Compact Lossy Trapdoor Functions and Selective Opening Security From LWE [⋆]

Dennis Hofheinz, Kristina Hostáková[0000−0001−5235−8416], Julia Kastner[0000−0002−8879−8226], Karen Klein, and Akin Ünal[0000−0002−8929−0221]

Department of Computer Science
ETH Zurich, Switzerland
{hofheinz,kristina.hostakova,julia.kastner,karen.klein,akin.uenal}
@inf.ethz.ch

Abstract. Selective opening (SO) security is a security notion for public-key encryption schemes that captures security against adaptive corruptions of senders. SO security comes in chosen-plaintext (SO-CPA) and chosen-ciphertext (SO-CCA) variants, neither of which is implied by standard security notions like IND-CPA or IND-CCA security.

In this paper, we present the first SO-CCA secure encryption scheme that combines the following two properties: (1) it has a constant ciphertext expansion (i.e., ciphertexts are only larger than plaintexts by a constant factor), and (2) its security can be proven from a standard assumption. Previously, the only known SO-CCA secure encryption scheme achieving (1) was built from an ad-hoc assumption in the RSA regime.

Our construction builds upon LWE, and in particular on a new and surprisingly simple construction of compact lossy trapdoor functions (LTFs). Our LTF can be converted into an "all-but-many LTF" (or ABM-LTF), which is known to be sufficient to obtain SO-CCA security. Along the way, we fix a technical problem in that previous ABM-LTF-based construction of SO-CCA security.

1 Introduction

Selective opening security. Selective opening (SO) security [19, 5] is a security notion for public-key encryption that models adaptive corruptions of senders. For instance, consider a scenario in which a number of small devices send data (such as measurements) to a single receiver. Each device encrypts its messages using the public key of the receiver. However, each single sending device can be also corrupted, in which case an adversary may learn its complete internal state.

More abstractly, SO security considers a scenario in which an adversary gets a number of ciphertexts (for messages jointly chosen from an adversarially chosen message distribution), and can then ask for *openings* of a subset of those ciphertexts. Here, an opening yields both the encrypted messages *and* the used encryption random coins. After providing these openings, we require that the

[⋆] This work was partially supported by ERC Project PREP-CRYPTO 724307.

unopened ciphertexts remain secure, in the sense that the adversary does not obtain any information about the corresponding unopened messages that does not follow from the opened messages. This latter property is somewhat tedious to formalize, and several concrete SO security notions have been proposed in the literature; see [9] for an overview.

The hardness of obtaining SO security. Curiously, SO security does not follow from standard security notions such as semantic security [4, 35, 34]. Indeed, while there are a number of constructions of SO-secure schemes (e.g., [5, 20, 7, 27, 32, 36, 22, 33, 12, 38]), the most efficient of those schemes are still considerably less efficient than state-of-the-art IND-CCA-secure encryption schemes. To be more precise: when aiming at SO security against chosen-ciphertext attacks (i.e., SO-CCA security), then all of the known constructions except the one from [32] suffer from large ciphertexts with a super-constant ciphertext/message size ratio. The only exception, [32], is tied to the RSA regime, and relies on an ad-hoc and non-standard assumption.[1] One reason for this inefficiency is that SO security does not appear to permit hybrid encryption techniques (notwithstanding positive results in idealized computational models [30, 31]).

Our goal. Our goal in this work will be to provide an SO-CCA secure public-key encryption scheme with compact ciphertexts (i.e., with a constant ciphertext/message size ratio) from the LWE assumption. Obviously, more compact ciphertexts are always desirable, but in a setting like the SO application sketched above (with many small devices sending data to a base station), keeping transmitted messages compact seems particularly desirable. To achieve our goal, we will follow the high-level approach of [32, 38], and construct an SO-CCA secure scheme from a variant of lossy trapdoor functions.

Lossy trapdoor functions. A lossy trapdoor function (LTF [45]), is a family of functions f_{ek} parameterized by an evaluation key ek, and such that the following holds:

- If ek is chosen using an "injective key generation algorithm" $\mathsf{LTF.IGen}$, then f_{ek} is invertible using a trapdoor also output by $\mathsf{LTF.IGen}$ alongside ek.
- But if ek is chosen using a "lossy key generation algorithm" $\mathsf{LTF.LGen}$ (that does not output any trapdoors alongside ek), then f_{ek} is highly non-injective.
- The keys ek output by $\mathsf{LTF.IGen}$ are computationally indistinguishable from those output by $\mathsf{LTF.LGen}$.

This indistinguishability of injective and lossy keys ek already implies one-wayness of the function f_{ek}, and hence LTFs imply trapdoor one-way functions (TD-OWFs). In particular, all applications of TD-OWFs are also applications of LTFs. However, from a theoretical perspective, LTFs seem to be more powerful

[1] Specifically, [32] assumes in a very strong sense that the Paillier encryption scheme is *not* multiplicatively homomorphic.

objects than TD-OWFs as LTFs imply collision-resistant hash functions which is not known to be implied (in a black-box way) from TD-OWFs.

From a practical perspective, LTFs are also a convenient way to obtain lossy encryption [44, 5], deterministic encryption [10], or chosen-ciphertext-secure encryption [45, 41], possibly secure even against selective openings [32]. There are also several concrete constructions of LTFs from number-theoretic assumptions, including from Decisional Diffie-Hellman [45] (or related group-based assumptions [21]), the Decisional Composite Residuosity (DCR) assumption [21] (or other RSA-related assumptions [21, 37, 28, 8, 3]), and even from the Learning With Errors (LWE) assumption [45, 6, 1, 12, 38, 18].

Properties of LTFs. There are two particularly interesting quantitative properties of an LTF in view of our SO application: its expansion, and its lossiness. To explain these attributes of an LTF, let us simplify things a bit and assume that inputs and outputs of f_{ek} are bits, i.e., we have $f_{ek} : \{0,1\}^{\nu} \to \{0,1\}^{\mu}$. Then we may call the fraction $\chi = \mu/\nu$ the (multiplicative) "expansion" of the LTF. Of course, $\chi \geq 1$ because at least with injective keys, f_{ek} is injective.

We may also define the "lossiness" of f_{ek} for lossy keys ek as $\ell = \nu - \log_2 |\mathcal{IG}|$, where $\mathcal{IG} := f_{ek}(\{0,1\}^{\nu})$ is the actual image of the function. Intuitively, ℓ denotes the average number of input bits that are lost by evaluating f_{ek} with a lossy key. We can also define the "relative lossiness" of f_{ek} as $L = \ell/\nu$. In view of applications, of course a larger (relative) lossiness, and a smaller expansion are desirable. For instance, in encryption applications, typically $y := f_{ek}(x)$ (for a random x) is part of the ciphertext, and entropy from x is used to hide a message to be encrypted. Hence, the ciphertext grows with $|y| = \mu$ (and thus with χ), and generally a larger ℓ means more entropy in x (when ek is lossy), and thus a larger potential message.[2]

Achieving large (relative) lossiness with a small expansion seems to be difficult. With one exception, all known group-based LTFs [45, 21] process the preimage in a bitwise fashion (which results in large outputs). The one exception is the group-based LTF from [18], which achieves large lossiness and optimal expansion asymptotically, but uses several abstractions and is comparatively involved to evaluate. Known lattice-based LTFs either also process their input bitwise [45], suffer from a relatively small relative lossiness [6, 1, 12, 38], or are again comparatively complex to evaluate [18].[3] The situation in the RSA regime is somewhat brighter: some RSA-based LTFs achieve a constant relative lossiness *and* a constant expansion simultaneously.[4]

[2] Of course, in many encryption applications, hybrid encryption is possible, and thus x only needs to have enough entropy (even given y) that a symmetric encryption key can be extracted. However, in certain applications like selective-opening security, hybrid encryption does not seem to be useful, and thus a larger lossiness leads to larger messages.

[3] We note that while this may seem promising, it is also not possible to boost relative lossiness generically, e.g., by repetition [46].

[4] It is not easy to give a more quantitative comparison, since all mentioned LTF constructions offer tradeoffs regarding efficiency, relative lossiness, and compactness.

It also seems hard to construct RSA-based LTFs with additional properties, and in particular "all-but-many LTFs" (ABM-LTFs [32, 12, 38]). Such ABM-LTFs are particularly useful to obtain selective-opening security. However, the only ABM-LTF construction in the RSA regime [32] relies on an ad-hoc and nonstandard assumption related to the *absence* of multiplicative homomorphisms in the Paillier encryption scheme [42].[5]

Our contribution. To obtain our goal of compact SO-CCA secure encryption, we first construct a conceptually extremely simple LTF from LWE that achieves both constant relative lossiness (arbitrarily close to 1) *and* a constant expansion. Building on ideas from [12, 38], we also extend our LTF to an ABM-LTF. We use this ABM-LTF to construct a public-key encryption scheme that combines the following properties:

- it is chosen-ciphertext selective-opening secure,
- it has a constant ciphertext expansion (i.e., ciphertexts are larger than plaintexts only by a constant factor),
- its security is based on a standard assumption (in this case: LWE).

We stress that our main claim regarding this ABM-LTF is conceptual simplicity and efficiency. Indeed, while it seems plausible that existing works (such as [13, 23], when combined with [12, 38]) yield an asymptotically similar result, our construction is simple and efficient.

1.1 Technical overview

In this section, we will give more details on our ideas and techniques, and in particular on the application of our LTF on selective-opening secure encryption.

Starting point: LWE trapdoors. Assume a modulus $q \in \mathbb{Z}$, dimensions $n, m \in \mathbb{N}$ with $m \gg n$, and a uniformly distributed matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$. Let $\mathfrak{D} \subset \mathbb{Z}_q$ be a set of short "noise values", i.e., of values $e \in \mathbb{Z}_q$ with $|e| \ll q$. The function $g_{\mathbf{A}}$ with

$$g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e}$$

(for $\mathbf{s} \in \mathbb{Z}_q^n$ and short $\mathbf{e} \in \mathfrak{D}^m$) is injective with high probability over the choice of $\mathbf{A}$. In fact, $g_{\mathbf{A}}$ is efficiently invertible with a suitable trapdoor $\tau_{\mathbf{A}}$ that can be generated alongside $\mathbf{A}$, e.g., as in [40].

It has been noted before that, for suitable choices of q, n, m, and "shortness" of vectors, the function $g_{\mathbf{A}}$ above is already a lossy trapdoor function [43, 6, 1, 38]. Indeed, setting up $\mathbf{A}' = \mathbf{B}\mathbf{D} + \mathbf{E}$ for "very flat" $\mathbf{D} \in \mathbb{Z}_q^{k \times n}$ with $k \ll n$ and short

[5] It would seem promising to construct LTFs and ABM-LTFs from suitable homomorphic encryption schemes, and in particular from "Rate-1 FHE schemes" [13, 23], using the blueprint of [29] and [12, 38]. This may indeed lead to asymptotically compact LTFs and ABM-LTFs with large lossiness, but the corresponding constructions have an involved evaluation procedure and will require large inputs to play out their asymptotic properties.

$\mathbf{E} \in \mathfrak{D}^{n \times m}$ leads to a highly non-injective $g_{\mathbf{A}'}$ [25]. The corresponding matrices $\mathbf{A}'$ are computationally indistinguishable from uniformly random $\mathbf{A}$ under the LWE assumption and can hence be used as lossy keys.

Unfortunately, since $m \gg n$, this $g_{\mathbf{A}}$ has large images $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^m$ for relatively small preimages $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times \mathfrak{D}^m$, and thus a large expansion of around $\log_2 q$ for typical parameters. This is particularly problematic in settings in which q is large (i.e., superpolynomial), e.g., [38].[6]

Our basic lossy trapdoor function. We first borrow an idea that has been used in the context of dual-mode commitments [16] (and later found use also in lossy encryption schemes [44, 27]). In a nutshell, we can convert any publicly re-randomizable encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ into a lossy commitment or encryption scheme as follows. The public key is $pk = (c_0 = \mathsf{Enc}(0), c_1 = \mathsf{Enc}(1))$, and to encrypt a message $b \in \{0, 1\}$, we simply output a re-randomized version of c_b. This yields a fresh encryption of b, which Dec can decrypt as usual. Lossy (public) keys are of the form $pk' = (c_0 = \mathsf{Enc}(0), c_1 = \mathsf{Enc}(0))$, such that encryptions are *always* fresh 0-encryptions.

This trick easily scales to larger message spaces when assuming (additively) homomorphic encryption. For instance, we can publish $pk = c_1 = \mathsf{Enc}(1)$ and compute a fresh encryption of any M homomorphically (as a re-randomized version of $M \cdot c_1$). In our setting, we can implement the encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with a dual version of Regev's encryption (as done in [44]), and omit the final re-randomization step. This yields a deterministic encryption scheme, which we can immediately interpret as a lossy trapdoor function.[7]

After resolving a few technical obstacles[8], we obtain the following scheme. Injective keys are of the form

$$ek = \mathbf{C} = \mathbf{A}\mathbf{S} + \mathbf{E} + \mathbf{G}$$

for suitably-sized uniform $\mathbf{A}$, short $\mathbf{E}$, and the "gadget matrix" $\mathbf{G}$ from [40]. For this overview, it is only important that $\mathbf{G}$ allows for an efficiently computable "bit decomposition" operation $\mathbf{G}^{-1} : \mathbb{Z}_q^m \to \{0, 1\}^{m \lceil \log_2 q \rceil}$ that satisfies $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{z}) = \mathbf{z}$ for all $\mathbf{z}$.

[6] It is tempting to rely on $\mathbf{e}$ (and not $\mathbf{s}$) as a means to transport information. In particular, making $\mathfrak{D}$ larger improves expansion (since preimages carry more information). However, at least with the arguments above, we can only argue that information about $\mathbf{s}$ (not $\mathbf{e}$) is lost in lossy mode, i.e., with $\mathbf{A}' = \mathbf{B}\mathbf{D} + \mathbf{E}$. Hence, making $\mathfrak{D}$ larger improves expansion, but at the same time hurts (relative) lossiness.

[7] This is similar to [29], who interpret lossy encryption schemes as lossy trapdoor functions (by deriving encryption random coins deterministically from the encrypted message). Our setting is considerably simpler, however, since our final encryption scheme is deterministic.

[8] For instance, in (dual) Regev encryption, noise terms in ciphertexts grow with homomorphic computations. This is a problem with large factors M as above, but can be avoided by the use of a "gadget matrix" [40]. Furthermore, we are using a more economic, "batched" version of (dual) Regev encryption as with [2].

This $\mathbf{C}$ can be viewed as an economic dual-Regev encryption of $\mathbf{G}$. In this context, it is also helpful to point out that $\mathbf{S}$ will be a "very flat" matrix (with many more columns than rows). Together with the fact that $\mathbf{E}$ is short (i.e., has a small norm), this means that the encryption randomness in this encryption is much smaller (in, say, overall bitsize) than the encrypted message. In the upcoming lossiness analysis, this will be crucial and enable an argument similar to the one in [29].

In order to evaluate the resulting LTF g_{ek} on an input $\mathbf{x}$, we encode $\mathbf{x}$ as $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix}$ for a suitable constant c and then simply compute

$$g_{ek}(\mathbf{x}) = \mathbf{C} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}}) = \mathbf{A}\left(\mathbf{S} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})\right) + \mathbf{E} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}}) + \tilde{\mathbf{x}}.$$

This is an economic dual-Regev encryption of $\mathbf{x}$ from which $\mathbf{x}$ can be retrieved using a decryption key (that incorporates trapdoor information about $\mathbf{A}$). This evaluation can also be viewed as a variant of the LWE-based LTF from [45], however with a different encoding $\tilde{\mathbf{x}}$ of inputs. This encoding, along with its "flattening" $\mathbf{G}^{-1}(\tilde{\mathbf{x}})$, essentially allows to use a [45]-like evaluation strategy on non-short input vectors $\mathbf{x}$ while containing noise growth.

Lossy keys, however, are of the form

$$ek' = \mathbf{C}' = \mathbf{AS} + \mathbf{E},$$

and their indistinguishability from injective keys readily follows from LWE. When evaluating $g_{ek'}$ with such lossy keys ek', we obtain

$$g_{ek'}(\mathbf{x}) = \mathbf{C} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}}) = \mathbf{A}\left(\mathbf{S} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})\right) + \mathbf{E} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}}),$$

which leaks information about $\mathbf{x}$ only through the terms $\mathbf{S} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})$ and $\mathbf{E} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})$. But the former of these terms will be a vector that has much fewer entries than $\mathbf{x}$ (due to the "very flat" nature of $\mathbf{S}$), and the latter term is a small-norm vector. A careful analysis (in Section 3 and Appendix B) will indeed show that any constant relative lossiness $L < 1$ (with constant expansion) can be achieved by setting q, n, m up as suitable polynomials in the security parameter. Larger values of q enable a relative lossiness even closer to 1 (see Appendix B for details), which also implies quantitative improvements for certain LTF applications (see [46]).

Extension to all-but-many lossy trapdoor functions. We will now sketch how to use methods from [12, 38] to convert our LTF into an "all-but-many LTF" (ABM-LTF). In a nutshell, ABM-LTFs are *tagged* LTFs (i.e., LTFs in which evaluation also takes as input a tag t) that are lossy or injective depending on that tag. It should be hard to generate lossy tags without a trapdoor, while it should be easy to publicly sample injective tags.[9]

[9] Moreover, random tags should be injective with high probability, and it should even be possible to explain any tag (injective or not) as having been randomly sampled.

Tags will be of the form $t = (t_c, t_a)$ with a core part t_c and an auxiliary part t_a, and will be lossy if and only if $t_c = F_K(t_a)$ for a pseudorandom function (PRF) F with a key $K = (K_i)_{i=1}^{\lambda} \in \{0,1\}^{\lambda}$ that is encrypted (bit-wise) in the LTF overall key. More specifically, our ABM-LTF evaluation key will be of the form

$$ek = (\mathbf{C}_i)_{i=1}^{\lambda} = (\mathbf{A}\mathbf{S}_i + \mathbf{E}_i + K_i \cdot \mathbf{G})_{i=1}^{\lambda} ,$$

i.e., consist of bit-wise encryptions (with independent $\mathbf{S}_i, \mathbf{E}_i$) of the bits of K. We have chosen to encrypt the K_i in this wasteful fashion to enable fully homomorphic computations with the K_i. In fact, these encryptions can be viewed as ciphertexts of the "dual GSW" fully homomorphic encryption scheme [24, 39]. Hence, we can publicly derive ciphertexts that encrypt the result of the binary "test" function $T(K, t)$ with $T(K, (t_c, t_a)) = 0$ if and only if $t_c = F_K(t_a)$ (and $T(K, (t_c, t_a)) = 1$ otherwise).

To evaluate this ABM-LTF on a tag $t = (t_c, t_a)$ and input $\mathbf{x}$, we first (deterministically, using fully homomorphic operations) compute a ciphertext

$$\mathbf{C}_t = \mathbf{A}\mathbf{S}_t + \mathbf{E}_t + T(K, t) \cdot \mathbf{G}$$

and then evaluate $g_{ek,t}(x) = \mathbf{C}_t \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})$ for $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix}$, as with our LTF. Observe that random tags (t_c, t_a) lead to injective LTF keys, and lossy tags (of the form $(F_K(t_a), t_a)$) can be generated using the PRF key K as trapdoor.

We provide full details in Section 4.

Achieving selective-opening security. Plugging both our LTF and ABM-LTF above in the construction of [32], however, yields the first SO-CCA secure encryption scheme with constant ciphertext expansion. We provide a full analysis in Section 5. Since the lossiness of our ABM-LTF is significantly larger than that of the (similarly LWE-based) ABM-LTFs from [12, 38], we end up with a conceptually simpler encryption scheme.[10] As a side note, we also identify and fix a minor problem in the original SO-CCA security proof of [32] (which is also inherited by [12, 38]) along the way, see Section 5, "Mending a gap in [32]". We admit that while the resulting scheme significantly improves in efficiency upon [12, 38], it is still not overly practical. Like [12, 38], we rely on FHE computations, and the resulting ciphertext expansion is constant but still considerably larger than the (moderate) expansion of the involved LTF and ABM-LTF (see Appendix B for a detailed and more quantitative discussion). However, we also believe that our ideas may open the door to further improvements, e.g., for more efficient transformations from LTFs to ABM-LTFs.

[10] For instance, the treatment of leakage through noise terms in [38] is quite involved, which causes also a more complex construction. In our case, the involved error terms leak little information (relative to the ABM-LTF input), and we can afford a simpler construction and analysis.

2 Preliminaries

2.1 Notation

We denote by $x \xleftarrow{\$} \mathfrak{D}$ for a distribution $\mathfrak{D}$ that x is sampled at random according to $\mathfrak{D}$. For a set X we denote by $x \xleftarrow{\$} X$ that x is sampled uniformly at random from X. We denote by $x \coloneqq y$ that x is deterministically assigned the value y.

For a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}^n$ and $c \geq 0$, we use the following notation:

$$\mathbf{x} \text{ div } c := \left(\left\lfloor \frac{x_1}{c} \right\rceil, \ldots, \left\lfloor \frac{x_n}{c} \right\rceil \right),$$

where $\lfloor \frac{x_i}{c} \rceil := \lfloor \frac{x_i}{c} + \frac{1}{2} \rfloor$.

Definition 1 (Infinity-Norm). *Let $\mathbf{x} = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$. The ∞-**norm** of $\mathbf{x}$ is given by $||\mathbf{x}||_\infty := \max_{i \in [n]} |x_i|$. The corresponding operator norm of $\mathbf{A}$ is given by $||\mathbf{A}||_\infty := \max_{\substack{\mathbf{x} \in \mathbb{R}^n \\ \mathbf{x} \neq 0}} \frac{||\mathbf{A}\mathbf{x}||_\infty}{||\mathbf{x}||_\infty}$.*

Proposition 1 ([26]). *For $\mathbf{A} = (a_{i,j})_{i,j} \in \mathbb{R}^{m \times n}$, $||\mathbf{A}||_\infty = \max\limits_{i \in [m]} \sum\limits_{j=1}^{n} |a_{i,j}|$.*

In particular, if $\mathbf{A} \in \{-1, 0, 1\}^{m \times n}$, we get $||\mathbf{A}||_\infty \leq n$.

Further, we have $\left|\left|\mathbf{A}^T\right|\right|_\infty \leq m \cdot ||\mathbf{A}||_\infty$ for any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$.

