# All-But-Many Lossy Trapdoor Functions and Selective Opening Chosen-Ciphertext Security from LWE

Benoît Libert[1,2], Amin Sakzad[3], Damien Stehlé[2], and Ron Steinfeld[3]

[1] CNRS, Laboratoire LIP, France
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France
[3] Faculty of Information Technology, Monash University, Australia

**Abstract.** Selective opening (SO) security refers to adversaries that receive a number of ciphertexts and, after having corrupted a subset of the senders (thus obtaining the plaintexts and the senders' random coins), aim at breaking the security of remaining ciphertexts. So far, very few public-key encryption schemes are known to provide simulation-based selective opening (SIM-SO-CCA2) security under chosen-ciphertext attacks and most of them encrypt messages bit-wise. The only exceptions to date rely on *all-but-many* lossy trapdoor functions (as introduced by Hofheinz; Eurocrypt'12) and the Composite Residuosity assumption. In this paper, we describe the first all-but-many lossy trapdoor function with security relying on the presumed hardness of the Learning-With-Errors problem (LWE) with standard parameters. Our construction exploits homomorphic computations on lattice trapdoors for lossy LWE matrices. By carefully embedding a lattice trapdoor in lossy public keys, we are able to prove SIM-SO-CCA2 security under the LWE assumption. As a result of independent interest, we describe a variant of our scheme whose multi-challenge CCA2 security tightly relates to the hardness of LWE and the security of a pseudo-random function.

**Keywords.** LWE, lossy trapdoor functions, chosen-ciphertext security, selective-opening security, tight security reductions.

## 1 Introduction

Lossy Trapdoor Functions. As introduced by Peikert and Waters [76], lossy tradpoor functions (LTFs) are function families where injective functions – which can be inverted using a trapdoor – are indistinguishable from lossy functions, where the image is much smaller than the domain. The last decade, they received continuous attention (see, e.g., [45, 54, 57, 81, 4, 82]) and found many amazing applications in cryptography. These include black-box realizations of cryptosystems with chosen-ciphertext (IND-CCA2) security [76], deterministic public-key encryption in the standard model [22, 30, 78] and encryption schemes retaining some security in the absence of reliable randomness [9, 11]. As another prominent application, they enabled the design [12, 17] of encryption schemes secure against selective-opening (SO) adversaries, thereby providing an elegant solution to a 10 year-old problem raised by Dwork *et al.* [42].

When it comes to constructing CCA2-secure [77] encryption schemes, LTFs are often combined with *all-but-one* trapdoor functions (ABO-LTFs) [76], which enable a variant of the two-key simulation paradigm [73] in the security proof. In ABO-LTF families, each function takes as arguments an input $x$ and a tag $t$ in such a way that the function $f_{\mathsf{abo}}(t, \cdot)$ is injective for any $t$, except a special tag $t^*$ for which $f_{\mathsf{abo}}(t^*, \cdot)$ behaves as a lossy function. In the security proof of [76], the lossy tag $t^*$ is used to compute the challenge ciphertext,

whereas decryption queries are handled by inverting $f_{\mathsf{abo}}(t, \cdot)$ for all injective tags $t \neq t^\star$. One limitation of ABO-LTFs is the uniqueness of the lossy tag $t^\star$ which must be determined at key generation time. As such, ABO-LTFs are in fact insufficient to prove security in attack models that inherently involve multiple challenge ciphertexts: examples include the key-dependent message [18] and selective opening [12] settings, where multi-challenge security does *not* reduce to single-challenge security via the usual hybrid argument [8].

To overcome the aforementioned shortcoming, Hofheinz [57] introduced *all-but-many* lossy trapdoor functions (ABM-LTFs) which extend ABO-LTFs by allowing the security proof to dynamically create arbitrarily many lossy tags using a trapdoor. Each tag $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ is comprised of an auxiliary component $t_{\mathsf{a}}$ and a core component $t_{\mathsf{c}}$ so that, by generating $t_{\mathsf{c}}$ as a suitable function of $t_{\mathsf{a}}$, the reduction is able to assign a lossy (but random-looking) tag to each challenge ciphertext while making sure that the adversary will be unable to create lossy tags by itself in decryption queries. Using carefully designed ABM-LTFs and variants thereof [58], Hofheinz gave several constructions [57, 58] of public-key encryption schemes in scenarios involving multiple challenge ciphertexts.

Selective Opening Security. In the context of public-key encryption, selective opening (SO) attacks take place in a scenario involving a receiver and $N$ senders. Those encrypt possibly correlated messages $(\mathsf{Msg}_1, \ldots, \mathsf{Msg}_N)$ under the receiver's public key $PK$ and, upon receiving the ciphertexts $(\mathbf{C}_1, \ldots, \mathbf{C}_N)$, the adversary decides to corrupt a subset of the senders. Namely, by choosing $I \subset [N]$, it obtains the messages $\{\mathsf{Msg}_i\}_{i \in I}$ as well as the random coins $\{r_i\}_{i \in I}$ for which $\mathbf{C}_i = \mathsf{Encrypt}(PK, \mathsf{Msg}_i, r_i)$. Then, the adversary aims at breaking the security of unopened ciphertexts $\{\mathbf{C}_i\}_{i \in [N] \setminus I}$. It is tempting to believe that standard notions like semantic security carry over to such adversaries due to the independence of random coins $\{r_i\}_{i \in [N]}$. However, this is not true in general [33] as even the strong standard notion of IND-CCA security [77] was shown [10, 63] not to guarantee anything under selective openings. Proving SO security turns out to be a challenging task for two main reasons. The first one is that the adversary must also obtain the random coins $\{r_i\}_{i \in I}$ of opened ciphertexts (and not only the underlying plaintexts) as reliably erasing them can be very difficult in practice. Note that having the reduction guess the set $I$ of corrupted senders beforehand is not an option since it is only possible with negligible probability $1/\binom{N}{N/2}$. The second difficulty arises from the potential correlation between $\{\mathsf{Msg}_i\}_{i \in I}$ and $\{\mathsf{Msg}_i\}_{i \in [N] \setminus I}$, which hinders the use of standard proof techniques and already makes selective opening security non-trivial to formalize.

Towards defining SO security, the indistinguishability-based (IND-SO) approach [12, 17] demands that unopened plaintexts $\{\mathsf{Msg}_i\}_{i \in [N] \setminus I}$ be indistinguishable from independently resampled ones $\{\mathsf{Msg}'_i\}_{i \in [N] \setminus I}$ conditionally on the adversary's view. However, such definitions are not fully satisfactory. Since $\{\mathsf{Msg}_i\}_{i \in [N]}$ may be correlated, the resampling of $\{\mathsf{Msg}'_i\}_{i \in [N] \setminus I}$ must be conditioned on $\{\mathsf{Msg}_i\}_{i \in I}$ to make the adversary's task non-trivial. This implies that, in the security experiment, the challenger can only be efficient for message distributions that admit efficient conditional resampling, which is a much stronger restriction than efficient samplability. Indeed, many natural message distributions (e.g., where some messages are hard-to-invert functions of other messages) do not support efficient conditional resampling.

Bellare *et al.* [17, 12] defined a stronger, simulation-based (SIM-SO) flavor of selective opening security. This notion mandates that, whatever the adversary outputs after having seen $\{\mathbf{C}_i\}_{i\in[N]}$ and $\{(\mathsf{Msg}_i, r_i)\}_{i\in I}$ can be efficiently simulated from $\{\mathsf{Msg}_i\}_{i\in I}$, without seeing the ciphertexts nor the public key. Unlike its indistinguishability-based counterpart, SIM-SO security does not imply any restriction on the message distributions. While clearly preferable, it turns out to be significantly harder to achieve. Indeed, Böhl *et al.* [21] gave an example of IND-SO-secure scheme that fails to achieve SIM-SO security.

On the positive side, simulation-based chosen-plaintext (SIM-SO-CPA) security was proved attainable under standard number theoretic assumptions like Quadratic Residuosity [17], Composite Residuosity [53] or the Decision Diffie-Hellman assumption [17, 62]. In the chosen-ciphertext (SIM-SO-CCA) scenario, additionally handling decryption queries makes the problem considerably harder: indeed, very few constructions achieve this security property and most of them [44, 64, 66, 68] proceed by encrypting messages in a bit-by-bit manner. The only exceptions [57, 46] to date rely on all-but-many lossy trapdoor functions and Paillier's Composite Residuosity assumption [74].

In this paper, we provide SIM-SO-CCA-secure realizations that encrypt many bits at once under lattice assumptions. Our constructions proceed by homomorphically evaluating a low-depth pseudorandom function (PRF) using the fully homomorphic encryption (FHE) scheme of Gentry, Sahai and Waters [49].

### 1.1 Our Results

Our contribution is three-fold. We first provide an all-but-many lossy trapdoor function based on the Learning-With-Errors ($\mathsf{LWE}$) assumption [79]. We tightly relate the security of our ABM-LTF to that of the underlying PRF and the hardness of the $\mathsf{LWE}$ problem.

As a second result, we use our ABM-LTF to pave the way towards public-key encryption schemes with *tight* (or, more precisely, *almost tight* in the terminology of [36]) chosen-ciphertext security in the multi-challenge setting [8]. By "tight CCA security", as in [61, 67, 59, 47, 60], we mean that the multiplicative gap between the adversary's advantage and the hardness assumption only depends on the security parameter and not on the number of challenge ciphertexts. The strength of the underlying $\mathsf{LWE}$ assumption depends on the specific PRF used to instantiate our scheme. So far, known tightly secure lattice-based PRFs rely on rather strong $\mathsf{LWE}$ assumptions with exponential modulus and inverse error rate [6], or only handle polynomially-bounded adversaries [41] (and hence do not fully exploit the conjectured exponential hardness of $\mathsf{LWE}$). However, any future realization of low-depth PRF with tight security under standard $\mathsf{LWE}$ assumptions (i.e., with polynomial approximation factor) could be plugged into our scheme so as to obtain tight CCA security under the same assumption. Especially, if we had such a tightly secure PRF with an evaluation circuit in $\mathsf{NC}^1$, our scheme would be instantiable with a polynomial-size modulus by translating the evaluation circuit into a branching program via Barrington's theorem [7] and exploiting the asymmetric noise growth of the GSW FHE as in [31, 52].

As a third and main result, we modify our construction so as to prove it secure against selective opening chosen-ciphertext attacks in the indistinguishability-based (i.e., IND-SO-

CCA2) sense. By instantiating our system with a carefully chosen universal hash function, we finally upgrade it from IND-SO-CCA2 to SIM-SO-CCA2 security. For this purpose, we prove that the upgraded scheme is a *lossy encryption* scheme with *efficient opening*. As defined by Bellare *et al.* [17, 12], a lossy encryption scheme is one where normal public keys are indistinguishable from *lossy keys*, for which ciphertexts statistically hide the plaintext. It was shown in [17, 12] that any lossy cryptosystem is in fact IND-SO-CPA-secure. Moreover, if a lossy ciphertext $\mathbf{C}$ can be efficiently opened to any desired plaintext $\mathsf{Msg}$ (i.e., by finding plausible random coins $r$ that explain $\mathbf{C}$ as an encryption of $\mathsf{Msg}$) using the secret key, the scheme also provides SIM-SO-CPA security. We show that our IND-SO-CCA-secure construction has this property when we embed a lattice trapdoor [48, 69] in lossy secret keys.

This provides us with the first multi-bit $\mathsf{LWE}$-based public-key cryptosystem with SIM-SO-CCA security. So far, the only known method [68] to attain the same security notion under quantum-resistant assumptions was to apply a generic construction where each bit of plaintext requires a full key encapsulation (KEM) using a CCA2-secure KEM. In terms of ciphertext size, our system avoids this overhead and can be instantiated with a polynomial-size modulus as long as the underlying PRF can be evaluated in $\mathsf{NC}^1$. For example, the Banerjee-Peikert PRF [5] – which relies on a much weaker $\mathsf{LWE}$ assumption than [6] as it only requires on a slightly superpolynomial modulus – satisfies this condition when the input of the PRF is hardwired into the circuit.

As a result of independent interest, we show that lattice trapdoors can also be used to reach SIM-SO-CPA security in lossy encryption schemes built upon lossy trapdoor functions based on $\mathsf{DDH}$-like assumptions. This shows that techniques from lattice-based cryptography can also come in handy to obtain simulation-based security from conventional number theoretic assumptions.

### 1.2 Our Techniques

Our ABM-LTF construction relies on the observation – previously used in [13, 4] – that the $\mathsf{LWE}$ function $f_{\mathsf{LWE}} : \mathbb{Z}_q^n \times \mathbb{Z}^m \to \mathbb{Z}_q^m : (\mathbf{x}, \mathbf{e}) \to \mathbf{A} \cdot \mathbf{x} + \mathbf{e}$ is lossy. Indeed, under the $\mathsf{LWE}$ assumption, the random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ can be replaced by a matrix of the form $\mathbf{A} = \mathbf{B} \cdot \mathbf{C} + \mathbf{F}$, for a random $\mathbf{B} \in \mathbb{Z}_q^{m \times \ell}$ such that $\ell < n$ and a small-norm $\mathbf{F} \in \mathbb{Z}^{m \times n}$, without the adversary noticing. However, we depart from [13, 4] in several ways.

First, in lossy mode, we sample $\mathbf{C}$ uniformly in $\mathbb{Z}_q^{\ell \times n}$ (rather than as a small-norm matrix as in [13]) because, in order to achieve SIM-SO security, we need to generate $\mathbf{C}$ with a trapdoor. Our application to SIM-SO security also requires to sample $(\mathbf{x}, \mathbf{e})$ from discrete Gaussian distributions, rather than uniformly over an interval as in [13]. Second, we assume that the noise $\mathbf{e} \in \mathbb{Z}^m$ is part of the input instead of using the Rounding technique[4] [6] as in the lossy function of Alwen *et al.* [4]. The reason is that, in our ABM-LTF, we apply the $\mathsf{LWE}$-based function $(\mathbf{x}, \mathbf{e}) \to \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$ for tag-dependent matrices $\mathbf{A}_t$ and, if we were to use the rounding technique, the lower parts of matrices $\mathbf{A}_t$ would have to be statistically independent for different tags. Since we cannot guarantee this independence, we consider the noise term $\mathbf{e}$ to be part of the input. In this case, we can prove that, for any lossy tag, the

[4] The function of [4] maps $\mathbf{x}$ to $f_{\mathsf{LWR}}(\mathbf{x}) = \lfloor (p/q) \cdot \mathbf{A} \cdot \mathbf{x} \rfloor$, for some prime moduli $p < q$.

vector $\mathbf{x}$ retains at least $\Omega(n \log n)$ bits of min-entropy conditionally on $\mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$ and this holds even if $\{\mathbf{A}_t\}_t$ are not statistically independent for distinct lossy tags $t$.

One difficulty is that our ABM-LTF only loses less than half of its input bits for lossy tags, which prevents it from being correlation-secure in the sense of [80]. For this reason, our encryption schemes *cannot* proceed exactly as in [76, 57] by simultaneously outputting an ABM-LTF evaluation $f_{\mathsf{ABM}}(\mathbf{x}, \mathbf{e}) = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$ and a lossy function evaluation $f_{\mathsf{LTF}}(\mathbf{x}, \mathbf{e}) = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}$ as this would leak $(\mathbf{x}, \mathbf{e})$. Fortunately, we can still build CCA2-secure systems by evaluating $f_{\mathsf{LTF}}(\cdot)$ and $f_{\mathsf{ABM}}(\cdot)$ for the same $\mathbf{x}$ and distinct noise vectors $\mathbf{e}_0, \mathbf{e}$. In this case, we can prove that the two functions are jointly lossy: conditionally on $(f_{\mathsf{LTF}}(\mathbf{x}, \mathbf{e}_0), f_{\mathsf{ABM}}(\mathbf{x}, \mathbf{e}))$, the input $\mathbf{x}$ retains $\Omega(n \log n)$ bits of entropy, which allows us to blind the message as $\mathsf{Msg} + h(\mathbf{x})$ using a universal hash function $h$.

Our ABM-LTF extends the all-but-one trapdoor function of Alwen *et al.* [4] by homomorphically evaluating a pseudorandom function. Letting $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ be a lossy matrix and $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ denote the gadget matrix of Micciancio and Peikert [69], the evaluation key of our ABM-LTF contains Gentry-Sahai-Waters (GSW) encryptions $\mathbf{B}_i = \mathbf{R}_i \cdot \bar{\mathbf{A}} + K[i] \cdot \mathbf{G} \in \mathbb{Z}_q^{m \times n}$ of the bits $K[i]$ of a PRF seed $K \in \{0,1\}^\lambda$, where $\mathbf{R}_i \in \{-1, 1\}^{m \times m}$. Given a tag $t = (t_\mathsf{c}, t_\mathsf{a})$, the evaluation algorithm computes a GSW encryption $\mathbf{B}_t = \mathbf{R}_t \cdot \bar{\mathbf{A}} + h_t \cdot \mathbf{G} \in \mathbb{Z}_q^{m \times n}$ of the Hamming distance $h_t$ between $t_\mathsf{c}$ and $\mathsf{PRF}(K, t_\mathsf{a})$ before using $\mathbf{A}_t = [\bar{\mathbf{A}}^\top \mid \mathbf{B}_t^\top]^\top$ to evaluate $f_{\mathsf{ABM}}(\mathbf{x}, \mathbf{e}) = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$. In a lossy tag $t = (\mathsf{PRF}(K, t_\mathsf{a}), t_\mathsf{a})$, we have $h_t = 0$, so that the matrix $\mathbf{A}_t = [\bar{\mathbf{A}}^\top \mid (\mathbf{R}_t \cdot \bar{\mathbf{A}})^\top]^\top$ induces a lossy function $f_{\mathsf{ABM}}(t, \cdot)$. At the same time, any injective tag $t = (t_\mathsf{c}, t_\mathsf{a})$ satisfies $t_\mathsf{c} \neq \mathsf{PRF}(K, t_\mathsf{a})$ and thus $h_t \neq 0$, which allows inverting $f_{\mathsf{ABM}}(\mathbf{x}, \mathbf{e}) = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$ using the public trapdoor [69] of the matrix $\mathbf{G}$.

The pseudorandomness of the PRF ensures that: (i) Lossy tags are indistinguishable from random tags; (ii) They are computationally hard to find without the seed $K$. In order to prove both statements, we resort to the $\mathsf{LWE}$ assumption as the matrix $\bar{\mathbf{A}}$ is not statistically uniform over $\mathbb{Z}_q^{m \times n}$.

Our tightly IND-CCA2-secure public-key cryptosystem uses ciphertexts of the form $(f_{\mathsf{LTF}}(\mathbf{x}, \mathbf{e}_0), f_{\mathsf{ABM}}(\mathbf{x}, \mathbf{e}), \mathsf{Msg} + h(\mathbf{x}))$, where $t_\mathsf{a}$ is the verification key of the one-time signature. Instantiating this scheme with a polynomial-size modulus requires a tightly secure PRF which is computable in $\mathsf{NC}^1$ when the input of the circuit is the *key* (rather than the input of the PRF).[5] To overcome this problem and as a result of independent interest, we provide a tighter proof for the key-homomorphic PRF of Boneh *et al.* [25] (where the concrete security loss is made independent of the number of evaluation queries), which gives us tight CCA2-security under a strong $\mathsf{LWE}$ assumption.

In our IND-SO-CCA2 system, an additional difficulty arises since we cannot use one-time signatures to bind ciphertext components altogether. One alternative is to rely on the hybrid encryption paradigm as in [28] by setting $t_\mathsf{a} = f_{\mathsf{LTF}}(\mathbf{x}, \mathbf{e}_0)$ and encrypting $\mathsf{Msg}$ using a CCA-secure secret-key encryption scheme keyed by $h(\mathbf{x})$. In a direct adaptation of this technique, the chosen-ciphertext adversary can modify $f_{\mathsf{ABM}}(\mathbf{x}, \mathbf{e})$ by re-randomizing the underlying $\mathbf{e}$. Our solution to this problem is to apply the encrypt-then-MAC approach and incorporate

[5] Note that the same holds for the construction of [26], in which the PRF from [6] should be replaced by another one which is in $\mathsf{NC}^1$ as a function the key (e.g., the one from [25]).

$f_{\mathsf{ABM}}(\mathbf{x}, \mathbf{e})$ into the inputs of the MAC so as to prevent the adversary from randomizing $\mathbf{e}$. Using the lossiness of $f_{\mathsf{ABM}}(\cdot)$ and $f_{\mathsf{LTF}}(\cdot)$, we can indeed prove that the hybrid construction provides IND-SO-CCA2 security.

In order to obtain SIM-SO-CCA2 security, we have to show that lossy ciphertexts can be equivocated in the same way as a chameleon hash function. Indeed, the result of [12, 17] implies that any lossy encryption scheme with this property is simulation-secure and the result carries over to the chosen-ciphertext setting. We show that ciphertexts can be trapdoor-opened if we instantiate the scheme using a particular universal hash function $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^L$ which maps $\mathbf{x} \in \mathbb{Z}_q^n$ to $h(\mathbf{x}) = \mathbf{H}_{\mathcal{UH}} \cdot \mathbf{x} \in \mathbb{Z}_q^L$, for a random matrix $\mathbf{H}_{\mathcal{UH}} \in \mathbb{Z}_q^{L \times n}$. In order to generate the evaluation keys $ek'$ and $ek$ of $f_{\mathsf{LTF}}$ and $f_{\mathsf{ABM}}$, we use random matrices $\mathbf{B}_{\mathsf{LTF}} \in \mathbb{Z}_q^{2m \times \ell}$, $\mathbf{C}_{\mathsf{LTF}} \in \mathbb{Z}_q^{\ell \times n}$, $\mathbf{B}_{\mathsf{ABM}} \in \mathbb{Z}_q^{m \times \ell}$, $\mathbf{C}_{\mathsf{ABM}} \in \mathbb{Z}_q^{\ell \times n}$ as well as small-norm $\mathbf{F}_{\mathsf{LTF}} \in \mathbb{Z}^{2m \times n}$, $\mathbf{F}_{\mathsf{ABM}} \in \mathbb{Z}^{m \times n}$ so as to set up lossy matrices $\mathbf{A}_{\mathsf{LTF}} = \mathbf{B}_{\mathsf{LTF}} \cdot \mathbf{C}_{\mathsf{LTF}} + \mathbf{F}_{\mathsf{LTF}}$ and $\mathbf{A}_{\mathsf{ABM}} = \mathbf{B}_{\mathsf{ABM}} \cdot \mathcal{C}_{\mathsf{ABM}} + \mathbf{F}_{\mathsf{ABM}}$. The key idea is to run the trapdoor generation algorithm of [69] to generate a statistically uniform $\mathbf{C} = [\mathbf{C}_{\mathsf{LTF}}^\top \mid \mathbf{C}_{\mathsf{ABM}}^\top \mid \mathbf{H}_{\mathcal{UH}}^\top]^\top \in \mathbb{Z}_q^{(2\ell+L) \times n}$ together with a trapdoor allowing to sample short integer vectors in any coset of the lattice $\Lambda^\perp(\mathbf{C})$. By choosing the target vector $\mathbf{t} \in \mathbb{Z}_q^{2\ell+L}$ as a function of the desired message $\mathsf{Msg}_1$, the initial message $\mathsf{Msg}_0$ and the initial random coins $(\mathbf{x}, \mathbf{e}_0, \mathbf{e})$, we can find a short $\mathbf{x}' \in \mathbb{Z}^n$ such that $\mathbf{C} \cdot \mathbf{x}' = \mathbf{t} \mod q$ and subsequently define $(\mathbf{e}_0', \mathbf{e}') \in \mathbb{Z}^{2m} \times \mathbb{Z}^m$ so that they explain the lossy ciphertext as an encryption of $\mathsf{Msg}_1$ using the coins $(\mathbf{x}', \mathbf{e}_0', \mathbf{e}')$. Moreover, we prove that these have the suitable distribution conditionally on the lossy ciphertext and $\mathsf{Msg}_1$.

### 1.3 Related Work

While selective opening security was first considered by Dwork *et al.* [42], the feasibility of SOA-secure public-key encryption remained open until the work of Bellare, Hofheinz and Yilek [12, 17]. They showed that IND-SO security can be generically achieved from any lossy trapdoor function and, more efficiently, under the DDH assumption. They also achieved SIM-SO-CPA security under the Quadratic Residuosity and DDH assumptions, but at the expense of encrypting messages bitwise. In particular, they proved the SIM-SO security of the Goldwasser-Micali system [50] and their result was extended to Paillier [53]. Hofheinz, Jager and Rupp recently described space-efficient schemes under DDH-like assumption. Meanwhile, the notion of SIM-SO-CPA security was realized in the identity-based setting by Bellare, Waters and Yilek [16]. Recently, Hoang *et al.* [56] investigated the feasibility of SO security using imperfect randomness.

Selective opening security was considered for chosen-ciphertext adversaries in several works [44, 64, 57, 66, 68]. Except constructions [57, 46] based on (variants of) the Composite Residuosity assumption, all of them process messages in a bit-wise fashion, incurring an expansion factor $\Omega(\lambda)$. In the random oracle model [14], much more efficient solutions are possible. In particular, Heuer *et al.* [55] gave evidence that several practical schemes like RSA-OAEP [15] are actually secure in the SIM-SO-CCA sense.

The exact security of public-key encryption in the multi-challenge, multi-user setting was first taken into account by Bellare, Boldyreva and Micali [8] who proved that Cramer-Shoup [37] was tightly secure in the number of users, but not w.r.t. the number $Q$ of challenge

ciphertexts. Using ABM-LTFs, Hofheinz managed to obtain tight multi-challenge security [57] (i.e., without a security loss $\Omega(Q)$ between the advantages of the adversary and the reduction) at the expense of non-standard, variable-size assumptions. Under simple DDH-like assumptions, Hofheinz and Jager [61] gave the first feasibility results in groups with a bilinear map. More efficient tight multi-challenge realizations were given in [67, 59, 47, 60] but, for the time being, the only solutions that do not rely on bilinear maps are those of [47, 60]. In particular, constructions from lattice assumptions have remained lacking so far. By instantiating our scheme with a suitable PRF [6], we take the first step in this direction (albeit under a strong LWE assumption with an exponential approximation factor). Paradoxically, while we can tightly reduce the security of the underlying PRF to the multi-challenge security of our scheme, we do not know how to prove tight multi-user security.

A common feature between our security proofs and those of [67, 59, 47, 60] is that they (implicitly) rely on the technique of the Naor-Reingold PRF [71]. However, while they gradually introduce random values in semi-functional spaces (which do not appear in our setting), we exploit a different degree of freedom enabled by lattices, which is the homomorphic evaluation of low-depth PRFs.

The GSW FHE scheme [49] inspired homomorphic manipulations [24] of Micciancio-Peikert trapdoors [69], which proved useful in the design of attribute-based encryption (ABE) for circuits [24, 32] and fully homomorphic signatures [51]. In particular, the homomorphic evaluation of PRF circuits was considered by Brakerski and Vaikuntanathan [32] to construct an unbounded ABE system. Boyen and Li [26] used similar ideas to build tightly secure IBE and signatures from lattice assumptions. Our constructions depart from [26] in that PRFs are also used in the schemes, and not only in the security proofs. Another difference is that [32, 26] only need PRFs with binary outputs, whereas our ABM-LTFs require a PRF with an exponentially-large range in order to prevent the adversary from predicting its output with noticeable probability.

We finally remark that merely applying the Canetti-Halevi-Katz paradigm [34] to the Boyen-Li IBE [26] does not imply tight CCA2 security in the multi-challenge setting since the proof of [26] is only tight for one identity: in a game with $Q$ challenge ciphertexts, the best known reduction would still lose a factor $Q$ via the standard hybrid argument.

Concurrent Work. In a concurrent and independent paper, Boyen and Li [27] investigated the construction of all-but-many lossy trapdoor functions from LWE and their applications to (selective-opening) CCA2 security.

## 2 Background

For any $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers with addition and multiplication modulo $q$. We always set $q$ as a prime integer. If $\mathbf{x}$ is a vector over $\mathbb{R}$, then $\|\mathbf{x}\|$ denotes its Euclidean norm. If $\mathbf{M}$ is a matrix over $\mathbb{R}$, then $\|\mathbf{M}\|$ denotes its induced norm. We let $\sigma_n(\mathbf{M})$ denote the least singular value of $\mathbf{M}$, where $n$ is the rank of $\mathbf{M}$. For a finite set $S$, we let $U(S)$ denote the uniform distribution over $S$. If $X$ is a random variable over a countable domain, the min-

entropy of $X$ is defined as $H_\infty(X) = \min_x(-\log_2 \Pr[X = x])$. If $X$ and $Y$ are distributions over the same domain, then $\Delta(X, Y)$ denotes their statistical distance.

### 2.1 Randomness Extraction

We first recall the Leftover Hash Lemma, as it was stated in [1].

**Lemma 1 ([1]).** *Let $\mathcal{H} = \{h : X \to Y\}_{h\in\mathcal{H}}$ be a family of universal hash functions, for countable sets $X, Y$. For any random variable $T$ taking values in $X$, we have*

$$\Delta\big((h, h(T)), (h, U(Y))\big) \leq \frac{1}{2} \cdot \sqrt{2^{-H_\infty(T)} \cdot |Y|}.$$

*More generally, let $(T_i)_{i\leq k}$ be independent random variables with values in $X$, for some $k > 0$. We have $\Delta\big((h, (h(T_i))_{i\leq k}), (h, (U(Y))^{(i)})_{i\leq k})\big) \leq \frac{k}{2} \cdot \sqrt{2^{-H_\infty(T)} \cdot |Y|}$.*

A consequence of Lemma 1 was used by Agrawal *et al.* [1] to re-randomize matrices over $\mathbb{Z}_q$ by multiplying them with small-norm matrices.

**Lemma 2 ([1]).** *Let us assume that $m > 2n \cdot \log q$, for some prime $q > 2$. For any integer $k \in \mathsf{poly}(n)$, if $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{k\times n})$, $\mathbf{R} \hookleftarrow U(\{-1, 1\}^{k\times m})$, the distributions $(\mathbf{A}, \mathbf{R} \cdot \mathbf{A})$ and $(\mathbf{A}, \mathbf{B})$ are within $2^{-\Omega(n)}$ statistical distance.*

### 2.2 Reminders on Lattices

Let $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ be a symmetric definite positive matrix, and $\mathbf{c} \in \mathbb{R}^n$. We define the Gaussian function on $\mathbb{R}^n$ by $\rho_{\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) = \exp(-\pi(\mathbf{x} - \mathbf{c})^\top \mathbf{\Sigma}^{-1}(\mathbf{x} - \mathbf{c}))$ and if $\mathbf{\Sigma} = \sigma^2 \cdot \mathbf{I}_n$ and $\mathbf{c} = \mathbf{0}$ we denote it by $\rho_\sigma$. For an $n$-dimensional lattice $\Lambda$, we define $\eta_\varepsilon(\Lambda)$ as the smallest $r > 0$ such that $\rho_{1/r}(\widehat{\Lambda} \setminus \mathbf{0}) \leq \varepsilon$ with $\widehat{\Lambda}$ denoting the dual of $\Lambda$, for any $\varepsilon \in (0, 1)$. In particular, we have $\eta_{2^{-n}}(\mathbb{Z}^n) \leq O(\sqrt{n})$. We denote by $\lambda_1^\infty(\Lambda)$ the infinity norm of the shortest non-zero vector of $\Lambda$.

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, we define $\Lambda^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^\top \cdot \mathbf{A} = \mathbf{0} \bmod q\}$ and $\Lambda(\mathbf{A}) = \mathbf{A} \cdot \mathbb{Z}^n + q\mathbb{Z}^m$.

**Lemma 3 (Adapted from [48, Lemma 5.3]).** *Let $m \geq 2n$ and $q \geq 2$ prime. With probability $\geq 1 - 2^{-\Omega(n)}$, we have $\eta_{2^{-n}}(\Lambda^\perp(\mathbf{A})) \leq \eta_{2^{-m}}(\Lambda^\perp(\mathbf{A})) \leq O(\sqrt{m}) \cdot q^{n/m}$ and $\lambda_1^\infty(\Lambda(\mathbf{A})) \geq q^{1-n/m}/4$.*

Let $\Lambda$ be a full-rank $n$-dimensional lattice, $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ be a symmetric definite positive matrix, and $\mathbf{x}', \mathbf{c} \in \mathbb{R}^n$. We define the discrete Gaussian distribution of support $\Lambda + \mathbf{x}'$ and parameters $\mathbf{\Sigma}$ and $\mathbf{c}$ by $D_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) \sim \rho_{\mathbf{\Sigma},\mathbf{c}}(\mathbf{x})$, for every $\mathbf{x} \in \Lambda + \mathbf{x}'$. For a subset $S \subseteq \Lambda + \mathbf{x}'$, we denote by $D^S_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}$ the distribution obtained by restricting the distribution $D_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}$ to the support $S$. For $\mathbf{x} \in S$, we have $D^S_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) = D_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}(\mathbf{x})/p_a$, where $p_a(S) = D_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}(S)$. Assuming that $1/p_a(S) = n^{O(1)}$, membership in $S$ is efficiently testable and $D_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}$ is efficiently samplable, the distribution $D^S_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}$ can be efficiently sampled from using rejection sampling.

We will use the following standard results on lattice Gaussians.

**Lemma 4 (Adapted from [29, Lemma 2.3]).** *There exists a* ppt *algorithm that, given a basis* $(\mathbf{b}_i)_{i\le n}$ *of a full-rank lattice* $\Lambda$, $\mathbf{x}', \mathbf{c} \in \mathbb{R}^n$ *and* $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ *symmetric definite positive such that* $\Omega(\sqrt{\log n}) \cdot \max_i \|\mathbf{\Sigma}^{-1/2} \cdot \mathbf{b}_i\| \le 1$, *returns a sample from* $D_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}$.

**Lemma 5 (Adapted from [70, Lemma 4.4]).** *For any* $n$*-dimensional lattice* $\Lambda$, $\mathbf{x}', \mathbf{c} \in \mathbb{R}^n$ *and symmetric positive definite* $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ *satisfying* $\sigma_n(\sqrt{\mathbf{\Sigma}}) \ge \eta_{2^{-n}}(\Lambda)$, *we have* $\Pr_{\mathbf{x}\hookleftarrow D_{\Lambda+\mathbf{x}',\mathbf{\Sigma},\mathbf{c}}}[\|\mathbf{x}-\mathbf{c}\| \ge \sqrt{n}\cdot\|\sqrt{\mathbf{\Sigma}}\|] \le 2^{-n+2}$.

**Lemma 6 (Adapted from [70, Lemma 4.4]).** *For any* $n$*-dimensional lattice* $\Lambda$, $\mathbf{x}', \mathbf{c} \in \mathbb{R}^n$ *and symmetric positive definite* $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ *satisfying* $\sigma_n(\sqrt{\mathbf{\Sigma}}) \ge \eta_{2^{-n}}(\Lambda)$, *we have*

$$\rho_{\mathbf{\Sigma},\mathbf{c}}(\Lambda+\mathbf{x}') \;\in\; [1-2^{-n}, 1+2^{-n}]\cdot \det(\Lambda)/\det(\mathbf{\Sigma})^{1/2}.$$

We will also use the following result on the singular values of discrete Gaussian random matrices.

