYPT 2020, Part I. https://doi.org/10.1007/978-3-030-45721-1_14
29. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS. LNCS, Springer (2013)
30. Garg, S., Gay, R., Hajiabadi, M.: New techniques for efficient trapdoor functions and applications. In: EUROCRYPT 2019, Part III. https://doi.org/10.1007/978-3-030-17659-4_2
31. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: 40th ACM STOC. https://doi.org/10.1145/1374376.1374407
32. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: CRYPTO 2013, Part I. https://doi.org/10.1007/978-3-642-40041-4_5
33. Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: ICS 2010
34. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: CRYPTO 2015, Part II. https://doi.org/10.1007/978-3-662-48000-7_25
35. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: 47th ACM STOC. https://doi.org/10.1145/2746539.2746576
36. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing (4) (1999)
37. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Asiacrypt (2011)
38. Hemenway, B., Ostrovsky, R.: Extended-DDH and lossy trapdoor functions. In: PKC (2012)
39. Hofheinz, D.: Circular chosen-ciphertext security with compact ciphertexts. In: Eurocrypt (2013)
40. Hofheinz, D.: All-but-many lossy trapdoor functions. In: EUROCRYPT 2012. https://doi.org/10.1007/978-3-642-29011-4_14
41. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In: ACM CCS 2013. https://doi.org/10.1145/2508859.2516668
42. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under chosen-plaintext attack. In: CRYPTO 2010. https://doi.org/10.1007/978-3-642-14623-7_16

43. Libert, B., Qian, C.: Lossy algebraic filters with short tags. In: PKC (2019)
44. Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: CRYPTO 2017, Part III. https://doi.org/10.1007/978-3-319-63697-9_12
45. Libert, B., Stehlé, D., Titiu, R.: Adaptively secure distributed PRFs from LWE. In: TCC 2018, Part II. https://doi.org/10.1007/978-3-030-03810-6_15
46. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: EUROCRYPT 2012. https://doi.org/10.1007/978-3-642-29011-4_41
47. Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. In: PKC (2010)
48. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT'99. https://doi.org/10.1007/3-540-48910-X_16
49. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: 40th ACM STOC. https://doi.org/10.1145/1374376.1374406
50. Quach, W., Waters, B., Wichs, D.: Targeted lossy functions and applications. In: CRYPTO 2021, Part IV. https://doi.org/10.1007/978-3-030-84259-8_15
51. Raghunathan, A., Segev, G., Vadhan, S.: Deterministic public-key encryption for adaptively-chosen plaintext distributions. In: Eurocrypt (2013)
52. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (2009)
53. Vergnaud, D., Xiao, D.: Public-key encryption with weak randomness: Security against strong chosen distribution attacks. Cryptology ePrint Archive: Report 2013/681 (2013)
54. Wee, H.: Dual projective hashing and its applications - lossy trapdoor functions and more. In: Eurocrypt (2012)

A Additional Definitions

A.1 Homorphic Encryption

Definition 5. *A* homomorphic encryption *scheme* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{FHEval})$ *for a family of circuits* $\{\mathcal{C}_\tau\}_{\tau\in\mathbb{N}}$ *consists of four* ppt *algorithms:*

- $\mathsf{KeyGen}(1^\lambda, 1^\tau)$: *Given as inputs a security parameter* λ *and another parameter* τ*, output a pair* $(\mathsf{pk}, \mathsf{sk})$*.*
- $\mathsf{Enc}(\mathsf{pk}, b)$: *Given a bit* $b \in \{0,1\}$ *and the public key* pk*, output a ciphertext* c*.*
- $\mathsf{Dec}(\mathsf{sk}, c)$: *Given a ciphertext* c *and the secret key* sk*, output a bit* b*.*
- $\mathsf{FHEval}(\mathsf{pk}, C, \mathbf{c})$: *Given the public key* pk*, a circuit* $C \in \mathcal{C}_\tau$*, and a vector* $\mathbf{c}$ *of ciphertexts encrypting input bits of* C*, output a vector* $\mathbf{c}'$ *of ciphertexts.*

Correctness. An HE scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{FHEval})$ is *(perfectly) correct* for a family of circuits $\{C_\tau\}_{\tau\in\mathbb{N}}$ if for all $\lambda, \tau \in \mathbb{N}$ the following conditions hold:

- For any $b \in \{0,1\}$

$$\Pr[b \leftarrow \mathsf{Dec}(\mathsf{sk}, c) \ : \ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\tau);\ c \leftarrow \mathsf{Enc}(\mathsf{pk}, b)] = 1 \ .$$

- For any $C \in \mathcal{C}_\tau$ and t input bits $(b_1, \ldots, b_t) \in \{0,1\}^t$ of C

$$\Pr\left[C(b_1, \ldots, b_t) \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{FHEval}(\mathsf{pk}, C, (c_1, \ldots, c_t))) \ : \ \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\tau) \\ \forall i \in [t],\ c_i \leftarrow \mathsf{Enc}(\mathsf{pk}, b_i) \end{array} \right] = 1 \ .$$

The correctness can be relaxed by allowing the above probabilities to hold except for a negligible error.