Definition 2 (Discrete Gaussian Distribution). *For $\mathbf{x} \in \mathbb{R}^m$ and $\sigma > 0$, set $\rho_\sigma(\mathbf{x}) := \exp\left(-\pi \cdot ||\mathbf{x}||_2^2 \cdot \sigma^{-2}\right)$. Then, the series $\rho_\sigma(\mathbb{Z}^m) = \sum_{\mathbf{x} \in \mathbb{Z}^m} \rho_\sigma(\mathbf{x})$ converges.*

*The **discrete Gaussian distribution** D_σ^m with deviation σ is the probability distribution over $\mathbb{Z}^m$ that assigns to each integer vector $\mathbf{x}$ the probability $D_\sigma^m(\mathbf{x}) := \rho_\sigma(\mathbf{x}) / \rho_\sigma(\mathbb{Z}^m)$.*

Remark 1. D_σ^m is subgaussian with parameter σ. For each $t > 0$, we have

$$\Pr_{\mathbf{e} \xleftarrow{\$} D_\sigma^m} \left[||\mathbf{e}||_\infty > t \right] \leq 2m \cdot \exp\left(-\pi \frac{t^2}{\sigma^2} \right).$$

Further, we have for each $t > 0$

$$\Pr_{\mathbf{E} \xleftarrow{\$} D_\sigma^{m \times N}} \left[||\mathbf{E}||_\infty > N \cdot t \right] \leq 2mN \cdot \exp\left(-\pi \frac{t^2}{\sigma^2} \right).$$

By setting $t = \sqrt{\lambda} \cdot \sigma$, we therefore get

$$\Pr_{\mathbf{E} \xleftarrow{\$} D_\sigma^{m \times N}} \left[||\mathbf{E}||_\infty > \sqrt{\lambda} \cdot \sigma \cdot N \right] \leq 2mN \cdot \exp\left(-\pi \cdot \lambda \right).$$

2.2 LWE-Based Trapdoors

Definition 3 (Decisional Learning With Errors Assumption). *Let $n \in \mathbb{N}, m \in \mathsf{poly}(n)$ and $q = q(n) \in \mathbb{N}, \alpha = \alpha(n) \in (0,1)$. The **decisional learning with errors assumption** $\mathsf{LWE}_{n,q,\alpha,m}$ states that the advantage of each PPT adversary in distinguishing the matrix distributions*

$$(\mathbf{A}, \mathbf{b}) \text{ and } (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e} \mod q),$$

for $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \xleftarrow{\$} D_{\alpha q}^m, \mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^m$, is negligible in n. Given an adversary $\mathcal{A}$, we denote by $\mathsf{Adv}_{\mathsf{LWE},\mathcal{A}}^{n,q,\alpha,m}(\lambda)$ its advantage in distinguishing LWE samples from uniformly random matrices i.e.

$$\mathsf{Adv}_{\mathsf{LWE},\mathcal{A}}^{n,q,\alpha,m}(\lambda) := \Bigg| \Pr_{\mathbf{A}\xleftarrow{\$}\mathbb{Z}_q^{m\times n},\mathbf{b}\xleftarrow{\$}\mathbb{Z}_q^m}[\mathcal{A}(\mathbf{A},\mathbf{b}) = 1] \\ - \Pr_{\mathbf{A}\xleftarrow{\$}\mathbb{Z}_q^{m\times n},\mathbf{s}\xleftarrow{\$}\mathbb{Z}_q^n,\mathbf{e}\xleftarrow{\$}D_{\alpha q}^m}[\mathcal{A}(\mathbf{A},\mathbf{A}\mathbf{s}+\mathbf{e} \bmod q) = 1] \Bigg| .$$

Definition 4 (Gadget Matrix). *Let $n, q \in \mathbb{N}$. By $\mathbf{G}_{n,q} \in \mathbb{Z}_q^{n\times(n\cdot\lceil\log_2 q\rceil)}$ we denote the **gadget matrix** (for n and q) that is given by*

$$\mathbf{G}_{n,q} = \begin{pmatrix} 1\ 2 \ldots 2^{\lceil\log_2 q\rceil-1} & & & \\ & 1\ 2 \ldots 2^{\lceil\log_2 q\rceil-1} & & \\ & & \ddots & \\ & & & 1\ 2 \ldots 2^{\lceil\log_2 q\rceil-1} \end{pmatrix}.$$

Since each number $a \in \{0, \ldots, q-1\}$ has a binary decomposition $a = b_0 \cdot 1 + b_1 \cdot 2 + \ldots + b_{\lceil\log_2 q\rceil-1} \cdot 2^{\lceil\log_2 q\rceil-1}$ over the integers, for each $\mathbf{y} \in \mathbb{Z}_q^n$ there is a binary vector $\mathbf{x} \in \{0,1\}^{n\cdot\lceil\log_2 q\rceil}$ s.t.

$$\mathbf{y} = \mathbf{G}_{n,q} \cdot \mathbf{x}.$$

This vector $\mathbf{x}$ is uniquely determined by $\mathbf{y}$ and we set $\mathbf{G}_{n,q}^{-1}(\mathbf{y}) := \mathbf{x}$. Given $\mathbf{y}$, $\mathbf{G}_{n,q}^{-1}(\mathbf{y})$ can be computed efficiently. If $\mathbf{B} = (\mathbf{b}_1|\ldots|\mathbf{b}_N)$ is an $m \times N$-matrix, then we define $\mathbf{G}_{n,q}^{-1}(\mathbf{B})$ as

$$\mathbf{G}_{n,q}^{-1}(\mathbf{B}) = (\mathbf{G}_{n,q}^{-1}(\mathbf{b}_1)|\ldots|\mathbf{G}_{n,q}^{-1}(\mathbf{b}_N)) \in \{0,1\}^{\lceil\log_2 q\rceil m\times N}.$$

In general, we will omit the subscripts n, q and simply write $\mathbf{G}$ and $\mathbf{G}^{-1}$ instead of $\mathbf{G}_{n,q}$ and $\mathbf{G}_{n,q}^{-1}$ when n, q can be deduced from the current context.

Lemma 1. *Let $n, q \in \mathbb{N}$ with $q \geq 8$. For each $\mathbf{y} \in \mathbb{Z}_q^{n\cdot\lceil\log_2 q\rceil}$, there is at most one pair $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times \mathbb{Z}^{n\cdot\lceil\log_2 q\rceil}$ with $||\mathbf{e}||_\infty < \frac{q}{2\cdot\lceil\log_2 q\rceil}$ s.t.*

$$\mathbf{y} = \mathbf{G}^T\mathbf{s} + \mathbf{e} \mod q.$$

There is a PPT algorithm that, given $\mathbf{y} \in \mathbb{Z}_q^{n\cdot\lceil\log_2 q\rceil}$, can output such a pair $(\mathbf{s}, \mathbf{e})$ if it exists.

A proof of Lemma 1 is given in [40]. Additionally, a proof can be found in Appendix A.1.

Definition 5 (Trapdoor Sampling and Inversion). *Let $\mathfrak{R}$ be the distribution over $\{-1, 0, +1\}$ that draws $b_1, b_2 \xleftarrow{\$} \{0, 1\}$ and outputs $b_1 - b_2$.*

The LWE-based trapdoor scheme of Micciancio & Peikert [40] works as follows:

GenTrap: *Given numbers $n, \overline{u}, q \in \mathbb{N}$,* GenTrap *sets $w := n \lceil \log_2 q \rceil$ and $u := \overline{u} + w$. It samples $\overline{\mathbf{B}} \xleftarrow{\$} \mathbb{Z}_q^{\overline{u} \times n}$, $\mathbf{R} \xleftarrow{\$} \mathfrak{R}^{w \times \overline{u}}$ and outputs the trapdoor $\mathbf{R}$ and the $u \times n$-matrix*

$$\mathbf{B} := \begin{pmatrix} \overline{\mathbf{B}} \\ \mathbf{G}_{n,q}^T - \mathbf{R}\overline{\mathbf{B}} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_{\overline{u}} & \mathbf{0} \\ -\mathbf{R} & \mathbf{G}_{n,q}^T \end{pmatrix} \cdot \begin{pmatrix} \overline{\mathbf{B}} \\ \mathbf{I}_n \end{pmatrix} \mod q.$$

Eval: *Given $\mathbf{B} \in \mathbb{Z}_q^{u \times n}, \mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}^u$,* Eval *outputs*

$$\mathbf{y} := \mathbf{B}\mathbf{s} + \mathbf{e} \mod q.$$

Invert: *Given $\mathbf{B} \in \mathbb{Z}_q^{u \times n}, \mathbf{y} \in \mathbb{Z}_q^u$ and a trapdoor $\mathbf{R} \in \{-1, 0, 1\}^{w \times \overline{u}}$,* Invert *computes*

$$\widehat{\mathbf{y}} := \begin{pmatrix} \mathbf{R} & \mathbf{I}_w \end{pmatrix} \cdot \mathbf{y} \mod q.$$

It uses the algorithm of Lemma 1 to find $\widehat{\mathbf{s}} \in \mathbb{Z}_q^n$ and $\widehat{\mathbf{e}} \in \mathbb{Z}^w$ s.t.

$$\widehat{\mathbf{y}} = \mathbf{G}^T \cdot \widehat{\mathbf{s}} + \widehat{\mathbf{e}} \mod q$$

and outputs

$$\mathbf{s} := \widehat{\mathbf{s}} \textit{ and } \mathbf{e} = \mathbf{y} - \mathbf{B}\mathbf{s} \mod q.$$

Lemma 2. *Let $n, \overline{u}, q \in \mathbb{N}$. Set $w := n \lceil \log_2 q \rceil$ and $u := \overline{u} + w$.*

1. *For $(\mathbf{R}, \mathbf{B}) \xleftarrow{\$}$* GenTrap$(n, \overline{u}, q)$*, the statistical distance of $\mathbf{B}$ and $U(\mathbb{Z}_q^{u \times n})$ is bounded by $\leq \frac{w}{2}\sqrt{q^n / 2^{\overline{u}}}$.*
   If we have $\overline{u} \geq w + 2(n + \log_2(w) - 1)$, then the statistical distance is $\leq 2^{-n}$.
2. *For each $\mathbf{R} \in \{-1, 0, 1\}^{w \times \overline{u}}$, we have $||\mathbf{R}||_\infty \leq \overline{u}$.*
3. *Let $\mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in \mathbb{Z}^u$ and $(\mathbf{R}, \mathbf{B}) \xleftarrow{\$}$* GenTrap$(n, \overline{u}, q)$*. The algorithm* Invert$(\mathbf{B}, \mathbf{B}\mathbf{s} + \mathbf{e} \bmod q, \mathbf{R})$ *computes $(\mathbf{s}, \mathbf{e})$ if*

$$||\mathbf{e}||_\infty < \frac{q}{2 \cdot \log_2(q) \cdot (\overline{u} + 1)}.$$

Proof. The first claim follows from the Leftover Hash Lemma for matrices. The second point is a consequence of Proposition 1, and the last point follows from Lemma 1. A more detailed proof of the first claim can be found in [40].

2.3 Fully Homomorphic Encryption from Lattices

Lemma 3 (Barrington's Theorem). *Let $C : \{0,1\}^\eta \to \{0,1\}$ be a circuit of depth d that only consists of "NAND" gates. Then, there is a branching program of length 4^d that computes the same functionality as C. I.e., there is a function $\iota : [4^d] \to [\eta]$ and permutations $\sigma_{i,j} \in S_5, i \in [4^d], j \in \{0,1\}$, s.t. we have for each $\mathbf{x} \in \{0,1\}^\eta$*

$$C(x_1, \dots, x_\eta) = 1 \iff \sigma_{4^d, x_{\iota(4^d)}} \circ \cdots \circ \sigma_{1, x_{\iota(1)}}(1) = 1.$$

There is a polynomial time algorithm that – given a description of C – outputs ι and $(\sigma_{i,j})_{i \in [4^d], j \in \{0,1\}}$.

We will now introduce a fully homomorphic encryption (FHE) scheme that is mainly known as *dual GSW* [24, 39]. However, we note that the dual GSW FHE scheme is actually very close to the FHE scheme of Brakerski & Vaikuntanathan [14] (up to applications of the inverse of the gadget matrix).

We will in the following only describe the encryption and the homomorphic evaluation algorithm of the dual GSW FHE scheme (since we will not need the key generation and decryption algorithm for our lossy trapdoor functions).

Definition 6 (Dual GSW FHE). *Let $q, n, m \in \mathbb{N}$ and $\alpha > 0$. Set $N := m \cdot \lceil \log_2 q \rceil$. The dual GSW FHE scheme consists of the following two algorithms:*

$\mathsf{FHE.Enc}$: *Given a public key $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a message $\mu \in \{0,1\}$, $\mathsf{FHE.Enc}$ samples $\mathbf{E} \xleftarrow{\$} D_{\alpha q}^{m \times N}$, $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n \times N}$ and outputs the ciphertext*

$$\mathbf{C} := \mathbf{A}\mathbf{S} + \mathbf{E} + \mu\mathbf{G} \mod q \in \mathbb{Z}_q^{m \times N}.$$

$\mathsf{FHE.Eval}$: *We first describe how $\mathsf{FHE.Eval}$ evaluates negations, additions and multiplications:*

Negations: $\mathsf{FHE.Eval}$ *negates the message of a ciphertext $\mathbf{C}_\mu \in \mathbb{Z}_q^{m \times N}$ by computing*

$$\mathbf{C}_{\neg\mu} := \mathbf{G} - \mathbf{C}_\mu \mod q.$$

Additions: *Given two ciphertexts $\mathbf{C}_{\mu_1}, \mathbf{C}_{\mu_2} \in \mathbb{Z}_q^{m \times N}$, $\mathsf{FHE.Eval}$ adds their messages by*

$$\mathbf{C}_{\mu_1 + \mu_2} := \mathbf{C}_{\mu_1} + \mathbf{C}_{\mu_2} \mod q.$$

Multiplications: *Given two ciphertexts $\mathbf{C}_{\mu_1}, \mathbf{C}_{\mu_2} \in \mathbb{Z}_q^{m \times N}$, $\mathsf{FHE.Eval}$ multiplies their messages by*

$$\mathbf{C}_{\mu_1 \cdot \mu_2} := \mathbf{C}_{\mu_1} \cdot \mathbf{G}^{-1}(\mathbf{C}_{\mu_2}) \mod q$$

where $\mathbf{G}^{-1}(\mathbf{B})$ of an $m \times N$-matrix $\mathbf{B} = (\mathbf{b}_1 | \dots | \mathbf{b}_N)$ is the binary $N \times N$-matrix

$$\mathbf{G}^{-1}(\mathbf{B}) = (\mathbf{G}^{-1}(\mathbf{b}_1) | \dots | \mathbf{G}^{-1}(\mathbf{b}_N)).$$

If $\mathsf{FHE.Eval}$ *is given* η *input ciphertexts* $\mathbf{C}_1, \ldots, \mathbf{C}_\eta$ *and a circuit* $C : \{0,1\}^\eta \to \{0,1\}$ *of depth* d *that only consists of "NAND" gates, then* $\mathsf{FHE.Eval}$ *converts* C *to a branching program of length* $4d$*, which is described by a function* $\iota : [4^d] \to [\eta]$ *and permutations* $\sigma_{i,\mu} \in S_5, i \in [4d], \mu \in \{0,1\}$ *(see Lemma 3). For* $i, j \in [5]$*,* $\mathsf{FHE.Eval}$ *sets*

$$\mathbf{Q}_{i,j}^{(0)} := \begin{cases} \mathbf{G}_{m,q}, & \text{if } i = j, \\ \mathbf{0} \in \mathbb{Z}_q^{m \times N}, & \text{if } i \neq j. \end{cases}$$

Additionally, it computes negations $\mathbf{C}_{\neg i} := \mathbf{G} - \mathbf{C}_i$ *of the inputs for* $i = 1, \ldots, \eta$*.*
For $h = 1, \ldots, 4^d$ *and* $i, j \in [5]$*,* $\mathsf{FHE.Eval}$ *computes* $\mathbf{Q}_{i,j}^{(h)}$ *by setting*

$$\mathbf{Q}_{i,j}^{(h)} := \mathbf{C}_{\iota(h)} \cdot \mathbf{G}^{-1}\left(\mathbf{Q}_{\sigma_{h,1}^{-1}(i),j}^{(h-1)}\right) + \mathbf{C}_{\neg\iota(h)} \cdot \mathbf{G}^{-1}\left(\mathbf{Q}_{\sigma_{h,0}^{-1}(i),j}^{(h-1)}\right).$$

Finally, it outputs the result

$$\mathsf{FHE.Eval}(C, \mathbf{C}_1, \ldots, \mathbf{C}_\eta) := \mathbf{Q}_{1,1}^{(4^d)}.$$

Similar to Brakerski & Vaikuntanathan [14], we define the ***noise*** *of an* $m \times N$*-matrix* $\mathbf{C} \in \mathbb{Z}_q^{m \times N}$ *under a public key* $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ *and a message* $\mu \in \{0,1\}$ *by*

$$\mathsf{noise}_{\mathbf{A},\mu}(\mathbf{C}) := \min\left\{ ||\mathbf{C} - \mathbf{AS} - \mu\mathbf{G} \mod q||_\infty \mid \mathbf{S} \in \mathbb{Z}_q^{n \times N} \right\}$$

where we interpret $\mathbf{C} - \mathbf{AS} - \mu\mathbf{G} \mod q$ *as a real matrix in* $[\frac{-q}{2}, \frac{q}{2})^{m \times N}$*.*

Remark 2. Note, that we gave the scheme FHE above without a key generation or a decryption algorithm. This is, because we wanted to keep its definition simple and seperate it from Definition 5.

In our ABM-LTF scheme, we will use the algorithm $\mathsf{GenTrap}$ to generate the matrix $\mathbf{A}$ as a public key for FHE together with the trapdoor $\mathbf{R}$ as secret key. In fact, by using $\mathsf{GenTrap}$ for key generation, we get a full encryption scheme:

If $\overline{u} \geq w + 2(n + \log_2(w) - 1)$, $\mathbf{A}$ is close to a uniformly random matrix and the ciphertexts of FHE are indistinguishable from random as we will show in the next lemma. Further, with the trapdoor $\mathbf{R}$ it is possible to decrypt a given ciphertext with sufficiently small noise (we show this in Lemma 5 that can be found in Appendix A.2.).

Lemma 4. *Let* $q, n, m \in \mathbb{N}$ *and* $\alpha > 0$*. Set* $N := m \cdot \lceil \log_2 q \rceil$*.*

1. *For each* $b \in \{0,1\}$*, if* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ *is drawn uniformly at random, then for each algorithm* $\mathcal{A}$ *with time complexity* t *there is an LWE-distinguisher* $\mathcal{B}$ *with time complexity* $t + \mathsf{poly}(n, m, \log_2 q)$ *s.t.*

$$\left| \Pr_{\mathbf{C} \xleftarrow{\$} \mathsf{FHE.Enc}(\mathbf{A}, b)}[\mathcal{A}(\mathbf{C}) = 1] - \Pr_{\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m \times N}}[\mathcal{A}(\mathbf{C}) = 1] \right| \leq N \cdot \mathsf{Adv}_{\mathsf{LWE},\mathcal{B}}^{n,q,\alpha,m}(\lambda).$$

2. *Let* $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$*,* $\mu_1, \ldots, \mu_\eta \in \{0,1\}$ *and let* $C : \{0,1\}^\eta \to \{0,1\}$ *be a circuit consisting of "NAND" gates of depth* d*. Let* $\mathbf{C}_1, \ldots, \mathbf{C}_\eta \in \mathbb{Z}_q^{m \times N}$ *and compute*

$$\mathbf{C} := \mathsf{FHE.Eval}(C, \mathbf{C}_1, \ldots, \mathbf{C}_\eta).$$

We have

$$\mathsf{noise}_{\mathbf{A}, C(\mu_1, \ldots, \mu_\eta)}(\mathbf{C}) \leq 2 \cdot 4^d \cdot N \cdot \max_{i \in [\eta]} \left(\mathsf{noise}_{\mathbf{A}, \mu_i}(\mathbf{C}_i) \right).$$

We will give a proof of this lemma in Appendix A.2.

2.4 Lossy Trapdoor Functions

Definition 7 (Lossy trapdoor function). *A* lossy trapdoor function *(LTF)* LTF *with domain* $\mathcal{D}$ *and range* $\mathcal{RG}$ *consists of the following algorithms:*

Key generation. $\mathsf{LTF.IGen}(1^\lambda)$ *yields an evaluation key* ek *and an inversion key* ik*.*

Evaluation. $\mathsf{LTF.Eval}(ek, x)$ *(with* $x \in \mathcal{D}$*) yields an image* $y \in \mathcal{RG}$*. Write* $y = f_{ek}(x)$*.*

Inversion. $\mathsf{LTF.Invert}(ik, y)$ *outputs a preimage* x*. Write* $x = f_{ik}^{-1}(y)$*.*

Lossy key generation. $\mathsf{LTF.LGen}(1^\lambda)$ *outputs an evaluation key* ek'*.*

We consider the following properties of LTF*:*

Correctness. *We require for* $(ek, ik) \xleftarrow{\$} \mathsf{LTF.IGen}(1^\lambda)$*,* $x \in \mathcal{D}$*, that* $f_{ik}^{-1}(f_{ek}(x)) = x$ *with all-but-negligible probability over the random coins used by* $\mathsf{LTF.IGen}$*.*

Expansion. *We define the expansion of* LTF *as* $\chi := \log_2 |\mathcal{RG}| / \log_2 |\mathcal{D}|$*.*

Lossiness. *We say that* LTF *is* ℓ*-lossy if for* $ek' \xleftarrow{\$} \mathsf{LTF.LGen}(1^\lambda)$*, with all-but-negligible probability the image set* $f_{ek'}(\mathcal{D})$ *is of size at most* $|\mathcal{D}|/2^\ell$*. We define the relative lossiness as* $L := \ell / \log_2 |\mathcal{D}|$*.*

Indistinguishability. *We require that the first output of* $\mathsf{LTF.IGen}(1^\lambda)$ *is indistinguishable from the output of* $\mathsf{LTF.LGen}(1^\lambda)$*, i.e., for any PPT* $\mathcal{A}$

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LTF},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathcal{A}(1^\lambda, ek) = 1\right] - \Pr\left[\mathcal{A}(1^\lambda, ek') = 1\right] \right|$$

is negligible, for $(ek, ik) \xleftarrow{\$} \mathsf{LTF.IGen}(1^\lambda)$*,* $ek' \xleftarrow{\$} \mathsf{LTF.LGen}(1^\lambda)$*.*

LTF *is an* ℓ*-lossy LTF with expansion* χ *if it satisfies the above properties for* ℓ *and* χ*.*

2.5 All-but-many Lossy Trapdoor Functions (ABM-LTF)

We recall the definition of All-But-Many Lossy-Trapdoor-Function (ABM-LTF) put forward by Hofheinz [32].

Definition 8 (ABM-LTF [32]). *An* all-but-many lossy trapdoor function ABM *with domain* $\mathcal{D}$ *and range* $\mathcal{RG}$ *consists of four PPT algorithms* $(\mathsf{ABM.Gen}, \mathsf{ABM.Eval}, \mathsf{ABM.Invert}, \mathsf{ABM.LTag})$ *with the following syntax:*

$\mathsf{ABM.Gen}(1^\lambda)$**:** *On input the security parameter, outputs an evaluation key ek, an inversion key ik, and a tag key tk. The evaluation key ek defines a set* $\mathcal{T} = \mathcal{T}_\mathsf{c} \times \{0,1\}^*$ *that contains the disjoint sets of* lossy tags $\mathcal{T}_\mathsf{loss} \subseteq \mathcal{T}$ *and* injective tags $\mathcal{T}_\mathsf{inj} \subseteq \mathcal{T}$. *Tags are of the form* $t = (t_c, t_a)$, *where* $t_c \in \mathcal{T}_\mathsf{c}$ *is the* core part *of the tag, and* $t_a \in \{0,1\}^*$ *is the* auxiliary part *of the tag.*

$\mathsf{ABM.Eval}(ek, t, x)$**:** *On input an evaluation key ek, a tag* $t \in \mathcal{T}$ *and a preimage* $x \in \mathcal{D}$, *outputs an image* $y \in \mathcal{RG}$.

$\mathsf{ABM.Invert}(ik, t, y)$**:** *On input an inversion key ik, a tag* $t \in \mathcal{T}_\mathsf{inj}$ *and an image* y, *outputs a preimage* $x \in \mathcal{D}$.