**Lemma 7 ([2, Lemma 8]).** *Assume that* $m \ge 2n$. *Let* $\mathbf{F} \in \mathbb{Z}^{m\times n}$ *with each entry sampled from* $D_{\mathbb{Z},\sigma}$, *for some* $\sigma \ge \Omega(\sqrt{n})$. *Then with probability* $\ge 1-2^{-\Omega(n)}$, *we have* $\|\mathbf{F}\| \le O(\sqrt{m}\sigma)$ *and* $\sigma_n(\mathbf{F}) \ge \Omega(\sqrt{m}\sigma)$.

### 2.3 The Learning With Errors Problem

We recall the Learning With Errors problem [79]. Note that we make the number of samples $m$ explicit in our definition.

**Definition 1.** *Let* $\lambda \in \mathbb{N}$ *be a security parameter and let integers* $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$. *Let* $\chi = \chi(\lambda)$ *be an efficiently samplable distribution over* $\mathbb{Z}_q$. *The* $\mathsf{LWE}_{n,m,q,\chi}$ *assumption posits that the following distance is a negligible function for any* ppt *algorithm* $\mathcal{A}$:

$$\begin{aligned}\mathbf{Adv}^{\mathcal{A},\mathsf{LWE}}_{n,m,q,\chi}(\lambda) := \big|\Pr[\mathcal{A}(1^\lambda,\mathbf{A},\mathbf{u}) = 1 \mid \mathbf{A}\hookleftarrow U(\mathbb{Z}_q^{n\times m}), \mathbf{u}\hookleftarrow U(\mathbb{Z}_q^m)] \\ - \Pr[\mathcal{A}(1^\lambda,\mathbf{A},\mathbf{A}\cdot\mathbf{s}+\mathbf{e}) = 1 \mid \mathbf{A}\hookleftarrow U(\mathbb{Z}_q^{m\times n}), \mathbf{s}\hookleftarrow U(\mathbb{Z}_q^n), \mathbf{e}\hookleftarrow \chi^m]\big|.\end{aligned}$$

A typical choice for $\chi$ is the integer Gaussian distribution $D_{\mathbb{Z},\alpha\cdot q}$ for some parameter $\alpha \in (\sqrt{n}/q, 1)$. In particular, in this case, there exist reductions from standard lattice problems to $\mathsf{LWE}$ (see [79, 29]).

In [69], Micciancio and Peikert described a trapdoor mechanism for $\mathsf{LWE}$. Their technique uses a "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{m\times n}$ for which anyone can publicly sample short vectors $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{x}^\top\mathbf{G} = \mathbf{0}$. As in [69], we call $\mathbf{R} \in \mathbb{Z}^{m\times m}$ a $\mathbf{G}$-trapdoor for a matrix $\mathbf{A} \in \mathbb{Z}_q^{2m\times n}$ if $[\mathbf{R} \mid \mathbf{I}_m]\cdot\mathbf{A} = \mathbf{G}\cdot\mathbf{H}$ for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ which is referred to as the trapdoor tag. If $\mathbf{H} = \mathbf{0}$, then $\mathbf{R}$ is called a "punctured" trapdoor for $\mathbf{A}$.

**Lemma 8 ([69, Section 5]).** *Assume that* $m \ge 2n\log q$. *There exists a* ppt *algorithm* $\mathsf{GenTrap}$ *that takes as inputs matrices* $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m\times n}$, $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ *and outputs matrices* $\mathbf{R} \in \{-1,1\}^{m\times m}$ *and*

$$\mathbf{A} = \begin{bmatrix}\bar{\mathbf{A}} \\ -\mathbf{R}\bar{\mathbf{A}} + \mathbf{G}\mathbf{H}\end{bmatrix} \in \mathbb{Z}_q^{2m\times n}$$

*such that if* $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ *is invertible, then* $\mathbf{R}$ *is a* $\mathbf{G}$*-trapdoor for* $\mathbf{A}$ *with tag* $\mathbf{H}$*; and if* $\mathbf{H} = \mathbf{0}$*, then* $\mathbf{R}$ *is a punctured trapdoor.*

*Further, in case of a* $\mathbf{G}$*-trapdoor, one can efficiently compute from* $\mathbf{A}, \mathbf{R}$ *and* $\mathbf{H}$ *a basis* $(\mathbf{b}_i)_{i\leq 2m}$ *of* $\Lambda^{\perp}(\mathbf{A})$ *such that* $\max_i \|\mathbf{b}_i\| \leq O(m^{3/2})$*.*

Micciancio and Peikert also showed that a $\mathbf{G}$-trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{2m\times n}$ can be used to invert the LWE function $(\boldsymbol{s}, \boldsymbol{e}) \mapsto \mathbf{A}\cdot\boldsymbol{s}+\boldsymbol{e}$, for any $\boldsymbol{s} \in \mathbb{Z}_q^n$ and any sufficiently short $\boldsymbol{e} \in \mathbb{Z}^{2m}$.

**Lemma 9 ([69, Theorem 5.4]).** *There exists a deterministic polynomial time algorithm* $\mathsf{Invert}$ *that takes as inputs matrices* $\mathbf{R} \in \mathbb{Z}^{m\times m}$*,* $\mathbf{A} \in \mathbb{Z}_q^{2m\times n}$*,* $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ *such that* $\mathbf{R}$ *is a* $\mathbf{G}$*-trapdoor for* $\mathbf{A}$ *with invertible tag* $\mathbf{H}$*, and a vector* $\mathbf{A}\cdot\mathbf{s}+\mathbf{e}$ *with* $\mathbf{s} \in \mathbb{Z}_q^n$ *and* $\|\mathbf{e}\| \leq q/(10\cdot\|\mathbf{R}\|)$*, and outputs* $\mathbf{s}$ *and* $\mathbf{e}$*.*

As showed in [49, 24], homomorphic computations can be performed on $\mathbf{G}$-trapdoors with respect to trapdoor tags $\mathbf{H}_i$ corresponding to scalars. As observed in [31], when the circuit belongs to $\mathsf{NC}^1$, it is advantageous to convert the circuit into a branching program, using Barrington's theorem. This is interesting to allow for a polynomial modulus $q$ but imposes a circuit depth restriction (so that the evaluation algorithms are guaranteed to run in polynomial-time).

**Lemma 10 (Adapted from [49, 24]).** *Let* $C : \{0,1\}^{\kappa} \to \{0,1\}$ *be a NAND Boolean circuit of depth* $d$*. Let* $\mathbf{B}_i = \mathbf{R}_i \cdot \bar{\mathbf{A}} + x_i \cdot \mathbf{G} \in \mathbb{Z}_q^{m\times n}$ *with* $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m\times n}$*,* $\mathbf{R}_i \in \{-1,1\}^{m\times m}$ *and* $x_i \in \{0,1\}$*, for* $i \leq \kappa$*.*

- *There exist deterministic algorithms* $\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{CCT}}$ *and* $\mathsf{Eval}^{\mathsf{priv}}_{\mathsf{CCT}}$ *that satisfy:*

$$\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{CCT}}(C, (\mathbf{B}_i)_i) = \mathsf{Eval}^{\mathsf{priv}}_{\mathsf{CCT}}(C, (\mathbf{R}_i)_i) \cdot \bar{\mathbf{A}} + C(x_1, \ldots, x_\kappa) \cdot \mathbf{G},$$

  *and* $\|\mathsf{Eval}^{\mathsf{priv}}_{\mathsf{CCT}}(C, (\mathbf{R}_i)_i)\| \leq m^{O(d)}$*. These algorithms run in time* $\mathsf{poly}(|C|, \kappa, m, n, \log q)$
- *There exist deterministic algorithms* $\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{BP}}$ *and* $\mathsf{Eval}^{\mathsf{priv}}_{\mathsf{BP}}$ *that satisfy:*

$$\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{BP}}(C, (\mathbf{B}_i)_i) = \mathsf{Eval}^{\mathsf{priv}}_{\mathsf{BP}}(C, (\mathbf{R}_i)_i) \cdot \bar{\mathbf{A}} + C(x_1, \ldots, x_\kappa) \cdot \mathbf{G},$$

  *and* $\|\mathsf{Eval}^{\mathsf{priv}}_{\mathsf{BP}}(C, (\mathbf{R}_i)_i)\| \leq 4^d \cdot O(m^{3/2})$*. These algorithms run in time* $\mathsf{poly}(4^d, \kappa, m, n, \log q)$*.*

Note that we impose that the $\mathsf{Eval}^{\mathsf{pub}}$ and $\mathsf{Eval}^{\mathsf{priv}}$ algorithms are deterministic, although probabilistic variants are considered in the literature. This is important in our case, as it will be used in the function evaluation algorithm of our all-but-many lossy trapdoor function family LTF function evaluation.

### 2.4 Lossy Trapdoor Functions

We consider a variant of the notion of Lossy Trapdoor Functions (LTF) introduced by [76], for which the function input may be sampled from a distribution that differs from the uniform distribution. In our constructions, for lossiness security, we actually allow the function evaluation algorithm to sample from a larger domain $\mathsf{Dom}^E_\lambda$ than the domain $\mathsf{Dom}^D_\lambda$ on which the inversion algorithm guaranteed to succeed. A sample over $\mathsf{Dom}^E_\lambda$ has an overwhelming probability to land in $\mathsf{Dom}^D_\lambda$ with respect to the sampling distribution.

**Definition 2.** *For an integer $l(\lambda) > 0$, a family of $l$-lossy trapdoor functions* $\mathsf{LTF}$ *with security parameter $\lambda$, evaluation sampling domain $\mathsf{Dom}_\lambda^E$, efficiently samplable distribution $D_{\mathsf{Dom}_\lambda^E}$ on $\mathsf{Dom}_\lambda^E$, inversion domain $\mathsf{Dom}_\lambda^D \subseteq \mathsf{Dom}_\lambda^E$ and range $\mathsf{Rng}_\lambda$ is a tuple of* ppt *algorithms* $(\mathsf{IGen}, \mathsf{LGen}, \mathsf{Eval}, \mathsf{Invert})$ *with the following functionalities:*

**Injective key generation.** $\mathsf{LTF.IGen}(1^\lambda)$ *outputs an evaluation key $ek$ for an injective function together with an inversion key $ik$.*

**Lossy key generation.** $\mathsf{LTF.LGen}(1^\lambda)$ *outputs an evaluation key $ek$ for a lossy function. In this case, there is no inversion key and we define $ik = \perp$.*

**Evaluation.** $\mathsf{LTF.Eval}(ek, X)$ *takes as inputs the evaluation key $ek$ and a function input $X \in \mathsf{Dom}_\lambda^E$. It outputs an image $Y = f_{ek}(X)$.*

**Inversion.** $\mathsf{LTF.Invert}(ik, Y)$ *inputs the inversion key $ik \neq \perp$ and a $Y \in \mathsf{Rng}_\lambda$. It outputs the unique $X = f_{ik}^{-1}(Y)$ such that $Y = f_{ek}(X)$ (if it exists).*

*In addition,* $\mathsf{LTF}$ *has to meet the following requirements:*

**Inversion Correctness.** *For an injective key pair $(ek, ik) \leftarrow \mathsf{LTF.IGen}(1^\lambda)$, we have, except with negligible probability over $(ek, ik)$, that for all inputs $X \in \mathsf{Dom}_\lambda^D$, $X = f_{ik}^{-1}(f_{ek}(X))$.*

**Eval Sampling Correctness.** *For $X$ sampled from $D_{\mathsf{Dom}_\lambda^E}$, we have $X \in \mathsf{Dom}_\lambda^D$ except with negligible probability.*

**$l$-Lossiness.** *For $(ek, \perp) \leftarrow\!\!\!\shortmid\ \mathsf{LTF.LGen}(1^\lambda)$ and $X \leftarrow\!\!\!\shortmid\ D_{\mathsf{Dom}_\lambda^E}$, we have*

$$H_\infty(X \mid ek = \overline{ek}, f_{ek}(X) = \overline{y}) \geq l,$$

*for all $(\overline{ek}, \overline{y})$ except a set of negligible probability.*

**Indistinguishability.** *The distribution of lossy functions is computationally indistinguishable from that of injective functions, namely:*

$$\mathbf{Adv}^{\mathcal{A},\mathsf{LTF}}(\lambda) := \big|\Pr[\mathcal{A}(1^\lambda, ek) = 1 \mid (ek, ik) \leftarrow\!\!\!\shortmid\ \mathsf{LTF.IGen}(1^\lambda)] - \Pr[\mathcal{A}(1^\lambda, ek) = 1 \mid (ek, \perp) \leftarrow\!\!\!\shortmid\ \mathsf{LTF.LGen}(1^\lambda)]\big|$$

*is a negligible function for any* ppt *algorithm $\mathcal{A}$.*

### 2.5 All-But-Many Lossy Trapdoor Functions

We consider a variant of the definition of All-But-Many Lossy Trapdoor Functions (ABM-LTF) from [57], in which the distribution over the domain may not be the uniform one.

**Definition 3.** *For an integer $l(\lambda) > 0$, a family of all-but-many $l$- lossy trapdoor functions* $\mathsf{ABM}$ *with security parameter $\lambda$, evaluation sampling domain $\mathsf{Dom}_\lambda^E$, efficiently samplable distribution $D_{\mathsf{Dom}_\lambda^E}$ on $\mathsf{Dom}_\lambda^E$, inversion domain $\mathsf{Dom}_\lambda^D \subseteq \mathsf{Dom}_\lambda^E$, and range $\mathsf{Rng}_\lambda$ consists of the following* ppt *algorithms:*

**Key generation.** $\mathsf{ABM.Gen}(1^\lambda)$ *outputs an evaluation key $ek$, an inversion key $ik$ and a tag key $tk$. The evaluation key $ek$ defines a set $\mathcal{T} = \mathcal{T}_\mathsf{c} \times \mathcal{T}_\mathsf{a}$ containing the disjoint sets of lossy tags $\mathcal{T}_\mathsf{loss}$ and injective tags $\mathcal{T}_\mathsf{inj}$. Each tag $t = (t_\mathsf{c}, t_\mathsf{a})$ is described by a core part $t_\mathsf{c} \in \mathcal{T}_\mathsf{c}$ and an auxiliary part $t_\mathsf{a} \in \mathcal{T}_\mathsf{a}$.*

**Evaluation.** $\mathsf{ABM.Eval}(ek, t, X)$ *takes as inputs an evaluation key* $ek$*, a tag* $t \in \mathcal{T}$ *and a function input* $X \in \mathsf{Dom}_\lambda^E$*. It outputs an image* $Y = f_{ek,t}(X)$*.*

**Inversion.** $\mathsf{ABM.Invert}(ik, t, Y)$ *takes as inputs an inversion key* $ik$*, a tag* $t \in \mathcal{T}$ *and a* $Y \in \mathsf{Rng}_\lambda$*. It outputs the unique* $X = f_{ik,t}^{-1}(Y)$ *such that* $Y = f_{ek,t}(X)$*.*

**Lossy tag generation.** $\mathsf{ABM.LTag}(tk, t_\mathsf{a})$ *takes as input an auxiliary part* $t_\mathsf{a} \in \mathcal{T}_\mathsf{a}$ *and outputs a core part* $t_\mathsf{c}$ *such that* $t = (t_\mathsf{c}, t_\mathsf{a})$ *forms a lossy tag.*

*In addition,* ABM *has to meet the following requirements:*

**Inversion Correctness.** *For* $(ek, ik, tk)$ *produced by* $\mathsf{ABM.Gen}(1^\lambda)$*, we have, except with negligible probability over* $(ek, ik, tk)$*, that for all injective tags* $t \in \mathcal{T}_\mathsf{inj}$ *and all inputs* $X \in \mathsf{Dom}_\lambda^D$*, that* $X = f_{ik,t}^{-1}(f_{ek,t}(X))$*.*

**Eval Sampling Correctness.** *For* $X$ *sampled from* $D_{\mathsf{Dom}_\lambda^E}$*, we have* $X \in \mathsf{Dom}_\lambda^D$ *except with negligible probability.*

**Lossiness.** *For* $(ek, ik, tk) \hookleftarrow \mathsf{ABM.Gen}(1^\lambda)$*, any* $t_\mathsf{a} \in \mathcal{T}_\mathsf{a}$*,* $t_\mathsf{c} \hookleftarrow \mathsf{ABM.LTag}(tk, t_\mathsf{a})$ *and* $X \hookleftarrow D_{\mathsf{Dom}_\lambda^E}$*, we have that* $H_\infty(X \mid ek = \overline{ek}, f_{ek,(t_\mathsf{c},t_\mathsf{a})}(X) = \overline{y}) \geq l$*, for all* $(\overline{ek}, \overline{y})$ *except a set of negligible probability.*

**Indistinguishability.** *Multiple lossy tags are computationally indistinguishable from random tags, namely:*

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

*is negligible for any* ppt *algorithm* $\mathcal{A}$*, where* $(ek, ik, tk) \hookleftarrow \mathsf{ABM.Gen}(1^\lambda)$ *and* $\mathcal{O}_{\mathcal{T}_\mathsf{c}}(\cdot)$ *is an oracle that assigns a random core tag* $t_\mathsf{c} \hookleftarrow U(\mathcal{T}_\mathsf{c})$ *to each auxiliary tag* $t_\mathsf{a} \in \mathcal{T}_\mathsf{a}$ *(rather than a core tag that makes* $t = (t_\mathsf{c}, t_\mathsf{a})$ *lossy). Here* $Q$ *denotes the number of oracle queries made by* $\mathcal{A}$*.*

**Evasiveness.** *Non-injective tags are computationally hard to find, even with access to an oracle outputting multiple lossy tags, namely:*

$$\mathbf{Adv}_{Q_1,Q_2}^{\mathcal{A},\mathsf{eva}}(\lambda) := \Pr[\mathcal{A}(1^\lambda, ek)^{\mathsf{ABM.LTag}(tk,\cdot),\mathsf{ABM.IsLossy}(tk,\cdot)} \in \mathcal{T} \backslash \mathcal{T}_\mathsf{inj}]$$

*is negligible for legitimate adversary* $\mathcal{A}$*, where* $(ek, ik, tk) \hookleftarrow \mathsf{ABM.Gen}(1^\lambda)$ *and* $\mathcal{A}$ *is given access to the following oracles:*

- $\mathsf{ABM.LTag}(tk, \cdot)$ *which acts exactly as the lossy tag generation algorithm.*
- $\mathsf{ABM.IsLossy}(tk, \cdot)$ *that takes as input a tag* $t = (t_\mathsf{c}, t_\mathsf{a})$ *and outputs* $1$ *if* $t \in \mathcal{T} \backslash \mathcal{T}_\mathsf{inj}$ *and otherwise outputs* $0$*.*

*We denote by* $Q_1$ *and* $Q_2$ *the number of queries to these two oracles. By "legitimate adversary", we mean that* $\mathcal{A}$ *is* ppt *and never outputs a tag* $t = (t_\mathsf{c}, t_\mathsf{a})$ *such that* $t_\mathsf{c}$ *was obtained by invoking the* ABM.LTag *oracle on* $t_\mathsf{a}$*.*

As pointed out in [57], the evasiveness property mirrors the notion of strong unforgeability for signature schemes. Indeed, the adversary is considered successful even if it outputs a $(t_\mathsf{c}, t_\mathsf{a})$ such that $t_\mathsf{a}$ was submitted to $\mathsf{ABM.LTag}(tk, \cdot)$ as long as the response $t'_\mathsf{a}$ of the latter was such that $t'_\mathsf{a} \neq t_\mathsf{a}$.

In order to simplify the tight proof of our public-key encryption scheme, we slightly modified the original definition of evasiveness in [57] by introducing a lossiness-testing oracle $\mathsf{ABM.IsLossy}(tk, \cdot)$. When it comes to proving tight CCA security, it will save the reduction from having to guess which decryption query contradicts the evasiveness property of the underlying ABM-LTF.

### 2.6 Selective-Opening Chosen-Ciphertext Security

A public-key encryption scheme consists of a tuple (Par-Gen, Keygen, Encrypt,Decrypt) of ppt algorithms, where Par-Gen takes as input a security parameter $1^\lambda$ and generates common public parameters $\Gamma$, Keygen takes in $\Gamma$ and outputs a key pair $(SK, PK)$, while Encrypt and Decrypt proceed in the usual way.

As a first step, we will consider encryption schemes that provide SO security in the sense of an indistinguishability-based definition (or IND-SOA security). This notion is captured by a game where the adversary obtains $N(\lambda)$ ciphertexts, opens an arbitrary subset of these (meaning that it obtains both the plaintexts and the encryption coins) and asks that remaining ciphertexts be indistinguishable from messages that are independently re-sampled conditionally on opened ones. In the IND-SO-CCA2 scenario, this should remain true even if the adversary has a decryption oracle. A formal definition is recalled in Appendix B.3.

A stronger notion is that of simulation-based security, which demands that an efficient simulator be able to perform about as well as the adversary without seeing neither the ciphertexts nor the public key. Formally, two experiments are required to have indistinguishable output distributions.

In the real experiment, the challenger samples $\mathbf{Msg} = (\mathsf{Msg}_1, \ldots, \mathsf{Msg}_N) \leftarrow \mathcal{M}$ from the joint message distribution and picks random coins $r_1, \ldots, r_N \leftarrow \mathcal{R}$ to compute ciphertexts $\{\mathbf{C}_i \leftarrow \mathsf{Encrypt}(PK, \mathsf{Msg}_i, r_i)\}_{i\in[N]}$ which are given to the adversary $\mathcal{A}$. The latter responds by choosing a subset $I \subset [N]$ and gets back $\{(\mathsf{Msg}_i, r_i)\}_{i\in I}$. The adversary $\mathcal{A}$ outputs a string $out_{\mathcal{A}}$ and the output of the experiment is a predicate $\mathfrak{R}(\mathcal{M}, \mathbf{Msg}, out_{\mathcal{A}})$.

In the ideal experiment, the challenger samples $\mathbf{Msg} = (\mathsf{Msg}_1, \ldots, \mathsf{Msg}_N) \leftarrow \mathcal{M}$ from the joint message distribution. Without seeing any encryptions, the simulator chooses a subset $I$ and some state information $st$. After having seen the messages $\{\mathsf{Msg}_i\}_{i\in I}$ and the state information but without seeing any randomness, the simulator outputs a string $out_S$. The outcome of the ideal experiment is the predicate $\mathfrak{R}(\mathcal{M}, \mathbf{Msg}, out_S)$. As in [44, 62], we allow the adversary to choose the message distribution $\mathcal{M}$. While this distribution should be efficiently samplable, it is *not* required to support efficient conditional re-sampling.

**Definition 4 ([44, 62]).** *A public-key encryption scheme* (Par-Gen, Keygen, Encrypt, Decrypt) *provides* **simulation-based selective opening** *(SIM-SO-CPA) security if, for any* ppt *function* $\mathfrak{R}$ *and any* ppt *adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ *in the real experiment* $\mathbf{Exp}^{\mathsf{cpa\text{-}so\text{-}real}}(\lambda)$*, there is an efficient simulator* $S = (S_0, S_1, S_2)$ *in the ideal experiment* $\mathbf{Exp}^{\mathsf{so\text{-}ideal}}(\lambda)$ *s.t.*

$$|\Pr[\mathbf{Exp}^{\mathsf{cpa\text{-}so\text{-}real}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathsf{so\text{-}ideal}}(\lambda) = 1]|$$

*is negligible, where the two experiments are defined as follows:*

| $\mathbf{Exp}^{\mathsf{cpa\text{-}so\text{-}real}}(\lambda)$: | $\mathbf{Exp}^{\mathsf{so\text{-}ideal}}(\lambda)$: |
|---|---|
| $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$; | $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$; |
| $(PK, SK) \leftarrow \mathsf{Keygen}(\Gamma)$ | $(\mathcal{M}, st_0) \leftarrow S_0(\Gamma)$ |
| $(\mathcal{M}, st_0) \leftarrow \mathcal{A}_0(PK, \Gamma)$ | $\mathbf{Msg} = (\mathsf{Msg}_1, \ldots, \mathsf{Msg}_N) \leftarrow \mathcal{M}$ |
| $\mathbf{Msg} = (\mathsf{Msg}_1, \ldots, \mathsf{Msg}_N) \leftarrow \mathcal{M}$ | $(I, st_1) \leftarrow S_1(st_0, 1^{\lvert\mathsf{Msg}_i\rvert})$ |
| $r_1, \ldots, r_n \leftarrow \mathcal{R}$ | $out_S \leftarrow S_2\big(st_1, \{\mathsf{Msg}_i\}_{i\in I}\big)$ |
| $\mathbf{C}_i \leftarrow \mathsf{Encrypt}(PK, \mathsf{Msg}_i, r_i) \;\; \forall i \in [N]$, | Output $\mathfrak{R}(\mathcal{M}, \mathbf{Msg}, out_S)$ |
| $(I, st_1) \leftarrow \mathcal{A}_1\big(st_0, \mathbf{C}_1, \ldots, \mathbf{C}_N\big)$ | |
| $out_{\mathcal{A}} \leftarrow \mathcal{A}_2\big(st_1, (\mathsf{Msg}_i, r_i)_{i\in I}\big)$ | |
| Output $\mathfrak{R}(\mathcal{M}, \mathbf{Msg}, out_{\mathcal{A}})$ | |

*As usual, the adversarially-chosen message distribution $\mathcal{M}$ is efficiently samplable and encoded as a polynomial-size circuit.*

The notion of simulation-based chosen-ciphertext (SIM-SO-CCA) security is defined analogously. The only difference is in the real experiment $\mathbf{Exp}^{\mathsf{cca\text{-}so\text{-}real}}$, which is obtained from $\mathbf{Exp}^{\mathsf{cpa\text{-}so\text{-}real}}$ by granting the adversary access to a decryption oracle at all stages. Of course, the adversary is disallowed to query the decryption of any ciphertext in the set $\{\mathbf{C}_i\}_{i\in[N]}$ of challenge ciphertexts.

It is known [12] that SIM-SO-CPA security can be achieved from lossy encryption schemes [17] when there exists an efficient $\mathsf{Opener}$ algorithm which, using the lossy secret key, can explain a lossy ciphertext $\mathbf{C}$ as an encryption of any given plaintext. As observed in [17, 62], this $\mathsf{Opener}$ algorithm can use the initial coins used in the generation of $\mathbf{C}$ for this purpose. This property, formalized by Definition 11, is called efficient weak opening.

# 3 An All-But-Many Lossy Trapdoor Function from LWE

As a warm-up, we first describe a variant of the lossy trapdoor function suggested by Bellare *et al.* [13, Section 5.2] that is better suited to our needs. We then extend this LWE-based LTF into an ABM-LTF in Section 3.2.

## 3.1 An LWE-Based Lossy Trapdoor Function

All algorithms use a prime modulus $q > 2$, integers $n \in \mathsf{poly}(\lambda)$, $m \geq 2n \log q$ and $\ell > 0$, an $\mathsf{LWE}$ noise distribution $\chi$, and parameters $\sigma_x, \sigma_e, \gamma_x, \gamma_e > 0$. The function evaluation sampling domain $\mathsf{Dom}^E_\lambda = \mathsf{Dom}^E_x \times \mathsf{Dom}^E_e$ where $\mathsf{Dom}^E_x$ (resp. $\mathsf{Dom}^E_e$) is the set of $\mathbf{x}$ (resp. $\mathbf{e}$) in $\mathbb{Z}^n$ (resp. $\mathbb{Z}^{2m}$) with $\|\boldsymbol{x}\| \leq \gamma_x \cdot \sqrt{n} \cdot \sigma_x$ (resp. $\|\boldsymbol{e}\| \leq \gamma_e \sqrt{2m} \cdot \sigma_e$). Its inversion domain is $\mathsf{Dom}^D_\lambda = \mathsf{Dom}^D_x \times \mathsf{Dom}^D_e$, where $\mathsf{Dom}^D_x$ (resp. $\mathsf{Dom}^D_e$) is the set of $\mathbf{x}$ (resp. $\mathbf{e}$) in $\mathbb{Z}^n$ (resp. $\mathbb{Z}^{2m}$) with $\|\boldsymbol{x}\| \leq \sqrt{n} \cdot \sigma_x$ (resp. $\|\boldsymbol{e}\| \leq \sqrt{2m} \cdot \sigma_e$) and its range is $\mathsf{Rng}_\lambda = \mathbb{Z}_q^{2m}$. The function inputs are sampled from the distribution $D_{\mathsf{Dom}^E_\lambda} = D^{\mathsf{Dom}^E_x}_{\mathbb{Z}^n,\sigma_x} \times D^{\mathsf{Dom}^E_e}_{\mathbb{Z}^{2m},\sigma_e}$.

**Injective key generation.** $\mathsf{LTF.IGen}(1^\lambda)$ samples $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and runs $(\mathbf{A}, \mathbf{R}) \hookleftarrow \mathsf{GenTrap}(\bar{\mathbf{A}}, \mathbf{I}_n)$ to obtain $\mathbf{A} \in \mathbb{Z}_q^{2m\times n}$ together with a $\mathbf{G}$-trapdoor $\mathbf{R} \in \{-1, 1\}^{m\times m}$. It outputs $ek := \mathbf{A}$ and $ik := \mathbf{R}$.

**Lossy key generation.** $\mathsf{LTF.LGen}(1^\lambda)$ generates $\mathbf{A} \in \mathbb{Z}_q^{2m\times n}$ as a matrix of the form $\mathbf{A} = \mathbf{B}\cdot\mathbf{C} + \mathbf{F}$ with $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{2m\times\ell})$, $\mathbf{C} \hookleftarrow U(\mathbb{Z}_q^{\ell\times n})$ and $\mathbf{F} \hookleftarrow \chi^{2m\times n}$. It outputs $ek := \mathbf{A}$ and $ik :=\perp$.

**Evaluation.** $\mathsf{LTF.Eval}(ek, (\mathbf{x}, \mathbf{e}))$ takes as input a domain element $(\mathbf{x}, \mathbf{e}) \in \mathsf{Dom}_\lambda^E$ and maps it to $\mathbf{y} = \mathbf{A}\cdot\mathbf{x} + \mathbf{e} \in \mathbb{Z}_q^{2m}$.

**Inversion.** $\mathsf{LTF.Invert}(ik, \mathbf{y})$ inputs a vector $\mathbf{y} \in \mathbb{Z}_q^{2m}$, uses the $\mathbf{G}$-trapdoor $ik = \mathbf{R}$ of $\mathbf{A}$ to find the unique $(\mathbf{x}, \mathbf{e}) \in \mathsf{Dom}_\lambda^D$ such that $\mathbf{y} = \mathbf{A}\cdot\mathbf{x}+\mathbf{e}$. This is done by applying the $\mathsf{LWE}$ inversion algorithm from Lemma 9.

Note that the construction differs from the lossy function of [13] in two ways. First, in [13], the considered distribution over the function domain is uniform over a parallelepiped. We instead consider a discrete Gaussian distribution. Second, in [13], the matrix $\mathbf{C}$ is chosen as a small-norm integer matrix sampled from the $\mathsf{LWE}$ noise distribution. We instead sample it uniformly. Both modifications are motivated by our application to SO-CCA security. Indeed, in the security proof, we will generate $\mathbf{C}$ along with a lattice trapdoor (using $\mathsf{GenTrap}$), which we will use to simulate the function domain distribution conditioned on an image value.

We first study the conditional distribution of the pair $(\mathbf{x}, \mathbf{e})$ given its image under a lossy function. This will be used to quantify the lossiness of the LTF.

**Lemma 11.** *Let $\mathbf{C} \in \mathbb{Z}_q^{\ell\times n}$ and $\mathbf{F} \in \mathbb{Z}^{2m\times n}$. Sample $(\mathbf{x}, \mathbf{e}) \hookleftarrow D_{\mathbb{Z}^n,\sigma_x}^{\mathsf{Dom}_x} \times D_{\mathbb{Z}^{2m},\sigma_e}^{\mathsf{Dom}_e}$ and define $(\mathbf{u}, \mathbf{f}) = (\mathbf{C}\cdot\mathbf{x}, \mathbf{F}\cdot\mathbf{x} + \mathbf{e}) \in \mathbb{Z}_q^n \times \mathbb{Z}^{2m}$. Note that $\mathbf{e}$ is fully determined by $\mathbf{x}, \mathbf{u}$ and $\mathbf{f}$. Further, the conditional distribution of $\mathbf{x}$ given $(\mathbf{u}, \mathbf{f})$ is $D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\mathbf{\Sigma}},\mathbf{c}}^{S_{\mathbf{F},\mathbf{u},\mathbf{f}}}$, with support*

$$S_{\mathbf{F},\mathbf{u},\mathbf{f}} = \left\{\bar{\mathbf{x}} \in \Lambda^\perp(\mathbf{C}^\top) + \mathbf{x}' : \bar{\mathbf{x}} \in \mathsf{Dom}_x,\ \mathbf{f} - \mathbf{F}\cdot\bar{\mathbf{x}} \in \mathsf{Dom}_e\right\},$$

*where $\mathbf{x}'$ is any solution to $\mathbf{C}\cdot\mathbf{x}' = \mathbf{u}$ and:*

$$\mathbf{\Sigma} = \sigma_x^2\cdot\sigma_e^2\cdot(\sigma_x^2\cdot\mathbf{F}^\top\cdot\mathbf{F} + \sigma_e^2\cdot\mathbf{I}_n)^{-1}, \quad \mathbf{c} = \sigma_x^2\cdot(\sigma_x^2\cdot\mathbf{F}^\top\cdot\mathbf{F} + \sigma_e^2\cdot\mathbf{I}_n)^{-1}\cdot\mathbf{F}^\top\cdot\mathbf{f}.$$

*Proof.* We first remark that the support of $\mathbf{x}|(\mathbf{u}, \mathbf{f})$ is $S_{\mathbf{F},\mathbf{u},\mathbf{f}}$, since the set of solutions $\bar{\mathbf{x}} \in \mathbb{Z}^n$ to $\mathbf{u} = \mathbf{C}\cdot\mathbf{x} \in \mathbb{Z}_q^\ell$ is $\Lambda^\perp(\mathbf{C}^\top) + \mathbf{x}'$ and each such $\bar{\mathbf{x}}$ has a non-zero conditional probability if and only if the corresponding $\bar{\mathbf{e}} = \mathbf{f} - \mathbf{F}\cdot\mathbf{x}$ is in $\mathsf{Dom}_e$. Now, for $\bar{\mathbf{x}} \in \mathbb{Z}^n$ in the support $S_{\mathbf{F},\mathbf{u},\mathbf{f}}$, we have

$$\begin{aligned}\Pr[\mathbf{x} = \bar{\mathbf{x}}|(\mathbf{u}, \mathbf{f})] &\sim D_{\mathbb{Z}^n,\sigma_x}(\bar{\mathbf{x}})\cdot D_{\mathbb{Z}^{2m},\sigma_e}(\mathbf{f} - \mathbf{F}\cdot\bar{\mathbf{x}}) \\ &\sim \exp\left(-\pi\left(\frac{\|\bar{\mathbf{x}}\|^2}{\sigma_x^2} + \frac{\|\mathbf{f} - \mathbf{F}\cdot\bar{\mathbf{x}}\|^2}{\sigma_e^2}\right)\right) \\ &\sim \exp\left(-\pi\left((\bar{\mathbf{x}} - \mathbf{c})^\top\cdot\mathbf{\Sigma}^{-1}\cdot(\bar{\mathbf{x}} - \mathbf{c})\right)\right).\end{aligned}$$

The last equality follows from expanding the norms and collecting terms. □

We now formally state for which parameters we can prove that the scheme above is an LTF. The second part of the theorem will be useful for our SO-CCA encryption application.