Semantic security. An HE scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{FHEval})$ is *semantically secure* if for all $\lambda \in \mathbb{N}$ and for all ppt adversary $\mathcal{A}$, the following advantage is negligible in λ:

$$\mathbf{Adv}^{\mathsf{sem}}_{\mathcal{E},\mathcal{A}}(\lambda) := \left| \Pr\left[1 \leftarrow \mathcal{A}(\mathsf{pk}, c) \ : \ \begin{array}{l} 1^\tau \leftarrow \mathcal{A}(1^\lambda),\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\tau) \\ c \leftarrow \mathsf{Enc}(\mathsf{pk}, 0) \end{array} \right] \right.$$
$$\left. - \Pr\left[1 \leftarrow \mathcal{A}(\mathsf{pk}, c) \ : \ \begin{array}{l} 1^\tau \leftarrow \mathcal{A}(1^\lambda),\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\tau) \\ c \leftarrow \mathsf{Enc}(\mathsf{pk}, 1) \end{array} \right] \right| \ .$$

We note that we consider the parameter τ as an adversarial quantity that can be chosen by $\mathcal{A}$.

Leveled homomorphic encryption. We will consider mainly *leveled homomorphic encryption schemes*. An HE scheme $\mathcal{E}$ is said to be a leveled homomorphic encryption scheme if $\mathcal{E}$ is correct for a family of circuits $\{\mathcal{C}_\tau\}_{\tau\in\mathbb{N}}$ and for all $\tau \in \mathbb{N}$, the set $\mathcal{C}_\tau$ contains only circuits whose depths are at most τ.

A.2 Damgård-Jurik encryption scheme

The Damgård-Jurik encryption scheme is given below:

$\mathsf{KeyGen}(1^\lambda)$: Given as input 1^λ where $\lambda \in \mathbb{N}$ is a security parameter, choose a λ-bit modulus $N = pq$ such that $\gcd(N, \varphi(N)) = 1$. Output $\mathsf{pk} = (N, \zeta)$ and $\mathsf{sk} = (p, q)$.

$\mathsf{Enc}(\mathsf{pk}, m)$: Given a public key $\mathsf{pk} = (N, \zeta)$ and a plaintext $m \in \mathbb{Z}_{N^\zeta}$, sample $r \hookleftarrow U(\mathbb{Z}_N^*)$ and output $\psi_\zeta(m, r) = (1+N)^m r^{N^\zeta} \pmod{N^{\zeta+1}}$.

$\mathsf{Dec}(\mathsf{sk}, c)$: Given a secret key $\mathsf{sk} = (p, q)$ and a ciphertext $c \in \mathbb{Z}^*_{N^{\zeta+1}}$, compute $\mathrm{lcm}(p-1, q-1)$ and use the inversion algorithm in Theorem 2 to compute back m. Output m.

B Deferred Proofs

B.1 Proof of Lemma 10

First, since $\mathbf{B} \sim U(\mathbb{Z}^{\ell \times m})$, Lemma 8 implies that $(\mathbf{Dx})^\top \cdot \mathbf{B}$ is statistically close to uniform. By Lemma 7, we have

$$\Pr\left[\left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B} \right\rceil_p \neq \left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B} + \mathbf{x}^\top \mathbf{E} \right\rceil_p\right] \leq \frac{2p \cdot \beta \cdot m\alpha q}{q} = 2p \cdot m\alpha \ ,$$

where $\|\mathbf{E}\|_\infty \leq m \cdot \alpha q$ and $\|\mathbf{x}\|_\infty \leq \beta = 1$.

On the other hand, a similar argument as in [2, Proof of Theorem 7.3] establishes that $\mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}}$ is statistically close to uniform. Applying Lemma 8 again implies $(\mathbf{Dx})^\top \cdot \mathbf{B} \cdot \mathbf{R}_{\mathsf{tag}}$ is statistically close to uniform. We have shown above that

$$\|\mathbf{E} \cdot \mathbf{R}_{\mathsf{tag}}\|_\infty \leq n'(n'+1)^t \cdot m \cdot \alpha q$$

where $n' = n \log q$. Hence, by Lemma 7, we obtain

$$\begin{aligned} &\Pr\left[\left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B}\mathbf{R}_{\mathsf{tag}} \right\rceil_p \neq \left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B}\mathbf{R}_{\mathsf{tag}} + \mathbf{x}^\top \cdot \mathbf{E}\mathbf{R}_{\mathsf{tag}} \right\rceil_p\right] \\ &\quad \leq \frac{2p \cdot \beta \cdot n'(n'+1)^t \cdot m\alpha q}{q} = 2p \cdot n'(n'+1)^t \cdot m\alpha \ . \end{aligned}$$

Finally, a union bound over all possible $\mathsf{tag} \in \mathcal{T}_t = \{0,1\}^t$ and $\mathsf{tag} \neq \mathsf{tag}^*$ yields

$$\begin{aligned} \Pr[\mathsf{BAD}] &\leq (2^t - 1) \cdot \Big(\Pr\left[\left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B} \right\rceil_p \neq \left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B} + \mathbf{x}^\top \mathbf{E} \right\rceil_p\right] \\ &\qquad + \Pr\left[\left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B}\mathbf{R}_{\mathsf{tag}} \right\rceil_p \neq \left\lfloor (\mathbf{Dx})^\top \cdot \mathbf{B}\mathbf{R}_{\mathsf{tag}} + \mathbf{x}^\top \cdot \mathbf{E}\mathbf{R}_{\mathsf{tag}} \right\rceil_p\right]\Big) \\ &\leq 2^{t+1} \cdot p \cdot m\alpha \cdot \left(1 + n'(n'+1)^t\right) \ . \end{aligned} \tag{4}$$

The right-hand side of (4) is set up to be negligible in λ. □