$\mathsf{ABM.LTag}(tk, t_a)$**:** *On input the tag key tk and auxiliary part of the tag* $t_a \in \{0,1\}^*$, *outputs a core tag* $t_c \in \mathcal{T}_\mathsf{c}$ *such that the tag* $t := (t_c, t_a) \in \mathcal{T}_\mathsf{loss}$.

We consider the following properties of ABM*:*

Correctness. *We require for* $(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$, $t \in \mathcal{T}_\mathsf{inj}$, *and* $x \in \mathcal{D}$ *that with all-but-negligible probability over the choice of the random coins used by* $\mathsf{ABM.Gen}$ *we have* $\mathsf{ABM.Invert}(ik, t, (\mathsf{ABM.Eval}(ek, t, x))) = x$.

Expansion. *We define the expansion of* ABM *as* $\chi := \log_2 |\mathcal{RG}| / \log_2 |\mathcal{D}|$.

Lossiness. *We say that* ABM *is* ℓ*-lossy if for* $(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$, *and all lossy tags* $t \in \mathcal{T}_\mathsf{loss}$, *with all-but negligible probability the image set* $\{\mathsf{ABM.Eval}(ek, t, x) \mid x \in \mathcal{D}\}$ *is of size at most* $|\mathcal{D}|/2^\ell$. *We define the relative lossiness as* $L := \ell / \log_2 |\mathcal{D}|$.

Indistinguishability. *We require that even multiple lossy tags are indistinguishable from random tags. I.e.,*

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{ABM},\mathcal{A}}(\lambda) := \left| \Pr[\mathcal{A}(1^\lambda, ek)^{\mathsf{ABM.LTag}(tk,\cdot)} = 1] - \Pr[\mathcal{A}(1^\lambda, ek)^{\mathcal{O}_{\mathcal{T}_\mathsf{c}}(\cdot)} = 1] \right|$$

is negligible for all PPT $\mathcal{A}$, *where* $(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$, *and* $\mathcal{O}_{\mathcal{T}_\mathsf{c}}(\cdot)$ *returns a uniform and independent core tag* $t_c \xleftarrow{\$} \mathcal{T}_\mathsf{c}$ *at each new query and consistently returns the same* t_c *if given query* t_a *occurs more than once.*

Evasiveness. *We require that non-injective tags are hard to find, even given multiple lossy tags, and an oracle* $\mathsf{isLossy}$ *that on input of a tag returns* 1 *if the tag is lossy and* 0 *if not. I.e.,*

$$\mathsf{Adv}^{\mathsf{eva}}_{\mathsf{ABM},\mathcal{A}}(\lambda) := \Pr[\mathcal{A}(1^\lambda, ek)^{\mathsf{ABM.LTag}(tk,\cdot), \mathsf{isLossy}(tk,\cdot)} \in \mathcal{T} \setminus \mathcal{T}_\mathsf{inj}]$$

is negligible with $(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$, *and for any PPT algorithm* $\mathcal{A}$ *that never outputs tags obtained through oracle queries (i.e.,* $\mathcal{A}$ *never outputs tags* $t = (t_c, t_a)$, *where* t_c *has been obtained by an oracle query* t_a*).*

ABM *is an* ℓ*-lossy ABM-LTF with expansion* χ *if it satisfies the above properties for* ℓ *and* χ.

For the application of IND-SO-CCA security, we need a slight variant of ABM-LTFs. Concretely, we require that values that are revealed during a ciphertext opening can be explained as uniformly chosen "without ulterior motive," if only their distribution is uniform. (This is called "invertible sampling" by Damgård & Nielsen [15].)

Definition 9 (Efficiently samplable and explainable). *A finite set S is* efficiently samplable and explainable *if any element of S can be explained as the result of a uniform sampling. Formally, there are PPT algorithms Samp_S, Expl_S, such that*

1. $\mathsf{Samp}_S(1^\lambda)$ *uniformly samples from S, and*
2. *for any $s \in S$, $\mathsf{Expl}_S(s)$ outputs random coins for Samp that are uniformly distributed among all random coins R with $\mathsf{Samp}_S(1^\lambda; R) = s$.*

Definition 10 (ABM-LTF with explainable tags [32]). *An ABM-LTF has* explainable tags *if the core part of tags is efficiently samplable and explainable. Formally, if we write $\mathcal{T} = \mathcal{T}_\mathsf{c} \times \mathcal{T}_\mathsf{a}$, where $\mathcal{T}_\mathsf{c}$ and $\mathcal{T}_\mathsf{a}$ denote the core and auxiliary parts of tags, then $\mathcal{T}_\mathsf{c}$ is efficiently samplable and explainable.*

2.6 Lossy authenticated encryption

Since the construction by Hofheinz [32] follows a hybrid approach, we require a suitable symmetric encryption scheme (we use the same definition as [32]):

Definition 11 (Lossy authenticated encryption). *A* lossy authenticated encryption scheme $\mathsf{LAE} = (\mathsf{E}, \mathsf{D})$ *with key space $\{0,1\}^{2\kappa}$ and message space $\{0,1\}^{\kappa}$ for some $\kappa = \kappa(\lambda)$ consists of the following two PPT algorithms:*

Encryption. $\mathsf{E}(K, msg)$, *for a key $K \in \{0,1\}^{2\kappa}$ and a message $msg \in \{0,1\}^{\kappa}$, outputs a (symmetric) ciphertext* ct.

Decryption. $\mathsf{D}(K, \mathsf{ct})$, *for a key $K \in \{0,1\}^{2\kappa}$ and a (symmetric) ciphertext* ct, *outputs a message $msg \in \{0,1\}^{\kappa}$ or $\perp$. (In the latter case, we say that* D *rejects* ct.*)*

We require the following:

Correctness. *We have $\mathsf{D}(K, \mathsf{E}(K, msg)) = msg$ for all $K \in \{0,1\}^{2\kappa}$ and $msg \in \{0,1\}^{\kappa}$.*

Authentication. *For an adversary $\mathcal{A}$, we let $\mathsf{Adv}^{\mathsf{auth}}_{\mathsf{LAE},\mathcal{A}}(\lambda)$ denote the probability that $\mathcal{A}$ succeeds in the following experiment:*

1. *$\mathcal{A}$, on input 1^λ, chooses a message $msg \in \{0,1\}^{\kappa}$, and gets an encryption $\mathsf{ct} = \mathsf{E}(K, msg)$ of msg under a freshly chosen key $K \overset{\$}{\leftarrow} \{0,1\}^{2\kappa}$.*
2. *$\mathcal{A}$ gets (many-time) oracle access to a decryption oracle $\mathsf{D}(K, \cdot)$ with hardwired key K.*
3. *$\mathcal{A}$ wins iff it manages to submit a decryption query $\mathsf{ct}' \neq \mathsf{ct}$ to D that is not rejected (i.e., for which $\mathsf{D}(K, \mathsf{ct}') \neq \perp$).*

We require that $\mathsf{Adv}^{\mathsf{auth}}_{\mathsf{LAE},\mathcal{A}}(\lambda)$ is negligible for every PPT $\mathcal{A}$.

Lossiness. *For $msg \in \{0,1\}^\kappa$, let $\mathcal{D}_{msg}$ be the distribution of $\mathsf{E}(K, msg)$ (for random $K \xleftarrow{\$} \{0,1\}^{2\kappa}$). We require that for any two $msg, msg' \in \{0,1\}^\kappa$, the distributions $\mathcal{D}_{msg}$ and $\mathcal{D}_{msg'}$ are identical. (That is, when K is unknown, a ciphertext reveals no information about the plaintext.)*

Lossy authenticated encryption schemes exist unconditionally. For instance, if we parse $K = (K_1, K_2) \in (\{0,1\}^\kappa)^2$, we can set $\mathsf{E}(K, msg) = (\rho, \tau) = (msg \oplus K_1, \mathsf{MAC}(K_2, \rho))$ for a message authentication code MAC that is strongly existentially unforgeable under one-time chosen-message attacks.

2.7 Selective Opening security

We use the following definitions from [32]:

Definition 12 (Efficiently re-samplable). *Let $N = N(\lambda) > 0$, $\kappa = \kappa(\lambda)$, and let* dist *be a joint distribution over $(\{0,1\}^\kappa)^N$. We say that* dist *is* efficiently re-samplable *if there is a PPT algorithm* $\mathsf{ReSamp}_{\mathsf{dist}}$ *such that for any $\mathcal{I} \subseteq [N]$ and any partial vector $\mathbf{msg}'_{\mathcal{I}} := (msg'^{(i)})_{i \in \mathcal{I}} \in (\{0,1\}^\kappa)^{|\mathcal{I}|}$,* $\mathsf{ReSamp}_{\mathsf{dist}}(\mathbf{msg}'_{\mathcal{I}})$ *samples from the distribution* dist*, conditioned on $msg^{(i)} = msg'^{(i)}$ for all $i \in \mathcal{I}$.*

We recall the definition of IND-SO-CCA security. In this game, the adversary first gets to specify a distribution from which message vectors will be sampled along with a resampling algorithm. Then it will be provided with a vector of ciphertexts encrypting a vector of messages drawn from the distribution specified earlier. It then selects a set of indices $\mathcal{I}$ to be opened by the challenger (indicated by running the adversary with the input `select`). The challenger opens the indicated ciphertexts (by revealing the random coins used during encryption along with the messages), and either provides the adversary with the initially encrypted message vector, or with a message vector that has been resampled (with the restriction that the opened messages are the same). This phase of the adversary is triggered with the input `output`. The output of this phase is the adversary's decision bit, i.e. whether it believes that the message vector it received is the encrypted one (indicated by 0) or whether it received a resampled vector (indicated by 1). The adversary wins if it guessed correctly.

Definition 13 (IND-SO-CCA security). *A PKE scheme* $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ *is* IND-SO-CCA secure *iff for every polynomially bounded functions $N = N(\lambda) > 0$ and $\kappa = \kappa(\lambda)$, and every stateful PPT adversary $\mathcal{A}$, the function*

$$\mathsf{Adv}^{\mathsf{cca\text{-}so}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},\mathcal{A},N}(\lambda) = 1\right] - \frac{1}{2} \right|$$

is negligible. Here, the experiment $\mathsf{Exp}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},\mathcal{A},N}(\lambda)$ *is defined as follows:*

```
Experiment $\mathsf{Exp}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},\mathcal{A},N}$
$b \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathsf{PKE.Gen}(1^\lambda)$
$(\mathsf{dist}, \mathsf{ReSamp}_{\mathsf{dist}}) \xleftarrow{\$} \mathcal{A}^{\mathsf{PKE.Dec}(sk,\cdot)}(pk)$
$\mathbf{msg}_0 := (msg^{(i)})_{i\in[N]} \xleftarrow{\$} \mathsf{dist}$
$\mathbf{R} := (R^{(i)})_{i\in[N]} \xleftarrow{\$} (\mathcal{R}_{\mathsf{PKE.Enc}})^N$
$\mathbf{C} := (C^{(i)})_{i\in[N]} := (\mathsf{PKE.Enc}(pk, msg^{(i)}; R^{(i)}))_{i\in[N]}$
$\mathcal{I} \xleftarrow{\$} \mathcal{A}^{\mathsf{PKE.Dec}(sk,\cdot)}(\mathtt{select}, \mathbf{C})$
$\mathbf{msg}_1 := \mathsf{ReSamp}_{\mathsf{dist}}(\mathbf{msg}_{\mathcal{I}})$
$\mathsf{out}[\mathcal{A}] \xleftarrow{\$} \mathcal{A}^{\mathsf{PKE.Dec}(sk,\cdot)}(\mathtt{output}, (msg^{(i)}, R^{(i)})_{i\in\mathcal{I}}, \mathbf{msg}_b)$
return $(\mathsf{out}[\mathcal{A}] = b)$
```

We only allow adversaries $\mathcal{A}$ that

- *always output efficiently re-samplable distributions* dist *over* $(\{0,1\}^\kappa)^N$ *with corresponding efficient re-sampling algorithms* $\mathsf{ReSamp}_{\mathsf{dist}}$*,*
- *never submit a received challenge ciphertext* $C^{(i)}$ *to their decryption oracle* $\mathsf{PKE.Dec}(sk,\cdot)$*, and*
- *always produce binary final output* $\mathsf{out}[\mathcal{A}]$*.*

3 Lossy Trapdoor Function Construction

In this section, we define our LTF construction which is based on the dual version of Regev's encryption scheme. As discussed in the introduction, injective evaluation keys of our LTF can be seen as economic dual-Regev encryptions of the gadget matrix $\mathbf{G}$ (as defined in Definition 4), while lossy keys are encryptions of 0. The indistinguishability of injective and lossy keys then follows from LWE.

To evaluate our LTF on an input $\mathbf{x}$ under the evaluation key $ek = \mathbf{C}$, one computes $\mathbf{C} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})$, where $\tilde{\mathbf{x}} := \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix}$. Hence, in the injective mode, the image is an encryption of $\tilde{\mathbf{x}}$ (i.e., $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} + \tilde{\mathbf{x}}$ for some $\mathbf{s}$ and small $\mathbf{e}$), and $\mathbf{x}$ can be obtained using the public key $\mathbf{A}$ and its trapdoor. In the lossy mode, in constrast, the image has the form $\mathbf{A} \cdot \mathbf{s} + \mathbf{e}$, where only $\mathbf{s} = \mathbf{S} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})$ and $\mathbf{e} = \mathbf{E} \cdot \mathbf{G}^{-1}(\tilde{\mathbf{x}})$ leak information about $\mathbf{x}$. Now since $\mathbf{S}$ is "very flat" and $\mathbf{E}$ has small norm, we can argue that $\mathbf{A} \cdot \mathbf{s} + \mathbf{e}$ loses a lot of entropy of $\mathbf{x}$.

Although it is natural to think about the evaluation key ek being a dual-Regev ciphertext, it will be more convenient to view ek as a dual-GSW ciphertext (i.e., dual-GSW encryption of 1 in the injective case and dual-GSW encryption of 0 in the lossy case) in the formal LTF description and proofs. Looking ahead to Section 4, this view will help us to convert the LTF construction into a ABM-LTF construction, where we make use of the fully homomorphic properties of the dual-GSW encryption scheme.

Let n, m, q and α be the LWE parameters. As in the previous section, we denote $w := n\lceil \log q \rceil$ and $N := m\lceil \log q \rceil$. The domain $\mathcal{D}$ of our LTF is $\mathbb{Z}_p^{\overline{m}}$ with $p < q$ and $\overline{m} < m$; for convenience, let $u := m - \overline{m}$ and $c := \lfloor q/p \rfloor$. Our LTF construction is defined as follows:

$\mathsf{LTF.IGen}(1^\lambda)$**:** On input the security parameter, proceed as follows:

1. Define $\mathbf{A} := \begin{pmatrix} \overline{\mathbf{A}} \\ \underline{\mathbf{A}} \end{pmatrix} \in \mathbb{Z}_q^{m \times n}$, where $(\mathbf{R}, \overline{\mathbf{A}}) \xleftarrow{\$} \mathsf{GenTrap}(n, u - w, q)$ and $\underline{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{\overline{m} \times n}$.
2. Let $\mathbf{C}$ be the Dual-GSW encryption of 1; that is
   (a) Sample $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n \times N}$ and $\mathbf{E} \xleftarrow{\$} D_{\alpha q}^{m \times N}$.
   (b) Set $\mathbf{C} := \mathbf{AS} + \mathbf{E} + \mathbf{G} \in \mathbb{Z}_q^{m \times N}$.
3. Output $(\mathit{ek}, \mathit{ik})$, where $\mathit{ek} := \mathbf{C}$, $\mathit{ik} := (\mathbf{R}, \mathbf{A})$.

$\mathsf{LTF.Eval}(\mathit{ek}, \mathbf{x})$**:** On input the evaluation key $\mathit{ek} = \mathbf{C}$ and $\mathbf{x} \in \mathbb{Z}_p^{\overline{m}} \subset \mathbb{Z}_q^{\overline{m}}$, output

$$\mathbf{y} := \mathbf{C} \cdot \mathbf{G}^{-1} \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix} \in \mathbb{Z}_q^m.$$

$\mathsf{LTF.Invert}(\mathit{ik}, \mathbf{y})$**:** On input the inversion key ik and image $\mathbf{y}$, proceed as follows:

1. Parse $\mathit{ik} = \left(\mathbf{R}, \mathbf{A} = \begin{pmatrix} \overline{\mathbf{A}} \\ \underline{\mathbf{A}} \end{pmatrix}\right)$, where $\overline{\mathbf{A}} \in \mathbb{Z}_q^{u \times n}$ and $\underline{\mathbf{A}} \in \mathbb{Z}_q^{\overline{m} \times n}$, and $\mathbf{y} = \begin{pmatrix} \overline{\mathbf{y}} \\ \underline{\mathbf{y}} \end{pmatrix}$, where $\overline{\mathbf{y}} \in \mathbb{Z}_q^u$ and $\underline{\mathbf{y}} \in \mathbb{Z}_q^{\overline{m}}$.
2. Compute $(\mathbf{s}, \mathbf{e}) := \mathsf{Invert}(\overline{\mathbf{A}}, \overline{\mathbf{y}}, \mathbf{R})$
3. Output $\mathbf{x} \in \mathbb{Z}_p^{\overline{m}}$ defined as $\mathbf{x} := (\underline{\mathbf{y}} - \underline{\mathbf{A}}\mathbf{s} \bmod q) \operatorname{div} c$.

$\mathsf{LTF.LGen}(1^\lambda)$**:** On input the security parameter, proceed as follows:

1. Let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$.
2. Let $\mathbf{C}$ be the Dual-GSW encryption of 0; that is
   (a) Sample $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n \times N}$ and $\mathbf{E} \xleftarrow{\$} D_{\alpha q}^{m \times N}$.
   (b) Set $\mathbf{C} := \mathbf{AS} + \mathbf{E} \in \mathbb{Z}_q^{m \times N}$.
3. Output $\mathit{ek}' := \mathbf{C}$.

We now prove that the construction from above is an LTF with constant expansion. For interpretation of the parameter bounds and to see that arbitrary constant relative lossiness can be achieved, we refer to Appendix B.1.

Theorem 1. *Assuming hardness of* $\mathsf{LWE}_{n,q,\alpha,m}$*, the LTF construction* $\mathsf{LTF} = (\mathsf{LTF.IGen}, \mathsf{LTF.Eval}, \mathsf{LTF.Invert}, \mathsf{LTF.LGen})$ *is an* ℓ*-lossy LTF with constant expansion for parameters that satisfy the following constraints:*

$$\log q = O(\log p), \quad m = O(\overline{m}), \quad 2^{-n} = \mathsf{negl}(\lambda), \quad u \geq 2(w + n + \log(w) - 1),$$

$$\alpha < \frac{1}{\sqrt{\lambda} \cdot 2 \cdot N \max\{N, p\}}, \quad \ell = \overline{m}\log(p) - n \log q - m \log(2N\sqrt{\lambda}\alpha q + 1).$$

In particular, for any PPT adversary $\mathcal{A}$*, there exists an LWE-distinguisher* $\mathcal{B}$ *that runs in about the same time as* $\mathcal{A}$*, such that*

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LTF},\mathcal{A}}(\lambda) \leq 2^{-n} + 2 \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}}(\lambda).$$

Proof. In the following we prove each property of our ABM-LTF separately.
Constant expansion. Our LTF constructed from Dual-GSW, achieves constant expansion. For $\mathbf{x} \in \mathbb{Z}_p^{\overline{m}}$, we get $\mathbf{y} := \mathsf{LTF.Eval}(\mathit{ek}, \mathbf{x}) \in \mathbb{Z}_q^m$. Hence, the expansion χ can be computed as

$$\chi = \frac{m \cdot \lceil \log q \rceil}{\overline{m} \cdot \lceil \log p \rceil} = \frac{O(\overline{m}) \cdot O(\log p)}{\overline{m} \cdot \lceil \log p \rceil} = O(1).$$

Correctness. Let $(ek, ik) \xleftarrow{\$} \mathsf{LTF.IGen}(1^\lambda)$, $\mathbf{x} \in \mathbb{Z}_p^m$ and $\mathbf{y} := \mathsf{LTF.Eval}(ek, \mathbf{x})$. We know that $ik = (\mathbf{R}, \mathbf{A})$, where $\mathbf{A} = \begin{pmatrix} \overline{\mathbf{A}} \\ \underline{\mathbf{A}} \end{pmatrix}$, $\mathbf{R}$ is a trapdoor for $\overline{\mathbf{A}}$ and $\mathbf{y}$ can be expressed as

$$\mathbf{y} := \mathbf{C} \cdot \mathbf{G}^{-1} \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix} \mod q,$$

where $\mathbf{C}$ is a Dual-GSW encryption of 1, i.e., $\mathbf{C} = \mathbf{A} \cdot \mathbf{S} + \mathbf{E} + \mathbf{G}$. Setting $\mathbf{s} := \mathbf{S} \cdot \mathbf{G}^{-1} \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix}$ and $\mathbf{e} := \begin{pmatrix} \overline{\mathbf{e}} \\ \underline{\mathbf{e}} \end{pmatrix} := \mathbf{E} \cdot \mathbf{G}^{-1} \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix}$, we can express $\mathbf{y}$ as

$$\mathbf{y} = \begin{pmatrix} \overline{\mathbf{y}} \\ \underline{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{A}}\mathbf{s} + \overline{\mathbf{e}} \\ \underline{\mathbf{A}}\mathbf{s} + \underline{\mathbf{e}} + c \cdot \mathbf{x} \end{pmatrix} \mod q.$$

Since $\mathbf{G}^{-1} \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix}$ is a binary vector and $\mathbf{E} \xleftarrow{\$} D_{\alpha q}^{m \times N}$, we know that with all-but-negligible probability

$$||\overline{\mathbf{e}}||_\infty \leq ||\mathbf{e}||_\infty \leq ||\mathbf{E}||_\infty \leq N \cdot \sqrt{\lambda} \cdot \alpha q,$$

where the last inequality stems from Remark 1. Hence, by Lemma 2, with all-but-negligible probability $\mathsf{Invert}(\overline{\mathbf{A}}, \overline{\mathbf{y}}, \mathbf{R})$ returns $\mathbf{s}$ and $\overline{\mathbf{e}}$ whenever

$$||\overline{\mathbf{e}}||_\infty \leq N \cdot \sqrt{\lambda} \cdot \alpha q < q/(2 \cdot \log(q) \cdot (u - w + 1)).$$

Having $\mathbf{s}$, and hence $\underline{\mathbf{A}}\mathbf{s}$, one can compute $\mathbf{x}$ from $\underline{\mathbf{y}}$ as

$$(\underline{\mathbf{y}} - \underline{\mathbf{A}}\mathbf{s}) \text{ div } c = (\underline{\mathbf{A}}\mathbf{s} + \underline{\mathbf{e}} + c \cdot \mathbf{x} - \underline{\mathbf{A}}\mathbf{s}) \text{ div } c = (\underline{\mathbf{e}} + c \cdot \mathbf{x}) \text{ div } c = \mathbf{x},$$

where the last equality holds if $||\mathbf{e}||_\infty < c/2$.