**Theorem 1.** *Let $\chi = D_{\mathbb{Z},\beta/(2\sqrt{\lambda})}$ for some $\beta > 0$. Let us assume that $\ell \geq \lambda$, $n = \Omega(\ell \log q)$ and $m \geq 2n \log q$, $\gamma_x \geq 3\sqrt{m/n}$ and $\gamma_e \geq 3$. Assume further that $\sigma_x \geq \Omega(n)$, $\sigma_e \leq O(q/m^{3/2})$ and $\sigma_e \geq \Omega(\sqrt{mn} \cdot \beta \cdot \sigma_x)$. Then, under the $\mathsf{LWE}_{\ell,2m,q,\chi}$ hardness assumption, the above construction is an $l$-lossy LTF with $l \geq n \log \sigma_x - 2 - \ell \log q > \Omega(n \log n)$. Further, any $\mathsf{ppt}$ indistinguishability adversary $\mathcal{A}$ implies an $\mathsf{LWE}$ distinguisher $\mathcal{D}$ with comparable running time such that*

$$\mathbf{Adv}^{\mathcal{A},\mathsf{LTF}}(\lambda) \leq n \cdot \mathbf{Adv}^{\mathcal{D},\mathsf{LWE}}_{\ell,2m,q,\chi}(\lambda).$$

*Moreover, there is a $\mathsf{ppt}$ sampling algorithm, that given $(\mathbf{B}, \mathbf{C}, \mathbf{F})$ generated by $\mathsf{LTF.LGen}(1^\lambda)$, a trapdoor basis $(\boldsymbol{b}_i)_{i \leq n}$ for $\Lambda^\perp(\mathbf{C}^\top)$ such that $\max_i \|\boldsymbol{b}_i\| \leq \sigma_x \sigma_e / (\Omega(\log n) \cdot \sqrt{2mn\beta^2\sigma_x^2 + \sigma_e^2})$ and a function output $\boldsymbol{y} = \mathsf{LTF.Eval}(ek, (\mathbf{x}, \mathbf{e}))$ for an input $(\mathbf{x}, \mathbf{e}) \hookleftarrow D^{\mathsf{Dom}_x^E}_{\mathbb{Z}^n,\sigma_x} \times D^{\mathsf{Dom}_e^E}_{\mathbb{Z}^{2m},\sigma_e}$, outputs, with probability $\geq 1 - 2^{-\Omega(\lambda)}$ over $ek$ and $(\mathbf{x}, \mathbf{e})$, an independent sample $(\bar{\mathbf{x}}, \bar{\mathbf{e}})$ from the conditional distribution of $(\mathbf{x}, \mathbf{e})$ conditioned on $\boldsymbol{y} = \mathsf{LTF.Eval}(ek, (\mathbf{x}, \mathbf{e}))$.*

*Proof.* First, the construction is correct. Indeed, by Lemmas 4 and 5, if $\sigma_x \geq \Omega(\sqrt{m})$ and $\sigma_e \geq \Omega(\sqrt{m})$, the distribution $D_{\mathbb{Z}^n,\sigma_x} \times D_{\mathbb{Z}^{2m},\sigma_e}$ is efficiently samplable, and a sample from it belongs to $\mathsf{Dom}^E_\lambda$ with probability $\geq 1 - 2^{-\Omega(\lambda)}$, so $D_{\mathsf{Dom}^E_\lambda}$ is efficiently samplable. For inversion correctness, we consider $(\mathbf{x}, \mathbf{e}) \in \mathsf{Dom}^D_\lambda$, and set $\mathbf{y} = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}$. By Lemma 9, we can recover $(\mathbf{x}, \mathbf{e})$ from $\mathbf{y}$ using the $\mathbf{G}$-trapdoor $\mathbf{R}$ of $\mathbf{A}$ if $\|\mathbf{e}\| \leq q/(10 \cdot \|\mathbf{R}\|)$. The fact that $\|\mathbf{R}\| \leq m$ and the parameter choices guarantee this.

The lossy and injective modes are computationally indistinguishable under the $\mathsf{LWE}_{\ell,2m,q,\chi}$ assumption. A standard hybrid argument over the columns of $\mathbf{A} \in \mathbb{Z}_q^{2m \times n}$ provides the inequality between the respective success advantages.

We now focus on the lossiness property. Note that Lemma 11 describes the conditional distribution of $(\mathbf{x}, \mathbf{e})$ conditioned on $(\mathbf{C} \cdot \mathbf{x}, \mathbf{F} \cdot \mathbf{x} + \mathbf{e})$. We claim that, except with probability $\leq 2^{-\Omega(\lambda)}$ over $ek$ generated by $\mathsf{LTF.LGen}(1^\lambda)$, this is also the distribution of $(\mathbf{x}, \mathbf{e})$ conditioned on $\mathsf{LTF.Eval}(ek, (\mathbf{x}, \mathbf{e}))$. Indeed, $\mathsf{LTF.Eval}(ek, (\mathbf{x}, \mathbf{e})) = \mathbf{B} \cdot \mathbf{C} \cdot \mathbf{x} + \mathbf{F} \cdot \mathbf{x} + \mathbf{e} \in \mathbb{Z}_q^{2m}$ uniquely determines $\mathbf{u} = \mathbf{C} \cdot \mathbf{x} \in \mathbb{Z}_q^\ell$ and $\mathbf{f} = \mathbf{F} \cdot \mathbf{x} + \mathbf{e} \in \mathsf{Dom}_e$ if $\|\mathbf{f}\|_\infty < \lambda_1^\infty(\Lambda(\mathbf{B}))/2$ for all $(\mathbf{x}, \mathbf{e}) \in \mathsf{Dom}^E$. The latter condition is satisfied except with probability $\leq 2^{-\Omega(\lambda)}$ over the choice of $ek$. This is because $\|\mathbf{f}\|_\infty \leq \sqrt{2m} \cdot \beta\sqrt{n}\sigma_x + \sqrt{2m}\sigma_x \leq 2\sqrt{2m} \cdot \sigma_e < q/8$ except with probability $2^{-\Omega(\lambda)}$ over the choice of $\mathbf{F}$, and $\lambda_1^\infty(\Lambda(\mathbf{B}))/2 \geq q/4$ with probability $\leq 2^{-\Omega(\lambda)}$ over the choice of $\mathbf{B}$, by Lemma 3.

We now show that the conditional distribution $D^{S_{\mathbf{F},\mathbf{u},\mathbf{f}}}_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\boldsymbol{\Sigma}},\mathbf{c}}$ given by Lemma 11 for $\mathbf{x}$ conditioned on $\mathsf{LTF.Eval}(ek, (\mathbf{x}, \mathbf{e}))$ has min-entropy at least $l$ and is efficiently samplable. For every $\bar{\mathbf{x}} \in S_{\mathbf{F},\mathbf{u},\mathbf{f}}$, we have

$$D^{S_{\mathbf{F},\mathbf{u},\mathbf{f}}}_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\boldsymbol{\Sigma}},\mathbf{c}}(\bar{\mathbf{x}}) = \frac{1}{p_a} \cdot D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\boldsymbol{\Sigma}},\mathbf{c}}(\bar{\mathbf{x}}), \qquad p_a = D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\boldsymbol{\Sigma}},\mathbf{c}}(S_{\mathbf{F},\mathbf{u},\mathbf{f}}).$$

For min-entropy, we observe that, by Lemma 6, the point with highest probability in $D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\boldsymbol{\Sigma}},\mathbf{c}}$ has probability $\leq 2 \det(\Lambda^\perp(\mathbf{C}^\top)/\sqrt{\det(\boldsymbol{\Sigma})}$ . We can apply Lemma 6 because $\sigma_n(\sqrt{\boldsymbol{\Sigma}}) \geq \eta_{2^{-n}}(\Lambda^\perp(\mathbf{C}^\top))$ with overwhelming probability. Indeed, thanks to assumption on $\chi$,

we have $\|\mathbf{F}^\top \cdot \mathbf{F}\| \leq 2mn\beta^2$ with probability $\geq 1 - 2^{-\Omega(\lambda)}$. When this inequality holds, we have

$$\sigma_n(\sqrt{\mathbf{\Sigma}}) \geq \sigma_x\sigma_e/\sqrt{2mn\beta^2\sigma_x^2 + \sigma_e^2}.$$

Further, by Lemma 3, we have $\eta_{2^{-n}}(\Lambda^\perp(\mathbf{C}^\top)) \leq O(\sqrt{n}q^{\ell/n})$ with probability $\geq 1 - 2^{-\Omega(\ell)}$. Hence the assumption of Lemma 6 holds, thanks to our parameter choices. Overall, we obtain that the scheme is $l$-lossy for

$$l \geq \log\sqrt{\det(\mathbf{\Sigma})} - \log\det(\Lambda^\perp(\mathbf{C}^\top)) - 1 - \log(1/p_a).$$

By calculations similar to those above, we have that $\sqrt{\det\mathbf{\Sigma}} \leq \sigma_x^n$. Further, matrix $\mathbf{C}$ has rank $\ell$ with probability $\geq 1 - 2^{-\Omega(\ell)}$, and, when this is the case, we have $\det(\Lambda^\perp(\mathbf{C}^\top)) = q^\ell$. We obtain $l \geq n\log\sigma_x - 1 - \ell\log q - \log(1/p_a)$.

To complete the lossiness proof, we show that $p_a \geq 1 - 2^{-\Omega(\lambda)}$ so that $\log(1/p_a) \leq 1$, except with probability $\leq 2^{-\Omega(\lambda)}$ over $(\mathbf{F}, \mathbf{C}, \mathbf{x}, \mathbf{e})$. For this, we have by a union bound that $p_a \geq 1 - (p_x + p_e)$, where $p_x$ is the probability that a sample $\bar{\mathbf{x}}$ from $D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\mathbf{\Sigma}},\mathbf{c}}$ lands outside $\mathsf{Dom}_x^E$ (i.e., $\|\bar{\mathbf{x}}\| > \gamma_x \cdot \sqrt{n} \cdot \sigma_x$), and $p_e$ is the probability that a sample $\bar{\mathbf{x}}$ from $D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\mathbf{\Sigma}},\mathbf{c}}$ is such that $\mathbf{f} - \mathbf{F}\cdot\bar{\mathbf{x}}$ lands outside $\mathsf{Dom}_e^E$ (i.e., $\|\mathbf{f} - \mathbf{F}\cdot\bar{\mathbf{x}}\| > \gamma_e \cdot \sqrt{2m} \cdot \sigma_e$).

In order to bound $p_x$, we observe that it is at most

$$p'_x = \Pr_{\bar{\mathbf{x}}\leftarrow D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\mathbf{\Sigma}},\mathbf{c}}}[\|\bar{\mathbf{x}} - \mathbf{c}\| > \|\sqrt{\mathbf{\Sigma}}\| \cdot \sqrt{n}]$$

if $\gamma_x \cdot \sqrt{n} \cdot \sigma_x \geq \|\mathbf{c}\| + \|\sqrt{\mathbf{\Sigma}}\| \cdot \sqrt{n}$. Now, using that $\|\mathbf{F}\| \leq \sqrt{2mn} \cdot \beta, \|\mathbf{x}\| \leq \sqrt{n} \cdot \sigma_x$ and $\|\mathbf{e}\| \leq \sqrt{2m} \cdot \sigma_e$ except with probability $2^{-\Omega(\lambda)}$, by Lemma 5, we get with the same probability that $\|\mathbf{c}\| \leq (\sigma_x/\sigma_e)^2 \cdot \sqrt{2mn}\beta \cdot (\sqrt{2mn} \cdot \beta \cdot \sigma_x \cdot \sqrt{n} + \sigma_e \cdot \sqrt{2m})$. Furthermore, using $\|\sqrt{\mathbf{\Sigma}}\| \leq \sigma_x/\sigma_e$, we have that the condition $\gamma_x \cdot \sqrt{n} \cdot \sigma_x \geq \|\mathbf{c}\| + \|\sqrt{\mathbf{\Sigma}}\| \cdot \sqrt{n}$ is satisfied by our choice of parameters. Also, as shown above, we have $\sigma_n(\sqrt{\mathbf{\Sigma}}) \geq \eta_{2^{-n}}(\Lambda^\perp(\mathbf{C}^\top))$ with overwhelming probability, so that we can apply Lemma 5 to conclude that $p_x \leq p'_x \leq 2^{-n+2}$ with probability $\geq 1 - 2^{-\Omega(\lambda)}$.

In order to bound $p_e$, we follow a similar computation as for $p_x$. Namely, we first observe that, if $\bar{\mathbf{x}}$ is sampled from $D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\mathbf{\Sigma}},\mathbf{c}}$, then $\bar{\mathbf{e}} = \mathbf{f} - \mathbf{F}\cdot\bar{\mathbf{x}}$ is distributed as $D_{\mathbf{F}\cdot\Lambda^\perp(\mathbf{C}^\top)+\mathbf{f}-\mathbf{F}\cdot\mathbf{x}',\sqrt{\mathbf{F}\mathbf{\Sigma}\mathbf{F}^\top},\mathbf{f}-\mathbf{F}\cdot\mathbf{c}}$. Therefore, the probability $p_e$ is at most the probability $p'_e$ that a sample $\bar{\mathbf{e}}$ from $D_{\mathbf{F}\cdot\Lambda^\perp(\mathbf{C}^\top)+\mathbf{f}-\mathbf{F}\cdot\mathbf{x}',\sqrt{\mathbf{F}\mathbf{\Sigma}\mathbf{F}^\top},\mathbf{f}-\mathbf{F}\cdot\mathbf{c}}$ satisfies $\|\bar{\mathbf{e}} - (\mathbf{f} - \mathbf{F}\cdot\mathbf{c})\| > \|\sqrt{\mathbf{F}\mathbf{\Sigma}\mathbf{F}^\top}\| \cdot \sqrt{2m}$, assuming that the condition

$$\gamma_e \cdot \sqrt{2m} \cdot \sigma_e \geq \|\mathbf{f} - \mathbf{F}\cdot\mathbf{c}\| + \|\sqrt{\mathbf{F}\mathbf{\Sigma}\mathbf{F}^\top}\| \cdot \sqrt{2m}, \tag{1}$$

is satisfied. Now, using $\|\mathbf{f} - \mathbf{F}\cdot\mathbf{c}\| \leq \|\mathbf{f}\| + \|\mathbf{F}\| \cdot \|\mathbf{c}\|$ and the above bounds on $\|\mathbf{F}\|$, $\|\mathbf{f}\|$ and $\|\mathbf{c}\|$ and our choice of parameters, we have that condition (1) is satisfied with overwhelming probability. In order to apply Lemma 5 and bound $p'_e$, we also need the inequality $\sigma_n(\sqrt{\mathbf{F}\mathbf{\Sigma}\mathbf{F}^\top}) \geq \eta_{2^{-n}}(\mathbf{F}\cdot\Lambda^\perp(\mathbf{C}^\top))$ to be satisfied. Now, note that

$$\sigma_n(\sqrt{\mathbf{F}\mathbf{\Sigma}\mathbf{F}^\top}) = \sigma_x \cdot \sigma_e/\sqrt{\sigma_x^2 + \sigma_e^2/\sigma_n(\mathbf{F})^2}.$$

By Lemma 7, we have $\sigma_n(\mathbf{F}) \geq \Omega(\sqrt{m} \cdot \beta)$ with overwhelming probability. We conclude that $\sigma_n(\sqrt{\mathbf{F}\boldsymbol{\Sigma}\mathbf{F}^\top}) \geq \Omega(\sigma_x \cdot \sqrt{m} \cdot \beta)$. On the other hand, Lemma 7 also implies that

$$\eta_{2^{-n}}(\mathbf{F} \cdot \Lambda^\perp(\mathbf{C}^\top)) \leq \|\mathbf{F}\| \cdot \eta_{2^{-n}}(\Lambda^\perp(\mathbf{C}^\top)) = O(\|\mathbf{F}\| \cdot \sqrt{n}) \leq O(\beta \cdot \sqrt{m} \cdot n)$$

with overwhelming probability. Hence, the condition $\sigma_n(\sqrt{\mathbf{F}\boldsymbol{\Sigma}\mathbf{F}^\top}) \geq \eta_{2^{-n}}(\mathbf{F} \cdot \Lambda^\perp(\mathbf{C}^\top))$ holds with with the same probability thanks to our choice of parameters. We can thus apply Lemma 5 to conclude that $p_e \leq p'_e \leq 2^{-n+2}$ with overwhelming probability.

Overall, we have that $p_a \geq 1 - (p_x + p_e) \geq 1 - 2^{-\Omega(\lambda)}$ which completes the proof of lossiness. This also immediately implies that the conditional distribution $D^{S_{\mathbf{F},\mathbf{u},\mathbf{f}}}_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\boldsymbol{\Sigma}},\mathbf{c}}$ is efficiently samplable by rejection sampling, given an efficient sampler for $D_{\Lambda^\perp(\mathbf{C}^\top)+\mathbf{x}',\sqrt{\boldsymbol{\Sigma}},\mathbf{c}}$. The latter sampler can be implemented with a $\mathsf{ppt}$ algorithm by Lemma 4 and the fact that $\max_i \|\boldsymbol{b}_i\| < \sigma_n(\boldsymbol{\Sigma})$ with overwhelming probability by the bound on $\sigma_n(\sqrt{\boldsymbol{\Sigma}})$. □

### 3.2 An All-But-Many Lossy Trapdoor Function from LWE

Parameters and domains are defined as in Section 3.1.

**Key generation.** $\mathsf{ABM.Gen}(1^\lambda)$ conducts the following steps.

1. For parameters $n, \ell, m, \gamma, \chi$, generate $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ as $\bar{\mathbf{A}} = \mathbf{B} \cdot \mathbf{C} + \mathbf{F}$ with $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{m \times \ell})$, $\mathbf{C} \hookleftarrow U(\mathbb{Z}_q^{\ell \times n})$ and $\mathbf{F} \hookleftarrow \chi^{m \times n}$.
2. Choose a PRF family $\mathsf{PRF} : \{0,1\}^\lambda \times \{0,1\}^k \to \{0,1\}^\lambda$ with input length $k = k(\lambda)$ and key length $\lambda$. Choose a seed $K \hookleftarrow U(\{0,1\}^\lambda)$ for $\mathsf{PRF}$.
3. Sample matrices $\mathbf{R}_1, \ldots, \mathbf{R}_\lambda \hookleftarrow U(\{-1,1\}^{m \times m})$ and compute

$$\mathbf{B}_i = \mathbf{R}_i \cdot \bar{\mathbf{A}} + K[i] \cdot \mathbf{G} \ \in \mathbb{Z}_q^{m \times n} \qquad \forall i \leq \lambda.$$

4. Output the evaluation key $ek$, the inversion key $ik$ and the lossy tag generation key $tk$, which consist of

$$ek := \Big(\bar{\mathbf{A}}, (\mathbf{B}_i)_{i \leq \lambda}\Big), \qquad ik := \big((\mathbf{R}_i)_{i \leq \lambda}, K\big), \qquad tk := K. \tag{2}$$

A tag $t = (t_\mathsf{c}, t_\mathsf{a}) \in \{0,1\}^\lambda \times \{0,1\}^k$ will be injective whenever $t_\mathsf{c} \neq \mathsf{PRF}(K, t_\mathsf{a})$.

**Lossy tag generation.** $\mathsf{ABM.LTag}(tk, t_\mathsf{a})$ takes as input an auxiliary tag component $t_\mathsf{a} \in \{0,1\}^k$ and uses $tk = K$ to compute and output $t_\mathsf{c} = \mathsf{PRF}(K, t_\mathsf{a})$.

**Evaluation.** $\mathsf{ABM.Eval}(ek, t, (\mathbf{x}, \mathbf{e}))$ takes in the function input $(\mathbf{x}, \mathbf{e}) \in \mathsf{Dom}^E_\lambda$, the tag $t = (t_\mathsf{c}, t_\mathsf{a}) \in \{0,1\}^\lambda \times \{0,1\}^k$ and proceeds as follows.

1. For each $j \leq \lambda$, let $C_{\mathsf{PRF},j}(t_\mathsf{a}) : \{0,1\}^\lambda \to \{0,1\}$ be the NAND Boolean circuit, where $t_\mathsf{a} \in \{0,1\}^k$ is hard-wired, which evaluates the $j$-th bit of $\mathsf{PRF}(\widetilde{K}, t_\mathsf{a}) \in \{0,1\}^\lambda$ for any $\widetilde{K} \in \{0,1\}^\lambda$. Run the public evaluation algorithm of Lemma 10 to obtain[6] $\mathbf{B}_{\mathsf{PRF},j} \leftarrow \mathsf{Eval}^{\mathsf{pub}}(C_{\mathsf{PRF},j}(t_\mathsf{a}), (\mathbf{B}_i)_{i \leq \lambda})$.

[6] One may use either $\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{CCT}}$ or $\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{BP}}$, but the choice must be consistent with the $\mathsf{Eval}^{\mathsf{priv}}$ variant used in function inversion.

2. Define the matrix

$$\mathbf{A}_t = \left[\frac{\bar{\mathbf{A}}}{\sum_{j\leq\lambda}\left((-1)^{t_\mathsf{c}[j]}\cdot\mathbf{B}_{\mathsf{PRF},j} + t_\mathsf{c}[j]\cdot\mathbf{G}\right)}\right] \in \mathbb{Z}_q^{2m\times n},$$

and compute the output $\mathbf{y} = \mathbf{A}_t\cdot\mathbf{x}+\mathbf{e} \ \in \mathbb{Z}_q^{2m}$.

**Inversion.** $\mathsf{ABM.Invert}(ik, t, \mathbf{y})$ inputs the inversion key $ik := \big((\mathbf{R}_i)_{i\leq\lambda}, K\big)$, the tag $t = (t_\mathsf{c}, t_\mathsf{a}) \in \{0,1\}^\lambda \times \{0,1\}^k$ and $\mathbf{y} \in \mathsf{Rng}_\lambda$, and proceeds as follows.

1. Return $\perp$ if $t_\mathsf{c} = \mathsf{PRF}(K, t_\mathsf{a})$.
2. Otherwise, for each $j \leq \lambda$, run the private evaluation algorithm from Lemma 10 to obtain $\mathbf{R}_{\mathsf{PRF},j} \leftarrow \mathsf{Eval}^{\mathsf{priv}}(C_{\mathsf{PRF},j}(t_\mathsf{a}), (\mathbf{R}_i)_{i\leq\lambda})$ and compute the (small-norm) matrix $\mathbf{R}_t = \sum_{j\leq\lambda}(-1)^{t_\mathsf{c}[j]}\cdot\mathbf{R}_{\mathsf{PRF},j} \ \in \mathbb{Z}^{m\times m}$.
3. Let $h_t$ denote the Hamming distance between $t_\mathsf{c}$ and $\mathsf{PRF}(K, t_\mathsf{a})$. Use the $\mathbf{G}$-trapdoor $\mathbf{R}_t$ of $\mathbf{A}_t$ with tag $h_t$ to find the unique $(\mathbf{x}, \mathbf{e}) \in \mathsf{Dom}_\lambda^D$ such that $\mathbf{y} = \mathbf{A}_t\cdot\mathbf{x}+\mathbf{e}$. This is done by applying the $\mathsf{LWE}$ inversion algorithm of Lemma 9.

All algorithms involved run in polynomial-time, if one uses $\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{CCT}}$ and $\mathsf{Eval}^{\mathsf{priv}}_{\mathsf{CCT}}$ from Lemma 10. If the circuits $C_{\mathsf{PRF},j}(t_\mathsf{a})$ (having the PRF key as input, and the PRF input hardwired) have logarithmic depth $d \leq O(\log\lambda)$, then it is preferable to use $\mathsf{Eval}^{\mathsf{pub}}_{\mathsf{BP}}$ and $\mathsf{Eval}^{\mathsf{priv}}_{\mathsf{BP}}$ instead. Indeed, under this small-depth assumption, these algorithms still run in polynomial-time, and have the advantage of leading to smaller $\mathbf{R}_t$'s. This eventually allows one to set $q$ as a polynomial function of $\lambda$. In the rest of this section, we choose these variants of $\mathsf{Eval}^{\mathsf{pub}}$ and $\mathsf{Eval}^{\mathsf{priv}}$. The results can be readily adapted to the other option.

**Theorem 2.** *Let $\chi = D_{\mathbb{Z},\beta/(2\sqrt{\lambda})}$ for some $\beta > 0$. Assume that $\mathsf{PRF}$ has depth $d = O(\log\lambda)$ when the circuit input is the key and the PRF input is hard-coded in the circuit. Assume that $\ell \geq \lambda$, $n = \Omega(\ell\log q)$ and $m \geq 2n\log q$, $\gamma_x \geq 3\sqrt{m/n}$ and $\gamma_e \geq 3$. Assume also that $\sigma_x \geq \Omega(n)$, $\sigma_e \geq \Omega(4^d\cdot m^2\cdot\beta\cdot\sqrt{n}\cdot\sigma_x)$ and $\sigma_e \leq O(q/(\lambda\cdot 4^d\cdot m^2))$. Then, under the PRF security and $\mathsf{LWE}_{\ell,2m,q,\chi}$ hardness assumptions, the above function is an $l$-lossy ABM LTF with $l = \Omega(n\log n)$.*

The theorem follows from the lemmas below.

**Lemma 12 (Correctness).** *Let us assume that and $q/\sigma_e \geq \lambda\cdot 4^d\cdot O(m^2)$. Assume that $\mathsf{PRF}$ has logarithmic depth $O(\log\lambda)$ when the circuit input is the key and the PRF input is hard-coded in the circuit. Then, for any triple $(ek, ik, tk)$ produced by $\mathsf{ABM.Gen}(1^\lambda)$, for any tag $t = (t_\mathsf{c}, t_\mathsf{a}) \in \{0,1\}^\lambda \times \{0,1\}^k$ satisfying $t_\mathsf{c} \neq \mathsf{PRF}(K, t_\mathsf{a})$ and for any input $(\mathbf{x}, \mathbf{e}) \in \mathsf{Dom}_\lambda^D$, the inversion correctness condition $(\mathbf{x}, \mathbf{e}) = \mathsf{ABM.Invert}(ik, t, \mathsf{ABM.Eval}(ek, t, (\mathbf{x}, \mathbf{e})))$ is satisfied.*

*Proof.* By Lemma 10, we have $\|\mathbf{R}_t\| \leq \lambda\cdot 4^d\cdot O(m^{3/2})$ and

$$\mathbf{A}_t = \left[\frac{\bar{\mathbf{A}}}{\mathbf{R}_t\cdot\mathbf{A} + h_t\cdot\mathbf{G}}\right] \bmod q,$$

where $h_t$ is the Hamming distance between $t_\mathsf{c}$ and $\mathsf{PRF}(K, t_\mathsf{a}) \in \{0,1\}^\lambda$. As $q > \lambda$ is prime, integer $h_t$ is invertible modulo $q$, and $\mathbf{R}_t$ is a $\mathbf{G}$-trapdoor with tag $h_t$ for $\mathbf{A}_t$. Thanks to our parameters, we have $\|\mathbf{e}\| \leq q/(10\cdot\|\mathbf{R}_t\|)$ and hence algorithm $\mathsf{Invert}$ from Lemma 9 recovers $(\mathbf{x}, \mathbf{e})$. □

Our ABM-LTF provides evasiveness unless the PRF family is not unpredictable, which would contradict its pseudorandomness. In order to meaningfully rely on the pseudorandomness of $\mathsf{PRF}$, the proof of Lemma 13 also appeals to the $\mathsf{LWE}$ assumption so as to first move to a game where the lossy matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ is traded for a random matrix. Since the matrices $\mathbf{B}_i = \mathbf{R}_i \cdot \bar{\mathbf{A}} + K[i] \cdot \mathbf{G}$ depend the bits of the seed $K$, moving to a uniform matrix $\bar{\mathbf{A}}$ is necessary to make sure that the evaluation key $ek$ is statistically independent of $K$.

**Lemma 13 (Evasiveness).** *Assume that $m \geq 2n \log q$. Any* ppt *evasiveness adversary $\mathcal{A}$ making $Q_1$ and $Q_2$ queries to* $\mathsf{ABM.LTag}$ *and* $\mathsf{ABM.IsLossy}$*, respectively, implies an* $\mathsf{LWE}$ *distinguisher $\mathcal{D}_1$ and a PRF distinguisher $\mathcal{D}_2$ such that*

$$\mathbf{Adv}^{\mathcal{A},\mathsf{eva}}_{Q_1,Q_2}(\lambda) \leq n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{LWE}}_{\ell,m,q,\chi}(\lambda) + \mathbf{Adv}^{\mathcal{D}_2,\mathsf{PRF}}_{Q_1+Q_2}(\lambda) + \frac{Q_2+1}{2^\lambda}.$$

*Proof.* Let us assume the existence of a ppt evasiveness adversary $\mathcal{A}$ with non-negligible advantage $\mathbf{Adv}^{\mathcal{A},\mathsf{eva}}_{Q_1,Q_2}(\lambda)$. Without loss of generality, we can consider the adversary $\mathcal{A}$ as successful as soon as it manages to query the $\mathsf{ABM.IsLossy}(tk, \cdot)$ oracle with input a non-trivial lossy tag $t^\star = (t^\star_{\mathsf{c}}, t^\star_{\mathsf{a}})$, i.e., a tag such that $t^\star_{\mathsf{a}}$ has never been submitted to the $\mathsf{ABM.LTag}(tk, \cdot)$ oracle. Note that, for any input $(t'_{\mathsf{c}}, t_{\mathsf{a}}) \neq (t_{\mathsf{c}}, t_{\mathsf{a}})$ such that $t_{\mathsf{c}} \leftarrow \mathsf{ABM.LTag}(tk, t_{\mathsf{a}})$, the oracle $\mathsf{ABM.IsLossy}(tk, \cdot)$ returns 0 since $t_{\mathsf{c}}$ is a deterministic function of $t_{\mathsf{a}}$.

We show that $\mathcal{A}$ implies either an $\mathsf{LWE}_{\ell,m,q,\chi}$ distinguisher $\mathcal{D}_1$ or a PRF distinguisher $\mathcal{D}_2$ with noticeable advantage. To this end, we consider a sequence of games that begins with the real evasiveness game and ends with a game where the adversary's advantage is statistically negligible. For each $i$, we call $W_i$ the event that the adversary wins, in which case the challenger outputs 1.

**Game** 0: This is the real evasiveness game where the adversary $\mathcal{A}$ is fed with a real evaluation $ek$ and interacts with the real $\mathsf{ABM.LTag}(tk, \cdot)$ oracle. We call $W_0$ the event that, at some point of the game, the adversary $\mathcal{A}$ queries $\mathsf{ABM.IsLossy}(tk, \cdot)$ for a non-trivial input $t^\star = (t^\star_{\mathsf{c}}, t^\star_{\mathsf{a}})$ such that $t^\star_{\mathsf{c}} = \mathsf{PRF}(K, t^\star_{\mathsf{a}})$, where $tk = K$, and $t^\star_{\mathsf{a}}$ has never been queried to $\mathsf{ABM.LTag}(tk, \cdot)$. By assumption, we know that $\mathcal{A}$ wins with probability $\Pr[W_0] = \mathbf{Adv}^{\mathcal{A},\mathsf{eva}}_{Q_1,Q_2}(\lambda)$.

**Game** 1: This game is identical Game 0, except that we modify the distribution of the evaluation key $ek$. Namely, the matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ is now sampled uniformly. Under the $\mathsf{LWE}_{\ell,m,q,\chi}$ assumption, this change does not significantly affect $\mathcal{A}$'s winning probability. Using the hybrid argument, we obtain that $|\Pr[W_1] - \Pr[W_0]| \leq n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{LWE}}_{\ell,m,q,\chi}(\lambda)$, for some efficient algorithm $\mathcal{D}_1$.

**Game** 2: This game is identical to Game 1 with the difference that the matrices $(\mathbf{B}_i)_{i \leq \lambda}$ are now sampled uniformly in $\mathbb{Z}_q^{m \times n}$. Lemma 2 implies that the distribution of $ek$ remains statistically unchanged: we have $|\Pr[W_2] - \Pr[W_1]| \leq 2^{-\lambda}$. We remark that the PRF seed $tk = K$ is now perfectly independent of $ek$.

**Game** 3: This game is like Game 2 but we modify the $\mathsf{ABM.LTag}(tk, \cdot)$ and $\mathsf{ABM.IsLossy}(tk, \cdot)$ oracles. Instead of returning $t_{\mathsf{c}} = \mathsf{PRF}(K, t_{\mathsf{a}})$ at each query of the form $\mathsf{ABM.LTag}(tk, t_{\mathsf{a}})$, the $\mathsf{ABM.LTag}(tk, \cdot)$ oracle outputs $R(t_{\mathsf{a}}) \in \{0,1\}^\lambda$, where $R : \{0,1\}^k \to \{0,1\}^\lambda$ is a uniformly

random function, which the challenger lazily defines by uniformly sampling a $\lambda$-bit string at each new query $t_{\mathsf{a}} \in \{0,1\}^k$. At each query to $\mathsf{ABM.IsLossy}(tk, \cdot)$, adversary $\mathcal{A}$ chooses a tag $t$ and we call $W_3$ the event that one of these tags $t^\star = (t^\star_{\mathsf{c}}, t^\star_{\mathsf{a}})$ satisfies $t^\star_{\mathsf{c}} = R(t^\star_{\mathsf{a}})$ although $t^\star_{\mathsf{a}}$ has never been queried to $\mathsf{ABM.LTag}(tk, \cdot)$. Given that $R(\cdot)$ is a truly random function, we have $\Pr[W_3] = Q_2/2^\lambda$, where $Q_2$ is the number of queries to $\mathsf{ABM.IsLossy}(tk, \cdot)$. Moreover, as explained below, there exists a PRF distinguisher $\mathcal{D}_2$ that makes at most $Q_1 + Q_2$ PRF evaluation queries and such that $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathcal{D}_2,\mathsf{PRF}}_{Q_1+Q_2}(\lambda)$.

Algorithm $\mathcal{D}_2$ interacts with a PRF challenger that chooses a uniform key $K^\star \hookleftarrow U(\{0,1\}^\lambda)$ and either always outputs $\mathsf{PRF}(K^\star, M)$ at each query $M$ or always outputs uniform values in the range $\{0,1\}^\lambda$ of $\mathsf{PRF}(K^\star, \cdot)$.

To generate the evaluation key for $\mathcal{A}$, $\mathcal{D}_2$ chooses random matrices $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\mathbf{B}_i \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ for each $i \leq \lambda$ and runs $\mathcal{A}$ on input of $ek = \big(\bar{\mathbf{A}}, (\mathbf{B}_i)_{i\leq\lambda}\big)$. Note that the distribution of $ek$ is identical to that of Game 2. Whenever $\mathcal{A}$ invokes $\mathsf{ABM.LTag}(tk, \cdot)$ on input of an auxiliary tag part $t_{\mathsf{a}} \in \{0,1\}^k$, distinguisher $\mathcal{D}_2$ submits $t_{\mathsf{a}}$ to its own PRF challenger and relays the response back to $\mathcal{A}$. At each query $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ made by $\mathcal{A}$ to $\mathsf{ABM.IsLossy}(tk, \cdot)$, distinguisher $\mathcal{D}_2$ submits the corresponding $t_{\mathsf{a}} \in \{0,1\}^k$ to its PRF challenger. When obtaining the response $v \in \{0,1\}^\lambda$, it checks if $t_{\mathsf{c}} = v$. If so, distinguisher $\mathcal{D}_2$ halts, and outputs 1 to declare the adversary successful. If $\mathcal{A}$ terminates without having queried $\mathsf{ABM.IsLossy}(tk, \cdot)$ on such a lossy tag, distinguisher $\mathcal{D}_2$ outputs 0.