Thus, the correctness property is satisfied for any parameters such that

$$N \cdot \sqrt{\lambda} \cdot \alpha q < \min \left\{ \frac{q}{2 \cdot \log(q) \cdot (u - w + 1)}, \frac{c}{2} \right\}.$$

Since $N \geq m \log(q) \geq (u - w + 1) \log(q)$, the above inequality is satisfied if

$$\alpha < \min \left\{ \frac{1}{\sqrt{\lambda} \cdot 2 \cdot N^2}, \frac{c}{q \cdot \sqrt{\lambda} \cdot 2 \cdot N} \right\} \leq \frac{1}{\sqrt{\lambda} \cdot 2 \cdot N \max\{N, p\}}.$$

Lossiness. In order to show that our LTF construction is ℓ-lossy, we need to prove that for $ek' \xleftarrow{\$} \mathsf{LTF.LGen}(1^\lambda)$, with all-but-negligible probability, the image set $\{\mathsf{LTF.Eval}(ek', \mathbf{x}) \mid \mathbf{x} \in \mathcal{D}\}$ is of size at most $|\mathcal{D}|/2^\ell$.

Let $\mathbf{y}$ be an arbitrary element in $\mathsf{LTF.Eval}(ek', \mathcal{D})$. Since $ek' = \mathbf{C}$ is lossy (i.e., $\mathbf{C} = \mathbf{AS} + \mathbf{E}$), setting $\mathbf{s}$ and $\mathbf{e}$ as in the correctness proof above, we can now express $\mathbf{y}$ as $\mathbf{y} = \mathbf{As} + \mathbf{e} \mod q$. Since $\mathbf{s} \in \mathbb{Z}_q^n$ and $||\mathbf{e}||_\infty \leq ||\mathbf{E}||_\infty \leq N \cdot \sqrt{\lambda} \cdot \alpha q$ with all-but-negligible probability, we obtain

$$|\{\mathsf{LTF.Eval}(ek', \mathbf{x}) \mid \mathbf{x} \in \mathcal{D}\}| \leq q^n \cdot (2 \cdot N \cdot \sqrt{\lambda} \cdot \alpha q + 1)^m$$

with all-but-negligible probability. On the other hand, we have $|\mathcal{D}| = p^{\overline{m}}$. Hence, we obtain ℓ-lossiness if we choose parameters such that

$$q^n \cdot (2 \cdot N \cdot \sqrt{\lambda} \cdot \alpha q + 1)^m \leq \frac{p^{\overline{m}}}{2^\ell}.$$

Indistinguishability. The fact that the first output $ek = \mathbf{AS} + \mathbf{E} + \mathbf{G}$ of $\mathsf{LTF.IGen}(1^\lambda)$ is indistinguishable from the output $ek' = \mathbf{AS}+\mathbf{E}$ of $\mathsf{LTF.LGen}(1^\lambda)$ follows straight-forward from Lemma 2 and the $\mathsf{LWE}_{n,q,\alpha,m}$ assumption. Formally, we can proceed by defining a series of game hops, where we denote by one_i the event that the adversary $\mathcal{A}$ outputs 1 in the i-th game.

Game 1**:** This is the game, where the adversary $\mathcal{A}$ receives an injective evaluation key ek.

Game 2**:** The game is defined as Game 1, except that the injective evaluation key ek is generated differently. Concretely, the generation of the matrix $\mathbf{A}$ in step 1 of $\mathsf{LTF.IGen}$ is replaced by sampling $\mathbf{A}$ uniformly at random, i.e., $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$. Since $\mathsf{GenTrap}$ is called with $u-w$ and we assume that $u \geq 2(w+n+\log(w)-1)$, we can apply Lemma 2 to argue that the statistical distance between the matrix $\mathbf{A}$ generated as in Game 1 and matrix $\mathbf{A}$ chosen uniformly at random as in Game 2 is 2^{-n} and hence $|\Pr[\mathsf{one}_2] - \Pr[\mathsf{one}_1]| \leq 2^{-n}$.

Game 3**:** In this game, the adversary $\mathcal{A}$ receives a uniformly random matrix $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m\times N}$ instead of an injective evaluation key ek. We have by Lemma 4 $|\Pr[\mathsf{one}_3] - \Pr[\mathsf{one}_2]| \leq N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}_1}(\lambda)$, for an LWE-distinguisher $\mathcal{B}_1$ that runs in about the same time as $\mathcal{A}$.

Game 4**:** This is the game, where the adversary $\mathcal{A}$ receives a lossy evaluation key ek'. Again, by Lemma 4, we have $|\Pr[\mathsf{one}_4] - \Pr[\mathsf{one}_3]| \leq N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}_2}(\lambda)$, for an LWE-distinguisher $\mathcal{B}_2$ that runs in about the same time as $\mathcal{A}$.
Combining these bounds, we obtain

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LTF},\mathcal{A}}(\lambda) = |\mathsf{one}_1 - \mathsf{one}_4| \leq 2^{-n} + 2 \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}}(\lambda)$$

for an LWE-distinguisher $\mathcal{B}$ that runs in about the same time as $\mathcal{A}$.

4 All-But-Many Lossy Trapdoor Function Construction

Using the methods from [12, 38], we now convert the LTF construction into an ABM-LTF. That is, our ABM-LTF tags are of the form $t = (t_c, t_a) \in \{0,1\}^\lambda \times \{0,1\}^*$ and are lossy if and only if $t_c = \mathsf{PRF}_K(t_a)$, for a pseudorandom function PRF and key $K \in \{0,1\}^\lambda$. The evaluation key of the ABM-LTF then consists of λ dual-GSW ciphertexts, each of which encrypts one bit of the PRF key K. During ABM-LTF evaluation, we make use of the full homomorphism of the dual-GSW encryption scheme and evaluate the PRF through homomorphic computations on the encrypted key bits.

To this end, we need to assume a PRF family $\mathsf{PRF}\colon \{0,1\}^\lambda \times \{0,1\}^* \to \{0,1\}^\lambda$, such that for each fixed input $t_a \in \{0,1\}^*$, the map $K \mapsto \mathsf{PRF}(K, t_a)$ can be computed by a circuit of NAND-gates of depth $d \in O(\log\lambda)$. We can instantiate such a PRF family using the LWE-based PRF construction of Boneh et al. [11] by first hashing the input using a collision resistant hash function.

The domain $\mathcal{D}$ of our ABM-LTF is the same as for the LTF, namely $\mathbb{Z}_p^{\overline{m}}$ with $p < q$ and $\overline{m} < m$. As in the previous sections, let n, m, q and α be the LWE parameters and let us denote $w = n\lceil \log q \rceil$ and $N = m\lceil \log q \rceil$, $u := m - \overline{m}$ and $c := \lfloor q/p \rfloor$. We define our ABM-LTF as follows:

$\mathsf{ABM.Gen}(1^\lambda)$**:** On input the security parameter, proceed as follows:

1. Define $\mathbf{A} := \begin{pmatrix} \overline{\mathbf{A}} \\ \underline{\mathbf{A}} \end{pmatrix} \in \mathbb{Z}_q^{m \times n}$, for $(\mathbf{R}, \overline{\mathbf{A}}) \xleftarrow{\$} \mathsf{GenTrap}(n, u - w, q)$ and $\underline{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{\overline{m} \times n}$.
2. Sample a PRF key $K = (K[i])_{i \in [\lambda]} \xleftarrow{\$} \{0,1\}^\lambda$.
3. For each $i \in [\lambda]$, let $\mathbf{C}_i$ be the Dual-GSW encryption of $K[i]$; that is
   (a) Sample $\mathbf{S}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times N}$ and $\mathbf{E}_i \xleftarrow{\$} D_{\alpha q}^{m \times N}$.
   (b) Set $\mathbf{C}_i := \mathbf{A}\mathbf{S}_i + \mathbf{E}_i + K[i] \cdot \mathbf{G} \in \mathbb{Z}_q^{m \times N}$.
4. Output (ek, ik, tk), where $ek := (\mathbf{C}_i)_{i \in [\lambda]}$, $ik := (\mathbf{R}, \mathbf{A})$, $tk := K$.

The tag space is $\mathcal{T} = \{0,1\}^\lambda \times \{0,1\}^*$, i.e., $\mathcal{T}_{\mathsf{c}} = \{0,1\}^\lambda$. The lossy tags are $\mathcal{T}_{\mathsf{loss}} = \{(t_c, t_a) \mid t_c = \mathsf{PRF}_K(t_a)\}$ and injective tags are $\mathcal{T}_{\mathsf{inj}} = \mathcal{T} \setminus \mathcal{T}_{\mathsf{loss}}$.

$\mathsf{ABM.LTag}(t_a, K)$**:** On input an auxiliary tag $t_a \in \{0,1\}^*$ and a PRF key $K \in \{0,1\}^\lambda$, output a core tag $t_c := \mathsf{PRF}_K(t_a)$.

$\mathsf{ABM.Eval}(ek, t, \mathbf{x})$**:** On input the evaluation key $ek = (\mathbf{C}_i)_{i \in [\lambda]}$, tag $t = (t_c, t_a) \in \mathcal{T}$ and $\mathbf{x} \in \mathbb{Z}_p^{\overline{m}} \subset \mathbb{Z}_q^{\overline{m}}$, proceed as follows

1. Let RC_t be a circuit consisting of "NAND" gates of depth d computing the function $f_{\mathsf{RC}}(\cdot, t)$ defined as
$$f_{\mathsf{RC}}(K, t) := \begin{cases} 0, & \text{if } t_c = \mathsf{PRF}_K(t_a), \\ 1, & \text{otherwise.} \end{cases}$$
2. Using the FHE scheme, evaluate RC_t on the PRF key in its encrypted form, i.e., $\mathbf{C}_t \leftarrow \mathsf{FHE.Eval}(\mathsf{RC}_t, (\mathbf{C}_i)_{i \in [\lambda]})$.
3. Output $\mathbf{y} := \mathbf{C}_t \cdot \mathbf{G}_{m,q}^{-1} \begin{pmatrix} \mathbf{0} \\ c \cdot \mathbf{x} \end{pmatrix} \in \mathbb{Z}_q^m$.

$\mathsf{ABM.Invert}(ik, \mathbf{y}, t)$**:**

1. Parse $ik = \left(\mathbf{R}, \mathbf{A} = \begin{pmatrix} \overline{\mathbf{A}} \\ \underline{\mathbf{A}} \end{pmatrix}\right)$, where $\overline{\mathbf{A}} \in \mathbb{Z}_q^{u \times n}$ and $\underline{\mathbf{A}} \in \mathbb{Z}_q^{\overline{m} \times n}$, $\mathbf{y} = \begin{pmatrix} \overline{\mathbf{y}} \\ \underline{\mathbf{y}} \end{pmatrix}$, where $\overline{\mathbf{y}} \in \mathbb{Z}_q^u$ and $\underline{\mathbf{y}} \in \mathbb{Z}_q^{\overline{m}}$, and $t = (t_c, t_a)$.
2. Compute $(\mathbf{s}_t, \mathbf{e}_t) := \mathsf{Invert}(\overline{\mathbf{A}}, \overline{\mathbf{y}}, \mathbf{R})$.
3. Output $\mathbf{x} \in \mathbb{Z}_p^{\overline{m}}$ defined as $\mathbf{x} := (\underline{\mathbf{y}} - \underline{\mathbf{A}}\mathbf{s}_t \mod q) \operatorname{div} c$.

We now prove that the construction from above is an ABM-LTF with constant expansion. For interpretation of the parameter bounds and to see that arbitrarily large constant relative lossiness $L := \ell / \log_2 |\mathcal{D}|$ can be achieved, we refer to Appendix B.2.

Theorem 2. *Assuming hardness of* $\mathsf{LWE}_{n,q,\alpha,m}$ *and the security of* PRF*, the construction* $\mathsf{ABM} = (\mathsf{ABM.Gen}, \mathsf{ABM.Eval}, \mathsf{ABM.Invert}, \mathsf{ABM.LTag})$ *is an* ℓ*-lossy ABM-LTF with constant expansion for parameters that satisfy the following constraints:*

$$\begin{aligned}\log q = O(\log p), \quad m = O(\overline{m}), \quad 2^{-n} = \mathsf{negl}(\lambda), \quad d \in O(\log \lambda),\\ \alpha < \frac{1}{\sqrt{\lambda}\cdot 4^{d+1}\cdot N^2 \max\{N,p\}}, \quad u \geq 2(w+n+\log(w)-1),\\ \ell = \overline{m}\log(p) - n\log(q) - m\log(\alpha q\cdot\sqrt{\lambda}\cdot 4^{d+1}\cdot N^2+1).\end{aligned}$$

In particular, for any PPT adversary $\mathcal{A}_{\mathsf{ind}}$*, there exists an LWE-distinguisher* $\mathcal{B}$ *and a PRF-distinguisher* $\mathcal{C}$ *that run in about the same time as* $\mathcal{A}_{\mathsf{ind}}$*, such that*

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{ABM},\mathcal{A}_{\mathsf{ind}}}(\lambda) \leq 2^{-n+1} + 2\cdot\lambda\cdot N\cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{PRF},\mathcal{C}}(\lambda),$$

where $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{PRF},\mathcal{C}}$ *denotes the advantage of* $\mathcal{C}$ *in the PRF security experiment. Moreover, for any PPT adversary* $\mathcal{A}_{\mathsf{eva}}$*, there exists an LWE-distinguisher* $\mathcal{B}$ *and a PRF-distinguisher* $\mathcal{C}$ *that run in about the same time as* $\mathcal{A}_{\mathsf{eva}}$*, such that*

$$\mathsf{Adv}^{\mathsf{eva}}_{\mathsf{ABM},\mathcal{A}_{\mathsf{eva}}}(\lambda) \leq 2^{-n} + \lambda\cdot N\cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{PRF},\mathcal{C}}(\lambda) + \frac{Q}{2^{\lambda}},$$

where Q *denotes the number of queries* $\mathcal{A}_{\mathsf{eva}}$ *made to* $\mathsf{isLossy}$ *oracle.*

Proof. The proof of constant expansion is exactly the same as for our LTF construction (see proof of Theorem 1).
Correctness. Let $(ek, ik, tk) \overset{\$}{\leftarrow} \mathsf{ABM.Gen}(1^\lambda)$, $t = (t_c, t_a) \in \mathcal{T}_{\mathsf{inj}}$, $\mathbf{x} \in \mathbb{Z}_p^{\overline{m}}$ and $\mathbf{y} := \mathsf{ABM.Eval}(ek, t, \mathbf{x})$. We know that $tk = K$ is the PRF key and $ik = (\mathbf{R}, \mathbf{A})$, where $\mathbf{A} = \begin{pmatrix}\overline{\mathbf{A}}\\ \underline{\mathbf{A}}\end{pmatrix}$, $\mathbf{R}$ is a trapdoor for $\overline{\mathbf{A}}$. Moreover, $\mathbf{y}$ can be expressed as $\mathbf{y} := \mathbf{C}_t \cdot \mathbf{G}^{-1}\begin{pmatrix}\mathbf{0}\\ c\cdot\mathbf{x}\end{pmatrix} \mod q$, where $\mathbf{C}_t$ is an encryption of $\mathsf{RC}_t(K)$ with noise bounded by Lemma 4 as

$$\mathsf{noise}_{\mathbf{A},\mathsf{RC}_t(K)}(\mathbf{C}_t) \leq 2\cdot 4^d \cdot N \cdot \max_{i\in[\eta]}(\mathsf{noise}_{\mathbf{A},\mu_i}(\mathbf{C}_i)) =: B.$$

Since t is injective (i.e., $t_c \neq \mathsf{PRF}_K(t_a)$), we know that $\mathsf{RC}_t(K) = 1$, and thus there exist $\mathbf{S}_t$ and $\mathbf{E}_t$ with $||\mathbf{E}_t||_\infty \leq B$ such that $\mathbf{C}_t = \mathbf{A}\cdot\mathbf{S}_t + \mathbf{E}_t + \mathbf{G}$. Setting $\mathbf{s}_t := \mathbf{S}_t \cdot \mathbf{G}^{-1}\begin{pmatrix}\mathbf{0}\\ c\cdot\mathbf{x}\end{pmatrix}$ and $\mathbf{e}_t := \begin{pmatrix}\overline{\mathbf{e}}_t\\ \underline{\mathbf{e}}_t\end{pmatrix} := \mathbf{E}_t\cdot\mathbf{G}^{-1}\begin{pmatrix}\mathbf{0}\\ c\cdot\mathbf{x}\end{pmatrix}$, we can now express $\mathbf{y}$ as $\mathbf{y} = \begin{pmatrix}\overline{\mathbf{y}}\\ \underline{\mathbf{y}}\end{pmatrix} = \begin{pmatrix}\overline{\mathbf{A}}\mathbf{s}_t + \overline{\mathbf{e}}_t\\ \underline{\mathbf{A}}\mathbf{s}_t + \underline{\mathbf{e}}_t + c\cdot\mathbf{x}\end{pmatrix} \mod q$. Since $\mathbf{G}^{-1}\begin{pmatrix}\mathbf{0}\\ c\cdot\mathbf{x}\end{pmatrix}$ is a binary vector, we know that $||\mathbf{e}_t||_\infty \leq ||\mathbf{E}_t||_\infty \leq B$. Hence, by Lemma 2, $\mathsf{Invert}(\overline{\mathbf{A}}, \overline{\mathbf{y}}, \mathbf{R})$ returns $\mathbf{s}_t$ and $\overline{\mathbf{e}}_t$ whenever $B < q/(2\cdot\log(q)\cdot(u-w+1))$. Having $\mathbf{s}_t$, and hence $\underline{\mathbf{A}}\mathbf{s}_t$, one can compute $\mathbf{x}$ from $\underline{\mathbf{y}}$ as

$$(\underline{\mathbf{y}} - \underline{\mathbf{A}}\mathbf{s}_t) \text{ div } c = (\underline{\mathbf{A}}\mathbf{s}_t + \underline{\mathbf{e}}_t + c\cdot\mathbf{x} - \underline{\mathbf{A}}\mathbf{s}_t) \text{ div } c = (\underline{\mathbf{e}}_t + c\cdot\mathbf{x}) \text{ div } c = \mathbf{x},$$

where the last equality holds if $||\mathbf{e}_t||_\infty < c/2$. Thus, we obtain correctness if

$$B < \min\left\{\frac{q}{2 \cdot \log(q) \cdot (u - w + 1)}, \frac{c}{2}\right\}. \tag{1}$$

Since $\mathbf{C}_i = \mathbf{A}\mathbf{S}_i + \mathbf{E_i} + K[i] \cdot \mathbf{G}$ with $\mathbf{E}_i \xleftarrow{\$} D_{\alpha q}^{m \times N}$, we have $\mathsf{noise}_{\mathbf{A},K[i]}(\mathbf{C}_i) = ||\mathbf{E}_i||_\infty \leq N \cdot \sqrt{\lambda} \cdot \alpha q$ for all $i \in [\lambda]$ with all-but-negligible (in λ) probability, and therefore the correctness property is satisfied for any parameters such that

$$2 \cdot 4^d \cdot N^2 \cdot \alpha q \cdot \sqrt{\lambda} < \min\left\{\frac{q}{2 \cdot \log(q) \cdot (u - w + 1)}, \frac{c}{2}\right\};$$

in particular if $\alpha < \min\left\{\frac{1}{\sqrt{\lambda} \cdot 4^{d+1} \cdot N^3}, \frac{c}{q \cdot \sqrt{\lambda} \cdot 4^{d+1} \cdot N^2}\right\} \leq \frac{1}{\sqrt{\lambda} \cdot 4^{d+1} \cdot N^2 \max\{N,p\}}$.

Lossiness. In order to show that ABM is ℓ-lossy, we need to prove that for $(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$ and all lossy tags $t = (t_c, t_a) \in \mathcal{T}_{\mathsf{loss}}$, with all-but-negligible probability the image set $\{\mathsf{ABM.Eval}(ek, t, \mathbf{x}) \mid \mathbf{x} \in \mathcal{D}\}$ is of size at most $|\mathcal{D}|/2^\ell$.

Let t be lossy and $\mathbf{y} := \mathbf{C}_t \cdot \mathbf{G}^{-1}\begin{pmatrix}\mathbf{0} \\ c \cdot \mathbf{x}\end{pmatrix} \mod q$, be an arbitrary element in $\{\mathsf{ABM.Eval}(ek, t, \mathbf{x}) \mid \mathbf{x} \in \mathcal{D}\}$. Since t is lossy (i.e., $t_c = \mathsf{PRF}_K(t_a)$), we know that $\mathsf{RC}_t(K) = 0$, and thus there exist $\mathbf{S}_t$ and $\mathbf{E}_t$ with $||\mathbf{E}_t||_\infty \leq B$ such that $\mathbf{C}_t = \mathbf{A} \cdot \mathbf{S}_t + \mathbf{E}_t$. Setting $\mathbf{s}_t := \mathbf{S}_t \cdot \mathbf{G}^{-1}\begin{pmatrix}\mathbf{0} \\ c \cdot \mathbf{x}\end{pmatrix}$ and $\mathbf{e}_t := \mathbf{E}_t \cdot \mathbf{G}^{-1}\begin{pmatrix}\mathbf{0} \\ c \cdot \mathbf{x}\end{pmatrix}$, we can now express $\mathbf{y}$ as $\mathbf{y} = \mathbf{A}\mathbf{s}_t + \mathbf{e}_t \mod q$. Since $\mathbf{s}_t \in \mathbb{Z}_q^n$ and $||\mathbf{e}_t||_\infty \leq ||\mathbf{E}_t||_\infty \leq B$, we obtain

$$\begin{aligned} |\{\mathsf{ABM.Eval}(ek, t, \mathbf{x}) \mid \mathbf{x} \in \mathcal{D}\}| &\leq q^n \cdot (2B + 1)^m \\ &\leq q^n \cdot \left(4^{d+1} \cdot N^2 \cdot \alpha q \cdot \sqrt{\lambda} + 1\right)^m \end{aligned}$$

with all-but-negligible probability. On the other hand, we have $|\mathcal{D}| = p^{\overline{m}}$. Hence, we obtain ℓ-lossiness if we choose parameters such that

$$q^n \cdot \left(\alpha q \cdot \sqrt{\lambda} \cdot 4^{d+1} \cdot N^2 + 1\right)^m \leq \frac{p^{\overline{m}}}{2^\ell}.$$

Indistinguishability. The proof very closely follows the proof of indistinguishability of the GSW-based scheme of Libert et al. [38, Lemma 14].

Let $\mathcal{A}$ be a PPT adversary and let us fix $(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$. We proceed by defining a series of game hops, where we denote by one_i the event that the adversary $\mathcal{A}$ outputs 1 in the i-th game.

Game 1: $\mathcal{A}$ is given ek and interacts with the tag oracle $\mathsf{ABM.LTag}(tk, \cdot)$ that on input an auxiliary tag t_a outputs $t_c := \mathsf{PRF}_K(t_a)$ for $K = tk$.

Game 2: The game is defined as Game 1, except that the adversary $\mathcal{A}$ is given a differently generated evaluation key. Concretely, the generation of the matrix $\mathbf{A}$ in step 1 of $\mathsf{ABM.Gen}$ is replaced by sampling $\mathbf{A}$ uniformly at random, i.e., $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$. Since $\mathsf{GenTrap}$ is called with $u - w$ and we assume that $u \geq 2(w+n+\log(w)-1)$, we can apply Lemma 2 to argue that the statistical distance between the matrix $\mathbf{A}$ generated as in Game 1 and matrix $\mathbf{A}$ chosen uniformly at random as in Game 2 is 2^{-n} and hence $|\Pr[\mathsf{one}_2] - \Pr[\mathsf{one}_1]| \leq 2^{-n}$.

Game 3: In this game, the evaluation key given to $\mathcal{A}$ is generated by sampling independent and uniform matrices $\mathbf{C}_i \xleftarrow{\$} \mathbb{Z}_q^{m \times N}$, $i \in [\lambda]$. The rest of the game is defined as Game 2. Indistinguishability of Game 3 from Game 2 follows from Lemma 4 by a standard hybrid argument: For $j \in \{0, \dots, \lambda\}$, let G_j denote the distribution of $\{\mathbf{C}_i\}_{i \in \lambda}$ defined as follows

$$\begin{aligned} \mathbf{C}_i &\xleftarrow{\$} \mathsf{FHE.Enc}(\mathbf{A}, b_i), \text{ for } i \in [1, \lambda - j] \\ \mathbf{C}_i &\xleftarrow{\$} \mathbb{Z}_q^{m \times N}, \text{ for } i \in [\lambda - j + 1, \lambda]. \end{aligned}$$

By Lemma 4 there exists an LWE-distinguisher $\mathcal{B}_1$ that runs in about the same time as $\mathcal{A}$ such that the advantage of the adversary $\mathcal{A}$ to distinguish G_j from G_{j+1}, for every $j \in \{0, \dots, \lambda - 1\}$ is upper bounded by $N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}_1}(\lambda)$. By triangular inequality, the advantage of the adversary $\mathcal{A}$ to distinguish G_0 from G_λ is upper bounded by $\lambda \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}_1}(\lambda)$. It follows $|\Pr[\mathsf{one}_3] - \Pr[\mathsf{one}_2]| \leq \lambda \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}_1}(\lambda)$.

Game 4: Compared to Game 3, the only change is in the oracle available to the adversary. The oracle $\mathsf{ABM.LTag}(tk, \cdot)$ is now replaced by the oracle $\mathcal{O}_{\mathcal{T}_\mathsf{c}}(\cdot)$ that returns a uniform and independent core tag $t_c \xleftarrow{\$} \mathcal{T}_\mathsf{c}$ at each new query and consistently returns the same t_c if the given query t_a occurs more than once.

Let us construct a PRF distinguisher $\mathcal{C}$ that uses $\mathcal{A}$ to win the PRF game. $\mathcal{C}$ first samples independent and uniform matrices $\{\mathbf{C}_i\}_{i \in [\lambda]}$ and sends them as ek to $\mathcal{A}$. When $\mathcal{A}$ makes a call t_a to its tag oracle, the distinguisher $\mathcal{C}$ forwards t_a to the PRF challenger and relays the reply back to $\mathcal{A}$. The distinguisher $\mathcal{C}$ outputs whatever $\mathcal{A}$ outputs.

If the PRF challenger is returning PRF values, the view of $\mathcal{A}$ is the same as in Game 3 as $\mathcal{C}$ always replies with $\mathsf{PRF}_{K^*}(t_a)$ when queried on t_a. Otherwise the view of $\mathcal{A}$ is the same as in Game 4 because $\mathcal{C}$ always replies with a random value when queried on t_a. Hence $|\Pr[\mathsf{one}_4] - \Pr[\mathsf{one}_3]| \leq \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{PRF},\mathcal{C}}(\lambda)$.

Game 5: This game is defined as Game 4 except that we change back how matrices $\mathbf{C}_i$ are generated. They are not sampled uniformly at random any more, but according to step 3 of the $\mathsf{ABM.Gen}$ algorithm. By a similar argument as for the indistinguishability of Game 2 and Game 3, we conclude that there exists an LWE-distinguisher $\mathcal{B}_2$ that runs in about the same time as $\mathcal{A}$ such that $|\Pr[\mathsf{one}_5] - \Pr[\mathsf{one}_4]| \leq \lambda \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}_2}(\lambda)$.

Game 6: The game is defined as Game 5 except that the adversary gets ek that was generated via $\mathsf{ABM.Gen}$. This, in particular, means that $\overline{\mathbf{A}}$ is not sampled uniformly at random as in Game 5 but via $\mathsf{GenTrap}(n, u - w, q)$. Again by Lemma 2, we know that the statistical distance between the two matrices is 2^{-n} and hence $|\Pr[\mathsf{win}_6] - \Pr[\mathsf{win}_5]| \leq 2^{-n}$.

To conclude, we have

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{ABM},\mathcal{A}}(\lambda) = |\mathsf{one}_1 - \mathsf{one}_6| \leq 2^{-n+1} + 2 \cdot \lambda \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{PRF},\mathcal{C}}(\lambda)$$

for some LWE-distinguisher $\mathcal{B}$ and some PRF distinguisher $\mathcal{C}$ that run in about the same time as $\mathcal{A}$.

Evasiveness. The proof very closely follows the proof of evasiveness of the GSW-based scheme of Libert et al. [38, Lemma 13].

Let $\mathcal{A}$ be a PPT adversary and let $(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$. We proceed by defining a series of game hops, where we denote by win_i the event that the adversary wins the i-th game, i.e, $\mathcal{A}$ outputs a tag $t^* = (t_c^*, t_a^*) \in \mathcal{T} \setminus \mathcal{T}_{\mathsf{inj}}$ that has not been obtained through a query to the $\mathsf{ABM.LTag}$ oracle.

Game 1: This game corresponds to the evasiveness experiment as of Definition 8. Namely, the adversary $\mathcal{A}$ is given ek and interacts with the tag oracle $\mathsf{ABM.LTag}(tk, \cdot)$ that on input an auxiliary tag t_a outputs $t_c := \mathsf{PRF}_K(t_a)$ for $K = tk$. Moreover, $\mathcal{A}$ has access to the $\mathsf{isLossy}(tk, \cdot)$ oracle that on input $t = (t_c, t_a)$ returns 1 iff $t_c = \mathsf{PRF}_K(t_a)$ and 0 otherwise. We have $\mathsf{Adv}^{\mathsf{eva}}_{\mathsf{ABM},\mathcal{A}}(\lambda) = \Pr[\mathsf{win}_1]$.

Game 2: The game is defined as Game 1, except that the adversary $\mathcal{A}$ is given a differently generated evaluation key. Concretely, the generation of the matrix $\mathbf{A}$ in step 1 of $\mathsf{ABM.Gen}$ is replaced by sampling $\mathbf{A}$ uniformly at random, i.e., $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$. Since $\mathsf{GenTrap}$ is called with $u - w$ and we assume that $u \geq 2(w+n+\log(w)-1)$, by Lemma 2 we have $|\Pr[\mathsf{win}_2] - \Pr[\mathsf{win}_1]| \leq 2^{-n}$.

Game 3: In this game, the evaluation key given to $\mathcal{A}$ is generated by sampling independent and uniform matrices $\mathbf{C}_i \xleftarrow{\$} \mathbb{Z}_q^{m \times N}$. The rest of the game is defined as Game 2. By Lemma 4 and a hybrid argument (similar to the argument justifying the switch from Game 2 to Game 3 in the proof of the indistinguishability property above), there exists an LWE-distinguisher $\mathcal{B}$ that runs in about the same time as $\mathcal{A}$ such that $|\Pr[\mathsf{win}_3] - \Pr[\mathsf{win}_2]| \leq \lambda \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}}(\lambda)$.

Game 4: Compared to Game 3, the only changes are in the oracles available to the adversary. Namely, instead of returning $t_c := \mathsf{PRF}_K(t_a)$ at each query $\mathsf{ABM.LTag}(tk, t_a)$, the $\mathsf{ABM.LTag}$ oracle returns $t_c := R(t_a) \in \{0,1\}^\lambda$, where R is a random function lazily defined by sampling a uniform λ-bit string at each new query $t_a \in \{0,1\}^*$. At each query $\mathsf{isLossy}(tk, t)$, the oracle outputs 1 iff $t_c = R(t_a)$ for $t = (t_c, t_a)$, and 0 otherwise. Given that R is a truly random function, $\Pr[\mathsf{win}_4] = \frac{Q}{2^\lambda}$, where Q is the number of queries to the $\mathsf{isLossy}(tk, \cdot)$ oracle. We now show that $|\Pr[\mathsf{win}_4] - \Pr[\mathsf{win}_3]|$ is bounded by the probability of breaking the security of the PRF.

Let us construct a PRF distinguisher $\mathcal{C}$ that uses $\mathcal{A}$ to win the PRF game. The distinguisher $\mathcal{C}$ first samples independent and uniform matrices $\{\mathbf{C}_i\}_{i \in [\lambda]}$ and sends them as ek to $\mathcal{A}$. When $\mathcal{A}$ makes a call t_a to $\mathsf{ABM.LTag}(tk, \cdot)$, the distinguisher $\mathcal{C}$ forwards t_a to the PRF challenger and relays the reply back to $\mathcal{A}$. When $\mathcal{A}$ makes a call $t = (t_c, t_a)$ to $\mathsf{isLossy}(tk, \cdot)$, the distinguisher $\mathcal{C}$ forwards t_a to the PRF challenger. If the reply equals t_c, $\mathcal{C}$ replies to $\mathcal{A}$ with 1. Otherwise $\mathcal{C}$ replies with 0. Once $\mathcal{A}$ outputs a tag $t = (t_c, t_a)$, $\mathcal{C}$ submits t_a to its challenger. If the reply equals t_c, $\mathcal{C}$ outputs 1. Otherwise $\mathcal{C}$ outputs 0.

If the PRF challenger is returning PRF values, the view of $\mathcal{A}$ is the same as in Game 3 as $\mathcal{C}$ always replies with $\mathsf{PRF}_{K^*}(t_a)$ when queried on t_a. Otherwise the view of $\mathcal{A}$ is the same as in Game 4 because $\mathcal{C}$ always replies with a random value when queried on t_a. Hence the advantage $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{PRF},\mathcal{C}}(\lambda)$ of the distinguisher $\mathcal{C}$ is at least $|\Pr[\mathsf{win}_4] - \Pr[\mathsf{win}_3]|$ as required.

To summarize, we have

$$\mathsf{Adv}^{\mathsf{eva}}_{\mathsf{ABM},\mathcal{A}}(\lambda) \leq 2^{-n} + \lambda \cdot N \cdot \mathsf{Adv}^{n,q,\alpha,m}_{\mathsf{LWE},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{PRF},\mathcal{C}}(\lambda) + \frac{Q}{2^{\lambda}}.$$

Remark 3. Note that the core tag space $\mathcal{T}_{\mathsf{c}} = \{0,1\}^{\lambda}$ is efficiently samplable and explainable. Hence ABM is an ABM-LTF with explainable tags according to Definition 10.

5 IND-SO-CCA security from ABM-LTFs

Having our ABM-LTF construction at hand, we are now prepared to present a IND-SO-CCA secure PKE with compact ciphertexts which is the main goal of our paper. To this end, let us recall the construction of IND-SO-CCA secure PKE based on ABM-LTFs by Hofheinz [32] which we follow. We require the following ingredients:

1. an LTF $\mathsf{LTF} = (\mathsf{LTF.IGen}, \mathsf{LTF.Eval}, \mathsf{LTF.Invert}, \mathsf{LTF.LGen})$ with domain $\{0,1\}^n$ (as in Definition 7) that is ℓ'-lossy,
2. an ABM-LTF with explainable tags $\mathsf{ABM} = (\mathsf{ABM.Gen}, \mathsf{ABM.Eval}, \mathsf{ABM.Invert}, \mathsf{ABM.LTag})$ with domain $\{0,1\}^n$ and tag set $\mathcal{T} = \mathcal{T}_{\mathsf{c}} \times \{0,1\}^*$ (as in Definition 10) that is ℓ-lossy,
3. a family $\mathcal{UH}$ of universal hash functions $h : \{0,1\}^n \to \{0,1\}^{2\kappa}$ for some $\kappa = \kappa(\lambda)$, so that for any $f : \{0,1\}^n \to \{0,1\}^{2n-(\ell+\ell')}$, it holds that $\mathsf{SD}\left((h, f(x), h(x))\,;\,(h, f(x), U)\right) = O(2^{-\lambda})$, where $h \overset{\$}{\leftarrow} \mathcal{UH}$, $x \overset{\$}{\leftarrow} \{0,1\}^n$, and $U \overset{\$}{\leftarrow} \{0,1\}^{2\kappa}$, and
4. a statistically secure lossy authenticated encryption scheme $\mathsf{LAE} = (\mathsf{E}, \mathsf{D})$ (see Definition 11) with 2κ-bit keys K, κ-bit messages msg, and ciphertexts of size 2κ.

Remark 4. The requirement in Item 3 above can be fulfilled for n linear in λ due to the Leftover Hash Lemma. We detail this in Appendix B.3

The PKE scheme from [32] $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ works as shown in Fig. 1.

We give a brief intuition of how the scheme works. The LTF and the ABM-LTF are used to "encrypt" a hash-pre-image of a symmetric key. The key is then used to encrypt the message using the LAE scheme. This allows for switching the LTF to lossy mode, and the ABM-LTF to lossy tags for the challenge ciphertexts in the security proof, allowing the reduction to ultimately switch the symmetric keys to random keys that are unrelated to the remaining ciphertext components.

```
Alg. $\mathsf{PKE.Gen}(1^\lambda)$
$(ek', ik') \xleftarrow{\$} \mathsf{LTF.lGen}(1^\lambda)$
$(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$
$h \xleftarrow{\$} \mathcal{UH}$
$pk := (ek', ek, h)$
$sk := (ik', ek, h)$
return $(pk, sk)$
```

```
Alg. $\mathsf{PKE.Enc}(pk, msg)$
parse $pk =: (ek', ek, h)$
$x \xleftarrow{\$} \{0,1\}^n$
$K := h(x)$
$\mathsf{ct} \xleftarrow{\$} \mathsf{E}(K, msg)$
$y' := f_{ek'}(x)$
$t_c := \mathsf{Samp}_{\mathcal{T}_c}(1^\lambda; R_{t_c})$
$y := f_{ek,(t_c,y')}(x)$
$C := (\mathsf{ct}, y', t_c, y)$
return $C$
```

```
Alg. $\mathsf{PKE.Dec}(sk, C)$
parse $sk =: (ik', ek, h)$,
    $C =: (\mathsf{ct}, y', t_c, y)$
$x \xleftarrow{\$} f^{-1}_{ik'}(y')$
if $y \neq f_{ek,(t_c,y')}(x)$
    return $\perp$
$K := h(x)$
$msg \xleftarrow{\$} \mathsf{D}(K, \mathsf{ct})$
return $msg$
```

Fig. 1. The construction of IND-SO-CCA secure encryption by Hofheinz [32].

Mending a gap in [32]. We rewrite the proof instead of using the one from the original work, as there is a gap in the original work. In particular, in Game 7 of the original work, the keys of the lossy authenticated encryption scheme used for generating challenge ciphertexts are switched to truly random keys. The proof does not mention however how challenge ciphertexts that were generated like this can be opened, as the preimage x of the key K would need to be revealed. We close this gap by introducing an additional game where the preimages x of the keys are resampled using an inefficient opening algorithm. When the keys are switched to random, we continue to use this inefficient opening algorithm to output preimages x matching the key K as well as the LTF and ABM-LTF images y' and y. This change means that all following games are inefficient, and we need to rely on statistical security for the game hops after this game.

Theorem 3. *If* LTF *is an LTF,* ABM *an ABM-LTF with explainable tags,* $\mathcal{UH}$ *an UHF family as described, and* LAE *a lossy authenticated encryption scheme, then* PKE *is IND-SO-CCA secure. In particular, for every IND-SO-CCA adversary* $\mathcal{A}$ *on* PKE *that makes at most* $q_{\mathsf{PKE.Dec}} = q_{\mathsf{PKE.Dec}}(\lambda)$ *decryption queries, there exist adversaries* $\mathcal{B}$, $\mathcal{C}$, $\mathcal{F}$, *and* $\mathcal{E}$ *of roughly same complexity as* $\mathcal{A}$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cca\text{-}so}}_{\mathsf{PKE},\mathcal{A}}(\lambda) \leq & \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{ABM},\mathcal{B}}(\lambda) + q_{\mathsf{PKE.Dec}}(\lambda) \cdot \mathsf{Adv}^{\mathsf{eva}}_{\mathsf{ABM},\mathcal{C}}(\lambda) + \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LTF},\mathcal{F}}(\lambda) \\ & + N \cdot \mathsf{Adv}^{\mathsf{auth}}_{\mathsf{LAE},\mathcal{E}}(\lambda) + O(N/2^\lambda). \end{aligned} \tag{2}$$

Remark 5. While the reduction depends on the number N of challenge ciphertexts, these dependencies only appear in statistical terms (when using the unconditionally secure lossy authenticated encryption from Section 2). On the other hand, the number of an adversary's decryption queries goes linearly into the reduction factor.

Remark 6. We note that when instantiated with our construction of an LTF and an ABM-LTF from sections Sections 3 and 4, respectively, we achieve constant expansion, i.e. the ciphertext size is only by a constant factor larger than the plaintext size.

We now turn to the proof of Theorem 3.

Proof. The proof largely follows [45, 27, 32]. Assume $N = N(\lambda) > 0$ and an IND-SO-CCA adversary $\mathcal{A}$ that makes exactly $q_{\mathsf{PKE.Dec}}$ decryption queries, where $q_{\mathsf{PKE.Dec}} = q_{\mathsf{PKE.Dec}}(\lambda)$ is a suitable polynomial. We proceed in games, and start with the real IND-SO-CCA experiment $\mathsf{Exp}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},\mathcal{A},N}$ as **Game** 1. An overview of how the game works can be seen in Definition 13 and the implementations of the algorithms used to generate keys, encrypt the challenges, respond to decryption queries, and open ciphertexts can be found in Fig. 2. If we denote with one_i the output of Game i, we get $\left|\Pr[\mathsf{one}_1] - \frac{1}{2}\right| = \mathsf{Adv}^{\mathsf{cca\text{-}so}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ (3).

In **Game** 2 we make a change to how the decryption oracle handles decryption queries with a tag that has been *copied* from a challenge ciphertext. We say that a tag (t_c, y') is copied if it occurs already in $C^{(i)}$ for some i. We reject decryption queries $C = (\mathsf{ct}, y', t_c, y)$ if (t_c, y') is copied, but y is not the same as in the challenge ciphertext that it was copied from (see Fig. 2 for details). For copied tags where y was copied as well, we use the key $K^{(i)}$ from the challenge ciphertext to decrypt ct. (That way, neither y nor y' have to be inverted when processing decryption queries with copied tag.)

Since y' uniquely determines x (and thus y) at this point, these changes are purely conceptual, and we have $\Pr[\mathsf{one}_2] = \Pr[\mathsf{one}_1]$ (4).

In **Game** 3, we output random coins for the tag generation via $R_{t_c} \xleftarrow{\$} \mathsf{Expl}_{\mathcal{T}_\mathsf{c}}(1^\lambda, t_c)$ instead of the random coins that were used to sample t_c originally (see Fig. 2). Since $\mathcal{T}_\mathsf{c}$ is efficiently samplable and explainable, we get $\Pr[\mathsf{one}_3] = \Pr[\mathsf{one}_2]$ (5).

In **Game** 4, we generate the ABM tags used in the challenge ciphertexts $C^{(i)}$ as lossy tags, see Fig. 2. A straightforward reduction shows $|\Pr[\mathsf{one}_4] - \Pr[\mathsf{one}_3]| = \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{ABM},\mathcal{B}}(\lambda)$ (6) for a suitable adversary $\mathcal{B}$ on ABM's indistinguishability.

In **Game** 5, we switch from using the inversion key ik' to invert y' and obtain $x = f^{-1}_{ik'}(y')$ and then checking that $f_{ek,(t_c,y')}(x) = y$ to using the inversion key ik of ABM and then checking consistency using LTF. These changes can be seen in Fig. 2. (Note that by the changes from Game 2, we may assume that the tag (t_c, y') is fresh.) By the correctness properties of LTF and ABM, these procedures yield the same results, *unless* the adversary submits a decryption query with a non-injective, non-copied tag. We thus need to bound $\Pr[\mathsf{bad}_{\mathsf{ninj}}]$, where $\mathsf{bad}_{\mathsf{ninj}}$ denotes the event that $\mathcal{A}$ submits a decryption query with a non-injective ABM tag $t = (t_c, y')$ that is not copied. However, the evasiveness property of ABM guarantees that $|\Pr[\mathsf{one}_5] - \Pr[\mathsf{one}_4]| \leq \Pr[\mathsf{bad}_{\mathsf{ninj}}] \leq q_{\mathsf{PKE.Dec}}(\lambda) \cdot \mathsf{Adv}^{\mathsf{eva}}_{\mathsf{ABM},\mathcal{C}}(\lambda)$ (7) is negligible, where $\mathcal{C}$ is a suitable adversary against the evasiveness property of ABM. (Concretely, $\mathcal{C}$ simulates Game 4, chooses $i \in [q_{\mathsf{PKE.Dec}}]$ uniformly, and outputs the tag $t = (t_c, y')$ from $\mathcal{A}$'s ith decryption query if it is not copied. Note that $\mathcal{C}$ can use its $\mathsf{ABM.LTag}$ oracle to produce lossy ABM tags.)

In **Game** 6, we generate LTF's evaluation key ek' as a lossy key, via $ek' \xleftarrow{\$} \mathsf{LTF.LGen}(1^\lambda)$. Since in Game 5, LTF's inversion key ik' is never used, a straightforward reduction shows $|\Pr[\mathsf{one}_5] - \Pr[\mathsf{one}_6]| = \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LTF},\mathcal{F}}(\lambda)$ (8) for a suitable PPT adversary on LTF's indistinguishability.

```
PKE.Gen($1^\lambda$)
$(ek', ik') \xleftarrow{\$} \mathsf{LTF.IGen}(1^\lambda)$
$(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$
$h \xleftarrow{\$} \mathcal{UH}$
$pk := (ek', ek, h)$
$sk := (ik', ek, h)$
return $(pk, sk)$

Open($\mathcal{I}$)
for $i \in \mathcal{I}$
    $R_{t_c}^{(i)\prime} \xleftarrow{\$} \mathsf{Expl}(t_c)$
return $\mathbf{R} = (x^{(i)}, R_{\mathsf{LAE}}^{(i)}, R_{t_c}^{(i)\prime})_{i \in \mathcal{I}}$
return $\mathbf{R} = (x^{(i)}, R_{\mathsf{LAE}}^{(i)}, R_{t_c}^{(i)})_{i \in \mathcal{I}}$ //until $\boxed{2}$

PKE.Enc($pk, msg^{(i)}; R = (x, R_{\mathsf{LAE}}, R_{t_c})$)
parse $pk =: (ek', ek, h)$
$K^{(i)} := h(x)$
$\mathsf{ct} := \mathsf{E}(K, msg; R_{\mathsf{LAE}})$
$y' := f_{ek'}(x)$
$t_c := \mathsf{Samp}_{\mathcal{T}_c}(1^\lambda; R_{t_c})$ //until 3
$t_c \xleftarrow{\$} \mathcal{T}_{\mathsf{loss}}(y')$
$y := f_{ek,(t_c,y')}(x)$
$C := (\mathsf{ct}, y', t_c, y)$
return $C$

PKE.Dec($sk, C$)
if $\exists i \colon C^{(i)} = C$
    return $\perp$
parse $sk =: (ik', ek, h)$,
    $C =: (\mathsf{ct}, y', t_c, y)$
if $\exists i \colon C^{(i)} = (\widetilde{\mathsf{ct}}, \widetilde{y'}, \widetilde{t_c}, \widetilde{y})$ :
        $(t_c, y') = (\widetilde{t_c}, \widetilde{y'})$
    if $y \neq \widetilde{y}$
        return $\perp$
    $K := K^{(i)}$
else
    if $t_c \in \mathcal{T}_{\mathsf{loss}}(y')$
        $\mathsf{bad}_{\mathsf{ninj}} = 1$
        return $\perp$
    $x := f_{ik,(t_c,y')}^{-1}(y)$
    if $f_{ek'}(x) \neq y'$
        return $\perp$
    $x := f_{ik'}^{-1}(y')$ //until 4
    if $y \neq f_{ek,(t_c,y')}(x)$ //until 4
        return $\perp$ //until 4
    $K := h(x)$
$msg := \mathsf{D}(K, \mathsf{ct})$
return $msg$
```

Fig. 2. Handling of encryption, decryption and opening in the games **Game** 1, $\boxed{\textbf{Game } 2}$, **Game** 3, **Game** 4, **Game** 5

In **Game** 7 we sample x for the opened ciphertexts through an inefficient opening algorithm Opener that given K, y, y' outputs x s.t. $h(x) = K, f_{ek'}(x) = y', f_{ek,(t_c,y')}(x) = y$. (It is here that our proof diverges from that of [32]). The opening algorithm Opener can for example be implemented as follows: First compute the set $\mathbf{X}_{K,y,y'} := \{x \mid h(x) = K, f_{ek,t_c,y'}(x) = y, f_{ek'}(x) = y'\}$ of possible values for x, then sample $x^{(i)\prime} \xleftarrow{\$} \mathbf{X}_{K,y,y'}$ uniformly at random.