Let us assume that $\mathcal{D}_2$'s challenger always outputs pseudo-random values $\mathsf{PRF}(K^\star, t_{\mathsf{a}})$ when invoked on the input $t_{\mathsf{a}}$. In this case, $\mathcal{A}$'s view is the same as in Game 2, so that $\mathcal{D}_2$ outputs 1 with probability $\Pr[W_2]$. If $\mathcal{D}_2$'s challenger always returns evaluations of a perfectly random function, distinguisher $\mathcal{D}_2$ outputs 1 with probability $\Pr[W_3]$ as $\mathcal{A}$'s view corresponds to Game 3. It comes that $\mathcal{D}_2$'s advantage as a PRF distinguisher is at least $|\Pr[W_3] - \Pr[W_2]|$, as claimed.

Putting the above altogether completes the proof of the lemma. □

The pseudo-randomness of core tag components also guarantees that lossy tags are computationally indistinguishable from uniformly random tags. The proof of Lemma 14 also relies on the $\mathsf{LWE}$ assumption since the evaluation key $ek$ only hides the PRF seed $K$ in the computational sense. It follows the same strategy as the proof of Lemma 13.

**Lemma 14 (Indistinguishability).** *Assume that $m > 2n \log q$. Then* ppt *indistinguishability adversary $\mathcal{A}$ implies either either an* $\mathsf{LWE}$ *distinguisher $\mathcal{D}_1$ or a PRF distinguisher $\mathcal{D}_2$ such that:*

$$\mathbf{Adv}^{\mathcal{A},\mathsf{ind}}_{Q}(\lambda) \leq 2n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{LWE}}_{\ell,m,q,\chi}(\lambda) + \mathbf{Adv}^{\mathcal{D}_2,\mathsf{PRF}}_{Q}(\lambda) + \frac{1}{2^{\lambda-1}},$$

*where $Q$ denotes the number of (genuine or uniform) lossy tag generation queries.*

*Proof.* Let $\mathcal{A}$ be an adversary that has non-negligible advantage in distinguishing outputs of the lossy tag generation oracle $\mathsf{ABM.LTag}(tk, \cdot)$ from uniform elements in $\mathcal{T}_{\mathsf{c}} = \{0,1\}^\lambda$. We show that $\mathcal{A}$ implies either an $\mathsf{LWE}_{\ell,m,q,\chi}$ distinguisher $\mathcal{D}_1$ or a PRF distinguisher $\mathcal{D}_2$. We prove this claim using a sequence of games that begins with a game where $\mathcal{A}$ interacts with

an oracle $\mathsf{ABM.LTag}(tk, \cdot)$ that always outputs the unique $t_\mathsf{c}$ such that $t = (t_\mathsf{c}, t_\mathsf{a}) \in \mathcal{T}_{\mathsf{loss}}$ at each query $t_\mathsf{a}$. In the final game, adversary $\mathcal{A}$ interacts with an oracle $\mathcal{O}_{\mathcal{T}_c}(\cdot)$ that outputs uniform strings in $\{0,1\}^\lambda$. For each $i$, we call $W_i$ the event that the adversary outputs 1.

**Game** 0: In this game, adversary $\mathcal{A}$ is given a real evaluation key $ek$ and interacts with an oracle $\mathsf{ABM.LTag}(tk, \cdot)$ that outputs $t_\mathsf{c} = \mathsf{PRF}(K, t_\mathsf{a})$, where $tk = K$, at each query $t_\mathsf{a} \in \{0,1\}^k$. When $\mathcal{A}$ halts, it outputs a bit $b \in \{0,1\}$ and we call $W_0$ the event that $b = 1$.

**Game** 1: This game is like Game 0 except that we modify the distribution of the evaluation key $ek$. Namely, the matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ is sampled uniformly. As in the proof of Lemma 13, we have $|\Pr[W_1] - \Pr[W_0]| \leq n \cdot \mathbf{Adv}^{\mathcal{D}_1, \mathsf{LWE}}_{\ell,m,q,\chi}(\lambda)$, for some efficient algorithm $\mathcal{D}_1$

**Game** 2: This game is like Game 1 except that the matrices $(\mathbf{B}_i)_{i \leq \lambda}$ are now sampled uniformly in $\mathbb{Z}_q^{m \times n}$. As in the proof of Lemma 13, we have $|\Pr[W_2] - \Pr[W_1]| \leq 2^{-\lambda}$.

**Game** 3: This game is identical to Game 2 but we replace the $\mathsf{ABM.LTag}(tk, \cdot)$ oracle by the oracle $\mathcal{O}_{\mathcal{T}_c}(\cdot)$ that outputs random $\lambda$-bit strings at each new query and consistently returns the same outputs if a given query $t_\mathsf{a}$ occurs more than once. Instead of returning $t_\mathsf{c} = \mathsf{PRF}(K, t_\mathsf{a})$ at each query $t_\mathsf{a}$, the oracle thus outputs $R(t_\mathsf{a}) \in \{0,1\}^\lambda$, where $R : \{0,1\}^k \to \{0,1\}^\lambda$ is a uniformly random function, which is lazily defined by sampling random strings in $\{0,1\}^\lambda$ at each new query $t_\mathsf{a} \in \{0,1\}^k$. When $\mathcal{A}$ halts, it outputs a bit $b \in \{0,1\}$ and we call $W_3$ the event that $b = 1$.

It can be seen that $\Pr[W_3]$ is close to $\Pr[W_2]$ if $\mathsf{PRF}$ is a pseudo-random function family, as there is a distinguisher $\mathcal{D}_2$ such that $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathcal{D}_2, \mathsf{PRF}}_Q(\lambda)$. In short, distinguisher $\mathcal{D}_2$ interacts with a PRF challenger that chooses a key $K^\star \hookleftarrow U(\{0,1\}^\lambda)$ and either always outputs $\mathsf{PRF}(K^\star, M)$ at each query $M$ or always outputs random strings $\{0,1\}^\lambda$ at each fresh query. To generate the evaluation key for $\mathcal{A}$, distinguisher $\mathcal{D}_2$ samples $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{B}_i \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ for each $i \leq \lambda$ and feeds $\mathcal{A}$ with $ek = \big(\bar{\mathbf{A}}, (\mathbf{B}_i)_{i \leq \lambda}\big)$. Whenever $\mathcal{A}$ queries $\mathsf{ABM.LTag}(tk, \cdot)$ on input of an auxiliary tag $t_\mathsf{a} \in \{0,1\}^k$, distinguisher $\mathcal{D}_2$ queries $t_\mathsf{a}$ to its PRF challenger and relays the answer to $\mathcal{A}$. When $\mathcal{A}$ terminates, distinguisher $\mathcal{D}_2$ outputs whatever $\mathcal{A}$ outputs. If $\mathcal{D}_2$'s challenger always outputs pseudo-random values $\mathsf{PRF}(K^\star, t_\mathsf{a})$ on input of $t_\mathsf{a}$, then $\mathcal{A}$'s view is clearly the same as in Game 2. If $\mathcal{D}_2$'s challenger always outputs uniformly random strings in $\{0,1\}^\lambda$, then $\mathcal{D}_2$ is providing $\mathcal{A}$ with the same view as in Game 3. We thus have $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathcal{D}_2, \mathsf{PRF}}_Q(\lambda)$, as claimed.

**Game** 4: This game is like Game 3, except that we change again the generation of matrices $(\mathbf{B}_i)_{i \leq \lambda}$ in the evaluation key $ek$. For each $i \leq \lambda$, we step back to computing $\mathbf{B}_i = \mathbf{R}_i \cdot \bar{\mathbf{A}} + K[i] \cdot \mathbf{G} \in \mathbb{Z}_q^{m \times n}$, with $\mathbf{R}_i \hookleftarrow U(\{-1,1\}^{m \times m})$ and $K[i] \hookleftarrow U(\{0,1\})$. By the same argument as in Game 2, the view of $\mathcal{A}$ remains statistically unchanged and we have $|\Pr[W_4] - \Pr[W_3]| \leq 2^{-\lambda}$.

**Game** 5: We restore the matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ back to its original distribution. Under the $\mathsf{LWE}_{\ell,m,q,\chi}$ assumption, we have $|\Pr[W_5] - \Pr[W_4]| \leq n \cdot \mathbf{Adv}^{\mathcal{D}_1, \mathsf{LWE}}_{\ell,m,q,\chi}(\lambda)$, for some efficient algorithm $\mathcal{D}_1$.

Combining the above leads to the claimed bound on $\mathbf{Adv}^{\mathcal{A}, \mathsf{ind}}_Q(\lambda)$. □

The proof of lossiness is essentially identical to that of the LTF (Theorem 1).

**Lemma 15 (Lossiness).** *Let $\chi = D_{\mathbb{Z},\beta/(2\sqrt{\lambda})}$ for some $\beta > 0$. Assume that the depth $d$ of* $\mathsf{PRF}$ *is in $O(\log \lambda)$, when the circuit input is the key and the PRF input is hardwired in the circuit. Let us assume that $\ell \geq \lambda$ and $n = \Omega(\ell \log q)$. Assume also that $\sigma_e \geq \Omega(4^d \cdot m^2 \cdot \beta \cdot \sigma_x \cdot \sqrt{n})$. Then, for any lossy tag $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$, the above ABM-LTF is $l$-lossy with $l = \Omega(n \log n)$.*

*Proof.* We rely on the fact that, for any lossy tag $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ (i.e., for which $t_{\mathsf{c}} = \mathsf{PRF}(K, t_{\mathsf{a}})$), we have

$$\mathbf{A}_t = \left[\frac{\bar{\mathbf{A}}}{\mathbf{R}_t \cdot \bar{\mathbf{A}}}\right] = \left[\frac{\mathbf{B}}{\mathbf{R}_t \cdot \mathbf{B}}\right] \cdot \mathbf{C} + \left[\frac{\mathbf{F}}{\mathbf{R}_t \cdot \mathbf{F}}\right], \tag{3}$$

where $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{m \times \ell})$, $\mathbf{C} \hookleftarrow U(\mathbb{Z}_q^{\ell \times n})$, $\mathbf{F} \hookleftarrow \chi^{m \times n}$ and $\mathbf{R}_t$ is as in the $\mathsf{ABM.Invert}$ description.

As a consequence, by the same argument as in the proof of Theorem 1, the distribution of the input $(\mathbf{x}, \mathbf{e})$ conditioned on $\mathsf{ABM.Eval}(ek, t, (\mathbf{x}, \mathbf{e}))$ is the same as the distribution of $(\mathbf{x}, \mathbf{e})$ conditioned on $(\mathbf{C} \cdot \mathbf{x}, \mathbf{F} \cdot \mathbf{x} + \mathbf{e})$. From this point, the proof is identical to that of Theorem 1, with $\mathbf{F}_{new} = [\mathbf{F}^\top \mid (\mathbf{R}_t \cdot \mathbf{F})^\top]^\top$ playing the role of $\mathbf{F}$ in the original proof. The two properties of $\mathbf{F}_{new}$ used in the proof are $\|\mathbf{F}_{new}\| \leq (1 + \|\mathbf{R}_t\|) \cdot \|\mathbf{F}\| \leq O(4^d \cdot m^{3/2}) \cdot \|\mathbf{F}\|$, using Lemma 10, which leads to a larger $\sigma_e$ by the factor $O(4^d \cdot m^{3/2})$. The other property is a lower bound on $\sigma_n(\mathbf{F}_{new})$ and since the latter is $\geq \sigma_n(\mathbf{F})$, no parameters are affected. □

In [4, Section 7], Alwen *et al.* used the a rounding technique [6] to build an all-but-one trapdoor function. While our construction bears resemblance with theirs, our proof of lossiness is very different. In [4, Theorem 7.3], they consider a matrix of the form (3) and crucially rely on the statistical independence of the rows of $[\mathbf{B}^\top \mid (\mathbf{R}_0 \cdot \mathbf{B})^\top]^\top$, for some $\mathbf{R}_0 \in \{-1, 1\}^{m \times m}$, conditionally on $\mathbf{R}_0 \cdot \mathbf{F}$. Here, we cannot guarantee that matrices $\mathbf{R}_t \cdot \mathbf{B}$ be statistically independent for different tags $t$, and hence it does not seem possible to directly use the rounding technique from [4]. Fortunately, the proof of Lemma 15 does not require the rows of the matrix $[\mathbf{B}^\top \mid (\mathbf{R}_t \cdot \mathbf{B})^\top]^\top$ to be statistically independent and neither does it rely on the independence of $\mathbf{R}_t \cdot \mathbf{B}$ for different tags $t$.

### 3.3 Joint Use of Lossy and All-But-Many Functions

We remark that our LTF and ABM-LTF are not lossy enough to be correlation-secure in the sense of Rosen and Segev [80]: indeed, the result of [80, Theorem 3.3] requires lossy functions that lose at least half of their input. In particular, we cannot reveal $\mathbf{y}_0 = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}$ and $\mathbf{y} = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$ for the same input $(\mathbf{x}, \mathbf{e})$ as this would expose $\mathbf{y} - \mathbf{y}_0 = (\mathbf{A} - \mathbf{A}_t) \cdot \mathbf{x}$, which would leak $(\mathbf{x}, \mathbf{e})$. However, we can safely reveal $\mathbf{y}_0 = \mathsf{LTF.Eval}(ek', (\mathbf{x}, \mathbf{e}_0)) = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}_0$ and $\mathbf{y} = \mathsf{ABM.Eval}(ek, t, (\mathbf{x}, \mathbf{e})) = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$ for distinct Gaussian terms $\mathbf{e}_0, \mathbf{e} \in \mathbb{Z}^{2m}$.

Indeed, conditionally on $\mathsf{LTF.Eval}(ek', (\mathbf{x}, \mathbf{e}_0))$ and $\mathsf{ABM.Eval}(ek, t, (\mathbf{x}, \mathbf{e}))$, the distribution of $\mathbf{x}$ retains $l$ bits of min-entropy, where $l = \Omega(n \cdot \log n)$. As in the proof of Theorem 1, this follows by observing that the residual distribution on $\mathbf{x}$ is a discrete Gaussian (by Lemma 15) whose covariance matrix is above the smoothing parameter of the support.

**Lemma 16.** *The LTF of Section 3.1 and the above ABM-LTF are jointly lossy when they share the first part* $\mathbf{x}$ *of their inputs.*

*Let* $\chi = D_{\mathbb{Z},\beta/(2\sqrt{\lambda})}$ *for some* $\beta > 0$*. Assume that the depth* $d$ *of* $\mathsf{PRF}$ *is in* $O(\log \lambda)$*, when the circuit input is the key and the PRF input is hardwired in the circuit. Let us assume that* $\ell \geq \lambda$ *and* $n = \Omega(\ell \log q)$*. Assume also that* $\sigma_e \geq \Omega(4^d \cdot m^2 \cdot \beta \cdot \sqrt{n} \cdot \sigma_x)$*. Then, except with probability* $\leq 2^{-\Omega(\lambda)}$ *over the choice of* $ek' \hookleftarrow \mathsf{LTF.LGen}(1^\lambda)$*,* $ek \hookleftarrow \mathsf{ABM.Gen}(1^\lambda)$*,* $\mathbf{x} \hookleftarrow \mathsf{Dom}_x$*, and* $\mathbf{e}_0, \mathbf{e} \hookleftarrow \mathsf{Dom}_e$*, we have, for any lossy tag* $t$*:*

$$H_\infty\big(\mathbf{x} \mid \mathsf{LTF.Eval}(ek', (\mathbf{x}, \mathbf{e}_0)),\ \mathsf{ABM.Eval}(ek, t, (\mathbf{x}, \mathbf{e}))\big) \geq n \cdot \log \sigma_x - 2 - \ell \log q > \Omega(n \cdot \log n).$$

*Proof.* The result follows by generalizing the proofs of Theorem 1 and Lemma 15 in a straightforward manner. If $\mathbf{A}_{\mathsf{LTF}} = \mathbf{B}_{\mathsf{LTF}} \cdot \mathbf{C}_{\mathsf{LTF}} + \mathbf{F}_{\mathsf{LTF}} \in \mathbb{Z}_q^{2m \times n}$ and $\bar{\mathbf{A}} = \mathbf{B}_{\mathsf{ABM}} \cdot \mathbf{C}_{\mathsf{ABM}} + \mathbf{F}_{\mathsf{ABM}} \in \mathbb{Z}_q^{m \times n}$ are the lossy matrices of both functions, the information revealed by $\mathsf{LTF.Eval}(ek', (\mathbf{x}, \mathbf{e}_0))$ and $\mathsf{ABM.Eval}(ek, t, (\mathbf{x}, \mathbf{e}))$ is

$$\left[\begin{array}{c|c} \mathbf{B}_{\mathsf{LTF}} & \mathbf{0}^{2m \times \ell} \\ \hline \mathbf{0}^{m \times \ell} & \mathbf{B}_{\mathsf{ABM}} \\ \mathbf{0}^{m \times \ell} & \mathbf{R}_t \cdot \mathbf{B}_{\mathsf{ABM}} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{C}_{\mathsf{LTF}} \\ \hline \mathbf{C}_{\mathsf{ABM}} \end{array}\right] \cdot \mathbf{x} + \left[\begin{array}{c} \mathbf{F}_{\mathsf{LTF}} \\ \hline \mathbf{F}_{\mathsf{ABM}} \\ \mathbf{R}_t \cdot \mathbf{F}_{\mathsf{ABM}} \end{array}\right] \cdot \mathbf{x} + \left[\begin{array}{c} \mathbf{e}_0 \\ \hline \mathbf{e} \end{array}\right].$$

It is thus entirely determined by the product $[\mathbf{C}_{\mathsf{LTF}}^\top \mid \mathbf{C}_{\mathsf{ABM}}^\top]^\top \cdot \mathbf{x} \in \mathbb{Z}_q^{2\ell}$ and the integer vector $[\mathbf{F}_{\mathsf{LTF}}^\top \mid \mathbf{F}_{\mathsf{ABM}}^\top \mid (\mathbf{R}_t \cdot \mathbf{F}_{\mathsf{ABM}})^\top]^\top \cdot \mathbf{x} + [\mathbf{e}_0^\top \mid \mathbf{e}_1^\top]^\top \in \mathbb{Z}^{4m}$. We obtain the result by repeating the arguments in the proof of Theorem 1 and Lemma 15. □

## 4 Selective Opening Chosen-Ciphertext Security

We now combine our ABM-LTF and the $\mathsf{LWE}$-based LTF of Section 3 to build an IND-SO-CCA2-secure public-key encryption scheme from the $\mathsf{LWE}$ assumption. The scheme can be seen as instantiating a variant of the Peikert-Waters methodology [76], as generalized by Hofheinz [57, Section 6.3] to the case of multiple lossy tags. In [57], ciphertexts consists of $(f_{\mathsf{lossy}}(x), f_{\mathsf{ABM}}(t, x), \mathsf{Msg} \oplus h(x))$, where $f_{\mathsf{lossy}}(x)$ (resp. $f_{\mathsf{ABM}}(t, x)$) is a lossy (resp. all-but-many) function of $x$; $t$ is the tag of the ciphertext; and $h(x)$ is a universal hash of $x$.

Nevertheless, our scheme is *not* a generic instantiation of this paradigm as we cannot use exactly the same input $x$ in the two functions $f_{\mathsf{lossy}}(\cdot)$ and $f_{\mathsf{ABM}}(t, \cdot)$. As we mentioned earlier, we cannot give out function outputs $\mathbf{y}_0 = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}$ and $\mathbf{y} = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$ for the same input $(\mathbf{x}, \mathbf{e})$. For this reason, our lossy and ABM functions have to use distinct noise terms $(\mathbf{e}_0, \mathbf{e})$ in the two evaluations $\mathbf{y}_0 = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}_0$ and $\mathbf{y} = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e}$. The decryption algorithm can proceed by inverting $(\mathbf{x}, \mathbf{e}_0) \leftarrow f_{\mathsf{lossy}}^{-1}(\mathbf{y}_0)$ as before. However, instead of simply testing if $\mathbf{y} = f_{\mathsf{ABM}}(t, (\mathbf{x}, \mathbf{e}_0))$ by evaluating $f_{\mathsf{ABM}}(t, .)$ in the forward direction as in [76, 57], the receiver has to test whether $\mathbf{y} - \mathbf{A}_t \cdot \mathbf{x}$ is a small-norm vector, analogously to [75, Section 4.4]. For this reason, the message $\mathsf{Msg}$ is hidden by the universal hash of $\mathbf{x}$ only, which is sufficient in our security proof. Moreover, our extension to SIM-SO-CCA2 security requires $h(\cdot)$ to operate on $\mathbf{x}$ alone.

Unlike [76], we cannot use one-time signatures to bind ciphertext components in a non-malleable manner. Indeed, at each corruption query, the challenger would have to reveal the one-time secret keys of the challenge ciphertexts, which would allow the adversary to make decryption queries for lossy tags.

Instead, we rely on hybrid encryption and proceed analogously to Boyen *et al.* [28]: namely, we define the auxiliary tags to be the output $\mathbf{y}_0 = \mathit{\Pi}^{\mathsf{LTF}}.\mathsf{Eval}(ek', (\mathbf{x}, \mathbf{e}_0))$ of the lossy function while resorting to the hybrid encryption paradigm and authenticate the message-carrying part $\mathbf{c}_0 = \mathsf{Msg} + h(\mathbf{x})$ of the ciphertext via the encrypt-then-MAC approach. One difficulty is that, since $\mathbf{y}_0 = \mathit{\Pi}^{\mathsf{LTF}}.\mathsf{Eval}(ek', (\mathbf{x}, \mathbf{e}_0))$ and $\mathbf{y} = \mathit{\Pi}^{\mathsf{ABM}}.\mathsf{Eval}(ek, t, (\mathbf{x}, \mathbf{e}))$ involve distinct small-norm vectors $\mathbf{e}_0$, $\mathbf{e}$, we must find a different way to prevent the adversary from tampering with $\mathbf{e}$ in one of the challenge ciphertexts (indeed, the ABM-LTF output $\mathbf{y}$ is no longer authenticated by a one-time signature). Our solution to this problem is to include $\mathbf{y} = \mathit{\Pi}^{\mathsf{ABM}}.\mathsf{Eval}(ek, t, (\mathbf{x}, \mathbf{e}))$ in the input of the MAC, which simultaneously authenticates $\mathbf{y}$ and $\mathbf{c}_0$. For simplicity, we assume MACs with the uniqueness property but the proof can be adapted to rely on any strongly unforgeable MAC.

As mentioned in [57, Section 6], the application to IND-SO-CCA2 security requires the core tag space $\mathcal{T}_{\mathsf{c}}$ of ABM-LTFs to be efficiently samplable and explainable. As defined in [57, Definition 6.2], "explainability" (a.k.a. "invertible samplability" [38]) means that any core tag $t_{\mathsf{c}}$ can be explained by the challenger as having been uniformly chosen "without ulterior motive" when the adversary opens a given ciphertext in the game of Definition 10. Our ABM-LTF clearly satisfies this property since core tags $t_{\mathsf{c}}$ are just random $\lambda$-bit strings.

### 4.1 Description

**Par-Gen**$(1^\lambda)$**:** Selects public parameters consisting of:

- A modulus $q > 2$, integers $\ell, \ell_0, \ell_1, n \in \mathsf{poly}(\lambda)$, $m = \lceil cn \cdot \log q \rceil$, for some constant $c > 0$, and parameters $\beta, \sigma_x, \sigma_e > 0$.
- The specification $\mathsf{MAC} = (\mathsf{KG}, \mathsf{Sig}, \mathsf{Ver})$ of a unique message authentication code with message space $\mathsf{MsgSp}^{mac} := \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{\ell_0}$ and key space $\mathcal{K}^{mac} := \mathbb{Z}_q^{\ell_1}$.
- A family $\mathcal{UH}$ of universal hash functions $h : [-\sigma_x\sqrt{n}, \sigma_x\sqrt{n}]^n \to \mathbb{Z}_q^{\ell_0+\ell_1}$ that range over $\mathsf{MsgSp} := \mathbb{Z}_q^{\ell_0}$.

The public parameters $\mathit{\Gamma} = \{\ell, \ell_0, \ell_1, n, m, q, \beta, \sigma_x, \sigma_e, \mathsf{MAC}\}$ define the plaintext space $\mathsf{MsgSp} := \mathbb{Z}_q^{\ell_0}$ and will be shared by the LWE-based LTF of Section 3.1 and our ABM-LTF of Section 3.2.

**Keygen**$(\mathit{\Gamma})$**:** Let $\mathit{\Pi}^{\mathsf{LTF}} = (\mathsf{IGen}, \mathsf{LGen}, \mathsf{Eval}, \mathsf{Invert})$ be an instance of the LTF of Section 3.1 and let $\mathit{\Pi}^{\mathsf{ABM}} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert}, \mathsf{LTag})$ be an instance of the ABM-LTF of Section 3.2. We assume $\mathit{\Pi}^{\mathsf{LTF}}$ and $\mathit{\Pi}^{\mathsf{ABM}}$ both operate over the domain

$$\mathsf{Dom}_\lambda^D := \{(\mathbf{x}, \mathbf{e}) \in \mathbb{Z}^n \times \mathbb{Z}^{2m} \mid \|\mathbf{x}\| \leq \sigma_x\sqrt{n},\ \|\mathbf{e}\| \leq \sigma_e\sqrt{2m}\}.$$

The public key is generated via the following steps.

1. Generate a pair $(ek', ik') \leftarrow \mathit{\Pi}^{\mathsf{LTF}}.\mathsf{IGen}(1^\lambda)$ for an injective function of the lossy trapdoor function family $\mathit{\Pi}^{\mathsf{LTF}}$.

2. Generate $(ek, ik, tk) \leftarrow \Pi^{\mathsf{ABM}}.\mathsf{Gen}(1^\lambda)$ as an ABM-LTF key pair. We assume that the space of auxiliary tags is $\mathcal{T}_\mathsf{a} = \mathbb{Z}_q^m$
3. Choose a random member $h \leftarrow \mathcal{UH}$ of the universal hash family.

Output $(PK, SK)$ where $PK = \big(ek', ek, h\big)$ and $SK = ik'$.

**Encrypt**$(PK, \mathsf{Msg})$**:** To encrypt $\mathsf{Msg} \in \mathbb{Z}_q^{\ell_0}$, choose $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^n,\sigma_x}$, $\mathbf{e}_0 \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_e}$, $\mathbf{e} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{2m},\sigma_e}$ and do the following.

1. Compute $\mathbf{y}_0 = \Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek', (\mathbf{x}, \mathbf{e}_0)) = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}_0 \ \in \mathbb{Z}_q^{2m}$.
2. Define $t_\mathsf{a} = \mathbf{y}_0$ and choose a random $t_\mathsf{c} \hookleftarrow U(\mathcal{T}_\mathsf{c})$. Then, let $t = (t_\mathsf{c}, t_\mathsf{a})$ and compute $\mathbf{y} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}(ek, t, (\mathbf{x}, \mathbf{e})) = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e} \ \in \mathbb{Z}_q^{2m}$.
3. Compute $(\mathbf{k}^{sym}, \mathbf{k}^{mac}) = h(\mathbf{x}) \in \mathbb{Z}_q^{\ell_0} \times \mathbb{Z}_q^{\ell_1}$.
4. Set $\mathbf{c}_0 = \mathsf{Msg} + \mathbf{k}^{sym} \ \in \mathbb{Z}_q^{\ell_0}$ and $\mathbf{c}_1 = \mathsf{MAC.Sig}(\mathbf{k}^{mac}, (\mathbf{y}, \mathbf{c}_0))$.

Output the ciphertext $\mathbf{C} = (t_\mathsf{c}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0, \mathbf{y})$.

**Decrypt**$(SK, C)$**:** To decrypt $\mathbf{C} = (t_\mathsf{c}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0, \mathbf{y})$ using $SK = ik'$,

1. Compute $(\mathbf{x}, \mathbf{e}_0) \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{Invert}(ik', \mathbf{y}_0)$. Return $\perp$ if the vector $\mathbf{y}_0 \in \mathbb{Z}_q^{2m}$ is not in the range[7] of $\Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek', \cdot)$ or if $(\mathbf{x}, \mathbf{e}_0) \notin \mathsf{Dom}_\lambda^D$.
2. Define the tag $t = (t_\mathsf{c}, \mathbf{y}_0)$. If $\|\mathbf{y} - \mathbf{A}_t \cdot \mathbf{x}\| > \sigma_e\sqrt{2m}$, return $\perp$.
3. Compute $(\mathbf{k}^{sym}, \mathbf{k}^{mac}) = h(\mathbf{x}) \ \in \mathbb{Z}_q^{\ell_0} \times \mathbb{Z}_q^{\ell_1}$.
4. If $\mathsf{MAC.Ver}(\mathbf{k}^{mac}, (\mathbf{y}, \mathbf{c}_0), \mathbf{c}_1) = 0$, return $\perp$. Otherwise, compute and output the plaintext $\mathsf{Msg} = \mathbf{c}_0 - \mathbf{k}^{sym} \in \mathbb{Z}_q^{\ell_0}$.

We note that, while the encryption algorithm requires to homomorphically compute $\lambda$ circuits, these evaluations can take place in an off-line phase, where $\mathbf{y}_0$ and $\mathbf{y}$ are pre-computed before knowing $\mathsf{Msg}$.

In order to instantiate the scheme with a polynomial-size modulus $q$, we need a PRF with an evaluation circuit in $\mathsf{NC}^1$, which translates into a polynomial-length branching program. By applying Lemma 10 and exploiting the asymmetric noise growth of the GSW FHE as in [31], we can indeed keep $q$ small.

For this purpose, the Banerjee-Peikert PRF [5] is a suitable candidate. While its evaluation circuit is in $\mathsf{NC}^2$ in general, we can still homomorphically evaluate input-dependent circuits $C_{\mathsf{PRF},j}(\cdot)$ over the encrypted key $K$ using an $\mathsf{NC}^1$ circuit. For public moduli $p, q$ and matrices $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n \times n\lceil \log q \rceil}$, their PRF maps an input $x \in \{0,1\}^k$ to $\lfloor (p/q) \cdot (\mathbf{k}^\top \cdot \mathbf{A}_x \bmod q) \rfloor$, where $\mathbf{k} \in \mathbb{Z}_q^n$ is the secret key and the input-dependent matrix $\mathbf{A}_x$ is publicly computable from $\mathbf{A}_0, \mathbf{A}_1$. This allows hard-coding $\mathbf{A}_x$ into an $\mathsf{NC}^1$ circuit to be evaluated over the "encrypted" bits of $\mathbf{k}$ in order to obtain "encryptions" of the bits of $\lfloor (p/q) \cdot \mathbf{k}^\top \cdot \mathbf{A}_x \rfloor$. Indeed, matrix-vector products and rounding can both be computed in $\mathsf{TC}^0 \subseteq \mathsf{NC}^1$, which allows using a polynomial-size $q$ by applying Lemma 10. The resulting instantiation relies on the same $\mathsf{LWE}$ assumption as the Banerjee-Peikert PRF [5], where the modulus-to-noise ratio is only slightly super-polynomial.

[7] Note that $\mathbf{y}_0$ may be far from the image of $\mathbf{A}$ in an invalid ciphertext but the inversion algorithm can detect this using $ik'$.

### 4.2 Indistinguishability-Based (IND-SO-CCA2) Security

We first prove that the scheme provides IND-SO-CCA2 security. While we can tightly relate the IND-SO-CCA security of the scheme to the pseudorandomness of the underlying PRF, the reduction from the unforgeability of the MAC loses a factor proportional to the number of challenges.

**Theorem 3.** *The scheme provides IND-SO-CCA2 security assuming that: (i)* $\Pi^{\mathsf{LTF}}$ *is a LTF; (ii)* $\Pi^{\mathsf{ABM}}$ *is an ABM-LTF; (iii)* $\mathsf{PRF}$ *is a pseudorandom function family; (iv)* $\mathsf{MAC}$ *provides sUF-OT-CMA security. In our instantiation, for any adversary* $\mathcal{A}$*, there exists an* $\mathsf{LWE}_{\ell,m,q,\chi}$ *distinguisher* $\mathcal{D}_1$*, a PRF adversary* $\mathcal{D}_2$ *and a MAC forger* $\mathcal{B}$ *with comparable running time and such that*

$$\mathbf{Adv}^{\text{IND-SO-CCA2}}_{\mathcal{A}}(\lambda) \leq 4n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{lwe}}_{\ell,m,q,\chi}(\lambda) + 2 \cdot \mathbf{Adv}^{\mathcal{D}_2,\text{prf}}_{N+Q_D}(\lambda) + \frac{Q_D + 2 + N \cdot (Q_D + 1)}{2^{\lambda-2}} + N \cdot \mathbf{Adv}^{\text{mac},Q_D}_{\mathcal{B}}(\lambda),$$

*where* $N$ *is the number of challenge ciphertexts and* $Q_D$ *is the number of decryption queries made by the adversary.* (The proof is given in Appendix D.1.)

In Appendix C, we describe a variant of the scheme which, while not secure under selective openings, can be proved tightly CCA2-secure in the multi-challenge setting (cf. Definition 5 in Appendix B.1) as long as the PRF is itself tightly secure. In order to enable instantiations with a polynomial-size modulus $q$, we give a tighter security proof for the PRF of [25] in Appendix F.

### 4.3 Achieving Simulation-Based (SIM-SO-CCA2) Security

We show that our scheme can be instantiated so as to achieve the stronger notion of SIM-SO-CCA2 security. To this end, we show that it is in fact a lossy encryption scheme with weak efficient opening. We first detail the lossy key generation algorithm (which can be used in the final game in the proof of IND-SO-CCA2 security) and the $\mathsf{Opener}$ algorithm.

In order for $\mathsf{Opener}$ to run efficiently, we instantiate our scheme with a universal hash family $\mathcal{UH}$, where each function $h : [-\sigma_x\sqrt{n}, \sigma_x\sqrt{n}]^n \to \mathbb{Z}_q^{\ell_0+\ell_1}$ is keyed by a public matrix $\mathbf{H}_{\mathcal{UH}} \in \mathbb{Z}_q^{(\ell_0+\ell_1)\times n}$, which is included in the public key $PK_{\mathsf{loss}}$ and allows evaluating

$$h_{\mathbf{H}_{\mathcal{UH}}}(\mathbf{x}) = \begin{bmatrix} \mathbf{k}^{sym} \\ \mathbf{k}^{mac} \end{bmatrix} = \mathbf{H}_{\mathcal{UH}} \cdot \mathbf{x} \mod q$$

before computing $\mathbf{c}_0 = \mathsf{Msg} + \mathbf{k}^{sym} \in \mathbb{Z}_q^{\ell_0}$ and $\mathbf{c}_1 = \mathsf{MAC.Sig}(\mathbf{k}^{sym}, (\mathbf{y}, \mathbf{c}_0))$.

We also require $\mathsf{Par\text{-}Gen}$ to output public parameters $\ell, \ell_0, n$ satisfying the constraint $n > 2 \cdot (2\ell + \ell_0 + \ell_1) \cdot \log q$, where $\ell_0$ is the message length, $\ell_1$ is the key length of the MAC and $\ell$ is the dimension of the underlying $\mathsf{LWE}$ assumption.

**Keygen**$(\Gamma, \mathsf{loss})$**:** Given public parameters $\Gamma = \{\ell, \ell_0, \ell_1, n, m, q, \beta, \sigma_x, \sigma_e\}$ containing integers $\ell, \ell_0, n, m$ such that $n > 2 \cdot (2\ell + \ell_0 + \ell_1) \cdot \lceil \log q \rceil$ and $m > 2(n+\ell) \cdot \log q$, conduct the following steps.