It is easy to see that it holds that $\forall x \in \{0,1\}^n$

$$\Pr_{x' \xleftarrow{\$} \{0,1\}^n}[x' = x] = \Pr_{\substack{x'' \xleftarrow{\$} \{0,1\}^n \\ x' \xleftarrow{\$} \mathbf{X}_{h(x''), f_{ek,(t_c,y')}(x''), f_{ek'}(x'')}}}[x' = x]$$

and thus $\Pr[\mathsf{one}_7] = \Pr[\mathsf{one}_6]$ (9).

In **Game** 8, we compute the keys K used during encryption as independently and truly random keys $K \in \{0,1\}^{2\kappa}$, instead of setting $K = h(x)$. (Note that by our rules from Game 2, this also means that upon a decryption query with a copied tag (t_c, y'), that same random key K used during encryption is used to decrypt.)

To justify our change, observe that in Game 7, all evaluations $y' = f_{ek'}(x)$, resp. $y = f_{ek,(t_c,y')}(x)$ that $\mathcal{A}$ receives in the challenge ciphertexts are made with respect to lossy keys, resp. tags. In particular, at this point, the values $h(x)$ generated during encryption of *msg* are statistically close to uniform, *even given y' and y*. This is due to the requirement we made in Item 3. Hence, the difference between Game 7 and Game 8 is only statistical:

$$|\Pr[\mathsf{one}_8] - \Pr[\mathsf{one}_7]| \leq O(N/2^\lambda). \tag{10}$$

Finally, in **Game** 9, we reject all decryption queries with copied tags (t_c, y') (even if also y is copied from the same challenge ciphertext) unless said ciphertext has been opened. A difference to Game 8 only occurs if $\mathcal{A}$ manages to submit a decryption query $(\mathsf{ct}, y', t_c, y)$ with the following properties:

- the values t_c, y', y are all copied from the same previous unopened challenge ciphertext $C^{(i)}$, and
- ct decrypts correctly to some message under the key K used in that challenge ciphertext $C^{(i)}$.

Let us call $\mathsf{bad}_{\mathsf{auth}}$ the event that $\mathcal{A}$ places such a decryption query. We can bound the probability that $\mathsf{bad}_{\mathsf{auth}}$ occurs using LAE's authentication property. Namely, a hybrid argument over all challenge ciphertexts shows that

$$|\Pr[\mathsf{one}_9] - \Pr[\mathsf{one}_8]| \leq \Pr[\mathsf{bad}_{\mathsf{auth}}] \leq N \cdot \mathsf{Adv}^{\mathsf{auth}}_{\mathsf{LAE},\mathcal{E}}(\lambda) \tag{11}$$

for an adversary $\mathcal{E}$ that simulates Game 8, and embeds its own challenge ciphertext as one of the IND-SO-CCA challenge ciphertexts of Game 8.

Now observe that in Game 9, $\mathcal{A}$ receives only lossy LAE ciphertexts made with independently random keys K (that are never used again for any decryption queries). The message vectors $\mathbf{msg}_0$ and $\mathbf{msg}_1$ from $\mathsf{Exp}^{\mathsf{ind\text{-}so\text{-}cca}}_{\mathsf{PKE},\mathcal{A}}$ are thus identically distributed (even given $\mathcal{A}$'s view), and we finally obtain $\Pr[\mathsf{one}_9] = \frac{1}{2}$ (12). Taking (Eq. (3)-Eq. (12)) together shows (Eq. (2)).

References

1. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited - new reduction, properties and applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 57–74. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40041-4_4
2. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (Aug 2009). https://doi.org/10.1007/978-3-642-03356-8_35
3. Auerbach, B., Kiltz, E., Poettering, B., Schoenen, S.: Lossy trapdoor permutations with improved lossiness. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 230–250. Springer, Heidelberg (Mar 2019). https://doi.org/10.1007/978-3-030-12612-4_12

```
$\mathsf{PKE.Gen}(1^\lambda)$
$(ek', ik') \xleftarrow{\$} \mathsf{LTF.IGen}(1^\lambda)$ //until 5
$\boxed{ek' \xleftarrow{\$} \mathsf{LTF.LGen}(1^\lambda)}$
$(ek, ik, tk) \xleftarrow{\$} \mathsf{ABM.Gen}(1^\lambda)$
$h \xleftarrow{\$} \mathcal{UH}$
$pk := (ek', ek, h)$
$sk := (ik', ek, h)$
return $(pk, sk)$

$\mathsf{Open}(\mathcal{I})$
for $i \in \mathcal{I}$
    $R_{t_c}^{(i)\prime} \xleftarrow{\$} \mathsf{Expl}(t_c)$
    $x^{(i)\prime} \xleftarrow{\$} \mathsf{Opener}(K^{(i)}, y, y')$
    return $\mathbf{R} = (x^{(i)\prime}, R_{\mathsf{LAE}}^{(i)}, R_{t_c}^{(i)\prime})_{i \in \mathcal{I}}$
return $\mathbf{R} = (x^{(i)}, R_{\mathsf{LAE}}^{(i)}, R_{t_c}^{(i)\prime})_{i \in \mathcal{I}}$

$\mathsf{PKE.Enc}(pk, msg^{(i)}; R = (x, R_{\mathsf{LAE}}, R_{t_c}))$
parse $pk =: (ek', ek, h)$
$K^{(i)} := h(x)$ // until 7
$K^{(i)} \xleftarrow{\$} \{0,1\}^{2\kappa}$
$\mathsf{ct} := \mathsf{E}(K, msg; R_{\mathsf{LAE}})$
$y' := f_{ek'}(x)$
$t_c \xleftarrow{\$} \mathcal{T}_{\mathsf{loss}}(y')$
$y := f_{ek,(t_c,y')}(x)$
$C := (\mathsf{ct}, y', t_c, y)$
return $C$
```

```
$\mathsf{PKE.Dec}(sk, C)$
if $\exists i\colon C^{(i)} = C$
    return $\bot$
parse $sk =: (ik', ek, h)$,
    $C =: (\mathsf{ct}, y', t_c, y)$
if $\exists i\colon C^{(i)} = (\widetilde{\mathsf{ct}}, \widetilde{y'}, \widetilde{t_c}, \widetilde{y})$:
        $(t_c, y') = (\widetilde{t_c}, \widetilde{y'})$
    if $i \notin \mathcal{I}$
        return $\bot$
    if $y \neq \widetilde{y}$
        return $\bot$
    $K := K^{(i)}$
else
    $x := f^{-1}_{ik,(t_c,y')}(y)$
    if $f_{ek'}(x) \neq y'$
        return $\bot$
    $K := h(x)$
$msg := \mathsf{D}(K, \mathsf{ct})$
return $msg$
```

Fig. 3. Handling of encryption, decryption and opening in the games **Game** 5, **Game** 6, **Game** 7, **Game** 8, **Game** 9

4. Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard security does not imply security against selective-opening. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 645–662. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_38
5. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (Apr 2009). https://doi.org/10.1007/978-3-642-01001-9_1
6. Bellare, M., Kiltz, E., Peikert, C., Waters, B.: Identity-based (lossy) trapdoor functions and applications. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 228–245. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_15
7. Bellare, M., Waters, B., Yilek, S.: Identity-based encryption secure against selective opening attack. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 235–252. Springer, Heidelberg (Mar 2011). https://doi.org/10.1007/978-3-642-19571-6_15
8. Benhamouda, F., Herranz, J., Joye, M., Libert, B.: Efficient cryptosystems from 2^k-th power residue symbols. Journal of Cryptology **30**(2), 519–549 (Apr 2017). https://doi.org/10.1007/s00145-016-9229-5

9. Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 522–539. Springer, Heidelberg (May 2012). https://doi.org/10.1007/978-3-642-30057-8_31
10. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85174-5_19
11. Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 410–428. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40041-4_23
12. Boyen, X., Li, Q.: All-but-many lossy trapdoor functions from lattices and applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 298–331. Springer, Heidelberg (Aug 2017). https://doi.org/10.1007/978-3-319-63697-9_11
13. Brakerski, Z., Döttling, N., Garg, S., Malavolta, G.: Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part II. LNCS, vol. 11892, pp. 407–437. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-36033-7_16
14. Brakerski, Z., Vaikuntanathan, V.: Lattice-based fhe as secure as pke. In: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science. p. 1–12. ITCS '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2554797.2554799, https://doi.org/10.1145/2554797.2554799
15. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (Aug 2000). https://doi.org/10.1007/3-540-44598-6_27
16. Damgård, I., Nielsen, J.B.: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 581–596. Springer, Heidelberg (Aug 2002). https://doi.org/10.1007/3-540-45708-9_37
17. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (May 2004). https://doi.org/10.1007/978-3-540-24676-3_31
18. Döttling, N., Garg, S., Ishai, Y., Malavolta, G., Mour, T., Ostrovsky, R.: Trapdoor hash functions and their applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 3–32. Springer, Heidelberg (Aug 2019). https://doi.org/10.1007/978-3-030-26954-8_1
19. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th FOCS. pp. 523–534. IEEE Computer Society Press (Oct 1999). https://doi.org/10.1109/SFFCS.1999.814626
20. Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 381–402. Springer, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_20
21. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. In: Nguyen, P.Q., Pointcheval,

D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (May 2010). https://doi.org/10.1007/978-3-642-13013-7_17

22. Fujisaki, E.: All-but-many encryption - A new framework for fully-equipped UC commitments. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 426–447. Springer, Heidelberg (Dec 2014). https://doi.org/10.1007/978-3-662-45608-8_23
23. Gentry, C., Halevi, S.: Compressible FHE with applications to PIR. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part II. LNCS, vol. 11892, pp. 438–464. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-36033-7_17
24. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40041-4_5
25. Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: Yao, A.C.C. (ed.) ICS 2010. pp. 230–240. Tsinghua University Press (Jan 2010)
26. Gustafsson, B.: Scientific Computing - A Historical Perspective. Springer, Berlin, Heidelberg (2018)
27. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (Dec 2011). https://doi.org/10.1007/978-3-642-25385-0_4
28. Hemenway, B., Ostrovsky, R.: Extended-DDH and lossy trapdoor functions. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 627–643. Springer, Heidelberg (May 2012). https://doi.org/10.1007/978-3-642-30057-8_37
29. Hemenway, B., Ostrovsky, R.: Building lossy trapdoor functions from lossy encryption. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 241–260. Springer, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-42045-0_13
30. Heuer, F., Jager, T., Kiltz, E., Schäge, S.: On the selective opening security of practical public-key encryption schemes. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 27–51. Springer, Heidelberg (Mar / Apr 2015). https://doi.org/10.1007/978-3-662-46447-2_2
31. Heuer, F., Poettering, B.: Selective opening security from simulatable data encapsulation. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 248–277. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_9
32. Hofheinz, D.: All-but-many lossy trapdoor functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 209–227. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_14
33. Hofheinz, D., Jager, T., Rupp, A.: Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 146–168. Springer, Heidelberg (Oct / Nov 2016). https://doi.org/10.1007/978-3-662-53644-5_6
34. Hofheinz, D., Rao, V., Wichs, D.: Standard security does not imply indistinguishability under selective opening. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 121–145. Springer, Heidelberg (Oct / Nov 2016). https://doi.org/10.1007/978-3-662-53644-5_5

35. Hofheinz, D., Rupp, A.: Standard versus selective opening security: Separation and equivalence results. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 591–615. Springer, Heidelberg (Feb 2014). https://doi.org/10.1007/978-3-642-54242-8_25
36. Huang, Z., Liu, S., Qin, B.: Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 369–385. Springer, Heidelberg (Feb / Mar 2013). https://doi.org/10.1007/978-3-642-36362-7_23
37. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under chosen-plaintext attack. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 295–313. Springer, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-14623-7_16
38. Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 332–364. Springer, Heidelberg (Aug 2017). https://doi.org/10.1007/978-3-319-63697-9_12
39. Mahadev, U.: Classical homomorphic encryption for quantum circuits. In: Thorup, M. (ed.) 59th FOCS. pp. 332–338. IEEE Computer Society Press (Oct 2018). https://doi.org/10.1109/FOCS.2018.00039
40. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_41
41. Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 296–311. Springer, Heidelberg (May 2010). https://doi.org/10.1007/978-3-642-13013-7_18
42. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (May 1999). https://doi.org/10.1007/3-540-48910-X_16
43. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 333–342. ACM Press (May / Jun 2009). https://doi.org/10.1145/1536414.1536461
44. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85174-5_31
45. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 187–196. ACM Press (May 2008). https://doi.org/10.1145/1374376.1374406
46. Pietrzak, K., Rosen, A., Segev, G.: Lossy functions do not amplify well. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 458–475. Springer, Heidelberg (Mar 2012). https://doi.org/10.1007/978-3-642-28914-9_26
47. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (sep 2009). https://doi.org/10.1145/1568318.1568324, https://doi.org/10.1145/1568318.1568324

Supplementary Material

A Additional Preliminaries

A.1 On Gadget Matrices

Proof of Lemma 1

Proof. We replicate here the construction of [40].

Set $k = \lceil \log_2 q \rceil$.Let $\mathbf{y} = \mathbf{G}^T\mathbf{s} + \mathbf{e} \mod q$ for $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}^{nk}$. Then, $\mathbf{y}$ has the following shape

$$\mathbf{y}^T = (s_1 + e_1,\ 2s_1 + e_2,\ \ldots,\ 2^{k-1}s_1 + e_k,\ s_2 + e_{k+1},\ 2s_2 + e_{k+2},\ \ldots).$$

Therefore, it suffices to prove the claim for the case $n = 1$. Set $\mathbf{g} = (1, 2, \ldots, 2^{k-1})$, then $\mathbf{y} = \mathbf{g}s + \mathbf{e}$ for some number $s \in \mathbb{Z}_q$ and noise $\mathbf{e} \in \mathbb{Z}^k$.

If q is a power of 2, i.e. $q = 2^k$, set

$$\mathbf{S} = \begin{pmatrix} 2 & -1 & & & \\ & 2 & -1 & & \\ & & \ddots & & \\ & & & 2 & -1 \\ & & & & 2 \end{pmatrix}.$$

Otherwise, if $q \neq 2^k$, consider the $k \times k$-matrix

$$\mathbf{S} = \begin{pmatrix} 2 & -1 & & & & \\ & 2 & -1 & & & \\ & & & \ddots & & \\ & & & & 2 & -1 \\ q_0 & q_1 & q_2 & \cdots & q_{k-2} & q_{k-1} \end{pmatrix}$$

where $q_0, \ldots, q_{k-1} \in \{0, 1\}$ is the binary representation of $q = q_0 + 2q_1 + \ldots + 2^{k-1}q_{k-1}$. Note, that the ∞-norm of $\mathbf{S}$ is bounded by $\max\{3, k\}$. Since $q \geq 8$ we have $||\mathbf{S}||_\infty \leq k$.

Over $\mathbb{Z}_q$, we have $\mathbf{S} \cdot \mathbf{g} = 0 \mod q$. However, $\mathbf{S}$ is invertible over the reals. This leads to the following algorithm:

1. Given $\mathbf{y} \in \mathbb{Z}_q^k$, compute $\mathbf{b} := \mathbf{S} \cdot \mathbf{y} \mod q$.
2. Treat $\mathbf{b}$ like a real vector in $[-\frac{q}{2}, \frac{q}{2})^k$ and compute $\mathbf{e}' := \mathbf{S}^{-1} \cdot \mathbf{b}$.
3. Output $\mathbf{e}'$ and $s' := y_1 - e'_1 \mod q$.

In this algorithm, we have $\mathbf{e} = \mathbf{e}'$ (and $s = s'$) if $\mathbf{b}$ equals the product $\mathbf{S} \cdot \mathbf{e}$ over the integers. Since $\mathbf{b}$ is computed by

$$\mathbf{b} = \mathbf{S}\mathbf{y} = \mathbf{S}(\mathbf{g}s + \mathbf{e}) = \mathbf{S}\mathbf{e} \mod q,$$

it equals $\mathbf{Se}$ if the ∞-norm of $\mathbf{Se}$ is smaller than $\frac{q}{2}$. Since we demanded $||\mathbf{e}||_\infty < \frac{q}{2\cdot\lceil\log_2 q\rceil}$, we have

$$||\mathbf{Se}||_\infty \leq ||\mathbf{S}||_\infty \cdot ||\mathbf{e}||_\infty < k \cdot \frac{q}{2k} = \frac{q}{2}.$$

Therefore, the correctness of the algorithm follows.

A.2 Dual GSW

Proof of Lemma 4

Proof. 1. We can prove the first statement of Lemma 4 by a hybrid argument (with N game hops).
For a given $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, $i \in 0, \ldots, N$ and $\beta \in \{0,1\}$ define the distribution $H_{i,\beta}$ as follows:
(a) Sample $\mathbf{u}_1, \ldots, \mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^m$.
(b) Sample $\mathbf{s}_{i+1}, \ldots, \mathbf{s}_N \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_{i+1}, \ldots, \mathbf{e}_N \xleftarrow{\$} D_{\alpha q}^m$ and set for $j = i+1, \ldots, N$

$$\mathbf{b}_j := \mathbf{A}\mathbf{s}_j + \mathbf{e}_j \mod q.$$

(c) Output $(\mathbf{u}_1|\ldots|\mathbf{u}_i|\mathbf{b}_{i+1}|\ldots|\mathbf{b}_N) + \beta\mathbf{G}_{m,q}$.
We have the following equalities of distributions (in the sense that these distributions are identical):

$$H_{0,0} = \mathsf{FHE.Enc}(\mathbf{A}, 0),$$
$$H_{0,1} = \mathsf{FHE.Enc}(\mathbf{A}, 1).$$

By the triangle-inequality, there must be an $i \in [N]$ and $\beta \in \{0,1\}$ s.t.

$$\left| \Pr_{\mathbf{C}\xleftarrow{\$}H_{i-1,\beta}}[\mathcal{A}(\mathbf{C}) = 1] - \Pr_{\mathbf{C}\xleftarrow{\$}H_{i,\beta}}[\mathcal{A}(\mathbf{C}) = 1] \right|$$
$$\geq \frac{1}{N} \cdot \left| \Pr_{\mathbf{C}\xleftarrow{\$}\mathsf{FHE.Enc}(\mathbf{A},0)}[\mathcal{A}(\mathbf{C}) = 1] - \Pr_{\mathbf{C}\xleftarrow{\$}\mathsf{FHE.Enc}(\mathbf{A},1)}[\mathcal{A}(\mathbf{C}) = 1] \right|.$$

However, $\left|\Pr_{\mathbf{C}\xleftarrow{\$}H_{i-1,\beta}}[\mathcal{A}(\mathbf{C}) = 1] - \Pr_{\mathbf{C}\xleftarrow{\$}H_{i,\beta}}[\mathcal{A}(\mathbf{C}) = 1]\right|$ is equal to the advantage of an LWE-distinguisher $\mathcal{B}$ that implements an LWE challenge in the i-th column of a sample of $H_{i-1,\beta}$. Therefore, we have

$$\left| \Pr_{\mathbf{C}\xleftarrow{\$}H_{i-1,\beta}}[\mathcal{A}(\mathbf{C}) = 1] - \Pr_{\mathbf{C}\xleftarrow{\$}H_{i,\beta}}[\mathcal{A}(\mathbf{C}) = 1] \right| \leq \mathsf{Adv}_{\mathsf{LWE}}^{n,q,\alpha,m}(\mathcal{B}).$$

2. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, $\mu_1, \mu_2 \in \{0,1\}$ and $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{m\times N}$. We first show the correctness for computing negations, sums and products:
Negations: It is easy to see that

$$\mathsf{noise}_{\mathbf{A},1-\mu_1}(\mathbf{G} - \mathbf{C}_1) = \mathsf{noise}_{\mathbf{A},\mu_1}(\mathbf{C}_1). \quad (13)$$

Additions: We will assume here that μ_1 and μ_2 are not simultaneously one, i.e. $\mu_1 + \mu_2 \in \{0,1\}$, since the case $\mu_1 = \mu_2 = 1$ is not relevant for the homomorphic evaluation of circuits. Assuming $\mu_1 + \mu_2 \in \{0,1\}$, it is easy to verify that

$$\mathsf{noise}_{\mathbf{A},\mu_1+\mu_2}(\mathbf{C}_1 + \mathbf{C}_2) \leq \mathsf{noise}_{\mathbf{A},\mu_1}(\mathbf{C}_1) + \mathsf{noise}_{\mathbf{A},\mu_2}(\mathbf{C}_2). \tag{14}$$

Multiplications: Let $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{Z}_q^{n\times N}$ and $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{Z}^{m\times N}$ s.t.

$$\mathbf{C}_1 = \mathbf{A}\mathbf{S}_1 + \mathbf{E}_1 + \mu_1\mathbf{G} \;\text{ and }\; \mathbf{C}_2 = \mathbf{A}\mathbf{S}_2 + \mathbf{E}_2 + \mu_2\mathbf{G}$$

and

$$||\mathbf{E}_1||_\infty = \mathsf{noise}_{\mathbf{A},\mu_1}(\mathbf{C}_1) \;\text{ and }\; ||\mathbf{E}_2||_\infty = \mathsf{noise}_{\mathbf{A},\mu_2}(\mathbf{C}_2).$$

$\mathsf{FHE.Eval}$ computes products by

$$\begin{aligned}\mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) =& \mathbf{A} \cdot (\mathbf{S}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1\mathbf{S}_2)\\ &+ (\mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1\mathbf{E}_2) + \mu_1\mu_2\mathbf{G}.\end{aligned}$$

It therefore suffices to bound the error $\mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1\mathbf{E}_2$. In fact, we have

$$\begin{aligned}&\mathsf{noise}_{\mathbf{A},\mu_1\mu_2}(\mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2))\\ \leq& \left|\left|\mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1\mathbf{E}_2\right|\right|_\infty\\ \leq& \left|\left|\mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2)\right|\right|_\infty + ||\mu_1\mathbf{E}_2||_\infty\\ \leq& ||\mathbf{E}_1||_\infty \cdot \left|\left|\mathbf{G}^{-1}(\mathbf{C}_2)\right|\right|_\infty + \mu_1 ||\mathbf{E}_2||_\infty\\ \leq& ||\mathbf{E}_1||_\infty \cdot N + \mu_1 ||\mathbf{E}_2||_\infty\\ =& N \cdot \mathsf{noise}_{\mathbf{A},\mu_1}(\mathbf{C}_1) + \mu_1 \cdot \mathsf{noise}_{\mathbf{A},\mu_2}(\mathbf{C}_2)\end{aligned} \tag{15}$$

Now, let $\iota : [L] \to [\eta], (\sigma_{h,\mu})_{i\in[L],\mu\in\{0,1\}}$ be a branching program of length L. Let $\mathbf{C}_1, \ldots, \mathbf{C}_\eta \in \mathbb{Z}_q^{m\times N}$ be ciphertexts for bits $\mu_1, \ldots, \mu_\eta \in \{0,1\}$. Set

$$B := \max_{i\in[\eta]} \;\mathsf{noise}_{\mathbf{A},\mu_i}(\mathbf{C}_i).$$

Let $\delta_{i,j}$ denote the Kronecker delta that is given by

$$\delta_{i,j} := \begin{cases}1, & \text{if } i = j,\\ 0, & \text{if } i \neq j.\end{cases}$$

Then, the permutation matrix $\mathbf{M}_\sigma \in \{0,1\}^{5\times 5}$ for a permutation $\sigma \in S_5$ has the entries $(\delta_{i,\sigma(j)})_{i,j\in[5]}$. It holds

$$\mathbf{M}_{\sigma_1} \cdot \mathbf{M}_{\sigma_2} = \mathbf{M}_{\sigma_1\circ\sigma_2}.$$

For $h = 1, \ldots, L$, set

$$\mathbf{P}^{(0)} := \mathbf{I}_5 \in \{0,1\}^{5\times 5},$$

$$\mathbf{P}^{(h)} := \mathbf{M}_{\sigma_{h,\mu_{\iota(h)}}} \cdot \mathbf{P}^{(h-1)} \in \{0,1\}^{5\times 5}.$$

Let $(p_{i,j}^{(h)})_{i,j\in[5]}$ denote the entries of $\mathbf{P}^{(h)}$.

We prove by induction on $h = 0, \dots, L$ that $\mathbf{Q}_{i,j}^{(h)}$ is an encryption of the (i,j)-th entry $p_{i,j}^{(h)}$ of $\mathbf{P}^{(h)}$ for each $i, j \in [5]$, i.e.

$$\mathsf{noise}_{\mathbf{A},p_{i,j}^{(h)}}\left(\mathbf{Q}_{i,j}^{(h)}\right) \le h \cdot 2NB.$$

The induction start is clear, since

$$\begin{pmatrix} \mathbf{Q}_{1,1}^{(0)} & \mathbf{Q}_{1,2}^{(0)} & \mathbf{Q}_{1,3}^{(0)} & \mathbf{Q}_{1,4}^{(0)} & \mathbf{Q}_{1,5}^{(0)} \\ \mathbf{Q}_{2,1}^{(0)} & \mathbf{Q}_{2,2}^{(0)} & \mathbf{Q}_{2,3}^{(0)} & \mathbf{Q}_{2,4}^{(0)} & \mathbf{Q}_{2,5}^{(0)} \\ \mathbf{Q}_{3,1}^{(0)} & \mathbf{Q}_{3,2}^{(0)} & \mathbf{Q}_{3,3}^{(0)} & \mathbf{Q}_{3,4}^{(0)} & \mathbf{Q}_{3,5}^{(0)} \\ \mathbf{Q}_{4,1}^{(0)} & \mathbf{Q}_{4,2}^{(0)} & \mathbf{Q}_{4,3}^{(0)} & \mathbf{Q}_{4,4}^{(0)} & \mathbf{Q}_{4,5}^{(0)} \\ \mathbf{Q}_{5,1}^{(0)} & \mathbf{Q}_{5,2}^{(0)} & \mathbf{Q}_{5,3}^{(0)} & \mathbf{Q}_{5,4}^{(0)} & \mathbf{Q}_{5,5}^{(0)} \end{pmatrix} = \begin{pmatrix} \mathbf{G} & & & & \\ & \mathbf{G} & & & \\ & & \mathbf{G} & & \\ & & & \mathbf{G} & \\ & & & & \mathbf{G} \end{pmatrix}$$

is a noise-free encryption of the identity matrix $\mathbf{P}^{(0)}$.

For the induction step, assume that the induction hypothesis holds for $(\mathbf{Q}_{i,j}^{(h-1)})_{i,j}$. Let $\mu = \mu_{\iota(h)}$ and note, that we have

$$\mathbf{M}_{\sigma_{h,\mu}} = \mu \cdot \mathbf{M}_{\sigma_{h,1}} + (1-\mu) \cdot \mathbf{M}_{\sigma_{h,0}}.$$

For the (i,j)-th entry of $\mathbf{P}^{(h)} = \mathbf{M}_\sigma \cdot \mathbf{P}^{(h-1)}$, we get

$$\begin{aligned} p_{i,j}^{(h)} =& \mu \cdot \left(\sum_{k=1}^{5} \delta_{i,\sigma_{h,1}(k)} \cdot p_{k,j}^{(h-1)} \right) + (1-\mu) \cdot \left(\sum_{k=1}^{5} \delta_{i,\sigma_{h,0}(k)} \cdot p_{k,j}^{(h-1)} \right) \\ =& \mu \cdot p_{\sigma_{h,1}^{-1}(i),j}^{(h-1)} + (1-\mu) \cdot p_{\sigma_{h,0}^{-1}(i),j}^{(h-1)}. \end{aligned}$$

For $\mathbf{Q}_{i,j}^{(h)}$, we have

$$\begin{aligned} & \mathsf{noise}_{\mathbf{A},p_{i,j}^{(h)}}(\mathbf{Q}_{i,j}^{(h)}) \\ = \ & \mathsf{noise}_{\mathbf{A},\mu\cdot p_{\sigma_{h,1}^{-1}(i),j}^{(h-1)} + (1-\mu)\cdot p_{\sigma_{h,0}^{-1}(i),j}^{(h-1)}} \left(\mathbf{C}_{\iota(h)} \cdot \mathbf{G}^{-1}\left(\mathbf{Q}_{\sigma_{h,1}^{-1}(i),j}^{(h-1)} \right) \right. \\ & \left. + \ \mathbf{C}_{\neg\iota(h)} \cdot \mathbf{G}^{-1}\left(\mathbf{Q}_{\sigma_{h,0}^{-1}(i),j}^{(h-1)} \right) \right) \\ \overset{Eq.\ (14)}{\le} & \mathsf{noise}_{\mathbf{A},\mu\cdot p_{\sigma_{h,1}^{-1}(i),j}^{(h-1)}} \left(\mathbf{C}_{\iota(h)} \cdot \mathbf{G}^{-1}\left(\mathbf{Q}_{\sigma_{h,1}^{-1}(i),j}^{(h-1)} \right) \right) \\ & + \ \mathsf{noise}_{\mathbf{A},(1-\mu)\cdot p_{\sigma_{h,0}^{-1}(i),j}^{(h-1)}} \left(\mathbf{C}_{\neg\iota(h)} \cdot \mathbf{G}^{-1}\left(\mathbf{Q}_{\sigma_{h,0}^{-1}(i),j}^{(h-1)} \right) \right) \\ \overset{Eq.\ (15)}{\le} & N \cdot \mathsf{noise}_{\mathbf{A},\mu}\left(\mathbf{C}_{\iota(h)} \right) + \mu \cdot \mathsf{noise}_{\mathbf{A},p_{\sigma_{h,1}^{-1}(i),j}^{(h-1)}} \left(\mathbf{Q}_{\sigma_{h,1}^{-1}(i),j}^{(h-1)} \right) \end{aligned}$$

$$\begin{aligned} &+ N \cdot \mathsf{noise}_{\mathbf{A},(1-\mu)}\left(\mathbf{C}_{\neg\iota(h)}\right) + (1-\mu)\cdot \mathsf{noise}_{\mathbf{A},p^{(h-1)}_{\sigma^{-1}_{h,0}(i),j}}\left(\mathbf{Q}^{(h-1)}_{\sigma^{-1}_{h,0}(i),j}\right) \\ \overset{(*)}{\leq}\ & N\cdot B + \mu\cdot(h-1)\cdot 2NB + N\cdot B + (1-\mu)\cdot(h-1)\cdot 2NB \\ =\ & h\cdot 2NB. \end{aligned}$$

The inequality at $(*)$ comes from the induction hypothesis that states

$$\mathsf{noise}_{\mathbf{A},p^{(h-1)}_{\sigma^{-1}_{h,1}(i),j}}\left(\mathbf{Q}^{(h-1)}_{\sigma^{-1}_{h,1}(i),j}\right), \mathsf{noise}_{\mathbf{A},p^{(h-1)}_{\sigma^{-1}_{h,0}(i),j}}\left(\mathbf{Q}^{(h-1)}_{\sigma^{-1}_{h,0}(i),j}\right) \leq (h-1)\cdot 2NB.$$

If $\iota : [L] \to [\eta], (\sigma_{i,j})_{i,j}$ is the branching program of a circuit $C : \{0,1\}^\eta \to \{0,1\}$, then we have $p^{(L)}_{1,1} = C(\mu_1, \ldots, \mu_\eta)$, since

$$p^{(L)}_{1,1} = 1 \iff \sigma_{4d,\mu_{\iota(L)}} \circ \ldots \circ \sigma_{1,\mu_{\iota(1)}}(1) = 1 \iff C(\mu_1, \ldots, \mu_\eta) = 1.$$

Therefore, we have

$$\mathsf{noise}_{\mathbf{A},C(\mu_1,\ldots,\mu_\eta)}(\mathbf{Q}^{(L)}_{1,1}) = \mathsf{noise}_{\mathbf{A},p^{(L)}_{1,1}}(\mathbf{Q}^{(L)}_{1,1}) \leq L\cdot 2NB$$

where $L = 4^d$ is four to the power of the depth of C.

Lemma 5. *Let $q, n, \overline{m} \in \mathbb{N}$ s.t. $\overline{m} > 0$, $q \geq 2^8$ and $\overline{m} \geq n \cdot \lceil \log_2 q \rceil$. Set $w = n \cdot \lceil \log_2 q \rceil$, $m = \overline{m} + w$ and $N := m \cdot \lceil \log_2 q \rceil$.*

Further, let $(\mathbf{R}, \mathbf{A})$ be sampled by $\mathsf{GenTrap}(n, \overline{m}, q)$.

There is a PPT algorithm that – given $\mathbf{R}, \mathbf{A}$ and a ciphertext $\mathbf{C} \in \mathbb{Z}_q^{m\times N}$ for a message $\mu \in \{0,1\}$ under $\mathbf{A}$ – will retrieve μ and matrices $\mathbf{S} \in \mathbb{Z}_q^{n\times N}$ and $\mathbf{E} \in \mathbb{Z}^{m\times N}$ s.t.

$$\mathbf{C} = \mathbf{AS} + \mathbf{E} + \mu\mathbf{G}$$

and

$$||\mathbf{E}||_\infty < \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}$$

if $\mathsf{noise}_{\mathbf{A},\mu}(\mathbf{C}) < \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}$.

Proof. We define the following sub-algorithm try:

1. Given matrices $\mathbf{R}, \mathbf{A}, \mathbf{C}$ and a bit μ, try sets
$$(\mathbf{c}_1|\ldots|\mathbf{c}_N) := \mathbf{C} - \mu\mathbf{G}.$$
2. For each $i \in [N]$, it uses $\mathsf{Invert}(\mathbf{A}, \mathbf{c}_i, \mathbf{R})$ to compute vectors $\mathbf{s}_i \in \mathbb{Z}_q^n, \mathbf{e}_i \in \mathbb{Z}^m$ s.t.
$$\mathbf{c}_i = \mathbf{As}_i + \mathbf{e}_i \mod q$$
and
$$||\mathbf{e}_i||_\infty < \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}.$$

3. If one of the calls $\mathsf{Invert}(\mathbf{A}, \mathbf{c}_i, \mathbf{A})$ fails or does not finish after a fixed polynomial number of computations or does not return a correct result, then try aborts.
4. Otherwise, try sets $\mathbf{S} = (\mathbf{s}_1|\dots|\mathbf{s}_N)$ and $\mathbf{E} = (\mathbf{e}_1|\dots|\mathbf{e}_N)$.
5. If $||\mathbf{E}||_\infty \geq \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}$, then try aborts. Otherwise, it returns $\mathbf{S}$ and $\mathbf{E}$.

Given $\mathbf{R}, \mathbf{A}$ and $\mathbf{C}$, our main-algorithm runs $\mathsf{try}(\mathbf{R}, \mathbf{A}, \mathbf{C}, 0)$ and $\mathsf{try}(\mathbf{R}, \mathbf{A}, \mathbf{C}, 1)$, and returns $\mathbf{E}$ and $\mathbf{S}$ of the one sub-algorithm that finished successfully (and the corresponding bit μ).

We claim, for each $(\mathbf{R}, \mathbf{A}) \overset{\$}{\leftarrow} \mathsf{GenTrap}(n, \overline{m}, q)$, for each matrix $\mathbf{C} \in \mathbb{Z}_q^{m\times N}$ and for each $\mu \in \{0,1\}$ s.t.

$$\mathsf{noise}_{\mathbf{A},\mu}(\mathbf{C}) < \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}$$

only $\mathsf{try}(\mathbf{R}, \mathbf{A}, \mathbf{C}, \mu)$ will succeed while $\mathsf{try}(\mathbf{R}, \mathbf{A}, \mathbf{C}, 1-\mu)$ will fail.

It suffices to show this for the case $\mu = 0$. Since $\mathsf{noise}_{\mathbf{A},0}(\mathbf{C}) < \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}$, there are matrices $\mathbf{S} \in \mathbb{Z}_q^{n\times N}$ and $\mathbf{E} \in \mathbb{Z}^{m\times N}$ s.t.

$$\mathbf{C} = \mathbf{A}\mathbf{S} + \mathbf{E} \mod q$$

and

$$||\mathbf{E}||_\infty < \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}.$$

Therefore, the ∞-norm of each column of $\mathbf{E}$ is suitably small s.t. $\mathsf{try}(\mathbf{R}, \mathbf{A}, \mathbf{C}, 0)$ will succeed according to Lemma 2 and return $\mathbf{S}$ and $\mathbf{E}$.

For the sake of contradiction, assume that $\mathsf{try}(\mathbf{R}, \mathbf{A}, \mathbf{C}, 1)$ succeeds, too, and returns matrices $\mathbf{S}', \mathbf{E}'$ s.t.

$$\mathbf{C} = \mathbf{A}\mathbf{S}' + \mathbf{E}' + \mathbf{G} \mod q$$

and

$$||\mathbf{E}'||_\infty < \frac{q}{2\log_2(q)\cdot(\overline{m}+1)}.$$

Then, we have

$$\mathbf{G} = \mathbf{A}(\mathbf{S} - \mathbf{S}') + \mathbf{E} - \mathbf{E}' \mod q.$$

We claim that there is a non-zero vector $\mathbf{x} \in \{-1, 0, 1\}^m$ s.t.

$$\mathbf{x}^T\mathbf{A} = \mathbf{0} \mod q.$$

In fact, since $\{0,1\}^m$ has more elements than $\mathbb{Z}_q^n$, there must exist two different vectors $\mathbf{a}, \mathbf{b} \in \{0,1\}^m$ s.t. $\mathbf{a}^T\mathbf{A} = \mathbf{b}^T\mathbf{A} \mod q$. Ergo, we can choose $\mathbf{x} = \mathbf{a} - \mathbf{b} \neq \mathbf{0}$.

Since $\mathbf{x}$ is non-zero, we can assume – without loss of generality – that $x_1 = 1$. We now have

$$\mathbf{x}^T\mathbf{G} = \mathbf{x}^T\mathbf{A}(\mathbf{S} - \mathbf{S}') + \mathbf{x}^T(\mathbf{E} - \mathbf{E}') = \mathbf{x}^T(\mathbf{E} - \mathbf{E}') \mod q.$$

Since $x_1 = 1$, the first $\lceil \log_2 q \rceil$ coordinates of $\mathbf{x}^T\mathbf{G}$ must look as follows

$$\mathbf{x}^T\mathbf{G} = (1 \;\; 2 \;\; \ldots \;\; 2^{\lceil \log_2 q \rceil - 2} \;\; 2^{\lceil \log_2 q \rceil - 1} \;\; \ldots).$$

For $\mathbf{x}^T(\mathbf{E} - \mathbf{E}')$, on the other hand, we have over the integers

$$\begin{aligned} &\left|\left|(\mathbf{E} - \mathbf{E}')^T\mathbf{x}\right|\right|_\infty \leq \left|\left|(\mathbf{E} - \mathbf{E}')^T\right|\right|_\infty \cdot ||\mathbf{x}||_\infty \leq m \cdot ||\mathbf{E} - \mathbf{E}'||_\infty \cdot 1 \\ \leq& m \cdot (||\mathbf{E}||_\infty + ||\mathbf{E}'||_\infty) \leq m \cdot 2 \cdot \frac{q}{2\log_2(q) \cdot (\overline{m} + 1)} \\ <& \frac{q}{\log_2 q} \cdot \frac{m}{\overline{m}} \leq \frac{q}{\log_2 q} \cdot 2 \leq \frac{q}{4}. \end{aligned}$$

In particular, each entry of the integer vector $\mathbf{x}^T(\mathbf{E} - \mathbf{E}')$ must lie in the open interval $(\frac{-q}{4}, \frac{q}{4})$. The entry $2^{\lceil \log_2 q \rceil - 2}$ of $\mathbf{x}^T\mathbf{G}$, however, must lie in the interval $[\frac{q}{4}, \frac{q}{2})$. Since the intervals $(\frac{-q}{4}, \frac{q}{4})$ and $[\frac{q}{4}, \frac{q}{2})$ don't intersect modulo q, the equality

$$\mathbf{x}^T\mathbf{G} = \mathbf{x}^T\mathbf{A}(\mathbf{S} - \mathbf{S}') + \mathbf{x}^T(\mathbf{E} - \mathbf{E}') \mod q$$

cannot hold. A contradiction!

Ergo, not both sub-algorithms can succeed. This implies that our main-algorithm always outputs a correct result.

B Parameters

In this section, we will give exhaustive overviews over the parameters used by our LTF construction Section 3, by our ABM-LTF construction Section 4 and by our IND-SO-CCA secure PKE construction Section 5.

We will gather all inequalities and constraints that the respective constructions need to fulfill and then give asymptotic choices of parameters that meet all those conditions.

The parameter λ is the security parameter in all of our constructions.

B.1 LTF Construction

Parameters. Our LTF scheme in Section 3 has the free parameters

$$n, \overline{m}, m, p, q \in \mathsf{poly}(\lambda) \text{ and } \alpha \in (0, 1).$$

By its construction, we have the following additional parameters:

$$\begin{aligned} w &:= n \cdot \lceil \log_2 q \rceil, \\ N &:= m \cdot \lceil \log_2 q \rceil, \\ u &:= m - \overline{m}, \\ c &:= \left\lfloor \frac{q}{p} \right\rfloor. \end{aligned}$$

We additionally introduce the helping parameter $\overline{u} := u - w$, that was only used implicitly in Section 3.

With those parameters, we achieve a lossiness of

$$\ell := \overline{m} \log_2 p - n \log_2 q - m \log_2(\alpha q \cdot 2\sqrt{\lambda} \cdot N + 1)$$

and an expansion of

$$\chi := \frac{m \cdot \log_2 q}{\overline{m} \cdot \log_2 p}.$$

We give a comprehensive overview of all numbers in Table 1.

Free Parameters		
Symbol	Meaning	Domain
n	Length of LWE Secrets	$\mathsf{poly}(\lambda)$
$\overline{m}$	Dimension of Preimages	$\mathsf{poly}(\lambda)$
m	Dimension of Images	$\mathsf{poly}(\lambda)$
p	Modulus of Preimages	$\mathsf{poly}(\lambda)$
q	Modulus of Images	$\mathsf{poly}(\lambda)$
α	Noise-Modulus Ratio at Key Generation	$(0,1)$
Constructed Parameters		
Symbol	Meaning	Definition
w	Rows of Lattice-Trapdoor	$n \cdot \lceil \log_2 q \rceil$
$\overline{u}$	Columns of Lattice-Trapdoor	$u - w = m - \overline{m} - w$
u	Rows of an Image that do not encrypt the Preimage	$m - \overline{m}$
N	Columns of Key Elements	$m \cdot \lceil \log_2 q \rceil$
c	Scaling Factor at Encryption	$\lfloor q/p \rfloor$
Properties of the Construction		
Symbol	Meaning	Definition
ℓ	Lossiness	$\overline{m} \log_2 p - n \log_2 q$ $-m \log_2(\alpha q \cdot 2\sqrt{\lambda} \cdot N + 1)$
χ	Expansion	$m \cdot \log_2 q / (\overline{m} \cdot \log_2 p)$

Table 1. An exhaustive overview of all parameters appearing in the LTF construction in Section 3. Under *Free Parameters*, we collect parameters that can be chosen freely (under some constraints) when constructing the LTF. Under *Constructed Parameters*, we collect parameters that are determined by free parameters. Under *Properties of the Construction*, we collect quantities that capture qualities of the LTF construction.

Constraints. In the following, we will collect all inequalities that our LTF-construction needs for correctness, indistinguishability, constant expansion and constant relative lossiness.

General Inequalities. For the construction to work at all, we need that the following basic inequalities hold:

$$p < q,$$
$$\overline{m} < m.$$

Correctness. For our LTF scheme to be correct, we need that the sampled gaussian noise matrices have sufficiently small infinity norms:

$$\alpha < \frac{c}{2\sqrt{\lambda} \cdot N \cdot q},$$
$$\alpha < \frac{1}{2\sqrt{\lambda} \cdot N^2}.$$

Indistinguishability. To guarantee that the lossy and injective keys of our LTF scheme are indistinguishable, we need to assume the LWE assumption for parameters n, q, α, m. However, to invoke the LWE assumption, we need that the matrix $A \in \mathbb{Z}_q^{m \times n}$ sampled by the LTF scheme is sufficiently close to uniform. If the inequality

$$\overline{u} \geq w + 2(n + \log_2(w) - 1)$$

is fulfilled, then the statistical distance between the distribution of A and uniform random matrices is bounded by $\leq 2^{-n}$.

Constant Expansion. To achieve a constant expansion factor for the ciphertexts of our LTF, the following asymptotic restrictions must hold:

$$\overline{m} \in \Theta(m),$$
$$\log_2 p \in \Theta(q).$$

Note, that $\overline{m}$ and p are smaller than m and q. So, the above inequalities are equivalent to the existence of constants $c_1, c_2 \in (0, 1)$ s.t.

$$c_1 \cdot m < \overline{m} < m,$$
$$c_2 \cdot \log_2 q < \log_2 p < \log_2 q.$$

Hardness of LWE. To invoke the worst case-average case reduction of [47] for LWE, we additionally include the inequalitiy

$$\alpha q \geq 2\sqrt{n}.$$

Constant Relative Lossiness. For our final IND-SO-CCA secure PKE scheme in Section 5 to have constant expansion, not just any lossiness ℓ does suffice. In fact, we need that both the LTF scheme and the ABM-LTF scheme together retain a number of entropy bits of the random value x that is at least linear in the bit-size of x.