1. Choose a random matrix $\mathbf{C}_0 \hookleftarrow U\big(\mathbb{Z}_q^{\bar{n} \times \bar{\ell}}\big)$, where $\bar{\ell} = (2\ell + \ell_0 + \ell_1)$ and $\bar{n} = n - \bar{\ell} \cdot \lceil \log q \rceil$ which is used to run the $(\mathbf{C}, \mathbf{R}_{sim}) \leftarrow \mathsf{GenTrap}(\mathbf{C}_0, \mathbf{I}_{\bar{\ell}}, \sigma_x)$ algorithm of Lemma 8 to produce a statistically uniform $\mathbf{C} \in \mathbb{Z}_q^{\bar{\ell} \times n}$ with a a small-norm $\mathbf{R}_{sim} \in \mathbb{Z}^{\bar{\ell} \cdot \lceil \log q \rceil \times \bar{n}}$ forming a $\mathbf{G}_{sim}$-trapdoor, where $\mathbf{G}_{sim} \in \mathbb{Z}_q^{\bar{\ell} \cdot \lceil \log q \rceil \times \bar{\ell}}$ is the gadget matrix of [69]. Parse $\mathbf{C} \in \mathbb{Z}_q^{\bar{\ell} \times n}$ as

$$\mathbf{C} = \left[\begin{array}{c} \mathbf{C}_{\mathsf{LTF}} \\ \hline \mathbf{C}_{\mathsf{ABM}} \\ \hline \mathbf{H}_{\mathcal{UH}} \end{array}\right] \in \mathbb{Z}_q^{\bar{\ell} \times n}, \tag{4}$$

   where $\mathbf{C}_{\mathsf{LTF}}, \mathbf{C}_{\mathsf{ABM}} \in \mathbb{Z}_q^{\ell \times n}$ and $\mathbf{H}_{\mathcal{UH}} \in \mathbb{Z}_q^{(\ell_0+\ell_1) \times n}$.
2. Sample matrices $\mathbf{B}_{\mathsf{LTF}} \hookleftarrow U(\mathbb{Z}_q^{2m \times \ell})$, $\mathbf{B}_{\mathsf{ABM}} \hookleftarrow U(\mathbb{Z}_q^{m \times \ell})$, $\mathbf{F}_{\mathsf{LTF}} \hookleftarrow \chi^{2m \times n}$, $\mathbf{F}_{\mathsf{ABM}} \hookleftarrow \chi^{m \times n}$ in order to define

$$\begin{aligned} \mathbf{A}_{\mathsf{LTF}} &= \mathbf{B}_{\mathsf{LTF}} \cdot \mathbf{C}_{\mathsf{LTF}} + \mathbf{F}_{\mathsf{LTF}} \ \in \mathbb{Z}_q^{2m \times n} \\ \mathbf{A}_{\mathsf{ABM}} &= \mathbf{B}_{\mathsf{ABM}} \cdot \mathbf{C}_{\mathsf{ABM}} + \mathbf{F}_{\mathsf{ABM}} \ \in \mathbb{Z}_q^{m \times n}, \end{aligned}$$

   which are statistically close to outputs of $\mathsf{Lossy}(1^n, 1^m, 1^\ell, q, \chi)$ as $\mathbf{C}_{\mathsf{LTF}}$ and $\mathbf{C}_{\mathsf{ABM}}$ are statistically uniform over $\mathbb{Z}_q^{\ell \times n}$.
3. Define $ek' = \mathbf{A}_{\mathsf{LTF}} \in \mathbb{Z}_q^{2m \times n}$ to be the evaluation key of $\Pi^{\mathsf{LTF}}$. Then, run Steps 2-4 of the key generation algorithm of $\Pi^{\mathsf{ABM}}$ while setting $\bar{\mathbf{A}} = \mathbf{A}_{\mathsf{ABM}} \in \mathbb{Z}_q^{m \times n}$ at Step 1. The resulting keys $(ek, ik, tk)$ consist of

$$ek := \Big(\mathbf{A}_{\mathsf{ABM}}, \{\mathbf{B}_i\}_{i=1}^{\lambda}\Big), \qquad ik := \big(\{\mathbf{R}_i\}_{i=1}^{\lambda}, K\big), \qquad tk := K$$

   and are statistically close to the output distribution (2) of $\Pi^{\mathsf{ABM}}.\mathsf{Gen}$.

Return $PK_{\mathsf{loss}} = (ek', ek, \mathbf{H}_{\mathcal{UH}})$ and

$$SK_{\mathsf{loss}} = \big(\mathbf{R}_{sim}, \mathbf{C}_0, \mathbf{B}_{\mathsf{LTF}}, \mathbf{B}_{\mathsf{ABM}}, \mathbf{F}_{\mathsf{LTF}}, \mathbf{F}_{\mathsf{ABM}}, ik\big). \tag{5}$$

**Opener**$(\Gamma, PK_{\mathsf{loss}}, SK_{\mathsf{loss}}, \mathsf{Msg}_0, (\mathbf{x}, \mathbf{e}_0, \mathbf{e}_1), \mathsf{Msg}_1)$**:** Parse $SK_{\mathsf{loss}}$ as in (5) and conduct the following steps.

1. Compute $\mathbf{t}_{\mathsf{LTF},\mathbf{x}} = \mathbf{C}_{\mathsf{LTF}} \cdot \mathbf{x} \in \mathbb{Z}_q^{\ell}$, $\mathbf{t}_{\mathsf{ABM},\mathbf{x}} = \mathbf{C}_{\mathsf{ABM}} \cdot \mathbf{x} \in \mathbb{Z}_q^{\ell}$ and

$$\begin{bmatrix} \mathbf{k}^{sym,\mathbf{x}} \\ \mathbf{k}^{mac,\mathbf{x}} \end{bmatrix} = \mathbf{H}_{\mathcal{UH}} \cdot \mathbf{x} \ \in \mathbb{Z}_q^{\ell_0+\ell_1}.$$

   Then, set $\mathbf{t}_{\mathsf{Msg},\mathbf{x}} = (\mathsf{Msg}_0 - \mathsf{Msg}_1) + \mathbf{k}^{sym,\mathbf{x}} \in \mathbb{Z}_q^{\ell_0}$ and define

$$\mathbf{t}_{\mathbf{x}} = \big[\ \mathbf{t}_{\mathsf{LTF},\mathbf{x}}^\top \mid \mathbf{t}_{\mathsf{ABM},\mathbf{x}}^\top \mid \mathbf{t}_{\mathsf{Msg},\mathbf{x}}^\top \mid \mathbf{k}^{mac,\mathbf{x}^\top}\ \big]^\top \in \mathbb{Z}_q^{\bar{\ell}}.$$

2. Using the trapdoor $\mathbf{R}_{sim} \in \mathbb{Z}^{\bar{\ell}\cdot\lceil \log q\rceil \times \bar{n}}$, sample a small-norm vector $\mathbf{x}' \hookleftarrow D^{S_{\underline{\mathbf{F}},\mathbf{t}_\mathbf{x},\mathbf{f}}}_{\Lambda^\perp(\mathbf{C})+\mathbf{z},\sqrt{\mathbf{\Sigma}},\mathbf{c}}$ so as to have a short integer vector $\mathbf{x}' \in \mathbb{Z}^n$ satisfying $\mathbf{C}\cdot\mathbf{x}' = \mathbf{t}_\mathbf{x} \bmod q$, using an arbitrary solution $\mathbf{z} \in \mathbb{Z}^n$ of $\mathbf{C}\cdot\mathbf{z} = \mathbf{t}_\mathbf{x} \in \mathbb{Z}_q^{\bar{\ell}}$, where $\mathbf{\Sigma}$ and $\mathbf{c}$ are defined based on Lemma 11, for

$$\underline{\mathbf{F}} := \left[\frac{\mathbf{F}_{\mathsf{LTF}}}{\frac{\mathbf{F}_{\mathsf{ABM}}}{\mathbf{R}_t\cdot\mathbf{F}_{\mathsf{ABM}}}}\right] \in \mathbb{Z}^{4m\times n}, \underline{\mathbf{e}} := \left[\frac{\mathbf{e}_0}{\mathbf{e}}\right] \in \mathbb{Z}^{4m},\ \mathbf{f} := \underline{\mathbf{F}}\cdot\mathbf{x} + \underline{\mathbf{e}} \in \mathbb{Z}^{4m}. \tag{6}$$

3. Output $(\mathbf{x}', \mathbf{e}'_0, \mathbf{e}')$ where

$$\begin{cases} \mathbf{e}'_0 = \mathbf{F}_{\mathsf{LTF}}\cdot(\mathbf{x}-\mathbf{x}') + \mathbf{e}_0 \ \in \mathbb{Z}^{2m} \\ \mathbf{e}' \ = \left[\frac{\mathbf{F}_{\mathsf{ABM}}}{\mathbf{R}_t\cdot\mathbf{F}_{\mathsf{ABM}}}\right]\cdot(\mathbf{x}-\mathbf{x}') + \mathbf{e} \ \in \mathbb{Z}^{2m} \end{cases} \tag{7}$$

We observe that algorithm $\mathsf{Opener}$ is efficient. In particular, at Step 2, it can compute the matrix $\mathbf{\Sigma}$ and the vector $\mathbf{c}$ of Lemma 11 by first reconstructing the matrix $\underline{\mathbf{F}} \in \mathbb{Z}^{4m\times n}$ of (6) and the vector $\mathbf{f} = \underline{\mathbf{F}}\cdot\mathbf{x} + \underline{\mathbf{e}} \in \mathbb{Z}^{4m}$, which requires to deterministically re-compute the integer matrix $\mathbf{R}_t$ obtained at Step 2 of $\mathsf{ABM.Invert}(ik, t, .)$ using $ik = ((\mathbf{R}_i)_{i\le\lambda}, K)$.

We easily check that, for any vector $\mathbf{x}'$ sampled at Step 2, the corresponding

$$\begin{bmatrix}\mathbf{k}^{sym,\mathbf{x}'} \\ \mathbf{k}^{mac,\mathbf{x}'}\end{bmatrix} = \mathbf{H}_{\mathcal{UH}}\cdot\mathbf{x}' \ \in \mathbb{Z}_q^{\ell_0+\ell_1}$$

satisfy $\mathbf{k}^{mac,\mathbf{x}'} = \mathbf{k}^{mac,\mathbf{x}_0}$ and $\mathbf{k}^{sym,\mathbf{x}'} = (\mathsf{Msg}_0 - \mathsf{Msg}_1) + \mathbf{k}^{sym,\mathbf{x}} \bmod q$.

As a consequence, if $C = (t_\mathsf{c}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0, \mathbf{y}) \leftarrow \mathsf{Encrypt}(PK_{\mathsf{loss}}, \mathsf{Msg}_0, (\mathbf{x}, \mathbf{e}_0, \mathbf{e}))$, the obtained ciphertext $C$ contains

$$\mathbf{c}_0 = \mathsf{Msg}_0 + \mathbf{k}^{sym,\mathbf{x}} \bmod q, \qquad \mathbf{c}_1 = \mathsf{MAC.Sig}(\mathbf{k}^{mac,\mathbf{x}}, (\mathbf{y}, \mathbf{c}_0)),$$

which coincide with $\mathbf{c}_0 = \mathsf{Msg}_1 + \mathbf{k}^{sym,\mathbf{x}'}$ and $\mathbf{c}_1 = \mathsf{MAC.Sig}(\mathbf{k}^{mac,\mathbf{x}'}, (\mathbf{y}, \mathbf{c}_0))$. Moreover, we also have $\mathbf{C}_{\mathsf{LTF}}\cdot\mathbf{x} = \mathbf{C}_{\mathsf{LTF}}\cdot\mathbf{x}'$ and $\mathbf{C}_{\mathsf{ABM}}\cdot\mathbf{x} = \mathbf{C}_{\mathsf{ABM}}\cdot\mathbf{x}'$.

The following theorem formally states the correctness of the $\mathsf{Opener}$ algorithm.

**Theorem 4.** *For any key pair $(PK_{\mathsf{loss}}, SK_{\mathsf{loss}})$ in the support of $\mathsf{Keygen}(\Gamma, \mathsf{loss})$, algorithm $\mathsf{Opener}$ outputs $(\mathbf{x}', \mathbf{e}'_0, \mathbf{e}')$ with the correct distribution conditionally on*

$$\mathsf{Encrypt}(PK_{\mathsf{loss},}, \mathsf{Msg}_0, (\mathbf{x}, \mathbf{e}_0, \mathbf{e})) = \mathsf{Encrypt}(PK_{\mathsf{loss},}, \mathsf{Msg}_1, (\mathbf{x}', \mathbf{e}'_0, \mathbf{e}')).$$

*Proof.* For any lossy tag $t = (t_\mathsf{c}, t_\mathsf{a})$, the matrix $\mathbf{A}_t$ used by $\Pi^{\mathsf{ABM}}.\mathsf{Eval}(ek, t, .)$ is of the form

$$\mathbf{A}_t = \left[\frac{\mathbf{A}_{\mathsf{ABM}}}{\mathbf{R}_t\cdot\mathbf{A}_{\mathsf{ABM}}}\right] = \left[\frac{\mathbf{B}_{\mathsf{ABM}}}{\mathbf{R}_t\cdot\mathbf{B}_{\mathsf{ABM}}}\right]\cdot\mathbf{C}_{\mathsf{ABM}} + \left[\frac{\mathbf{F}_{\mathsf{ABM}}}{\mathbf{R}_t\cdot\mathbf{F}_{\mathsf{ABM}}}\right], \tag{8}$$

where $\mathbf{R}_t \in \mathbb{Z}^{m\times m}$ is the integer matrix obtained in $\mathsf{ABM.Invert}(ik, t, .)$. At the same time, $ek'$ consists of a matrix of the form $\mathbf{A}_{\mathsf{LTF}} = \mathbf{B}_{\mathsf{LTF}}\cdot\mathbf{C}_{\mathsf{LTF}} + \mathbf{F}_{\mathsf{LTF}}$.

We now claim that, due to the way to sample $\mathbf{x}'$ at Step 2 of $\mathsf{Opener}$ and the definition of $\mathbf{e}'_0$ and $\mathbf{e}'$ at Step 3, the distribution of $\mathbf{y}'_0$ and $\mathbf{y}'$, with

$$\begin{cases} \mathbf{y}'_0 = \mathbf{A}_{\mathsf{LTF}} \cdot \mathbf{x}' + \mathbf{e}'_0 \ \in \mathbb{Z}^{2m} \\ \mathbf{y}' = \mathbf{A}_t \cdot \mathbf{x}' + \mathbf{e}' \ \in \mathbb{Z}^{2m} \end{cases} \tag{9}$$

is the same as that of the real encryptions explained in the beginning of this Section. By replacing $\mathbf{A}_{\mathsf{LTF}}, \mathbf{A}_t$ and $\mathbf{e}'_0$ and $\mathbf{e}'$ we get:

$$\begin{aligned} \mathbf{y}'_0 &= (\mathbf{B}_{\mathsf{LTF}} \cdot \mathbf{C}_{\mathsf{LTF}} + \mathbf{F}_{\mathsf{LTF}}) \cdot \mathbf{x}' + (\mathbf{F}_{\mathsf{LTF}} \cdot (\mathbf{x} - \mathbf{x}') + \mathbf{e}_0) \\ &= \mathbf{B}_{\mathsf{LTF}} \cdot \mathbf{C}_{\mathsf{LTF}} \cdot \mathbf{x}' + \mathbf{F}_{\mathsf{LTF}} \cdot \mathbf{x} + \mathbf{e}_0 \\ &= \mathbf{B}_{\mathsf{LTF}} \cdot \mathbf{C}_{\mathsf{LTF}} \cdot \mathbf{x} + \mathbf{F}_{\mathsf{LTF}} \cdot \mathbf{x} + \mathbf{e}_0 \\ &= \mathbf{A}_{\mathsf{LTF}} \cdot \mathbf{x} + \mathbf{e}_0 \ \in \mathbb{Z}^m \end{aligned}$$

and

$$\begin{aligned} \mathbf{y}' &= \left( \left[ \frac{\mathbf{B}_{\mathsf{ABM}}}{\mathbf{R}_t \cdot \mathbf{B}_{\mathsf{ABM}}} \right] \cdot \mathbf{C}_{\mathsf{ABM}} + \left[ \frac{\mathbf{F}_{\mathsf{ABM}}}{\mathbf{R}_t \cdot \mathbf{F}_{\mathsf{ABM}}} \right] \right) \cdot \mathbf{x}' \\ &\quad + \left( \left[ \frac{\mathbf{F}_{\mathsf{ABM}}}{\mathbf{R}_t \cdot \mathbf{F}_{\mathsf{ABM}}} \right] \cdot (\mathbf{x} - \mathbf{x}') + \mathbf{e} \right) \\ &= \left[ \frac{\mathbf{B}_{\mathsf{ABM}}}{\mathbf{R}_t \cdot \mathbf{B}_{\mathsf{ABM}}} \right] \cdot \mathbf{C}_{\mathsf{ABM}} \cdot \mathbf{x}' + \left[ \frac{\mathbf{F}_{\mathsf{ABM}}}{\mathbf{R}_t \cdot \mathbf{F}_{\mathsf{ABM}}} \right] \cdot \mathbf{x} + \mathbf{e} \\ &= \left[ \frac{\mathbf{B}_{\mathsf{ABM}}}{\mathbf{R}_t \cdot \mathbf{B}_{\mathsf{ABM}}} \right] \cdot \mathbf{C}_{\mathsf{ABM}} \cdot \mathbf{x} + \left[ \frac{\mathbf{F}_{\mathsf{ABM}}}{\mathbf{R}_t \cdot \mathbf{F}_{\mathsf{ABM}}} \right] \cdot \mathbf{x} + \mathbf{e} \\ &= \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e} \ \in \mathbb{Z}^{2m} \end{aligned} \tag{10}$$

It remains to show that $(\mathbf{x}', \mathbf{e}'_0, \mathbf{e}')$ have the correct distribution. Recall that $\mathbf{x}'$ is sampled by applying the second part of Theorem 1 at Step 2. By applying Lemma 11 to the matrix $\mathbf{C}$ of (4) with $\mathbf{u} = \mathbf{t}_{\mathbf{x}}$, the conditional distribution of $\mathbf{x}'$ given $(\mathbf{t}_{\mathbf{x}}, \underline{\mathbf{F}} \cdot \mathbf{x} + \underline{\mathbf{e}})$ is statistically close to $D^{S_{\underline{\mathbf{F}}, \mathbf{t}_{\mathbf{x}}, \mathbf{f}}}_{\Lambda^{\perp}(\mathbf{C}) + \mathbf{z}, \sqrt{\mathbf{\Sigma}}, \mathbf{c}}$, where $\mathbf{z}$ is an arbitrary solution of $\mathbf{C} \cdot \mathbf{z} = \mathbf{t}_{\mathbf{x}}$. Since $\mathbf{x}'$ can be efficiently sampled by Theorem 1, this provides the claimed result. □

In Appendix E, we show that lattice trapdoors can also be used to obtain SIM-SO-CPA security from LTFs based on $\mathsf{DDH}$-like assumptions.

## Acknowledgements

We thank Fabrice Benhamouda for useful discussions. Part of this research was funded by the French ANR ALAMBIC project (ANR-16-CE39-0006) and by BPI-France in the context of the national project RISQ (P141580). The third author was supported by ERC Starting Grant ERC-2013-StG-335086-LATTAC. The second and fourth authors were supported by Australian Research Council Discovery Grant DP150100285.

# References


1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt*, 2010.
2. S. Agrawal, C. Gentry, S. Halevi, and A. Sahai. Discrete gaussian leftover hash lemma over infinite domains. In *Asiacrypt*, LNCS, 2013.
3. J. Alwen, M. Hirt, U. Maurer, A. Patra, and P. Raykov. Key-indistinguishable message authentication codes. In *SCN*, 2014.
4. J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs. Learning with rounding, revisited: New reduction, properties and applications. In *Crypto*, 2013.
5. A. Banerjee and C. Peikert. New and improved key-homomorphic pseudo-random functions. In *Crypto*, 2014.
6. A. Banerjee, C. Peikert, and A. Rosen. Pseudo-random functions and lattices. In *Eurocrypt*, 2012.
7. D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. In *STOC*, 1986.
8. M. Bellare, S. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *Eurocrypt*, 2000.
9. M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *Asiacrypt*, 2009.
10. M. Bellare, R. Dowsley, B. Waters, and S. Yilek. Standard security does not imply security against selective-opening. In *Eurocrypt*, 2012.
11. M. Bellare and V.-T. Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In *Eurocrypt*, 2015.
12. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Eurocrypt*, 2009.
13. M. Bellare, E. Kiltz, C. Peikert, and B. Waters. Identity-based (lossy) trapdoor functions and applications. In *Eurocrypt*, 2012.
14. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM-CCS*, 1993.
15. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *Eurocrypt*, 1994.
16. M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In *TCC*, 2011.
17. M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack. Cryptology ePrint Archive: Report 2009/101, 2009.
18. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *SAC*, 2002.
19. O. Blazy, S. Kakvi, E. Kiltz, and J. Pan. Tightly-secure signatures from chameleon hash functions. In *PKC*, 2015.
20. F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *J. of Cryptology*, 28(1), 2015.
21. F. Böhl, D. Hofheinz, and D. Kraschewski. On definitions of selective opening security. In *PKC*, 2012.
22. S. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Crypto*, 2008.
23. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Crypto*, 2004.
24. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Eurocrypt*, 2014.
25. D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan. Key-homomorphic PRFs and their applications. In *Crypto*, 2013.
26. X. Boyen and Q. Li. Towards tightly secure lattice short signature and ID-based encryption. In *Asiacrypt*, 2016.
27. X. Boyen and Q. Li. All-but-many lossy trapdoor functions from lattices and applications. In *Crypto*, 2017.
28. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based technique. In *ACM-CCS*, 2005.
29. Z. Brakerski, A. Langlois, C. Peikert, Regev. O., and D. Stehlé. On the classical hardness of learning with errors. In *STOC*, 2013.
30. Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *Crypto*, 2011.
31. Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, 2014.

32. Z. Brakerski and V. Vaikuntanathan. Circuit-ABE from LWE: unbounded attributes and semi-adaptive security. In *Crypto*, 2016.
33. R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *STOC*, 1996.
34. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Eurocrypt*, 2004.
35. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt*, 2010.
36. J. Chen and H. Wee. Fully, (almost) tightly secure ibe and dual system groups. In *Crypto*, 2013.
37. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto*, 1998.
38. I. Damgård and J.-B. Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *Crypto*, 2000.
39. Y. Dodis, E. Kiltz, K. Pietrzak, and D. Wichs. Message authentication, revisited. In *Eurocrypt*, 2012.
40. Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt*, 2004.
41. N. Döttling and D. Schröder. Efficient pseudorandom functions via on-the-fly adaptation. In *Crypto*, 2015.
42. C. Dwork, M. Naor, O. Reingold, and L. Stockmeyer. Magic functions. *J. of the ACM*, 50(6), 2003.
43. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In *Crypto*, 2013.
44. S. Fehr, D. Hofheinz, E. Kiltz, and H. Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In *Eurocrypt*, 2010.
45. D. Freeman, O. Goldreich, E. Kiltz, and G. Rosen, A.and Segev. More constructions of lossy and correlation-secure trapdoor functions. *J. of Cryptology*, 26(1), 2013.
46. E. Fujisaki. All-but-many encryption - a new framework for fully-equipped UC commitments. In *Asiacrypt*, 2014.
47. R. Gay, D. Hofheinz, E. Kiltz, and H. Wee. Tightly CCA-secure encryption without pairings. In *Eurocrypt*, 2016.
48. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
49. C. Gentry, A. Sahai, and B.. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Crypto*, 2013.
50. S. Goldwasser and S. Micali. Probabilistic encryption. *J. of Computer and System Sciences*, 28, 1984.
51. S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, 2015.
52. S. Gorbunov and V. Vinayagamurthy. Riding on asymmetry: Efficient ABE for branching programs. In *Asiacrypt*, 2015.
53. B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *Asiacrypt*, 2011.
54. B. Hemenway and R. Ostrovsky. Extended-DDH and lossy trapdoor functions. In *PKC*, 2012.
55. F. Heuer, T. Jager, E. Kiltz, and S. Schäge. On the selective opening security of practical public-key encryption schemes. In *PKC*, 2015.
56. V.-T. Hoang, J. Katz, A. O'Neill, and M. Zaheri. Selective-opening security in the presence of randomness failures. In *Asiacrypt*, 2016.
57. D. Hofheinz. All-but-many lossy trapdoor functions. In *Eurocrypt*, 2012.
58. D. Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In *Eurocrypt*, 2013.
59. D. Hofheinz. Algebraic partitioning: Fully compact and (almost) tightly secure cryptography. In *TCC-A*, 2016.
60. D. Hofheinz. Adaptive partitioning. In *Eurocrypt*, 2017.
61. D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In *Crypto*, 2012.
62. D. Hofheinz, T. Jager, and A. Rupp. Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In *TCC-B*, 2016.
63. D. Hofheinz, V. Rao, and D. Wichs. Standard security does not imply indistinguishability under selective opening. In *TCC-B*, 2016.
64. Z. Huang, S. Liu, and B. Qin. Sender equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In *PKC*, 2013.
65. E. Kiltz, K. Pietrzak, D. Cash, A. Jain, and D. Venturi. Efficient authentication from hard learning problems. In *Eurocrypt*, 2011.
66. J. Lai, R. Deng, S. Liu, J. Weng, and Y. Zhao. Identity-based encryption secure against selective-opening chosen-ciphertext attack. In *Eurocrypt*, 2014.

67. B. Libert, T. Peters, M. Joye, and M. Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In *Asiacrypt*, 2015.
68. S. Liu and K. Paterson. Simulation-based selective opening CCA security for PKE from key encapsulation mechanisms. In *PKC*, 2015.
69. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.
70. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007.
71. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS*, 1997.
72. M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *Crypto*, 2009.
73. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, 1990.
74. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Eurocrypt*, 1999.
75. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, 2009.
76. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, 2008.
77. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Crypto*, 1991.
78. A. Raghunathan, G. Segev, and S. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In *Eurocrypt'13*, 2013.
79. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.
80. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *TCC*, 2009.
81. H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *Eurocrypt*, 2012.
82. M. Zhandry. The magic of ELFs. In *Crypto*, 2016.

# A Efficiency Improvements for the Scheme of Section 4 Using MACs Instead of PRFs

The bottleneck of our constructions is the need to homomorphically evaluate $O(\lambda)$ circuits when evaluating the all-but-many function in the encryption and decryption algorithms. In our ABM-LTF, the motivation for using a PRF was to prove the feasibility of tightly CCA2-secure public-key encryption under lattice assumptions. In our SIM-SO-CCA2-secure encryption scheme (which is not meant to be tightly secure), we can substantially improve the efficiency by trading the PRF for a MAC in our ABM-LTF construction.

Concretely, we can have the matrices $\mathbf{B}_i = \mathbf{R}_i \cdot \bar{\mathbf{A}} + K[i] \cdot \mathbf{G}$ encrypt the bits of the MAC secret key $K \in \{0,1\}^\lambda$ and define lossy tags $t = (t_\mathsf{c}, t_\mathsf{a})$ to be those for which $\mathsf{MAC.Ver}(K, t_\mathsf{a}, t_\mathsf{c}) = 1$. At Step 1 of the evaluation algorithm $\mathsf{ABM.Eval}(ek, t, \cdot)$, we define $C(t_\mathsf{c}, t_\mathsf{a}) : \{0,1\}^\lambda \to \{0,1\}$ to be the circuit, where $(t_\mathsf{c}, t_\mathsf{a})$ are hardwired, that outputs $1 - \mathsf{MAC.Ver}(K, t_\mathsf{a}, t_\mathsf{c}) \in \{0,1\}$. By doing so, we only need to evaluate *one* circuit.

In order to preserve the evasiveness property of the modified ABM-LTF, the MAC can be randomized but has to be strongly unforgeable. As for the indistinguishability property, we need the MAC to satisfy a stronger flavor of the notion of *key-indistinguishability*[8] introduced by Alwen *et al.* [3]. Namely, we need valid MACs to be computationally indistinguishable from random elements of the ambient space $\mathcal{T}_c$. If the MAC is both strongly unforgeable and pseudorandom, everything goes through in the proofs of IND-SO-CCA2 and SIM-SO-CCA2 security with the modified ABM-LTF.

[8] Key indistinguishability refers to the indistinguishability of two games. In the first one, the adversary has access to MAC and verification oracles for two independent keys while, in the second game, the two oracles use the same secret key.

In [3, Section 3.2], Alwen *et al.* proved the key-indistinguishability of the LPN-based MAC of [65] (of which $\mathsf{LWE}$-based variants were described in [65, Appendix A.3]) by implicitly showing it pseudorandom (see [3, Theorem 2]). Moreover, they also showed that this specific MAC can be made strongly unforgeable via a circuit-depth-preserving transformation due to Dodis *et al.* [39, Section 3.1], which maintains its pseudorandomness. Moreover, the verification algorithm of the resulting MAC fits within an $\mathsf{NC}^1$ circuit as it basically requires two matrix-vector products followed by a Hamming weight test.

# B Definitions for Cryptographic Primitives

## B.1 Public-Key Encryption in the Multi-Challenge Setting

Bellare, Boldyreva and Micali [8] considered the following security definition, which accounts for multiple users and multiple challenge ciphertexts per user.

**Definition 5.** *A public-key encryption scheme is $(\mu, Q_E, Q_D)$-IND-CCA secure, for integers $\mu, Q_E, Q_D$, if no* ppt *adversary has noticeable advantage in this game:*

1. *The challenger first generates $\Gamma \hookleftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$ and runs $(SK^{(j)}, PK^{(j)}) \hookleftarrow \mathsf{Keygen}(\Gamma)$ for $j \leq \mu$. It gives $(PK^{(j)})_{j \leq \mu}$ to the adversary $\mathcal{A}$ and retains $(SK^{(j)})_{j \leq \mu}$. In addition, the challenger initializes a set $\mathcal{Q} \leftarrow \emptyset$ and counters $i_e, i_d \leftarrow 0$. Finally, it chooses a random bit $b \hookleftarrow U(\{0,1\})$.*
2. *The adversary $\mathcal{A}$ adaptively makes queries to the following oracles on multiple occasions:*
   - *Encryption query: $\mathcal{A}$ chooses an index $j \leq \mu$ and a pair $(\mathsf{Msg}_0, \mathsf{Msg}_1)$ of equal-length messages; if $i_e = Q_E$, the oracle returns $\perp$; otherwise, it computes $C \hookleftarrow \mathsf{Encrypt}(PK^{(j)}, \mathsf{Msg}_b)$ and returns $C$; in addition, it sets $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(j, C)\}$ and $i_e \leftarrow i_e + 1$.*
   - *Decryption query: $\mathcal{A}$ chooses an index $j \leq \mu$ and an element $C$ of the ciphertext domain; if $i_d = Q_D$ or $(j, C) \in \mathcal{Q}$, the oracle returns $\perp$; otherwise, the oracle returns $\mathsf{Msg} \leftarrow \mathsf{Decrypt}(SK^{(j)}, C)$, which may be $\perp$ if $C$ is an invalid ciphertext, and sets $i_d \leftarrow i_d + 1$.*
3. *The adversary $\mathcal{A}$ outputs a bit $b'$ and is deemed successful if $b' = b$. As usual, $\mathcal{A}$'s advantage is measured as the distance*

$$\mathbf{Adv}^{\mathsf{CCA}}_{\mu, Q_E, Q_D}(\lambda) := |\Pr[b' = b] - 1/2|.$$

Using the random self-reducibility property of $\mathsf{DDH}$, well-known constructions (e.g., the Cramer-Shoup encryption scheme [37]) were shown [8] to provide tight $(\mathsf{poly}, 1, \mathsf{poly})$-IND-CCA security. Hofheinz and Jager [61] gave the first tight security result in the most general $(\mathsf{poly}, \mathsf{poly}, \mathsf{poly})$ case. Here, we restrict ourselves to the single-user $(1, \mathsf{poly}, \mathsf{poly})$ setting, in which we provide the first tight multi-challenge security results under lattice assumptions.

### B.2 One-Time Signatures, Message Authentication Codes and Pseudorandom Functions

A one time signature is a tuple of efficient algorithms $(\mathcal{G}, \mathcal{S}, \mathcal{V})$, where:

- $\mathcal{G}$ takes as input a security parameter $1^\lambda$ and, optionally, a set of public parameters $\Gamma$. It outputs a key pair $(\mathsf{SVK}, \mathsf{SSK})$
- $\mathcal{S}$ is a possibly randomized algorithm that takes as input a message $M$ and a secret key $\mathsf{SSK}$. It outputs a signature $sig$.
- $\mathcal{V}$ is a deterministic algorithm taking as input a verification key $\mathsf{SVK}$, a message $M$ and a candidate signature $sig$. It outputs 1 or 0.

We consider a security definition in the multi-signer setting (see, e.g., [59]). Note that, while security in the single-user setting implies security in the multi-user setting via a standard hybrid argument, this argument is not tight as it incurs a linear security loss w.r.t. $N$.

**Definition 6.** *A one-time signature $(\mathcal{G}, \mathsf{S}, \mathcal{V})$ provides one-time strong unforgeability in the multi-user setting if no* ppt *adversary has non-negligible advantage in the following game.*

1. *The challenger generates $N$ key pairs $(\mathsf{SVK}_i, \mathsf{SSK}_i) \leftarrow \mathcal{G}(\Gamma, 1^\lambda)$ and gives $(\mathsf{SVK}_i)_{i \leq N}$ to the adversary $\mathcal{A}$.*
2. *The adversary adaptively makes up to $N$ queries of the form $(i, M_i)$. At each query, it obtains $sig_i \leftarrow \mathcal{S}(\mathsf{SSK}_i, M_i)$. Note that a single query is allowed for each index $i \leq N$.*
3. *$\mathcal{A}$ outputs $(i^\star, M^\star, sig^\star)$ and wins if $\mathcal{V}(\mathsf{SVK}_{i^\star}, M^\star, sig^\star) = 1$ and $(M^\star, sig^\star) \neq (M_{i^\star}, sig_{i^\star})$.*

*The adversary's advantage $\mathbf{Adv}_N^{\mathcal{A}, \text{suf-OTS}}(\lambda)$ its probability of success taken over all random coins.*

A message authentication code (MAC) is a triple $\mathsf{MAC} = (\mathsf{KG}, \mathsf{Sig}, \mathsf{Ver})$ of efficient algorithms which are associated with a key space $\mathcal{K}$ and a message space $\mathcal{M}$. The syntax of these algorithms is the following:

**MAC.KG**$(1^\lambda)$ takes as input a security parameter $1^\lambda$ and outputs a random secret key $k \in \mathcal{K}$.

**MAC.Sig**$(k, M)$ is an algorithm taking as input a message $M \in \mathcal{M}$ and a secret key $k \in \mathcal{K}$. It outputs an authentication value $s \leftarrow \mathsf{MAC.Sig}(k, M)$.

**MAC.Ver**$(k, M, s)$ is a deterministic algorithm that takes in a secret key $k \in \mathcal{K}$, a message $M \in \mathcal{M}$ and a candidate MAC value $s$. It outputs 1 or 0.

**Definition 7.** *A MAC is strongly one-time unforgeable under chosen-message attacks (sUF-OT-CMA) if no* ppt *adversary has non-negligible advantage in the following game:*

1. *The challenger chooses a random key $k \in \mathcal{K}$. The adversary $\mathcal{A}$ is run on input of the security parameter $1^\lambda$.*
2. *Adversary $\mathcal{A}$ adaptively makes the following kinds of queries:*
   - *MAC queries: $\mathcal{A}$ chooses an arbitrary message $M \in \mathcal{M}$. The challenger computes $s \leftarrow \mathsf{MAC.Sig}(k, M)$ and returns $\sigma$.*

- *Verification queries:* $\mathcal{A}$ *chooses an arbitrary pair* $(M, s)$*. The challenger returns the output of* $\mathsf{MAC.Ver}(k, M, s) \in \{0,1\}$ *to* $\mathcal{A}$*.*

*While* $\mathcal{A}$ *can make arbitrarily many verification queries, only one MAC query is allowed and we call* $(M^\dagger, s^\dagger)$ *the input-output pair of that query.*

3. *When* $\mathcal{A}$ *halts, it outputs a pair* $(M^\star, s^\star)$ *and wins if the following conditions are satisfied: (i)* $(M^\star, s^\star) \neq (M^\dagger, s^\dagger)$*; (ii)* $\mathsf{MAC.Ver}(k, M^\star, s^\star) = 1$*.*

*We define the adversary's advantage* $\mathbf{Adv}^{\mathcal{B},\mathsf{MAC}}_{Q_V}(\lambda)$ *after* $Q_V$ *verification queries as its probability of success taken over all random choices.*

We will consider *unique* one-time MACs, where $\mathsf{MAC.Sig}$ does not use any random coins and each message $M$ has only one valid $s$. For example, any PRF is a unique MAC.

**Definition 8.** *Let* $\lambda$ *be a security parameter and let* $\kappa = \kappa(\lambda)$*. A pseudorandom function* $\mathsf{PRF} : \{0,1\}^\lambda \times \{0,1\}^\kappa \to \{0,1\}^\lambda$ *is an efficiently computable function where the first input* $K \in \{0,1\}^\lambda$ *is the key. Let* $\Omega$ *be the set of all functions that map* $\kappa$*-bit inputs to* $\lambda$*-bit strings. The advantage of a PRF distinguisher* $\mathcal{D}$ *making* $Q$ *evaluation queries is defined as*

$$\mathbf{Adv}^{\mathcal{D},\mathrm{prf}}_{Q}(\lambda) := |\Pr[\mathcal{D}^{\mathsf{PRF}(K,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{D}^{F(\cdot)}(1^\lambda) = 1]|,$$

*where the probability is taken over the random choice of* $K \hookleftarrow U(\{0,1\}^\lambda)$ *and* $F \hookleftarrow U(\Omega)$ *and the coin tosses of* $\mathcal{D}$*.*

### B.3 Definition of Indistinguishability-Based Selective Opening Chosen-Ciphertext (IND-SO-CCA2) Security

It was shown [12, 17] that, for message distributions supporting efficient conditional re-sampling, lossy trapdoor functions imply SOA security in the sense of an indistinguishability-based definition.

**Definition 9 ([57]).** *Let* $N = N(\lambda)$ *and let* $\mathcal{M}$ *be a joint distribution over* $\mathsf{MsgSp}(\lambda)^N$*. We say that* $\mathcal{M}$ *supports efficient conditional re-sampling if there is an efficient algorithm* $\mathsf{ReSamp}_{\mathcal{M}}$ *such that, for any* $I \subseteq [N]$*, and any set* $\mathbf{Msg}_I$ *of pairs* $\{(i, \mathsf{Msg}_i)\}_{i \in I}$*, with* $\mathsf{Msg}_i \in \mathsf{MsgSp}(\lambda)$ *for each* $i \in I$*,* $\mathsf{ReSamp}_{\mathcal{M}}(I, \mathbf{Msg}_I)$ *samples from the distribution* $\mathcal{M}$ *by outputting a* $N$*-vector* $\mathbf{Msg}' \in \mathsf{MsgSp}(\lambda)^N$ *such that* $\mathbf{Msg}'[i] = \mathsf{Msg}_i$ *for each* $i \in I$*.*

For such distributions, as in [57], we first aim at IND-SOA security under chosen-ciphertext attacks: i.e., when the adversary is granted access to a decryption oracle.

**Definition 10 ([53, 57]).** *A public-key encryption scheme* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *provides indistinguishability-based selective-opening security under chosen-ciphertext attacks (or IND-SO-CCA2 security) if, for any polynomial* $N \in \mathsf{poly}(\lambda)$ *and any message distribution* $\mathcal{M} \in \mathsf{MsgSp}(\lambda)^N$ *supporting efficient conditional re-sampling, no* ppt *adversary* $\mathcal{A}$ *has non-negligible advantage in this game.*

1. *The challenger flips a random coin* $b \hookleftarrow U(\{0,1\})$. *It generates public parameters* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$ *with a key pair* $(SK, PK) \leftarrow \mathsf{Keygen}(\Gamma)$ *and gives* $(\Gamma, PK)$ *to* $\mathcal{A}$ *which adaptively makes the following kinds of queries:*
   - ***Challenge Query:*** *let* $\mathcal{M}$ *be a message sampler for* $\mathsf{MsgSp}(\lambda)$. *The challenger samples* $\mathbf{Msg} = (\mathsf{Msg}_1, \ldots, \mathsf{Msg}_N) \leftarrow \mathcal{M}$ *and returns* $N$ *target ciphertexts*
   $$\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_N) \leftarrow \big(\mathsf{Encrypt}(PK, \mathsf{Msg}_1, r_1), \ldots, \mathsf{Encrypt}(PK, \mathsf{Msg}_N, r_N)\big).$$
   *which are computed using independent random coins* $r_1, \ldots, r_N \hookleftarrow \mathcal{R}$.
   - ***Corrupt Query:*** $\mathcal{A}$ *chooses an arbitrary subset* $I \subset \{1, \ldots, N\}$. *The challenger then reveals* $\{(\mathsf{Msg}_i, r_i)\}_{i \in I}$ *to* $\mathcal{A}$. *As for indexes* $i \in [N] \setminus I$, *the challenger does the following:*
     - *If* $b = 1$, *reveal* $\{\mathsf{Msg}_j\}_{j \notin I}$ *to* $\mathcal{A}$.
     - *In* $b = 0$, *re-sample* $(\mathsf{Msg}'_1, \ldots, \mathsf{Msg}'_N) \leftarrow \mathsf{ReSamp}_{\mathcal{M}}(I, \{(i, \mathsf{Msg}_i)\}_{i \in I})$ *and return the subset* $\{\mathsf{Msg}'_j\}_{j \notin I}$ *of re-sampled messages.*
   - ***Decryption Queries:*** $\mathcal{A}$ *chooses a ciphertext* $C$ *such that* $C \neq \mathbf{C}_i$ *for each* $i \in [N]$ *and obtains* $\mathsf{Decrypt}(SK, C) \in \mathsf{MsgSp}(\lambda) \cup \{\perp\}$.

*After polynomially-many decryption queries and exactly one challenge query followed by one corruption query,* $\mathcal{A}$ *outputs a bit* $b' \in \{0,1\}$ *and wins if* $b' = b$. *Its advantage is defined as the distance* $\mathbf{Adv}^{\text{IND-SO-CCA2}}_{\mathcal{A}}(\lambda) := |\Pr[b' = b] - 1/2|$, *where the probability is taken over all coin tosses.*

We insist that, in the above definition, the challenger is only efficient for distributions supporting efficient conditional resampling. It is known [63] that, under certain conditions, standard security notions for public-key encryption do *not* imply security under indistinguishability-based selective openings.

### B.4 Lossy Encryption

In [12] Bellare, Hofheinz and Yilek defined the notion of *lossy encryption*. In short, a lossy encryption scheme admits two computationally indistinguishable distributions of public keys. On injective keys, the system behaves in the usual way whereas, for lossy public keys, ciphertexts are statistically independent of the message they encrypt. Yet, no ppt adversary should be able to distinguish normal keys from lossy keys. It was proved in [12] that any lossy encryption scheme provides IND-SO-CPA security.

Bellare *et al.* [12] also consider a property, called *openability*, which allows a possibly inefficient algorithm Opener to open a ciphertext $\mathbf{C}$ generated under a lossy key to *any* arbitrary plaintext Msg by outputting coins $r \in \mathcal{R}$ such that $\mathbf{C} = \mathsf{Encrypt}(PK, \mathsf{Msg}, r)$. While lossy encryption is limited to provide IND-SO-CPA security, simulation-based security is also achieved when Opener is a ppt algorithm. As pointed out in [17], a relaxed notion of efficient openability (termed *weak efficient openability* by Hofheinz *et al.* [62]) suffices to ensure SIM-SO-CPA security: in this relaxed flavor of openability, the Opener algorithm is given access to the original message and random coins that were used to create the ciphertext, which must now be opened to a different plaintext. Here, we adapt this definition to the case of a non-uniform random coin distribution.

**Definition 11 ([12, 62]).** *A lossy PKE scheme with weak efficient opening consists of a tuple* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *of efficient algorithms such that*

- $\mathsf{Keygen}(\Gamma, \mathsf{inj})$ *outputs injective keys* $(PK_{\mathsf{inj}}, SK_{\mathsf{inj}})$.
- $\mathsf{Keygen}(\Gamma, \mathsf{loss})$ *outputs keys* $(PK_{\mathsf{loss}}, SK_{\mathsf{loss}})$, *which are called* lossy keys.

*Moreover, these algorithms must satisfy the following properties:*

1. *For any public parameters* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$, *if* $(PK, SK) \leftarrow \mathsf{Keygen}(\Gamma, \mathsf{inj})$, *for any* $\mathsf{Msg} \in \mathsf{MsgSp}$, *we have* $\mathsf{Decrypt}(SK, \mathsf{Encrypt}(PK, \mathsf{Msg})) = \mathsf{Msg}$ *with overwhelming probability.*
2. *Lossy public keys are computationally indistinguishable from injective ones. Namely, for any public parameters* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$, *we have*

$$\{PK \mid (PK, SK) \leftarrow \mathsf{Keygen}(\Gamma, \mathsf{inj})\} \approx_c \{PK \mid (PK, SK) \leftarrow \mathsf{Keygen}(\Gamma, \mathsf{loss})\}$$

3. *If* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$ *and* $(PK, SK) \leftarrow \mathsf{Keygen}(\Gamma, \mathsf{loss})$, *then for any distinct messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, *the two distributions*

$$\{C \mid C \leftarrow \mathsf{Encrypt}(PK_{\mathsf{loss}}, \mathsf{Msg}_0)\}, \qquad \{C \mid C \leftarrow \mathsf{Encrypt}(PK_{\mathsf{loss}}, \mathsf{Msg}_1)\}$$

   *are statistically close.*

4. *Let* $D_{\mathcal{R}}$ *denote the distribution of random coins input to* $\mathsf{Encrypt}$. *For any message* $\mathsf{Msg} \in \mathsf{MsgSp}$ *and ciphertext* $C$, *let* $D_{PK,\mathsf{Msg},C}$ *denote the probability distribution on randomness space* $\mathcal{R}$ *with support*

$$S_{PK,\mathsf{Msg},C} = \{\overline{R} \in \mathcal{R} : \mathsf{Encrypt}(PK, \mathsf{Msg}, \overline{R}) = C\},$$

   *and such that, for each* $\overline{R} \in S_{PK,\mathsf{Msg},C}$, *we have*

$$D_{PK,\mathsf{Msg},C}(\overline{R}) = \Pr_{R' \hookleftarrow D_{\mathcal{R}}}[R' = \overline{R} \mid \mathsf{Encrypt}(PK, \mathsf{Msg}, R') = C].$$

   *There exists a* ppt *sampling algorithm* $\mathsf{Opener}$ *such that, given public parameters* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$, *lossy keys* $(PK_{\mathsf{loss}}, SK_{\mathsf{loss}}) \leftarrow \mathsf{Keygen}(\Gamma, \mathsf{loss})$, *random coins* $R \hookleftarrow \mathcal{R}$ *and any two messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, *outputs, with probability* $\geq 1 - 2^{-\Omega(\lambda)}$ *over* $(SK_{\mathsf{loss}}, PK_{\mathsf{loss}}, R)$, *an independent sample* $\overline{R}$ *from the distribution* $D_{PK,\mathsf{Msg}_1,C}$, *where* $C = \mathsf{Encrypt}(PK_{\mathsf{loss}}, \mathsf{Msg}_0, R)$.

While weak efficient openability (i.e., property 4) is a weaker property than that of efficient openability, it suffices to guarantee simulation-based selective opening security.

**Lemma 17 ([12, 17]).** *Any lossy encryption scheme with efficient weak opening is SIM-SO-CPA secure.*

While the above result was only proved for chosen-plaintext adversaries, it carries over to the chosen-ciphertext scenario. Namely, any IND-SO-CCA2 secure lossy encryption scheme is also secure in the SIM-SO-CCA2 sense.

# C Tight Chosen-Ciphertext Security from LWE

Here, we combine our ABM-LTF and our LTF of Section 3 to obtain a public-key encryption scheme whose chosen-ciphertext security in the multi-challenge setting tightly relates to the LWE assumption if instantiated with a tightly secure LWE-based PRF.

The construction is similar to that of Section 4. The difference is that, instead of relying on the hybrid encryption paradigm, it uses one-time signatures, which allows for a tight reduction as we do not lose a factor $Q_E$ (i.e., the number of queries to the encryption oracle) w.r.t. the computational security of the MAC. Note that, in Section 4, we cannot rely on an unconditionally secure one-time MAC since the MAC secret key would have to be longer than the plaintext, which would not be compatible with the parameters of our Opener algorithm.

## C.1 Description

We assume w.l.o.g. that the verification key SVK of the one-time signature fits within the space of auxiliary tags of the ABM-LTF, since it can always be hashed for this purpose.

**Par-Gen**$(1^\lambda)$**.** Selects public parameters consisting of:

- A modulus $q > 2$, integers $\ell, \ell_0, n \in \mathsf{poly}(\lambda)$, $m = \lceil cn \cdot \log q \rceil$, for some constant $c > 0$, parameters $\sigma_x, \sigma_e > 0$, and an efficiently samplable LWE noise distribution.
- The specification $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ of a one-time signature.
- A family $\mathcal{UH}$ of universal hash functions $h : [-\sigma_x\sqrt{n}, \sigma_x\sqrt{n}]^n \to \mathbb{Z}_q^{\ell_0}$.

The public parameters $\Gamma = \{\ell, \ell_0, n, m, q, \sigma_x, \sigma_e, \chi, \Sigma, \mathcal{UH}\}$ define the plaintext space $\mathsf{MsgSp} := \mathbb{Z}_q^{\ell_0}$ and will be shared by the LTF of Section 3.1 and the ABM-LTF of Section 3.2.

**Keygen**$(\Gamma)$**.** Let $\Pi^{\mathsf{LTF}} = (\mathsf{IGen}, \mathsf{LGen}, \mathsf{Eval}, \mathsf{Invert})$ be an instance of the LTF family of Section 3.1 and let $\Pi^{\mathsf{ABM}} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert}, \mathsf{LTag})$ be an instance of the ABM-LTF family of Section 3.2. We assume that functions $\Pi^{\mathsf{LTF}}$ and $\Pi^{\mathsf{ABM}}$ operate over the common domain

$$\mathsf{Dom}_\lambda^D := \{(\mathbf{x}, \mathbf{e}) \in \mathbb{Z}^n \times \mathbb{Z}^{2m} \mid \|\mathbf{x}\| \leq \sigma_x \cdot \sqrt{n},\ \|\mathbf{e}\| \leq \sigma_e \cdot \sqrt{2m}\}.$$

The public key is generated via the following steps.

1. Generate $(ek', ik') \hookleftarrow \Pi^{\mathsf{LTF}}.\mathsf{IGen}(1^\lambda)$ for an injective function of the LTF family to obtain $ek' = \mathbf{A} \in \mathbb{Z}_q^{2m \times n}$ and $ik' = \mathbf{R} \in \{-1, 1\}^{m \times m}$.
2. Generate $(ek, ik, tk) \hookleftarrow \Pi^{\mathsf{ABM}}.\mathsf{Gen}(1^\lambda)$ to obtain $ek = (\bar{\mathbf{A}}, (\mathbf{B}_i)_{i \leq \lambda})$ and $ik = ((\mathbf{R}_i)_{i \leq \lambda}, K)$.
3. Sample an element $h \hookleftarrow U(\mathcal{UH})$ of the universal hash family.

Output $(PK, SK)$ where the public key is $PK = \big(ek', ek, h\big)$ and the underlying private key consists of $SK = ik'$.

**Encrypt**$(PK, \mathsf{Msg})$**.** To encrypt $\mathsf{Msg} \in \mathbb{Z}_q^{\ell_0}$, generate a key pair $(\mathsf{SVK}, \mathsf{SSK}) \hookleftarrow \mathcal{G}(1^\lambda)$ and do the following.

1. Sample $(\mathbf{x}, \mathbf{e}_0, \mathbf{e}) \hookleftarrow D_{\mathbb{Z}^n, \sigma_x} \times D_{\mathbb{Z}^{2m}, \sigma_e} \times D_{\mathbb{Z}^{2m}, \sigma_e}$.
2. Compute $\mathbf{c}_0 = \mathsf{Msg} + h(\mathbf{x}) \in \mathbb{Z}_q^{\ell_0}$.

3. Compute $\mathbf{y}_0 = \Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek', (\mathbf{x}, \mathbf{e}_0)) = \mathbf{A} \cdot \mathbf{x} + \mathbf{e}_0 \ \in \mathbb{Z}_q^{2m}$.
4. Define the auxiliary tag $t_{\mathsf{a}} = \mathsf{SVK}$ and choose a random $t_{\mathsf{c}} \hookleftarrow U(\mathcal{T}_{\mathsf{c}})$. Let $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ and compute $\mathbf{y} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}(ek, t, (\mathbf{x}, \mathbf{e})) = \mathbf{A}_t \cdot \mathbf{x} + \mathbf{e} \ \in \mathbb{Z}_q^{2m}$.
5. Generate a one-time signature $sig = \mathcal{S}(\mathsf{SSK}, (t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}))$.

Output the ciphertext $C = (\mathsf{SVK}, t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}, sig)$.

**Decrypt**$(SK, C)$**.** To decrypt $C = (\mathsf{SVK}, t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}, sig)$ using $SK = ik'$,

1. If $\mathcal{V}(\mathsf{SVK}, (t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}), sig) = 0$, return $\perp$.
2. Compute $(\mathbf{x}, \mathbf{e}_0) \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{Invert}(ik', \mathbf{y}_0)$ and return $\perp$ if $\mathbf{y}_0$ is not in the range of $\Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek', \cdot)$ or if $(\mathbf{x}, \mathbf{e}_0) \notin \mathsf{Dom}_\lambda^D$.
3. Define the tag $t = (t_{\mathsf{c}}, \mathsf{SVK})$. If $\|\mathbf{y} - \mathbf{A}_t \cdot \mathbf{x}_0\| > \sigma_e \cdot \sqrt{2m}$, return $\perp$.
4. Return $\mathsf{Msg} = \mathbf{c}_0 - h(\mathbf{x}) \in \mathbb{Z}_q^{\ell_0}$.

In order to instantiate the above construction with a polynomial-size modulus $q$, we need a PRF that can be evaluated via an $\mathsf{NC}^1$ circuit – and thus a polynomial-length branching program [7] – in order to apply Lemma 10. For this purpose, a tempting idea is to use the PRF[9] of Banerjee, Peikert and Rosen [6, Section 5], which has a tight reduction from an $\mathsf{LWE}$ assumption. However, as mentioned in [6, Section 5.2], evaluating this function in $\mathsf{NC}^1$ is possible when the key is fixed and hard-coded into the circuit. In our setting, the input of the circuit should be the seed $K$ – which is GSW-encrypted in the matrices $\{\mathbf{B}_i\}_{i \leq \lambda}$ contained in $ek$ – while it is the public string $t_{\mathsf{a}}$ that can be hard-wired in the circuit to be evaluated on the encrypted $K \in \{0,1\}$. In this case, the construction of [6, Section 5.2] seemingly requires an $\mathsf{NC}^2$ circuit.

Another PRF candidate is the key-homomorphic construction of Boneh *et al.* [25] as it is easily computable in $\mathsf{NC}^1$ when the input is fixed. One hurdle is that the security proof of [25] is not tight as it incurs a concrete security loss (i.e., the ratio between the advantages of the adversary and the reduction) proportional to the number of evaluation queries. Fortunately, in Appendix F, we give a tighter security proof which eliminates the multiplicative gap proportional to the number of evaluation queries. This new proof thus allows instantiating the above cryptosystem system with a polynomial-size modulus while retaining a tight proof of CCA2 security (albeit under a strong $\mathsf{LWE}$ assumption).

### C.2 Security

We show that the scheme provides almost tight (according to the terminology of [36]) CCA2 security in the multi-challenge setting (in the sense of Definition 5 in Appendix B.1) under the $\mathsf{LWE}$ assumption. Namely, the security bound does not depend on the number $Q_E$ of challenge ciphertext obtained by the adversary. In order to avoid a degradation factor proportional to the number of decryption queries (which appears in [57, Theorem 6.5]), the security proof exploits the fact that our definition of evasiveness involves an $\mathsf{ABM.IsLossy}$ oracle to detect when the adversary creates a breach in the evasiveness property. This saves the reduction from losing a factor $Q_D$, where $Q_D$ is the number of decryption queries.

[9] This PRF maps inputs $x \in \{0,1\}^\kappa$ to outputs $\lfloor (p/q) \cdot \mathbf{A} \cdot \prod_{i=1}^{\kappa} \mathbf{S}_i^{x_i} \rfloor$, for secret matrices $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{S}_i \in \mathbb{Z}^{n \times n}$ and prime moduli $p < q$.

**Theorem 5.** *The above scheme provides $(1, Q_E, Q_D)$-IND-CCA security assuming that: (i) $\Pi^{\mathsf{LTF}}$ is an LTF; (ii) $\Pi^{\mathsf{ABM}}$ is an ABM-LTF; (iii) $\mathsf{PRF}$ is a pseudo-random function family; (iv) $\Sigma$ is a strongly unforgeable one-time signature. In our instantiation, for any $\mathsf{ppt}$ adversary $\mathcal{A}$, there exists an $\mathsf{LWE}$ distinguisher $\mathcal{D}_1$, a PRF adversary $\mathcal{D}_2$ and a signature forger $\mathcal{B}$ which all run in about the same time as $\mathcal{A}$ and such that*

$$\mathbf{Adv}^{\mathcal{A},\mathsf{CCA}}_{1,Q_E,Q_D}(\lambda) \leq 4n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{LWE}}_{\ell,2m,q,\chi}(\lambda) + 2 \cdot \mathbf{Adv}^{\mathcal{D}_2,\mathsf{PRF}}_{Q_E+Q_D}(\lambda) + \mathbf{Adv}^{\mathcal{B},\text{suf-}\mathsf{OTS}}_{Q_E} + \frac{Q_D + Q_E + 2}{2^{\lambda-1}}. \quad (11)$$

To properly instantiate the above construction, we need a lattice-based one-time signature with a tight security proof in the multi-user setting (as defined in Appendix B.2) as the adversary can see signatures for up to $Q_E$ verification keys in the proof of Theorem 5. We provide a simple example in Appendix C.3.

*Proof (of Theorem 5).* The proof proceeds with a sequence of games. For each $i$, we let $S_i$ denote the event that the adversary $\mathcal{A}$ wins (by outputting $b' \in \{0,1\}$ such that $b' = b$) in Game $i$.

**Game** 0: This is the real game. It begins with the challenger $\mathcal{B}$ choosing a bit $b \hookleftarrow U(\{0,1\})$ and generating $PK = (ek', ek, h)$ while keeping $SK = (ik, ik')$ to itself. We may assume w.l.o.g. that the auxiliary tags $t_{\mathsf{a}} = \mathsf{SVK}^{(i)}$ of all challenge ciphertexts are chosen at the very beginning of the game. At the outset of the game, $\mathcal{B}$ thus samples

$$(\mathbf{x}^{(1)}, \mathbf{e}_0^{(1)}, \mathbf{e}^{(1)}), \ldots, (\mathbf{x}^{(Q_E)}, \mathbf{e}_0^{(Q_E)}, \mathbf{e}^{(Q_E)}) \hookleftarrow D_{\mathbb{Z}^n,\sigma_x} \times D_{\mathbb{Z}^{2m},\sigma_e} \times D_{\mathbb{Z}^{2m},\sigma_e}.$$

For each $i \in [Q_E]$, it then computes $\mathbf{y}_0^{(i)} = \Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek', (\mathbf{x}^{(i)}, \mathbf{e}^{(i)}))$ and $\mathbf{k}^{sym,(i)} = h(\mathbf{x}^{(i)})$. It also defines $t_{\mathsf{a}}^{(i)} = \mathsf{SVK}^{(i)}$, defines $t^{(i)} = (t_{\mathsf{c}}^{(i)}, t_{\mathsf{a}}^{(i)})$ for a random core tag $t_{\mathsf{c}}^{(i)} \hookleftarrow U(\mathcal{T}_{\mathsf{c}})$ and computes $\mathbf{y}^{(i)} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}\big(ek, t^{(i)}, (\mathbf{x}^{(i)}, \mathbf{e}^{(i)})\big)$.

At each encryption query $\big(\mathsf{Msg}_0^{(i)}, \mathsf{Msg}_1^{(i)}\big)$, the adversary obtains an encryption of $\mathsf{Msg}_b^{(i)}$. We denote by $C^{(i)} = (\mathsf{SVK}^{(i)}, t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}, sig^{(i)})$ the $i$-th challenge ciphertext, which is obtained as

$$\begin{aligned}
\mathbf{c}_0^{(i)} &= \mathsf{Msg}_b^{(i)} + \mathbf{k}^{sym,(i)} \mod q \\
\mathbf{y}_0^{(i)} &= \Pi^{\mathsf{LTF}}.\mathsf{Eval}\big(ek', (\mathbf{x}^{(i)}, \mathbf{e}_0^{(i)})\big) \\
\mathbf{y}^{(i)} &= \Pi^{\mathsf{ABM}}.\mathsf{Eval}\big(ek, t^{(i)}, (\mathbf{x}^{(i)}, \mathbf{e}^{(i)})\big).
\end{aligned}$$

All decryption queries are answered using the inversion trapdoor $ik'$ of $\Pi^{\mathsf{LTF}}$. When $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$ and we call $S_0$ the event that $b' = b$.

**Game** 1: In this game, $\mathcal{B}$ rejects all decryption queries $C = (\mathsf{SVK}, t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}, sig)$ such that $\mathsf{SVK} = \mathsf{SVK}^{(i)}$ for some $i \in [Q_E]$. It is easy to see that, if $\Sigma$ is a strongly unforgeable one-time signature, $\mathcal{B}$ does not reject any ciphertext that it would not also reject in Game 0. We have $|\Pr[S_1] - \Pr[S_0]| \leq \mathbf{Adv}^{\text{suf-ots}}(\lambda)$.

**Game** 2: We modify the encryption oracle. At each encryption query $(\mathsf{Msg}_0^{(i)}, \mathsf{Msg}_1^{(i)})$, the challenger encrypts $C^{(i)} = (\mathsf{SVK}^{(i)}, t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}, sig^{(i)})$ on a lossy tag $t^{(i)} = (t_{\mathsf{c}}^{(i)}, \mathsf{SVK}^{(i)})$. In the instantiation based on our ABM-LTF, this implies that the core tag component $t_{\mathsf{c}}^{(i)}$ is computed as a pseudo-random function $t_{\mathsf{c}}^{(i)} = \mathsf{PRF}(K, t_{\mathsf{a}}^{(i)})$ of the auxiliary tag component $t_{\mathsf{a}}^{(i)} = \mathsf{SVK}^{(i)}$ at the beginning of the game. The indistinguishability property of our ABM-LTF ensures that this change will not noticeably increase $\mathcal{A}$'s probability $\Pr[S_2]$ of success. We have $|\Pr[S_2] - \Pr[S_1]| \leq 2n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{lwe}}_{\ell,m,q,\chi}(\lambda) + \mathbf{Adv}^{\mathcal{D}_2,\mathrm{prf}}_{Q_E}(\lambda) + \frac{1}{2^{\lambda-1}}$, for some efficient distinguishers $\mathcal{D}_1$ and $\mathcal{D}_2$.

**Game** 3: In this game, we modify the decryption oracle. Namely, at each decryption query $C = (\mathsf{SVK}, t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}, sig)$, $\mathcal{B}$ still rejects $C$ if $\mathsf{SVK} \in \{\mathsf{SVK}^{(1)}, \ldots, \mathsf{SVK}^{(Q_E)}\}$. However, it also rejects $C$ in the event that $\mathsf{SVK} \notin \{\mathsf{SVK}^{(1)}, \ldots, \mathsf{SVK}^{(Q_E)}\}$ but the corresponding tag $t = \big(t_{\mathsf{c}}, \mathsf{SVK}\big)$ is a lossy tag. Clearly, Game 4 is identical to Game 3 until the event $F_3$ that $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game 3. In this case, $\mathcal{A}$ manages to break the evasiveness property of our ABM-LTF.

We claim that $\Pr[F_3] \leq \mathbf{Adv}^{\mathcal{A},\mathsf{eva}}_{Q_E,Q_D}(\lambda)$. To see this, we describe an evasiveness adversary $\mathcal{B}$ with advantage $\Pr[F_3]$. Concretely, $\mathcal{B}$ interacts with an evasiveness challenger that provides it with an evaluation key $ek$ and oracles $\mathsf{ABM.LTag}(tk, .)$ and $\mathsf{ABM.IsLossy}(tk, .)$. The adversary is run on input of the public encryption key $PK = \big(ek', ek, h\big)$ where the evaluation key $ek'$ of $\Pi^{\mathsf{LTF}}$ is obtained by generating $(ek', ik') \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{IGen}(1^\lambda)$. At the $i$-th encryption query $(\mathsf{Msg}_0^{(i)}, \mathsf{Msg}_1^{(i)})$ made by $\mathcal{A}$ during the game, $\mathcal{D}$ defines $t_{\mathsf{a}}^{(i)} = \mathsf{SVK}^{(i)}$, which it queries to its $\mathsf{ABM.LTag}(tk, .)$ oracle and obtains $t_{\mathsf{c}}^{(i)} \in \{0,1\}^\lambda$. Next, $\mathcal{B}$ fetches $(\mathbf{x}^{(i)}, \mathbf{e}_0^{(i)}, \mathbf{e}^{(i)})$ and computes $\mathbf{y}_0^{(i)} = \Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek', (\mathbf{x}^{(i)}, \mathbf{e}_0^{(i)}))$ as well as $\mathbf{y}^{(i)} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}(ek, t^{(i)}, (\mathbf{x}^{(i)}, \mathbf{e}^{(i)}))$, where $t^{(i)} = (t_{\mathsf{c}}^{(i)}, \mathsf{SVK}^{(i)})$. Then, it returns the ciphertext $C^{(i)} = (\mathsf{SVK}^{(i)}, t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}, sig^{(i)})$ to $\mathcal{A}$.

At each decryption query $C = (\mathsf{SVK}, t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}, sig)$, our evasiveness adversary $\mathcal{B}$ queries its $\mathsf{ABM.IsLossy}(tk, .)$ oracle on the input $(t_{\mathsf{c}}, \mathbf{y}_0)$. If the response is 1, then $\mathcal{B}$ halts and outputs $(t_{\mathsf{c}}, \mathsf{SVK})$. Otherwise, $\mathcal{B}$ continues and uses the LTF inversion key $ik'$ – which it knows from the key generation phase – to faithfully run the real decryption algorithm. If $\mathcal{B}$ did not halt by the time $\mathcal{A}$ terminates, it aborts. It is easy to see that $\mathcal{B}$ succeeds in breaking the evasiveness with probability $\Pr[F_3]$. This yields the announced lower bound $\mathbf{Adv}^{\mathcal{A},\mathsf{eva}}_{Q_E,Q_D}(\lambda) \geq \Pr[F_3]$.

Using our concrete $\mathsf{LWE}$-based ABM-LTF, the result of Lemma 13 implies the inequality

$$|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F_3] \leq n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{lwe}}_{\ell,m,q,\chi}(\lambda) + \mathbf{Adv}^{\mathcal{D}_2,\mathrm{prf}}_{Q_E+Q_D}(\lambda) + \frac{Q_D+1}{2^\lambda}.$$

In subsequent games, we do not rely on the pseudo-randomness of the PRF any more and we can henceforth explicitly use its seed $K$ to detect lossy tags when they show up.

**Game** 4: In this game, we modify again the decryption oracle. In this game, we do no longer use the inversion trapdoor $ik'$ of $\Pi^{\mathsf{LTF}}$ but rather use the trapdoor $ik$ of the ABM-LTF function $\Pi^{\mathsf{ABM}}$. At each decryption query $C = (\mathsf{SVK}, t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{y}_0, \mathbf{y}, sig)$, $\mathcal{B}$ still rejects $C$ – as in Game 3 – in the event that $\mathsf{SVK} \notin \{\mathsf{SVK}^{(1)}, \ldots, \mathsf{SVK}^{(Q_E)}\}$ and $t = (t_{\mathsf{c}}, \mathsf{SVK})$ is a lossy tag. Otherwise the tag $t = (t_{\mathsf{c}}, \mathsf{SVK})$ must be injective. This allows $\mathcal{B}$ to compute the pre-image $(\mathbf{x}, \mathbf{e}) = \Pi^{\mathsf{ABM}}.\mathsf{Invert}(ik, t, \mathbf{y})$. Having obtained $(\mathbf{x}, \mathbf{e})$, $\mathcal{B}$ returns $\perp$ if $(\mathbf{x}, \mathbf{e}) \notin \mathsf{Dom}^D_\lambda$ or

$\|\mathbf{y}_0 - \mathbf{A}\cdot\mathbf{x}\| > \sigma_e\sqrt{2m}$. Otherwise, it computes $\mathbf{k}^{sym} = h(\mathbf{x})$ and returns $\mathsf{Msg} = \mathbf{c}_0 - \mathbf{k}^{sym} \in \mathbb{Z}_q^{\ell_0}$. It is easy to see that the adversary's view is the same as in Game 3 since $\mathcal{B}$ does not reject any ciphertext that would not have been rejected in Game 3. We thus have $\Pr[S_4] = \Pr[S_3]$.

**Game** 5: In this game, we modify the key generation phase. Instead of choosing the evaluation/inversion keys $(ek', ik') \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{IGen}(1^\lambda)$ as in an injective function, $\mathcal{D}$ generates $(ek', \perp) \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{LGen}(1^\lambda)$ as the evaluation key of a lossy function. By the properties of the lossy trapdoor function of Section 3.1, we know that this change will not be noticeable to $\mathcal{A}$ under the $\mathsf{LWE}$ assumption. We have

$$|\Pr[S_5] - \Pr[S_4]| \leq \mathbf{Adv}^{\mathcal{A},\mathsf{LTF},\mathsf{ind}}(\lambda) \leq n \cdot \mathbf{Adv}^{\mathsf{lwe}}_{\mathsf{LTF},\mathcal{D}}(\lambda).$$

**Game** 6: We modify the preparation stage of the challenger. Instead of deriving the symmetric encryption keys $\mathbf{k}^{sym,(i)} = h(\mathbf{x}^{(i)})$ for each $i \in [Q_E]$, $\mathcal{B}$ samples them as uniformly random one-time keys $\mathbf{k}^{sym,(i)} \hookleftarrow U(\mathbb{Z}_q^{\ell_0})$ for each $i \in [Q_E]$.