Therefore, we demand here additionally that our LTF scheme retains at least 50%+δ of the entropy of x where $\delta > 0$ is constant. I.e., there must be a constant $\delta > 0$ s.t.

$$\ell \geq \left(\frac{1}{2} + \delta\right) \cdot \overline{m} \cdot \log_2 p.$$

Asymptotic Parameter Choice. We will now give an asymptotic choice for the parameters $n, q, m, \ldots$ of the LTF construction. I.e., instead of giving concrete numbers, we will give a set of functions for the parameters of our construction s.t. each of the above constraints is met. With our parameters we will achieve 60% relative lossiness, i.e.

$$\ell \geq 0.6 \cdot \overline{m} \cdot \log_2 p.$$

We will let q be a prime number in the interval $[n^{21}, 2n^{21}]$. The other free parameters will be parametrized in λ as follows:

$$\begin{aligned}
n &= \lambda \\
p &= \lceil \sqrt{q} \rceil \\
m &= 5n \cdot \lceil \log_2 q \rceil \\
\overline{m} &= 3n \cdot \lceil \log_2 q \rceil - 2(n + \lceil \log_2 q \rceil - 1) \\
\alpha &= 2\sqrt{n}/q.
\end{aligned}$$

For the other parameters, we now have

$$\begin{aligned}
c &= \left\lfloor \frac{q}{p} \right\rfloor = \left\lfloor \frac{q}{\lceil \sqrt{q} \rceil} \right\rfloor \geq \left\lfloor \frac{q}{\sqrt{q}+1} \right\rfloor = \frac{n^{21}}{n^{10.5}+1} \geq n^{10}. \\
\overline{u} &= m - \overline{m} - w = w + 2(n + \log_2 w - 1). \\
N &= 5n \cdot \lceil \log_2 q \rceil^2 .
\end{aligned}$$

We will now argue that all required constraints are fulfilled:

1. *Correctness:* To achieve correctness, we need

$$\alpha < \min\left(\frac{c}{2\sqrt{\lambda} \cdot N \cdot q}, \frac{1}{2\sqrt{\lambda} \cdot N^2}\right). \tag{16}$$

Since $\alpha q = 2\sqrt{n}$, this is equivalent to

$$2\sqrt{n} < \min\left(\frac{c}{2\sqrt{\lambda} \cdot N}, \frac{q}{2\sqrt{\lambda} \cdot N^2}\right). \tag{17}$$

This holds, since we have for both terms on the RHS

$$\frac{c}{2\sqrt{\lambda} \cdot N} \geq \frac{n^{10}}{2\sqrt{n} \cdot 5n \cdot \lceil \log_2 q \rceil^2} \geq n^8, \tag{18}$$

$$\frac{q}{2\sqrt{\lambda}\cdot N^2} \geq \frac{n^{20}}{2\sqrt{n}\cdot\left(5n\cdot\lceil\log_2 q\rceil^2\right)^2} \geq n^{18} \tag{19}$$

for n big enough.

2. *Indistinguishability:* To achieve indistinguishability of evaluation keys we need $\overline{u} \geq w + 2(n + \log_2(w) - 1)$. By our choice of parameters, this inequality is trivially satisfied.
3. *Constant Expansion:* We have

$$\frac{\log_2 q}{\log_2 p} = \frac{\log_2 q}{\log_2(\lceil\sqrt{q}\rceil)} \leq \frac{\log_2 q}{\log_2(\sqrt{q})} = 2$$

$$\frac{m}{\overline{m}} = \frac{5w}{3w - 2(n + \lceil\log_2 q\rceil - 1)} \leq 2$$

for n big enough. We therefore have an expansion of

$$\chi = \frac{\log_2 q}{\log_2 p}\cdot\frac{m}{\overline{m}} \leq 4.$$

4. *Hardness of LWE:* By our choice of parameters, we trivially have $\alpha q \geq 2\sqrt{n}$.
5. *Constant Relative Lossiness:* To achieve a relative lossiness of 60%, we need that

$$\ell = \overline{m}\log_2 p - n\log_2 q - m\log_2(\alpha q\cdot 2\sqrt{\lambda}\cdot N + 1) \geq 0.6\cdot\overline{m}\log_2 p.$$

This is equivalent to the inequality

$$0.4\cdot\overline{m}\log_2 p \geq n\log_2 q + m\log_2(\alpha q\cdot 2\sqrt{\lambda}\cdot N + 1). \tag{20}$$

By dividing with $\overline{m}$ on both sides, we get

$$0.4\cdot\log_2 p \geq \frac{n\log_2 q}{\overline{m}} + \frac{m}{\overline{m}}\log_2(\alpha q\cdot 2\sqrt{\lambda}\cdot N + 1). \tag{21}$$

We can lower bound the LHS by $0.4\cdot\log_2 p \geq 4.2\cdot\log_2 n$. Since

$$\frac{n\log_2 q}{\overline{m}} \leq \frac{w}{3w - 2(n + \lceil\log_2 q\rceil - 1)} \leq \frac{1}{2}$$

and $\frac{m}{\overline{m}} \leq 2$ for n big enough, we can asymptotically upper bound the RHS by

$$\begin{aligned}
&\frac{n\log_2 q}{\overline{m}} + \frac{m}{\overline{m}}\log_2(\alpha q\cdot 2\sqrt{\lambda}\cdot N + 1) && (22)\\
\leq&\frac{1}{2} + 2\log_2(\alpha q\cdot 2\sqrt{\lambda}\cdot N + 1) && (23)\\
\leq&O(1) + 2\log_2(\alpha q\cdot\sqrt{\lambda}) + 2 + 2\log_2 N + 2 && (24)\\
\leq&O(1) + 2\log_2 n + 2 + 2\log_2(5n\cdot\lceil\log_2 q\rceil^2) && (25)\\
\leq&O(1) + 4\log_2(\lceil\log_2 q\rceil) + 4\log_2 n + 2\log_2 5 && (26)\\
\leq&4.2\cdot\log_2 n. && (27)
\end{aligned}$$

Therefore, Eq. (21) does hold.

Stronger Lossiness-Guarantees. Assuming super-poly LWE, our LTF scheme can even achieve relative lossiness arbitrarily close to 1, i.e.

$$L = \frac{\ell}{\overline{m} \cdot \log_2 p} \overset{\lambda\to\infty}{\longrightarrow} 1.$$

In other words, for each $\delta \in (0,1)$, our scheme can have a relative lossiness of δ (for λ big enough).

However, for this end we need to set the modulus q to a super-polynomial prime in $[n^{21+2\omega(1)}, 2n^{21+2\omega(1)}]$. We leave the other parameters as above i.e. $n = \lambda, p = \lceil \sqrt{q} \rceil, m = 5n \cdot \lceil \log_2 q \rceil, \overline{m} = 3n \cdot \lceil \log_2 q \rceil - 2(n + \lceil \log_2 q \rceil - 1)$ and $\alpha = 2\sqrt{n}/q$. Then, the necessary inequalities for correctness, LWE hardness, indistinguishability and constant expansion are fulfilled.

For the relative lossiness, we get

$$L = \frac{\ell}{\overline{m} \cdot \log_2 p} = \frac{\overline{m} \log_2 p - n \log_2 q - m \log_2(\alpha q \cdot 2\sqrt{\lambda} \cdot N + 1)}{\overline{m} \cdot \log_2 p} \tag{28}$$

$$= 1 - \frac{n \log_2 q}{\overline{m}} \cdot \frac{1}{\log_2 p} - \frac{m}{\overline{m}} \cdot \frac{\log_2(\alpha q \cdot 2\sqrt{\lambda} \cdot N + 1)}{\log_2 p}. \tag{29}$$

We already know that $\frac{n \log_2 q}{\overline{m}} \leq \frac{1}{2}$ and $\frac{m}{\overline{m}} \leq 2$ for n large enough. Since $\log_2(\alpha q \cdot 2\sqrt{\lambda} \cdot N + 1) \in O(\log_2 n)$ and $\log_2 p \in \omega(1) \cdot \log_2 n$, we have now

$$\begin{aligned} L = \frac{\ell}{\overline{m} \cdot \log_2 p} &= 1 - \frac{n \log_2 q}{\overline{m}} \cdot \frac{1}{\log_2 p} - \frac{m}{\overline{m}} \cdot \frac{\log_2(\alpha q \cdot 2\sqrt{\lambda} \cdot N + 1)}{\log_2 p} \\ &\geq 1 - \frac{1}{2 \log_2 p} - 2 \cdot \frac{O(\log_2 n)}{\omega(1) \cdot \log_2 n} \geq 1 - \frac{1}{\omega(1)}. \end{aligned}$$

B.2 ABM-LTF Construction

Parameters. Our ABM-LTF scheme in Section 4 has the free parameters

$$n, \overline{m}, m, p, q \in \mathsf{poly}(\lambda), d \in O(\log_2 \lambda) \text{ and } \alpha \in (0,1).$$

By its construction, we have the following additional parameters:

$$\begin{aligned} w &:= n \cdot \lceil \log_2 q \rceil, \\ N &:= m \cdot \lceil \log_2 q \rceil, \\ u &:= m - \overline{m}, \\ c &:= \left\lfloor \frac{q}{p} \right\rfloor. \end{aligned}$$

Again, we introduce the implicit parameter $\overline{u} := u - w$.

We achieve a lossiness and an expansion of

$$\begin{aligned} \ell &:= \overline{m} \log_2 p - n \log_2 q - m \log_2(\alpha q \cdot \sqrt{\lambda} \cdot 4^{d+1} \cdot N^2 + 1), \\ \chi &:= \frac{m \cdot \log_2 q}{\overline{m} \cdot \log_2 p}. \end{aligned}$$

We give a comprehensive overview of all parameters in Table 2.

Free Parameters		
Symbol	Meaning	Domain
λ	Security Parameter and Length of PRF Keys and Outputs	λ
n	Length of LWE Secrets	$\mathsf{poly}(\lambda)$
$\overline{m}$	Dimension of Preimages	$\mathsf{poly}(\lambda)$
m	Dimension of Images	$\mathsf{poly}(\lambda)$
p	Modulus of Preimages	$\mathsf{poly}(\lambda)$
q	Modulus of Images	$\mathsf{poly}(\lambda)$
α	Noise-Modulus Ratio at Key Generation	$(0, 1)$
d	Maximum Length of RC_t for $t \in \{0,1\}^\lambda$	$O(\log_2 \lambda)$
Constructed Parameters		
Symbol	Meaning	Definition
w	Rows of Lattice-Trapdoor	$n \cdot \lceil \log_2 q \rceil$
$\overline{u}$	Columns of Lattice-Trapdoor	$u - w = m - \overline{m} - w$
u	Rows of an Image value that do not encrypt the Preimage	$m - \overline{m}$
N	Columns of Key Elements	$m \cdot \lceil \log_2 q \rceil$
c	Scaling Factor at Evaluation	$\lfloor q/p \rfloor$
Properties of the Construction		
Symbol	Meaning	Definition
ℓ	Lossiness	$\overline{m} \log_2 p - n \log_2 q$ $-m \log_2(\alpha q \cdot \sqrt{\lambda} \cdot 4^{d+1} \cdot N^2 + 1)$
χ	Expansion	$m \cdot \log_2 q / (\overline{m} \cdot \log_2 p)$

Table 2. An exhaustive overview of all parameters appearing in the ABM-LTF construction in Section 4.

Constraints. Our ABM-LTF-construction needs that the following constraints are met.

General Inequalities. Again, we need that basic inequalities hold:

$$p < q, \qquad \overline{m} < m.$$

Correctness. For correctness, we need this time the stricter inequalities

$$\alpha < \frac{c}{\sqrt{\lambda} \cdot 4^{d+1} \cdot N^2 \cdot q},$$
$$\alpha < \frac{1}{\sqrt{\lambda} \cdot 4^{d+1} \cdot N^3}.$$

Indistinguishability. For indistinguishability, we need the same constraint on $\overline{u}$ as in the LTF construction:

$$\overline{u} \geq w + 2(n + \log_2(w) - 1).$$

Constant Expansion. Again, the following asymptotic restrictions must hold:

$$\overline{m} \in \Theta(m), \qquad \log_2 p \in \Theta(q).$$

Hardness of LWE. We, again, include the inequalitiy

$$\alpha q \geq 2\sqrt{n}.$$

Constant Relative Lossiness. We demand that our ABM-LTF scheme retains at least $50\% + \delta$ of the entropy of x where $\delta > 0$ is constant. I.e., there must be a constant $\delta > 0$ s.t.

$$\ell \geq \left(\frac{1}{2} + \delta\right) \cdot \overline{m} \cdot \log_2 p.$$

Asymptotic Parameter Choice. We will now give an asymptotic choice for the parameters $n, q, m, \dots$ of the ABM-LTF construction, where we will only assume that d lies in $O(\log_2 \lambda)$. With our parameters we will achieve 60% relative lossiness.

We will let q be a prime number in the interval $[n^{31+20\frac{d}{\log_2 \lambda}}, 2n^{31+20\frac{d}{\log_2 \lambda}}]$. The other free parameters will be parametrized as in Appendix B.1:

$$\begin{aligned}
n &= \lambda \\
p &= \lceil \sqrt{q} \rceil \\
m &= 5n \cdot \lceil \log_2 q \rceil \\
\overline{m} &= 3n \cdot \lceil \log_2 q \rceil - 2(n + \lceil \log_2 q \rceil - 1) \\
\alpha &= 2\sqrt{n}/q.
\end{aligned}$$

Since most of our parameters coincide with the parameters in Appendix B.1, it can be seen that the necessary inequalities for constant expansion of 4, indistinguishability, and LWE hardness are again satisfied. We will therefore only show that the inequalities for correctness and constant relative lossiness do hold:

1. *Correctness:* To achieve correctness, we need

$$\alpha < \min\left(\frac{c}{\sqrt{\lambda}\cdot 4^{d+1}\cdot N^2\cdot q}, \frac{1}{\sqrt{\lambda}\cdot 4^{d+1}\cdot N^3}\right).$$

Since $\alpha q = 2\sqrt{n}$, this is equivalent to

$$8n\cdot 4^d\cdot N^2 < \min\left(c, \frac{q}{N}\right).$$

We rewrite $4^d = n^{2\cdot\frac{d}{\log_2\lambda}}$ and have that

$$200\cdot\lceil\log_2 q\rceil^2\cdot n^{3+2\cdot\frac{d}{\log_2\lambda}} < \min\left(c, \frac{q}{N}\right)$$

must hold.
This holds, since $q \geq n^{31+20\cdot\frac{d}{\log_2\lambda}}$.

2. *Constant Relative Lossiness:* To achieve a relative lossiness of 60%, we need that

$$\ell = \overline{m}\log_2 p - n\log_2 q - m\log_2(\alpha q\cdot\sqrt{\lambda}\cdot 4^{d+1}\cdot N^2+1) \geq 0.6\cdot\overline{m}\log_2 p.$$

Again, we consider the equivalent inequality

$$0.4\cdot\log_2 p \geq \frac{n\log_2 q}{\overline{m}} + \frac{m}{\overline{m}}\log_2(\alpha q\cdot\sqrt{\lambda}\cdot 4^{d+1}\cdot N^2+1). \tag{30}$$

Again, the terms on the RHS are bounded by $\frac{n\log_2 q}{\overline{m}} \leq \frac{1}{2}$ and $\frac{m}{\overline{m}} \leq 2$. The LHS can be lower bounded by

$$0.4\cdot\log_2 p \geq \frac{2}{5}\cdot\frac{1}{2}\cdot\left(31+20\frac{d}{\log_2\lambda}\right)\log_2 n = \frac{31}{5}\log_2 n + 4d.$$

Inequality Eq. (30) now follows by

$$\begin{aligned}
&\frac{n\log_2 q}{\overline{m}} + \frac{m}{\overline{m}}\log_2(\alpha q\cdot\sqrt{\lambda}\cdot 4^{d+1}\cdot N^2+1) && (31)\\
\leq&\frac{1}{2} + 2\log_2(\alpha q\cdot\sqrt{\lambda}\cdot 4^{d+1}\cdot N^2+1) && (32)\\
\leq&O(1) + 2\log_2(\alpha q\cdot\sqrt{\lambda}) + 4(d+1) + 4\log_2 N + 2 && (33)\\
\leq&O(1) + 2\log_2 n + 2 + 4d + 4\log_2(5n\cdot\lceil\log_2 q\rceil^2) && (34)\\
\leq&O(1) + 4\log_2(\lceil\log_2 q\rceil) + 6\log_2 n + 4d + 4\log_2 5 && (35)\\
\leq&\frac{31}{5}\log_2 n + 4d \leq 0.4\cdot\log_2 p. && (36)
\end{aligned}$$

Stronger Lossiness-Guarantees. Again, assuming super-poly LWE, our ABM-LTF scheme can achieve relative lossiness arbitrarily close to 1. If $q \in [n^{31+20\frac{d}{\log_2 \lambda}+\omega(1)}, 2n^{31+20\frac{d}{\log_2 \lambda}+\omega(1)}]$ is a prime, then the same calculations as in the case of our LTF yield that with this modulus our ABM-LTF achieves any relative lossiness in $(0, 1)$.

B.3 IND-SO-CCA Secure PKE

Parameters. Our PKE scheme has the free parameters λ, n, and message length κ. Additional parameters are the lossiness ℓ' and ℓ and the expansion factor χ of LTF and ABM. We give a comprehensive overview of all parameters in Table 3. A ciphertext of our PKE scheme consists of a ciphertext of LAE, an image of LTF, a core tag part and an image of ABM. Since LAE outputs ciphertexts of size 2κ, core tag parts have a length of λ and LTF and ABM have an expansion factor of χ, it follows that ciphertexts of our PKE scheme consist of $\lambda + 2 \cdot \kappa + 2 \cdot \chi \cdot n$ bits. Since our PKE scheme encrypts messages of length κ, we get a ciphertext expansion factor of

Symbol	Meaning	Domain
λ	Security Parameter and Length of Core Tags	λ
κ	Length of Messages	$\mathsf{poly}(\lambda)$
n	Length of LTF and ABM Preimages	$\mathsf{poly}(\lambda)$
ℓ'	Lossiness of LTF	$O(n)$
ℓ	Lossiness of ABM	$O(n)$
χ	Expansion Factor of LTF and ABM	$O(1)$

Table 3. An exhaustive overview of all parameters appearing in the PKE construction in Section 5.

$$2 + \frac{\lambda + 2 \cdot \chi \cdot n}{\kappa}$$

for our scheme.

Constraints.

Constant Ciphertext Expansion. To achieve a constant ciphertext expansion, we need

$$\kappa \in \Omega(\lambda) \quad \text{and} \quad n \in O(\kappa).$$

Looking back at our asymptotic parameter choices from Appendix B.1 and Appendix B.2 we assume that n is of order $\lambda(\log_2 \lambda)^2$, hence we have to use an LAE with

$$\kappa \in \Omega(\lambda(\log_2 \lambda)^2).$$

Invoking Leftover Hash Lemma. To invoke the Leftover Hash Lemma between Game 7 and Game 8 in the proof of Theorem 3, we need that a random preimage $x \xleftarrow{\$} \{0,1\}^n$ has enough average min-entropy when we expose its (lossy) images under LTF and ABM. With overwhelming probability over the generation of the evaluation keys ek', ek, both schemes will be ℓ' resp. ℓ-lossy. It can be shown that the average min-entropy of x is in that case at least $\ell' + \ell - n$ i.e.

$$\widetilde{\mathbf{H}}_\infty \left(x \mid ek', ek, f_{ek'}(x), t_c, f_{ek,(t_c, f_{ek'}(x))}(x)\right) \geq \ell' + \ell - n.$$

For given ek', ek, t_c, set $f(x) := (ek', ek, f_{ek'}(x), f_{ek,(t_c,f_{ek'}(x))}(x))$. Let $\mathcal{UH} : \{0,1\}^n \to \{0,1\}^{2\kappa}$ be a family of universal hash functions. Then, the generalized Leftover Hash Lemma [17] states that (with overwhelming probability over the generation of ek' and ek) we have the inequality

$$\mathsf{SD}\left((h, h(x), f(x))\,;\,(h, y, f(x))\right) \leq \frac{1}{2}\sqrt{2^{2\kappa - \widetilde{\mathbf{H}}_\infty(x \mid f(x))}} \leq \frac{1}{2}\sqrt{2^{2\kappa+n-\ell'-\ell}}$$

where $h \xleftarrow{\$} \mathcal{UH}, x \xleftarrow{\$} \{0,1\}^n, y \xleftarrow{\$} \{0,1\}^{2\kappa}$ and t_c is computed s.t. $(t_c, f_{ek'}(x))$ is a lossy tag for ABM. Hence, to achieve $\mathsf{SD}\left((h, h(x), f(x))\,;\,(h, y, f(x))\right) \leq 2^{-\lambda-1}$, we need that

$$\ell' + \ell - n \geq 2(\kappa + \lambda). \tag{37}$$

Asymptotic Parameter Choice. We assume here for simplicity that LTF and ABM have the same lossiness, i.e. $\ell = \ell'$. Denote by L the relative lossiness of both schemes, i.e. $L = \frac{\ell}{n}$. We assume that L is a constant in $(0.5, 1)$. Inequality Eq. (37) can then be rewritten as

$$\begin{aligned} (2L-1)n &\geq 2(\kappa+\lambda) \\ \iff n &\geq \frac{2}{2L-1} \cdot (\kappa + \lambda). \end{aligned}$$

By setting $n := \frac{2}{2L-1} \cdot (\kappa + \lambda)$, we have a ciphertext expansion factor of

$$2 + \frac{\lambda + 2\chi \cdot n}{\kappa} = 2 + \frac{\lambda}{\kappa} + \frac{4\chi(\kappa+\lambda)}{\kappa(2L-1)} \tag{38}$$

$$= 2\left(1 + \frac{2\chi}{2L-1}\right) + \frac{\lambda}{\kappa}\left(1 + \frac{4\chi}{2L-1}\right). \tag{39}$$

For messages of length $\kappa \in \Omega(\lambda)$ – in particular for $\kappa \in \Theta(\lambda \cdot (\log_2(\lambda))^2)$ – we obtain an expansion factor upper-bounded by a constant.

If we want to instantiate our PKE scheme with the asymptotic parameter choices we made in Appendix B.1 and Appendix B.2, we need to align the size of image values of LTF and ABM. We can do so by giving LTF the same parameters we gave to ABM in Appendix B.2. In that case, we achieve a relative lossiness of $L \geq 0.6$ and an expansion factor of $\chi \leq 4$. For λ and κ large enough this yields a ciphertext expansion factor of ≤ 83 for our PKE scheme.

In fact, by more careful calculations, we obtain an upper bound for the ciphertext expansion of our PKE scheme of ≤ 70 for $\lambda \geq 64$. This bound is independent of the depth of the PRF PRF used by ABM. Note that for larger messages we obtain a smaller expansion factor.

For a lower bound on Eq. (39) for large messages and optimized parameters, note that since χ must always be greater than 1 and L must always be less than 1, it follows that our PKE scheme must always have a ciphertext expansion factor of at least 6 (in fact, our LTF and ABM schemes can achieve any expansion $\chi > 1$ and lossiness $L < 1$ for appropriate parameters).