We claim that this change leaves $\mathcal{A}$'s view statistically unchanged. Indeed, for lossy tags $t^{(i)} = (t_{\mathsf{c}}^{(i)}, \mathsf{SVK}^{(i)})$, conditionally on

$$\mathbf{y}_0^{(i)} = \Pi^{\mathsf{LTF}}.\mathsf{Eval}\big(ek', (\mathbf{x}^{(i)}, \mathbf{e}_0^{(i)})\big), \qquad \mathbf{y}^{(i)} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}\big(ek, t^{(i)}, (\mathbf{x}^{(i)}, \mathbf{e}^{(i)})\big),$$

the input $\mathbf{x}^{(i)}$ retains at least $n$ bits of min-entropy. More precisely, Lemma 16 implies

$$H_\infty(\mathbf{x}^{(i)} \mid ek, ek', \mathbf{y}_0, \mathbf{y}) \geq n \cdot \log\sigma_x - 2 - \ell \cdot \log q > \Omega(n \cdot \log n) \tag{12}$$

Lemma 1 implies that we can bound the statistical distance as

$$\Delta\big((ek, ek', t^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}, h(\mathbf{x}^{(i)})), (ek, ek', t^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}, U(\mathbb{Z}_q^{\ell_0}))\big) \leq \frac{1}{2^{(n\cdot\log\sigma_x - (\ell+\ell_0)\cdot\log q)/2}}$$

which is smaller than $2^{-\lambda}$ as long as $n \cdot \log\sigma_x$ is sufficiently large. By repeating the same argument for each challenge ciphertext, we obtain $|\Pr[S_6] - \Pr[S_5]| \leq Q_E/2^\lambda$.

In Game 6, we finally remark that $\mathbf{c}_0^{(i)} = \mathsf{Msg}_b^{(i)} + \mathbf{k}^{sym,(i)} \in \mathbb{Z}_q^{\ell_0}$ perfectly hide the underlying messages in all encryption queries since the one-time symmetric keys act as one-time pads. We conclude that $b \in \{0,1\}$ is independent of $\mathcal{A}$'s view, so that $\Pr[S_6] = 1/2$. □

### C.3 Tightly Secure One-Time Signatures from the SIS Assumption

In this section, we describe a one-time signature with a tight security proof under the $\mathsf{SIS}$ assumption in the multi-user setting.

**Definition 12.** *Let $m, n, q, \beta$ be functions of a parameter $1^\lambda$. The Short Integer Solution problem $\mathsf{SIS}_{m,q,\beta}$ is as follows: Given $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, find a non-zero $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{x}^\top \cdot \mathbf{A} = \mathbf{0}^{1\times n}$ and $0 < \|\mathbf{x}\| \leq \beta$.*

Tightly secure signatures based on the SIS assumption (and hence the LWE assumption) were previously described [19] in the single-user setting. In the multi-user setting, we give a simple construction, which is inspired from a SIS-based signature due to Böhl *et al.* [20]. In particular, it combines a weakly secure one-time signature (i.e., which is only secure against non-adaptive attacks, where the adversary chooses the messages to be signed before seeing the public key) with a chameleon hash function suggested in [35].

We assume public parameters $\Gamma$ which contain random matrices $\mathbf{A}_0 \hookleftarrow U(\mathbb{Z}_q^{m/2\times n})$, $\mathbf{B}_0, \mathbf{B}_1 \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{k\times n})$, where $m > 4n\log q$ and $k = n\lceil \log q \rceil$.

$\mathcal{G}(\Gamma, 1^\lambda)$**:** Run $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{GenTrap}(\mathbf{A}_0, \mathbf{I}_n, \gamma)$ and choose $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$. Output $(\mathsf{SVK}, \mathsf{SSK})$, where $\mathsf{SVK} = (\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^n$ and $\mathsf{SSK} = \mathbf{R} \in \mathbb{Z}^{m/2\times m/2}$.

$\mathcal{S}(\mathsf{SSK}, \mathbf{m})$**:** To sign $\mathbf{m} \in \{0,1\}^m$, using $\mathsf{SSK} = \mathbf{R}$, do the following:

1. Compute a chameleon hash $\mathbf{c}_M^\top = \mathbf{m}^\top \cdot \mathbf{B}_0 + \mathbf{r}^\top \mathbf{B}_1 \bmod q$, using a short Gaussian vector $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^m,\gamma}$.
2. Define $\mathbf{u}_M^\top = \mathbf{u}^\top + \mathbf{g}^{-1}(\mathbf{c}_M)^\top \cdot \mathbf{D} \bmod q$, where $\mathbf{g}^{-1}(\mathbf{c}_M) \in \{0,1\}^k$ is the binary decomposition of $\mathbf{c}_M \in \mathbb{Z}_q^n$.
3. Using $\mathsf{SSK} = \mathbf{R}$, sample $\mathbf{v} \hookleftarrow D_{\Lambda_q^{\mathbf{u}_M}(\mathbf{A}),\gamma}$ to obtain a $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\|_2 \leq \gamma\sqrt{m}$ and

$$\mathbf{v}^\top \cdot \mathbf{A} = \mathbf{u}^\top + \mathbf{g}^{-1}\big(\mathbf{m}^\top \cdot \mathbf{B}_0 + \mathbf{r}^\top \cdot \mathbf{B}_1\big)^\top \cdot \mathbf{D} \mod q \tag{13}$$

and output $sig = (\mathbf{r}, \mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}^m$.

$\mathcal{V}(\mathsf{SVK}, \mathbf{m}, sig = (\mathbf{r}, \mathbf{v}))$**:** Given a candidate signature $sig = (\mathbf{r}, \mathbf{v})$ and $\mathbf{m} \in \{0,1\}^m$, return 1 if and only if $\|\mathbf{r}\|_2 \leq \gamma\sqrt{m}$, $\|\mathbf{v}\|_2 \leq \gamma\sqrt{m}$ and equality (13) is satisfied.

One disadvantage of the above construction is its long public key, which contains $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$. In our CCA2-secure encryption scheme, it requires to introduce a full $n \times m$ matrix in the ciphertext. Ideally, one would clearly prefer verification keys SVK that live in $\mathbb{Z}_q^n$. In standard lattices, however, we are not aware of any one-time signature with $O(n)$-size verification key, let alone with tight security. We leave it as an interesting open problem.

**Theorem 6.** *The above one-time signature is tightly secure under the* SIS *assumption.*

*Proof.* Assuming a forger $\mathcal{A}$ against the signature scheme in the game of Definition 6, we build a SIS solver $\mathcal{B}$ which takes as input a matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m\times n}$ and finds an integer vector $\mathbf{e} \in \mathbb{Z}^m$ of $\Lambda_q^\perp(\bar{\mathbf{A}})$ of norm $\|\mathbf{e}\|_2 \leq \beta$.

The reduction distinguishes two kinds of attacks.

**Type I attacks:** The adversary outputs $(i^\star, \mathbf{m}^\star, sig^\star = (\mathbf{r}^\star, \mathbf{v}^\star))$ such that

$$\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1 = \mathbf{m}^{(i^\star)\top} \cdot \mathbf{B}_0 + \mathbf{r}^{(i^\star)\top} \cdot \mathbf{B}_1 \mod q,$$

where $\big(\mathbf{m}^{(i^\star)}, sig^{(i^\star)} = (\mathbf{r}^{(i^\star)}, \mathbf{v}^{(i^\star)})\big)$ denotes the message-signature pair obtained by $\mathcal{A}$ for $\mathsf{SVK}^{(i^\star)} = (\mathbf{A}^{(i^\star)}, \mathbf{u}^{(i^\star)})$.

**Type II attacks:** The adversary outputs $(i^\star, \mathbf{m}^\star, sig^\star = (\mathbf{r}^\star, \mathbf{v}^\star))$ such that

$$\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1 \neq \mathbf{m}^{(i^\star)\top} \cdot \mathbf{B}_0 + \mathbf{r}^{(i^\star)\top} \cdot \mathbf{B}_1 \mod q. \tag{14}$$

Type I attacks yield a collision on the chameleon hash function, which has a tight security proof under $\mathsf{SIS}$ assumption [35]. We thus focus on Type II forgeries.

In order to build public parameters and $N$ verification keys $\mathsf{SVK}_i = \big(\mathbf{A}_i, \mathbf{u}_i\big)$, the reduction $\mathcal{B}$ first samples a random matrix $\mathbf{R}_D \hookleftarrow U(\{-1,1\}^{k\times m})$ and computes $\mathbf{D} = \mathbf{R}_D \cdot \bar{\mathbf{A}} \in \mathbb{Z}_q^{k\times n}$. It also chooses $\mathbf{B}_0 \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ at random and generates $\mathbf{B}_1 \in \mathbb{Z}_q^{m\times n}$ as a statistically uniform matrix for which it knows a trapdoor $\mathbf{T}_{\mathbf{B}_1}$. For each $i \in [N]$, $\mathcal{B}$ generates the $i$-th verification key $\mathsf{SVK}_i$ by sampling $\mathbf{R}_i \hookleftarrow U(\{-1,1\}^{m\times m})$, $\mathbf{v}_i \hookleftarrow D_{\mathbb{Z}^m,\gamma}$, $\mathbf{h}_i \hookleftarrow U(\mathbb{Z}_q^n)$. It sets $\mathbf{A}_i = \mathbf{R}_i \cdot \bar{\mathbf{A}} \in \mathbb{Z}_q^{m\times n}$ and $\mathbf{u}_i^\top = \mathbf{v}_i^\top \cdot \mathbf{A}_i - \mathbf{g}^{-1}(\mathbf{h}_i)^\top \cdot \mathbf{D} \bmod q$. The public keys $\mathsf{SVK}_i = (\mathbf{A}_i, \mathbf{u}_i)$ are given to the adversary $\mathcal{A}$.

When $\mathcal{A}$ makes a signing query $(i, \mathbf{m}_i)$, $\mathcal{B}$ uses the trapdoor $\mathbf{T}_{\mathbf{B}_1}$ to sample a vector $\mathbf{r}_i \in \mathbb{Z}^m$ of norm $\|\mathbf{r}_i\|_2 \leq \gamma\sqrt{m}$ such that $\mathbf{r}_i^\top \cdot \mathbf{B}_1 = \mathbf{h}_i - \mathbf{m}_i^\top \cdot \mathbf{B}_0 \mod q$ and returns $(\mathbf{v}_i, \mathbf{r}_i)$. By hypothesis, $\mathcal{A}$ outputs $(i^\star, \mathbf{m}^\star, sig^\star = (\mathbf{r}^\star, \mathbf{v}^\star))$ satisfying (14) Due to the injectivity of $\mathbf{g}^{-1}(\cdot)$, we know that

$$\mathbf{g}^{-1}(\mathbf{h}_{i^\star}) \neq \mathbf{g}^{-1}(\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1),$$

Since the verification equation $\mathbf{v}^{\star\top} \cdot \mathbf{A}_{i^\star} = \mathbf{u}_i^\top + \mathbf{g}^{-1}(\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1)^\top \cdot \mathbf{D}$ can be written

$$\begin{aligned}\mathbf{v}^{\star\top} \cdot \mathbf{R}_{i^\star} \cdot \bar{\mathbf{A}} = \big(\mathbf{v}_i^\top \cdot \mathbf{R}_{i^\star} - \mathbf{g}^{-1}(\mathbf{h}_{i^\star})^\top \cdot \mathbf{R}_D\big) \cdot \bar{\mathbf{A}} \\ + \mathbf{g}^{-1}(\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1) \cdot \mathbf{R}_D \cdot \bar{\mathbf{A}},\end{aligned} \tag{15}$$

we obtain

$$(\mathbf{v}^\star - \mathbf{v}_i)^\top \cdot \mathbf{R}_{i^\star} \cdot \bar{\mathbf{A}} = \big(\mathbf{g}^{-1}(\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1) - \mathbf{g}^{-1}(\mathbf{h}_{i^\star})\big) \cdot \mathbf{R}_D \cdot \bar{\mathbf{A}},$$

so that

$$\mathbf{w}^\top = (\mathbf{v}^\star - \mathbf{v}_i)^\top \cdot \mathbf{R}_{i^\star} + \big(\mathbf{g}^{-1}(\mathbf{h}_{i^\star}) - \mathbf{g}^{-1}(\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1)\big) \cdot \mathbf{R}_D \in \mathbb{Z}^{1\times m}$$

is in $\Lambda^\perp(\bar{\mathbf{A}})$ and has norm $\|\mathbf{w}\|_2 \leq m(1+2\gamma)$. Moreover, we argue that it is non-zero w.h.p. Since $\mathbf{g}^{-1}(\mathbf{h}_{i^\star}) \neq \mathbf{g}^{-1}(\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1)$, we must have $\mathbf{v}^\star \neq \mathbf{v}_i$ over $\mathbb{Z}$ unless the row vector

$$\big(\mathbf{g}^{-1}(\mathbf{m}^{\star\top} \cdot \mathbf{B}_0 + \mathbf{r}^{\star\top} \cdot \mathbf{B}_1) - \mathbf{g}^{-1}(\mathbf{h}_{i^\star})\big) \cdot \mathbf{R}_D$$

is itself a non-zero vector of $\Lambda^\perp(\bar{\mathbf{A}})$. Since each row of $\mathbf{R}_{i^\star} \in \{-1,1\}^{m\times m}$ has at least $m - n\log q > m/2$ bits of min-entropy conditionally on $\mathbf{A}_{i^\star} = \mathbf{R}_{i^\star} \cdot \bar{\mathbf{A}}$, each coordinate of $(\mathbf{v}^\star - \mathbf{v}_i)^\top \cdot \mathbf{R}_{i^\star}$ is statistically unpredictable from $\mathcal{A}$'s view and we can only have $\mathbf{w} = \mathbf{0}^m$ with probability smaller than $2^{-n}$. □

# D Deferred Proofs for the Scheme in Section 4

## D.1 Proof of Theorem 3

*Proof.* The proof proceeds with a sequence of games. For each $i$, we denote by $S_i$ the event that the adversary $\mathcal{A}$ wins (by outputting $b' \in \{0,1\}$ such that $b' = b$) in Game $i$.

**Game** 0: This is the real game. It begins with the challenger $\mathcal{B}$ choosing a bit $b \hookleftarrow U(\{0,1\})$ and generating $PK = (ek', ek, h)$ while keeping $SK = (ik, ik')$ to itself. Before starting its interaction with $\mathcal{A}$, $\mathcal{B}$ samples

$$(\mathbf{x}^{(1)}, \mathbf{e}_0^{(1)}, \mathbf{e}^{(1)}), \ldots, (\mathbf{x}^{(N)}, \mathbf{e}_0^{(N)}, \mathbf{e}^{(N)}) \hookleftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_x} \times D_{\mathbb{Z}^{2m},\sigma_e} \times D_{\mathbb{Z}^{2m},\sigma_e}.$$

For each $i \in [N]$, it computes $\mathbf{y}_0^{(i)} = \Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek', (\mathbf{x}^{(i)}, \mathbf{e}_0^{(i)}))$ and derives symmetric keys $(\mathbf{k}^{sym,(i)}, \mathbf{k}^{mac,(i)}) = h(\mathbf{x}^{(i)})$. Next, it defines $t_{\mathsf{a}}^{(i)} = \mathbf{y}_0^{(i)}$ and $t^{(i)} = (t_{\mathsf{c}}^{(i)}, t_{\mathsf{a}}^{(i)})$ for a randomly chosen core tag $t_{\mathsf{c}}^{(i)} \hookleftarrow U(\mathcal{T}_{\mathsf{c}})$ before computing $\mathbf{y}^{(i)} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}\big(ek, t^{(i)}, \mathbf{x}^{(i)}\big)$.

In response to its challenge query, the adversary $\mathcal{A}$ obtains encryptions $(\mathbf{C}_1, \ldots, \mathbf{C}_N)$ of messages $\mathbf{Msg} = (\mathsf{Msg}_1, \ldots, \mathsf{Msg}_N) \leftarrow \mathcal{M}$, where each ciphertext $\mathbf{C}_i = (t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{c}_1^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)})$ is obtained as

$$\mathbf{c}_0^{(i)} = \mathsf{Msg}_i + \mathbf{k}^{sym,(i)} \mod q, \qquad \mathbf{y}_0^{(i)} = \Pi^{\mathsf{LTF}}.\mathsf{Eval}\big(ek', (\mathbf{x}^{(i)}, \mathbf{e}_0^{(i)}\big)$$
$$\mathbf{y}^{(i)} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}\big(ek, t^{(i)}, (\mathbf{x}^{(i)}, \mathbf{e}^{(i)})\big), \qquad \mathbf{c}_1^{(i)} = \mathsf{MAC.Sig}(\mathbf{k}^{mac,(i)}, (\mathbf{y}^{(i)}, \mathbf{c}_0^{(i)}))$$

Then, $\mathcal{A}$ chooses a subset $I \subset [N]$ for which it receives $\{(\mathsf{Msg}_i, \mathbf{y}^{(i)})\}_{i \in I}$. At this point, the challenger $\mathcal{D}$ re-samples

$$(\mathsf{Msg}'_1, \ldots, \mathsf{Msg}'_N) \hookleftarrow \mathsf{ReSamp}_{\mathcal{M}}(I, \{(i, \mathsf{Msg}_i)\}_{i \in I}).$$

If $b = 1$, $\mathcal{D}$ reveals the real messages $\{\mathsf{Msg}_i\}_{[n]\setminus I}$ to $\mathcal{A}$. If $b = 0$, it rather returns $\{\mathsf{Msg}'_i\}_{[n]\setminus I}$ to $\mathcal{A}$. All decryption queries are answered using the inversion trapdoor $ik'$ of $\Pi^{\mathsf{LTF}}$. When $\mathcal{A}$ terminates, it outputs a bit $b \in \{0,1\}$. If we call $S_0$ the event that $b' = b$, we have $\mathbf{Adv}_{\mathcal{A}}^{\text{IND-SO-CCA2}}(\lambda) := |\Pr[S_0] - 1/2|$.

**Game** 1: We modify the generation of the challenge ciphertexts $(\mathbf{C}_1, \ldots, \mathbf{C}_N)$. For each $i \in [N]$, the challenger $\mathcal{B}$ encrypts $\mathbf{C}_i = (t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{c}_1^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)})$ on a lossy tag $t^{(i)} = (t_{\mathsf{c}}^{(i)}, t_{\mathsf{a}}^{(i)})$. In the instantiation based on our ABM-LTF, $t_{\mathsf{c}}^{(i)}$ is thus computed as a pseudo-random function $t_{\mathsf{c}}^{(i)} = \mathsf{PRF}(K, t_{\mathsf{a}}^{(i)})$. The indistinguishability of our ABM-LTF ensures that $\Pr[S_1]$ remains negligibly far apart from $\Pr[S_0]$. There exist efficient distinguishers $\mathcal{D}_1$ and $\mathcal{D}_2$ such that $|\Pr[S_1] - \Pr[S_0]| \leq 2n \cdot \mathbf{Adv}_{\ell,m,q,\chi}^{\mathcal{D}_1,\mathsf{lwe}}(\lambda) + \mathbf{Adv}_{Q_E}^{\mathcal{D}_2,\text{prf}}(\lambda) + \frac{1}{2^{\lambda-1}}$.

**Game** 2: We modify the decryption oracle. At each decryption query $\mathbf{C} = (t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0, \mathbf{y})$, $\mathcal{B}$ rejects $C$ if the corresponding tag $t = \big(t_{\mathsf{c}}, \mathbf{y}_0\big)$ is a lossy tag but $\mathbf{y}_0 \notin \{\mathbf{y}_0^{(1)}, \ldots, \mathbf{y}_0^{(N)}\}$. Game 2 is identical to Game 1 until the event $F_2$ that $\mathcal{B}$ rejects a ciphertext that would have deemed valid in Game 1. This event would break the evasiveness property of our ABM-LTF.

Indeed, the same argument as in the proof of Theorem 5 shows that any evasiveness adversary implies either an $\mathsf{LWE}_{\ell,m,q,\chi}$ distinguisher $\mathcal{D}_1$ or a PRF adversary $\mathcal{D}_2$ with non-negligible advantage after $N + Q_D$ evaluation queries. Lemma 13 implies

$$|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq n \cdot \mathbf{Adv}^{\mathcal{D}_1,\mathsf{lwe}}_{\ell,m,q,\chi}(\lambda) + \mathbf{Adv}^{\mathcal{D}_2,\mathrm{prf}}_{N+Q_D}(\lambda) + \frac{Q_D+1}{2^\lambda}.$$

From here on, we are done with relying on the pseudo-randomness of the PRF and we thus explicitly use the seed $K$ in subsequent games.

**Game** 3: In this game, we do not use the inversion trapdoor $ik'$ of $\Pi^{\mathsf{LTF}}$ any longer but rather use the trapdoor $ik$ of $\Pi^{\mathsf{ABM}}$ to answer decryption queries. For each decryption query $\mathbf{C} = (t_\mathsf{c}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0, \mathbf{y})$, $\mathcal{B}$ aborts (as in Game 2) if $\mathbf{y}_0 \notin \{\mathbf{y}_0^{(1)}, \ldots, \mathbf{y}_0^{(N)}\}$ and $t = (t_\mathsf{c}, \mathbf{y}_0)$ is a lossy tag. Otherwise, $\mathcal{B}$ distinguishes three kinds of queries:

a. If $\mathbf{y}_0 \notin \{\mathbf{y}_0^{(1)}, \ldots, \mathbf{y}_0^{(N)}\}$, the tag $t = (t_\mathsf{c}, \mathbf{y}_0)$ is injective as $C$ would have been rejected otherwise. This allows $\mathcal{B}$ to compute $(\mathbf{x}, \mathbf{e}) = \Pi^{\mathsf{ABM}}.\mathsf{Invert}(ik, t, \mathbf{y})$ and return $\perp$ in the following situations:
   - $\mathbf{y}$ is not in the range of $\Pi^{\mathsf{ABM}}.\mathsf{Eval}(ek, t, .)$;
   - $(\mathbf{x}, \mathbf{e}) \notin \mathsf{Dom}^D_\lambda$;
   - $\|\mathbf{y}_0 - \mathbf{A} \cdot \mathbf{x}\| > \sigma_e\sqrt{2m}$.

   If none of the above rejection rules applies, $\mathcal{B}$ derives $(\mathbf{k}^{sym}, \mathbf{k}^{mac}) = h(\mathbf{x})$ and returns $\perp$ if $\mathsf{MAC.Ver}(\mathbf{k}^{mac}, (\mathbf{y}, \mathbf{c}_0), \mathbf{c}_1) = 0$. If the MAC verifies, it returns $\mathsf{Msg} = \mathbf{c}_0 - \mathbf{k}^{sym} \in \mathbb{Z}_q^{\ell_0}$.

b. If $\mathbf{y}_0 = \mathbf{y}_0^{(i)}$, for some $i \in [N]$, but $t_c \neq t_\mathsf{c}^{(i)}$, the tag $t = (t_\mathsf{c}, \mathbf{y}_0^{(i)})$ is also injective since $t_\mathsf{c}^{(i)}$ is the only core tag that makes $t = (t_\mathsf{c}^{(i)}, \mathbf{y}_0^{(i)})$ lossy. This allows $\mathcal{B}$ to proceed by running $\Pi^{\mathsf{ABM}}.\mathsf{Invert}(ik, t, .)$ as in the previous case.

c. If there is an index $i \in [N]$ such that $\mathbf{y}_0 = \mathbf{y}_0^{(i)}$ and $t_c = t_\mathsf{c}^{(i)}$, $\mathcal{B}$ recalls the input $\mathbf{x}^{(i)}$ that was chosen in the preparation phase and returns $\perp$ if $\|\mathbf{y} - \mathbf{A}_t \cdot \mathbf{x}^{(i)}\| > \sigma_e\sqrt{2m}$. Otherwise, it recalls the symmetric key $(\mathbf{k}^{sym,(i)}, \mathbf{k}^{mac,(i)}) = h(\mathbf{x}^{(i)})$ that were derived from $\mathbf{x}^{(i)}$. As in Steps 3-4 of the actual decryption algorithm, $\mathcal{B}$ returns the plaintext $\mathsf{Msg} = \mathbf{c}_0 - \mathbf{k}^{sym,(i)} \in \mathbb{Z}_q^{\ell_0}$ if $\mathsf{MAC.Ver}(\mathbf{k}^{mac,(i)}, (\mathbf{y}, \mathbf{c}_0), \mathbf{c}_1) = 1$ and $\perp$ otherwise.

By inspection, it is easy to see that $\mathcal{A}$'s view is exactly as in Game 2, so that $\Pr[S_3] = \Pr[S_2]$.

**Game** 4: We modify the key generation phase and replace $ek'$ by the evaluation key of a lossy function. Instead of generating $(ek', ik') \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{IGen}(1^\lambda)$ as an injective function, $\mathcal{D}$ chooses it as $(ek', \perp) \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{LGen}(1^\lambda)$. Since the inversion key $ik'$ of $\Pi^{\mathsf{LTF}}$ was not used in Game 4, this change will not be noticeable to $\mathcal{A}$ under the $\mathsf{LWE}$ assumption. We have $|\Pr[S_5] - \Pr[S_4]| \leq n \cdot \mathbf{Adv}^{\mathsf{lwe}}(\lambda)$.

For all indexes $i \in [N]$, due to the lossiness of all challenge ciphertexts, Lemma 16 tell us that

$$H_\infty(\mathbf{x}^{(i)} \mid ek, ek', \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}) \geq n \cdot \log \sigma_x - 2 - \ell \cdot \log q \geq \Omega(n \cdot \log n), \tag{16}$$

By Lemma 1, this implies

$$\Delta\big((ek, ek', t^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}, h(\mathbf{x}^{(i)})), (ek, ek', t^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}, U(\mathbb{Z}_q^{\ell_0} \times \mathbb{Z}_q^{\ell_1}))\big) \leq \frac{1}{2^{(n \cdot \log \sigma_x - (\ell + \ell_0 + \ell_1) \cdot \log q)/2}} \tag{17}$$

which is smaller than $2^{-\lambda}$ when $n \cdot \log \sigma_x$ is sufficiently large.

**Game** 5: We change the treatment of decryption queries $\mathbf{C} = (t_{\mathsf{c}}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0, \mathbf{y})$ corresponding to cases a or b of Game 3. In both cases, $\mathcal{B}$ can compute $(\mathbf{x}, \mathbf{e}) = \Pi^{\mathsf{ABM}}.\mathsf{Invert}(ik, t, \mathbf{y})$ since the tag $t = (t_{\mathsf{c}}, \mathbf{y}_0)$ is injective. The modification is that $\mathcal{B}$ now rejects $C$ if $\mathbf{x} = \mathbf{x}^{(i)}$ for some $i \in [N]$. We claim that, except with negligible probability, $\mathcal{B}$ does not reject any ciphertext that would not be rejected in Game 4.

To see this, we note that, for lossy tags $t^{(i)}$,

$$\mathbf{y}_0^{(i)} = \Pi^{\mathsf{LTF}}.\mathsf{Eval}\big(ek', (\mathbf{x}^{(i)}, \mathbf{e}_0^{(i)})\big), \qquad \mathbf{y}^{(i)} = \Pi^{\mathsf{ABM}}.\mathsf{Eval}\big(ek, t^{(i)}, (\mathbf{x}^{(i)}, \mathbf{e}^{(i)})\big)$$

jointly leak a limited number bits about $\mathbf{x}^{(i)}$, which retains at least $\lambda$ bits of min-entropy due to (16). Taking a union bound over all decryption queries and all indices $i \in [N]$, the probability to reject a ciphertext that would not be rejected in Game 4 is at most $Q_D \cdot N/2^{\lambda}$, which implies $|\Pr[S_5] - \Pr[S_4]| \leq Q_D \cdot N/2^{\lambda}$.

**Game** 6: We modify the decryption oracle and let $\mathcal{B}$ reject all potentially harmful decryption queries, which could possibly reveal $h(\mathbf{x}^{(i)})$ for unopened ciphertexts. Namely, the challenger $\mathcal{B}$ introduces two rejection rules:

A. It rejects queries $(t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{c}_1, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)})$ such that $\mathbf{c}_1 \neq \mathbf{c}_1^{(i)}$, for some $i \in [N]$.
B. It rejects all decryption queries of the form $(t_{\mathsf{c}}^{(i)}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0^{(i)}, \mathbf{y})$, for some $i \in [N]$, such that $(\mathbf{y}, \mathbf{c}_0) \neq (\mathbf{y}^{(i)}, \mathbf{c}_0^{(i)})$ and which either
   - involve the index $i \in [N] \setminus I$ of an unopened ciphertext $\mathbf{C}_i$ after the corruption query;
   - occur between the challenge query and the corruption query (and thus before the adversary obtains the random coins of $\mathbf{C}_i$).

Due to the uniqueness property of the MAC, we easily see that rule A does not change anything to the adversary's view. So, Game 6 only departs from Game 5 when rule B rejects a ciphertext that would not be rejected in Game 5. If we call $F_6$ the latter event, Lemma shows that $\Pr[F_6] \leq N \cdot \big(\mathbf{Adv}_{\mathcal{B}}^{\mathrm{mac},Q_D}(\lambda) + \frac{1}{2^{\lambda}}\big)$.

In Game 6, we argue that $\Pr[S_6] = 1/2 + N/2^{\lambda}$, so that the adversary's advantage is statistically negligible. For each decryption query of the form $(t_{\mathsf{c}}^{(i)}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0^{(i)}, \mathbf{y})$, with $(\mathbf{y}, \mathbf{c}_0) \neq (\mathbf{y}^{(i)}, \mathbf{c}_0^{(i)})$ and for some $i \in [N] \setminus I$, the decryption oracle returns $\perp$. As for decryption queries $(t_{\mathsf{c}}, \mathbf{c}_0^{(i)}, \mathbf{c}_1, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)})$ such that $t_{\mathsf{c}} \neq t_{\mathsf{c}}^{(i)}$, they are also rejected when the inversion algorithm $\Pi^{\mathsf{ABM}}.\mathsf{Invert}(tk, (t_{\mathsf{c}}, \mathbf{y}_0^{(i)}), \mathbf{y}^{(i)})$ outputs $(\mathbf{x}^{(i)}, \cdot)$ due to the change introduced in Game 5. For all unopened ciphertexts $\{\mathbf{C}_i\}_{i \in [N] \setminus I}$, $\mathcal{A}$ never tricks the decryption oracle into revealing $h(\mathbf{x}^{(i)})$. Conditionally on the adversary's view, $\{\mathbf{k}^{sym,(i)}\}_{i \in [N] \setminus I}$ are thus within distance $2^{-\lambda}$ from the uniform distribution $U(\mathbb{Z}_q^{\ell_0})$ and they statistically hide $\{\mathsf{Msg}_i\}_{i \in [N] \setminus I}$. We conclude that $b \in \{0, 1\}$ is statistically independent of $\mathcal{A}$'s view, as claimed. $\square$

**Lemma 18.** *In Game* 6*, event* $F_6$ *occurs with negligible probability if* MAC *is a strongly secure one-time MAC. We have* $\Pr[F_6] \leq N \cdot \big(\mathbf{Adv}_{\mathcal{B}}^{\mathrm{mac},Q_D}(\lambda) + \frac{1}{2^\lambda}\big)$*, where* $\mathbf{Adv}_{\mathcal{B}}^{\mathrm{mac},Q_D}(\lambda)$ *denotes* $\mathcal{B}$*'s probability to break the strong unforgeability of the MAC after* $Q_D$ *verification queries.*

*Proof.* By hypothesis, we know that, at some point of Game 6, rule B will cause $\mathcal{B}$ to reject a ciphertext $\mathbf{C} = \big(t_{\mathsf{c}}^{(i)}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}\big)$ that would not have been rejected in Game 5. We build a MAC forger $\mathcal{B}$ with advantage $(\Pr[F_6] - 2^{-\lambda})/N$.

To this end, we need to consider two modifications of Game 6, which we call Game 6.1 and Game 6.2.

**Game 6.1:** This game is like Game 6 except that the challenger $\mathcal{B}$ randomly chooses $i^\star \hookleftarrow U([N])$ as a prediction that event $F_6$ will occur for the $i^\star$-th challenge ciphertext $\mathbf{C}_{i^\star}$. If $\mathcal{A}$ decides to open $\mathbf{C}_{i^\star}$, $\mathcal{B}$ aborts since its guess $i^\star \in [N]$ turns out to be wrong. If the first application of rule B involves an index $i \neq i^\star$, $\mathcal{B}$ also aborts for the same reason. Otherwise, rule B applies for the first time for the index $i^\star$. If we call $F_{6.1}$ this event, we have $\Pr[F_{6.1}] \geq \Pr[F_6]/N$ since $i^\star$ was chosen independently of $\mathcal{A}$'s view.

**Game 6.2:** We modify the preparation phase and replace the $i^\star$-th MAC key $\mathbf{k}^{mac,(i^\star)}$ by a uniformly random key $\mathbf{k}^{mac,\star} \hookleftarrow U(\mathbb{Z}_q^{\ell_1})$. Since $\mathbf{k}^{mac,(i^\star)}$ is statistically close to the uniform distribution for any unopened ciphertext, (17) implies $|\Pr[F_{6.2}] - \Pr[F_{6.1}]| \leq 2^{-\lambda}$ if $F_{6.2}$ denotes the counterpart of event $F_{6.1}$ in Game 6.2.

In Game 6.2, we can turn $\mathcal{B}$ into a MAC forger such that $\Pr[F_{6.2}] \geq \mathbf{Adv}_{\mathcal{B}}^{\mathrm{mac},Q_D}(\lambda)$.

This forger proceeds by initially choosing $i^\star \hookleftarrow U([N])$ as a predication of the index for which event $F_6$ will occur for the first time. It implicitly defines the $i^\star$-th MAC secret key $\mathbf{k}^{mac,(i^\star)}$ to be the key $\mathbf{k}^{mac,\star}$ of its MAC challenger. The remaining MAC keys $\{\mathbf{k}^{mac,(i)}\}_{i \neq i^\star}$ are chosen by computing $(\mathbf{k}^{sym,(i)}, \mathbf{k}^{mac,(i)}) = h(\mathbf{x}^{(i)})$ as in Game 6. In the challenge phase, $\mathcal{B}$ generates $(t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)})$, where $\mathbf{c}_0^{(i)} = \mathsf{Msg}_i + \mathbf{k}^{sym,(i)} \in \mathbb{Z}_q^{\ell_0}$, as in Game 6. The generation of $\mathbf{c}_1^{(i)}$ depends on the index $i \in [N]$ of the challenge ciphertext:

- If $i \neq i^\star$, $\mathcal{B}$ computes $\mathbf{c}_1^{(i)} = \mathsf{MAC.Sig}\big(\mathbf{k}^{mac,(i)}, (\mathbf{y}_0, \mathbf{c}_0^{(i)})\big)$ itself.
- If $i = i^\star$, $\mathcal{B}$ obtains $\mathbf{c}_1^{(i^\star)} = \mathsf{MAC.Sig}\big(\mathbf{k}^{mac,\star}, (\mathbf{y}^{(i^\star)}, \mathbf{c}_0^{(i^\star)})\big)$ from its MAC challenger.

In either case, $\mathcal{B}$ sets $\mathbf{C}_i = \big(t_{\mathsf{c}}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{c}_1^{(i)}, \mathbf{y}_0^{(i)}, \mathbf{y}^{(i)}\big)$ to $\mathcal{A}$.

At each decryption query of the form $C = \big(t_{\mathsf{c}}^{(i)}, \mathbf{c}_0, \mathbf{c}_1, \mathbf{y}_0^{(i)}, \mathbf{y}\big)$ for some $i \in [N]$, $\mathcal{B}$ queries the verification of $((\mathbf{y}, \mathbf{c}_0), \mathbf{c}_1)$ to its MAC challenger. If the response is 1, $\mathcal{B}$ aborts if $i \neq i^\star$. Otherwise (i.e., if $i = i^\star$), $\mathcal{B}$ outputs $((\mathbf{y}, \mathbf{c}_0), \mathbf{c}_1)$ and wins the game of Definition 7.

Since the choice of $i^\star \in [Q_E]$ is independent of $\mathcal{A}$'s view, we immediately obtain the claimed lower bound on $\mathcal{B}$'s advantage as a MAC forger. □

# E SIM-SO-CPA Security from DDH-based Lossy Trapdoor Functions

In this section, we show that lossy trapdoor functions based on the Decision Diffie-Hellman assumption (or, more generally, the matrix Diffie-Hellman assumption [43]) imply lossy en-

cryption schemes [17] which can be instantiated so as to admit an efficient opening algorithm. This provides evidence that, in the DDH setting, lossy trapdoor functions can also provide simulation-based selective opening security.

We illustrate this with the LTF of Freeman *et al.* [45], modulo a slight modification in its specification.

**Definition 13.** *Let $k, \ell \in \mathbb{N}$ with $\ell > k$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs matrices in $\mathbb{Z}_q^{\ell \times k}$ of full rank $k$ in polynomial time.*

We assume w.l.o.g. that the first $k$ rows of $\mathbf{A} \hookleftarrow U(\mathcal{D}_{\ell,k})$ form an invertible matrix over $\mathbb{Z}_q$.

In discrete-logarithm-hard groups, we rely on the matrix Diffie-Hellman assumption [43], which posits the infeasibility of distinguishing full-rank matrices from lower-rank matrices when they are given in the exponent.

**Definition 14 ($\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman Assumption $\mathcal{D}_{\ell,k}$-MDDH).** *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order $q$. We say that the $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman ($\mathcal{D}_{\ell,k}$-MDDH) assumption holds in $\mathbb{G}$ if, for any* ppt *distinguisher $\mathcal{A}$, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k}}(\mathcal{A}) := |\Pr[\mathcal{A}(\mathbb{G}, g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{w}}) = 1 \mid \mathbf{A} \hookleftarrow U(\mathcal{D}_{\ell,k}),\ \mathbf{w} \hookleftarrow U(\mathbb{Z}_q^k)] \\ - \Pr[\mathcal{A}(\mathbb{G}, g^{\mathbf{A}}, g^{\mathbf{u}}) = 1 \mid \mathbf{A} \hookleftarrow U(\mathcal{D}_{\ell,k}),\ \mathbf{u} \hookleftarrow U(\mathbb{Z}_q^\ell)]| \in \mathsf{negl}(\lambda).\end{aligned}$$

It is easy to see (see, e.g., [47]) that the above assumption implies the $Q$-fold $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman Assumption.

**Definition 15.** *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution and let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order $q$. The $Q$-fold $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman assumption holds in $\mathbb{G}$ says that, for any* ppt *distinguisher $\mathcal{A}$, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k}}(\mathcal{A}) := |\Pr[\mathcal{A}(\mathbb{G}, g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{W}}) = 1 \mid \mathbf{A} \hookleftarrow U(\mathcal{D}_{\ell,k}),\ \mathbf{W} \hookleftarrow U(\mathbb{Z}_q^{k\times Q})] \\ - \Pr[\mathcal{A}(\mathbb{G}, g^{\mathbf{A}}, g^{\mathbf{U}}) = 1 \mid \mathbf{A} \hookleftarrow U(\mathcal{D}_{\ell,k}),\ \mathbf{U} \hookleftarrow U(\mathbb{Z}_q^{\ell\times Q})]| \in \mathsf{negl}(\lambda).\end{aligned}$$

As in [45], we use a distribution $\mathcal{D}_{n,k} = \mathcal{U}_{n,k}$, which outputs uniformly random matrices $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times k})$, where $n$ denotes the input length. We rely on an $n$-fold $\mathcal{U}_{n,k}$-MDDH assumption saying that the distributions

$$\begin{aligned}D_0 &:= \{g, g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{W}} \mid \mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times k}), \mathbf{W} \hookleftarrow U(\mathbb{Z}_q^{k\times n})\} \\ D_1 &:= \{g, g^{\mathbf{A}}, g^{\mathbf{U}} \mid \mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times k}), \mathbf{U} \hookleftarrow U(\mathbb{Z}_q^{n\times n})\}\end{aligned}$$

are computationally indistinguishable. Naor and Segev [72] showed that this assumption is implied by the $k$-linear assumption [23] in $\mathbb{G}$, which is the infeasibility of distinguishing the distributions

$$D_0 := \{(g, g_1, \ldots, g_k, g_1^{a_1}, \ldots, g_k^{a_k}, g^{\sum_{i=1}^k a_i}) \mid g_1, \ldots, g_k \hookleftarrow U(\mathbb{G}),\ a_1, \ldots, a_k \hookleftarrow U(\mathbb{Z}_q)\},$$

$$D_1 := \{(g, g_1, \ldots, g_k, g_1^{a_1}, \ldots, g_k^{a_k}, g^b) \mid g_1, \ldots, g_k \hookleftarrow U(\mathbb{G}),\ a_1, \ldots, a_k, b \hookleftarrow U(\mathbb{Z}_q)\}$$

and which boils down to the DDH assumption for $k = 1$.

### E.1 Lossy Trapdoor Functions from Matrix Diffie-Hellman Assumptions

We describe a natural variant of a $\mathsf{DDH}$-like LTF suggested by Freeman *et al.* [45, Section 6]. The main difference between the construction below and the one of [45, Section 6] is the domain of the function family. While inputs consist of $n$-bit strings in [45], we use small-norm integer vectors sampled from a discrete Gaussian distribution: namely, we define $\mathsf{Dom}_\lambda^D = \{\mathbf{x} \in \mathbb{Z}^n \mid \|\mathbf{x}\| \leq \sigma\sqrt{n}\}$ and $\mathsf{Dom}_\lambda^E = \{\mathbf{x} \in \mathbb{Z}^n \mid \|\mathbf{x}\| \leq \gamma \cdot \sigma\sqrt{n}\}$, for some standard deviation $\sigma > 0$ and some parameter.

For $k = 1$, Peikert and Waters [76] previously suggested to use small-norm integer (rather than binary) vectors to improve the efficiency of their $\mathsf{DDH}$-based LTF. We further constrain them to have discrete Gaussian entries in order to efficiently open lossy ciphertexts in the resulting lossy encryption scheme.

**Injective Key generation.** $\mathsf{LTF.IGen}(1^\lambda)$ chooses a random $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times n})$, which is $\mathbb{Z}_q$-invertible w.h.p. (the process is repeated until an invertible matrix is found). It outputs the evaluation key $ek := g^{\mathbf{A}} \in \mathbb{G}^{n\times n}$ while the inversion key is $ik := \mathbf{A}^{-1} \in \mathbb{Z}_q^{n\times n}$.

**Lossy Key generation.** $\mathsf{LTF.LGen}(1^\lambda)$ chooses random matrices $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n\times k})$, $\mathbf{W} \hookleftarrow U(\mathbb{Z}_q^{k\times n})$ and defines $\mathbf{A} = \bar{\mathbf{A}} \cdot \mathbf{W} \in \mathbb{Z}_q^{n\times n}$. It outputs $ek := g^{\mathbf{A}} \in \mathbb{Z}_q^{n\times n}$ and $ik :=\perp$.

**Evaluation.** $\mathsf{LTF.Eval}(ek, \mathbf{x})$ takes as input $\mathbf{x} \in \mathsf{Dom}_\lambda^E$, which is mapped to the output $\mathbf{Y} = g^{\mathbf{A}\cdot\mathbf{x}} \in \mathbb{G}^n$.

**Inversion.** $\mathsf{LTF.Invert}(ik, \mathbf{Y})$ (with $\mathbf{Y} = g^{\mathbf{y}} \in \mathbb{G}^n$ for some $\mathbf{y} \in \mathbb{Z}_q^n$) uses the trapdoor $ik := \mathbf{A}^{-1} \in \mathbb{Z}_q^{n\times n}$ to compute

$$\mathbf{X} = \mathbf{Y}^{\mathbf{A}^{-1}} = g^{\mathbf{A}^{-1}\cdot\mathbf{y}} \in \mathbb{G}^n$$

From $\mathbf{X} \in \mathbb{G}^n$, compute $\mathbf{x} \in \mathsf{Dom}_\lambda^D$ such that $\mathbf{X} = g^{\mathbf{x}}$. If no such vector is found, output $\perp$. Otherwise, output $\mathbf{x}$.

It is easy to see that, in lossy mode, the output is completely determined by $\mathbf{W} \cdot \mathbf{x}$, so that the function has image size smaller than $k \cdot \log q$. The proof of the following Lemma is completely similar to that of [45, Lemma 6.3] and omitted.

**Lemma 19.** *Under the $n$-fold $\mathcal{U}_{n,k}$-$\mathsf{MDDH}$ assumption, the above function is an entropic lossy trapdoor function for which*

$$H_\infty(\mathbf{x} \mid g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{x}}) = H_\infty(\mathbf{x} \mid g^{\mathbf{A}}, g^{\mathbf{W}\cdot\mathbf{x}}) \geq H_\infty(\mathbf{x}) - k \cdot \log q.$$

Since a vector $\mathbf{x}$ sampled from the distribution $D_{\mathbb{Z}^n,\gamma}$ has at least $n$ bits of min-entropy, we have $H_\infty(\mathbf{x} \mid g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{x}}) \geq n - k \cdot \log q$.

### E.2 A Lossy Encryption Scheme with Efficient Opening from Matrix Diffie-Hellman Assumptions

For $k = 1$, the scheme coincides with the one of Bellare and Yilek [17, Section 5.2] instantiated with a specific universal hash functions, except that we modify the space of random coins.

**Par-Gen**$(1^\lambda)$**:** Choose a cyclic group $\mathbb{G}$ of prime order $q > 2^\lambda$ with $g \hookleftarrow U(\mathbb{G})$. The public parameters $\Gamma = \{\mathbb{G}, g, \ell_0, q, \mathcal{UH}, \sigma\}$ specify the message space $\mathsf{MsgSp} := \mathbb{Z}_q^{\ell_0}$, a universal hash function family $\mathcal{UH}$ and an integer $\gamma > 0$.

**Keygen**$(\Gamma)$**:** Let $\Pi^{\mathsf{LTF}} = (\mathsf{IGen}, \mathsf{LGen}, \mathsf{Eval}, \mathsf{Invert})$ be an instance of the LTF family recalled in Section E.1. Let $\mathsf{Dom}_\lambda^D = \{\mathbf{x} \in \mathbb{Z}^n \mid \|\mathbf{x}\| \leq \sigma\sqrt{n}\}$ be the inversion domain of the function. The public key is generated via the following steps.

1. Generate a pair $(ek, ik) \leftarrow \Pi^{\mathsf{LTF}}.\mathsf{IGen}(1^\lambda)$ for an injective function of the LTF family $\Pi^{\mathsf{LTF}}$.
2. Choose the key of a function $h : \mathsf{Dom}_\lambda \to \mathbb{Z}_q^{\ell_0}$ from the universal hash family, which consists of a random matrix $\mathbf{H}_{\mathcal{UH}} \hookleftarrow U(\mathbb{Z}_q^{\ell_0 \times n})$.

Output $(PK, SK)$, where $PK := \big(ek = g^{\mathbf{A}}, \mathbf{H}_{\mathcal{UH}}\big)$ and the underlying private key is $SK := ik = \mathbf{A}^{-1} \in \mathbb{Z}_q^{n \times n}$.

**Encrypt**$(PK, \mathsf{Msg})$**:** To encrypt $\mathsf{Msg} \in \mathbb{Z}_q^{\ell_0}$, do the following.

1. Choose $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^n, \sigma}$ and compute $\mathbf{Y} = \Pi^{\mathsf{LTF}}.\mathsf{Eval}(ek, \mathbf{x}) = g^{\mathbf{A} \cdot \mathbf{x}}$.
2. Compute $\mathbf{c}_0 = \mathsf{Msg} + \mathbf{H}_{\mathcal{UH}} \cdot \mathbf{x} \in \mathbb{Z}_q^{\ell_0}$.

Output the ciphertext $C = (\mathbf{Y}, \mathbf{c}_0)$.

**Decrypt**$(SK, C)$**:** To decrypt $C = (\mathbf{Y}, \mathbf{c}_0)$ using $SK = ik$, do the following.

1. Compute $\mathbf{x} = \Pi^{\mathsf{LTF}}.\mathsf{Invert}(ik, \mathbf{Y})$ and return $\perp$ if $\mathbf{x} \notin \mathsf{Dom}_\lambda^D$.
2. Return $\mathsf{Msg} = \mathbf{c}_0 - \mathbf{H}_{\mathcal{UH}} \cdot \mathbf{x} \in \mathbb{Z}_q^{\ell_0}$.

We now present the lossy key generation algorithm and the efficient opening procedure. We require that $n > 2(k + \ell_0) \cdot \lceil \log q \rceil$ so that the matrix $\mathbf{W} \in \mathbb{Z}_q^{k \times n}$ of the lossy mode can be generated with a Micciancio-Peikert trapdoor.

**Keygen**$(\Gamma, \mathsf{loss})$**:** Choose a random matrix $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n \times k})$. In order to generate $\mathbf{W} \in \mathbb{Z}_q^{k \times n}$, conduct the following steps.

1. Choose $\mathbf{C}_0 \hookleftarrow U\big(\mathbb{Z}_q^{\bar{n} \times \bar{\ell}}\big)$, where $\bar{\ell} = k + \ell_0$ and $\bar{n} = n - \bar{\ell} \cdot \lceil \log q \rceil$ which is used to run the $(\mathbf{C}, \mathbf{R}_{sim}) \leftarrow \mathsf{GenTrap}(\mathbf{C}_0, \mathbf{I}_{\bar{\ell}}, \sigma)$ algorithm of Lemma 8 to produce a statistically uniform matrix of the form

$$\mathbf{C}^\top = \left[\frac{\mathbf{C}_0}{-\mathbf{R}_{sim} \cdot \mathbf{C}_0 + \mathbf{G}_{sim}}\right] \in \mathbb{Z}_q^{n \times \bar{\ell}},$$

where $\mathbf{G}_{sim} \in \mathbb{Z}_q^{\bar{\ell} \cdot \lceil \log q \rceil \times \bar{\ell}}$ is the gadget matrix of [69] and a small-norm matrix $\mathbf{R}_{sim} \in \mathbb{Z}^{\bar{\ell} \cdot \lceil \log q \rceil \times \bar{n}}$. Then, parse $\mathbf{C} \in \mathbb{Z}_q^{\bar{\ell} \times n}$ as

$$\mathbf{C} = \left[\frac{\mathbf{W}}{\mathbf{H}_{\mathcal{UH}}}\right] \in \mathbb{Z}_q^{\bar{\ell} \times n},$$

where $\mathbf{W} \in \mathbb{Z}_q^{k \times n}$ and $\mathbf{H}_{\mathcal{UH}} \in \mathbb{Z}_q^{\ell_0 \times n}$.

2. Define $\mathbf{A} = \bar{\mathbf{A}} \cdot \mathbf{W} \in \mathbb{Z}_q^{n \times n}$ and compute $ek := g^{\mathbf{A}} \in \mathbb{G}^{n \times n}$.

Return $PK_{\mathsf{loss}} = (ek, \mathbf{H}_{\mathcal{UH}})$ and $SK_{\mathsf{loss}} = \big(\mathbf{R}_{sim}, \mathbf{C}_0, \bar{\mathbf{A}}\big)$.

**Opener**$(\Gamma, PK_{\mathsf{loss}}, SK_{\mathsf{loss}}, \mathsf{Msg}_0, \mathbf{x}, \mathsf{Msg}_1)$**:** Given $\mathbf{x} \in \mathsf{Dom}_\lambda^E$, parse the lossy secret key $SK_{\mathsf{loss}}$ as $\big(\mathbf{R}_{sim}, \mathbf{C}_0, \bar{\mathbf{A}}\big)$ and do the following.

1. Compute $\mathbf{c}_{\mathsf{LTF},\mathbf{x}} = \mathbf{W} \cdot \mathbf{x} \in \mathbb{Z}_q^k$ and $\mathbf{c}_{\mathbf{x}} = (\mathsf{Msg}_0 - \mathsf{Msg}_1) + \mathbf{H}_{\mathcal{UH}} \cdot \mathbf{x}$. Define

$$\mathbf{t}_{\mathbf{x}} = \left[ \begin{array}{c|c} \mathbf{c}_{\mathsf{LTF},\mathbf{x}}^\top & \mathbf{c}_{\mathbf{x}}^\top \end{array} \right]^\top \in \mathbb{Z}_q^{\bar{\ell}}.$$

2. Using the trapdoor $\mathbf{R}_{sim} \in \mathbb{Z}^{\bar{\ell} \cdot \lceil \log q \rceil \times \bar{n}}$, sample a small-norm vector $\mathbf{x}' \hookleftarrow D_{\Lambda_q^{\mathbf{t}_{\mathbf{x}}}(\mathbf{C}),\sigma}$ so as to have a short $\mathbf{x}' \in \mathbb{Z}^n$ such that

$$\left[\frac{\mathbf{W}}{\mathbf{H}_{\mathcal{UH}}}\right] \cdot \mathbf{x}' = \mathbf{t}_{\mathbf{x}} \mod q.$$

If $\mathbf{x}' \in \mathsf{Dom}_\lambda^D$, output $\mathbf{x}'$. Otherwise, repeat Step 2 until a suitable $\mathbf{x}'$ is found.

It is easy to see that, as long as **Opener** finds a suitable $\mathbf{x}'$ at Step 2, this vector satisfies $\mathsf{Encrypt}(PK_{\mathsf{loss}}, \mathsf{Msg}_1, \mathbf{x}') = \mathsf{Encrypt}(PK_{\mathsf{loss}}, \mathsf{Msg}_0, \mathbf{x})$. Moreover, the obtained vector $\mathbf{x}'$ has the required distribution since, by [48][Lemma 5.2], $\{\mathbf{C} \cdot \mathbf{x} \mid \mathbf{x} \hookleftarrow D_{\mathbb{Z}^n,\sigma}\}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{\bar{\ell}}$ and the two distributions

$$\{(\mathbf{x}', \mathbf{C} \cdot \mathbf{x}') \mid \mathbf{x}' \hookleftarrow D_{\mathbb{Z}^n,\sigma}\}, \quad \{(\mathbf{x}', \mathbf{C} \cdot \mathbf{x}') \mid \mathbf{t}_{\mathbf{x}} \hookleftarrow U(\mathbb{Z}_q^{\bar{\ell}}), \mathbf{x}' \hookleftarrow D_{\Lambda_q^{\mathbf{t}_{\mathbf{x}}}(\mathbf{C}),\sigma}\}$$

are statistically indistinguishable.

The above scheme thus provides an alternative construction with SIM-SO-CPA security under Matrix Diffie-Hellman assumptions. Solutions based on similar DDH-like assumptions were previously reported in [17, 12, 62]. The schemes of [62] feature short ciphertexts, but incur $O(|\mathsf{Msg}|^2)$ exponentiations to encrypt messages of bitlength $|\mathsf{Msg}|$. In comparison, our ciphertexts require $n > 2(k+\ell_0) \cdot \lceil \log q \rceil$ (which amounts to $2|\mathsf{Msg}|$ for $k = 1$) group elements and each encryption costs $O(\log \sigma \cdot |\mathsf{Msg}|^2)$ group operations.

While less efficient than the bit-wise DDH-based scheme of [17, Section 5.4] in terms of key sizes and computational overhead, we believe the above construction to be of interest. First, it shows that lattice trapdoors can come in handy to obtain simulation-based SOA security from traditional number-theoretic tools (i.e., which do not rely on lattice assumptions). Moreover, combining the above system with suitable pairing-based ABM-LTFs could open the way to new realizations providing SIM-SO-CCA security under discrete-logarithm-related assumptions.

# F Tighter Security Proof for the BLMR PRF

This section shows that the key-homomorphic PRF of Boneh *et al.* [25] enjoys a tighter security proof than previously known, under the same assumption as in [25]. By "tighter," we mean that the upper bound (19) on the adversary's advantage does not depend on the number of adversarial queries, except in a statistically negligible term.

**Definition 16 ([25]).** *Let $\lambda$ be a security parameter and let integers $k = k(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$. Let $\chi = \chi(\lambda)$ and $\eta = \eta(\lambda)$ be distributions over $\mathbb{Z}_q$. The Non-Uniform Learning-With-Errors ($\mathsf{NLWE}_{k,m,q,\chi,\eta}$) assumption posits that*

$$\mathbf{Adv}^{\mathcal{A},\mathsf{NLWE}}_{k,m,q,\chi,\eta}(\lambda) := \big|\Pr[\mathcal{A}(1^\lambda, \mathbf{A}, \mathbf{u}) = 1 \mid \mathbf{A} \hookleftarrow \eta^{m\times k}, \mathbf{u} \hookleftarrow U(\mathbb{Z}_q^m)] - \Pr[\mathcal{A}(1^\lambda, \mathbf{A}, \mathbf{A}\cdot\mathbf{s} + \mathbf{e}) = 1 \mid \mathbf{A} \hookleftarrow \eta^{m\times k}, \mathbf{s} \hookleftarrow U(\mathbb{Z}_q^k), \mathbf{e} \hookleftarrow \chi^m]\big|$$

*is a negligible function for any* ppt *algorithm $\mathcal{A}$.*

Boneh *et al.* [25] gave an advantage-preserving reduction from $\mathsf{LWE}$ to $\mathsf{NLWE}$ for several distributions $\eta$, called "coset-samplable," which include the binary uniform distribution $\eta_{\mathsf{bin}(k)} = U(\{0,1\})$. For this specific distribution, they showed that solving $\mathsf{NLWE}$ in dimension $k = n\lceil \log q\rceil$ is as hard as solving $\mathsf{LWE}$ in dimension $n$ for the same modulus $q$ (assuming that $2^{\lceil \log q\rceil}/q - 1$ is negligible, i.e., that $q$ is close to a power of 2).[10]

The BLMR PRF. For any $x \in \mathbb{Z}_q$, let $\lfloor x \rfloor_p := \lfloor (p/q)\cdot x \rfloor \in \mathbb{Z}_p$, the notation being extended to vectors in the obvious way. Boneh *et al.* [25] showed that, under the $\mathsf{NLWE}$ assumption, one can construct a PRF $F : \mathbb{Z}_q^m \times \{0,1\}^\kappa \to \mathbb{Z}_p^m$ keyed by a uniformly random vector $\mathbf{k} \hookleftarrow U(\mathbb{Z}_q^m)$, where $m = n\lceil \log q\rceil$, which maps a $\kappa$-bit input $\mathbf{x} = \mathbf{x}[1]\ldots\mathbf{x}[\kappa] \in \{0,1\}^\kappa$ to the output

$$F(\mathbf{k}, \mathbf{x}) = \left\lfloor \prod_{i=1}^{\kappa} \mathbf{A}_{\mathbf{x}[i]} \cdot \mathbf{k} \right\rfloor_p , \tag{18}$$

for distinct moduli $p, q$ such that $p|q$ and $\mathbf{A}_0, \mathbf{A}_1 \in \{0,1\}^{m\times m}$ are random $\mathbb{Z}_q$-invertible binary public matrices.

One advantage of this construction is that the matrices $\mathbf{A}_0, \mathbf{A}_1$ may be public. When the input $\mathbf{x}$ is fixed, a pre-processing phase allows hard-coding the product $\prod_{i=1}^{\kappa} \mathbf{A}_{\mathbf{x}[i]}$ into an $\mathsf{NC}^1$ circuit to be evaluated on an encrypted key $\mathbf{k}$ (note that the rounding step can be computed in constant depth).

Boneh *et al.* [25] proved the security of the above PRF family under the $\mathsf{NLWE}$ assumption via a reduction that loses a factor $Q$ between the advantage of the pseudorandomness adversary and the distinguishing advantage against the problem of Definition 16. Indeed, the proof of [25, Claim 5.4] uses an $\mathsf{NLWE}$ instance with $Q$ secrets, where $Q$ is the number of PRF evaluation queries. Analogously to $\mathsf{LWE}$, the only known reductions from $\mathsf{NLWE}$ to its multi-secret variants (where the secret $\mathbf{S}$ is a $k \times Q$ matrix) proceed via a hybrid argument over the columns of $\mathbf{S} \in \mathbb{Z}_q^{k\times Q}$, implying a multiplicative gap $\Omega(Q)$ in terms of concrete security.

In order to give a tighter reduction here, we need a variant of the Leftover Hash Lemma due to Dodis *et al.* [40], which is stated as in [1]

[10] Note that for inputs of length $k$, our modulus $q$ will be required to be of magnitude $2^{\Omega(k)}$. We may choose a $q$ that satisfies $2^{\lceil \log q\rceil}/q - 1 \leq 2^{-\Omega(k)}$.

**Lemma 20.** *Let $\mathcal{H} = \{h : X \to Y\}_{h\in\mathcal{H}}$ be a family of universal hash functions and $f : X \to Z$ be a function, for countable sets $X, Y, Z$. For any random variable $T$ taking values in $X$:*

$$\Delta\big((h, h(T), f(T)), (h, U(Y), f(T))\big) \leq \frac{1}{2} \cdot \sqrt{2^{-H_\infty(T)} \cdot |Y| \cdot |Z|}.$$

*More generally, let $(T_i)_{i\leq k}$ be independent random variables taking values in $X$. In this case, we have:*

$$\Delta\big((h, (h(T_i), f(T_i))_{i\leq k}), (h, (U(Y))^{(i)}, f(T_i))_{i\leq k}))\big) \leq \frac{k}{2} \cdot \sqrt{2^{-H_\infty(T)} \cdot |Y| \cdot |Z|}.$$

In the following, for any $\mathbf{x} = \mathbf{x}[1] \dots \mathbf{x}[\kappa] \in \{0,1\}^\kappa$ and any index $j \leq \kappa$, we let $\mathbf{x}|^j \in \{0,1\}^{\kappa-j+1}$ denote the sub-string comprised of the bits $\mathbf{x}[j], \dots, \mathbf{x}[\kappa]$ of $\mathbf{x}$.

As in the parameter choice recommended in [25], we may set $\kappa = n^\varepsilon / \log n$, $p = 2^{n^\varepsilon - \omega(\log n)}$ and choose $\chi = D_{\alpha q}$ with rejection of samples of magnitude $\geq B = \alpha q\sqrt{\lambda}$, for some $\alpha = 2^{-n^\varepsilon}$ with $0 < \varepsilon < 1$.

**Theorem 7.** *Assume that $q$ is odd. If $\alpha \leq 1/(4(m \log q + \lambda)\sqrt{\lambda q})$ and $\alpha \cdot (2m)^\kappa \cdot p \leq 2^{-\omega(\log n)}$, the BLMR PRF is secure under the* NLWE *assumption. Specifically, the advantage of any PRF distinguisher making at most $Q$ queries is smaller than*

$$\begin{aligned}\mathbf{Adv}^{\mathcal{D},\mathrm{prf}}_Q(\lambda) \leq\ & 2 \cdot (m \log q + \lambda) \cdot (\kappa + 1) \cdot \mathbf{Adv}^{\mathcal{D},\mathsf{nlwe}}_{m,2m,q,\chi,\eta}(\lambda) \\ & + 2^{\kappa+3} \cdot (\kappa + 1) \cdot m^\kappa \cdot \alpha \cdot p \cdot (m \log q + \lambda) + \frac{Q \cdot (\kappa + 1)}{2^{\lambda-1}}.\end{aligned} \tag{19}$$

*Proof.* To prove the result we consider a sequence of games. For each $j$, we call $W_j$ the event that the adversary $\mathcal{A}$ outputs 1 in Game $j$ for each $j \in \{0, \dots, \kappa + 1\}$. For convenience, we first describe Game $\kappa+1$ which corresponds to the adversary interacting with the real PRF.

**Game $\kappa + 1$:** In this game, the adversary interacts with the real function as defined by (18).

**Game $j$ $(0 \leq j \leq \kappa)$:** In this game, the adversary interacts with a "hybrid" function which, for each query $\mathbf{x} \in \{0,1\}^\kappa$, consists of a rounded product of

$$F^{(j)}(\mathbf{x}) = \left\lfloor \prod_{i=1}^{j-1} \mathbf{A}_{\mathbf{x}[i]} \cdot \mathbf{k}_{\mathbf{x}|^j} \right\rfloor_p,$$

where $\mathbf{k}_{\mathbf{x}|^j} = R(\mathbf{x}|^j) \in \mathbb{Z}_q^m$, for a truly uniform function $R : \{0,1\}^{\kappa-j+1} \to \mathbb{Z}_q^m$. The function $R : \{0,1\}^{\kappa-j+1} \to \mathbb{Z}_q^m$ can be lazily defined by choosing random outputs in $\mathbb{Z}_q^m$ as $\mathcal{A}$ makes queries and bookkeeping the queries and answers.

Note that, in Game 0, the adversary interacts with a truly random function

$$\begin{aligned} R_p : \{0,1\}^\kappa &\to \mathbb{Z}_p^m \\ \mathbf{x} &\to F^{(0)}(\mathbf{x}) = \lfloor R(\mathbf{x}) \rfloor_p, \end{aligned}$$

since $p|q$ and $R : \{0,1\}^\kappa \to \mathbb{Z}_q^m$ is itself truly random.

To prove the result, we show that, for each $j \in \{0, \dots, \kappa\}$, Game $j$ is computationally indistinguishable from Game $j + 1$. To this end, we define a modification of Game $j$.

**Game** $j.0$ $(1 \leq j \leq \kappa)$**:** In this game, we modify the way the adversary computes the hybrid function for each query $\mathbf{x} \in \{0,1\}^\kappa$. Before starting its interaction with the adversary, the challenger picks a uniformly random matrix $\mathbf{B} \hookleftarrow U\big(\mathbb{Z}_q^{2m \times (2m \log q + 2\lambda)}\big)$. At each query $\mathbf{x} \in \{0,1\}^\kappa$, the challenger looks up a list $\mathsf{L}$ (which is initially empty) and checks if it contains an entry $(\mathbf{x}|^{j+1}, \mathbf{t}_{\mathbf{x}|^j})$ for some vector $\mathbf{t}_{\mathbf{x}|^j} \in \mathbb{Z}_q^{2m}$. If so, it parses $\mathbf{t}_{\mathbf{x}|^j}$ as in (20) and uses it to compute $F^{(j)}(\mathbf{x})$ as in (21). If no such entry is in $\mathsf{L}$, the challenger samples $\mathbf{r}_{\mathbf{x}|^{j+1}} \hookleftarrow U(\{-1,1\}^{2m \log q + 2\lambda})$, computes $\mathbf{t}_{\mathbf{x}|^j} = \mathbf{B} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}} \in \mathbb{Z}_q^{2m}$, which it parses as

$$\mathbf{t}_{\mathbf{x}|^j} = \left[\frac{\mathbf{k}_{0||\mathbf{x}|^{j+1}}}{\mathbf{k}_{1||\mathbf{x}|^{j+1}}}\right], \tag{20}$$

where $\mathbf{k}_{0||\mathbf{x}|^{j+1}}, \mathbf{k}_{1||\mathbf{x}|^{j+1}} \in \mathbb{Z}_q^m$. Then, it stores $(\mathbf{x}|^{j+1}, \mathbf{t}_{\mathbf{x}|^j})$ in the list $\mathsf{L}$ and computes

$$F^{(j)}(\mathbf{x}) = \left\lfloor \prod_{i=1}^{j-1} \mathbf{A}_{\mathbf{x}[i]} \cdot \mathbf{k}_{\mathbf{x}[j]||\mathbf{x}|^{j+1}} \right\rfloor_p. \tag{21}$$

We remark that, since $\mathbf{B}$ is uniformly distributed over $\mathbb{Z}_q^{2m \times (2m \log q + 2\lambda)}$, the distribution of the vector $\mathbf{t}_{\mathbf{x}|^j} = \mathbf{B} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}$ is statistically close to the distribution $U(\mathbb{Z}_q^{2m})$ by the Leftover Hash Lemma.[11] By taking a union bound over the number $Q$ of evaluation queries, we obtain the inequality $|\Pr[W_j] - \Pr[W_{j.0}]| \leq Q \cdot 2^{-\lambda}$.

**Game** $j.1$ $(1 \leq j \leq \kappa)$**:** This game is identical to Game $j.0$ except that we replace the uniformly random matrix $\mathbf{B}$ by a matrix of the form

$$\mathbf{B} = \mathbf{A} \cdot \mathbf{S} + \mathbf{E} \ \in \mathbb{Z}_q^{2m \times (2m \log q + 2\lambda)},$$

where $\mathbf{A} = \left[\frac{\mathbf{A}_0}{\mathbf{A}_1}\right]$, $\mathbf{S} \hookleftarrow U\big(\mathbb{Z}_q^{m \times (2m \log q + 2\lambda)}\big)$ and $\mathbf{E} \hookleftarrow \chi^{2m \times (2m \log q + 2\lambda)}$.

Under the $\mathsf{NLWE}$ assumption, Game $j.1$ is indistinguishable from Game $j.0$: there exists an $\mathsf{NLWE}$ distinguisher such that $|\Pr[W_{j.0}] - \Pr[W_{j.1}]| \leq (2m \log q + 2\lambda) \cdot \mathbf{Adv}^{\mathcal{D},\mathsf{nlwe}}_{m,2m,q,\chi,\eta}(\lambda)$. We remark that in Game $j.1$, the vector $\mathbf{t}_{\mathbf{x}|^j}$ of (20) is of the form

$$\mathbf{t}_{\mathbf{x}|^j} = \left[\frac{\mathbf{k}_{0||\mathbf{x}|^{j+1}}}{\mathbf{k}_{1||\mathbf{x}|^{j+1}}}\right] = \left[\frac{\mathbf{A}_0 \cdot \mathbf{S} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}} + \mathbf{E}_0 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}}{\mathbf{A}_1 \cdot \mathbf{S} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}} + \mathbf{E}_1 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}}\right] \tag{22}$$

where $\mathbf{E} = \left[\frac{\mathbf{E}_0}{\mathbf{E}_1}\right] \hookleftarrow \chi^{2m \times (2m \log q + 2\lambda)}$.

**Game** $j.2$ $(1 \leq j \leq \kappa)$**:** We change the distribution of the vectors $\mathbf{t}_{\mathbf{x}|^j}$ of (22) which, for each query $\mathbf{x} \in \{0,1\}^\kappa$, are now computed as

$$\mathbf{t}_{\mathbf{x}|^j} = \left[\frac{\mathbf{A}_0 \cdot \mathbf{k}'_{\mathbf{x}|^{j+1}} + \mathbf{E}_0 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}}{\mathbf{A}_1 \cdot \mathbf{k}'_{\mathbf{x}|^{j+1}} + \mathbf{E}_1 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}}\right] \tag{23}$$

[11] Note that the assumption that $q$ is odd guarantees that we have a family of universal hash functions.

for some uniformly random vector $\mathbf{k}'_{\mathbf{x}|^{j+1}} \hookleftarrow U(\mathbb{Z}_q^m)$ which is a random function of the last $\kappa - j$ bits $\mathbf{x}[j+1], \ldots, \mathbf{x}[\kappa]$ of $\mathbf{x}$ (i.e., if these bits are the same, the output is the same, independently of the first $j$ bits)).

We observe that Game $j.2$ is statistically close to Game $j.1$ by Lemma 20. We know that each entry of $\mathbf{E} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}} \in \mathbb{Z}^{2m}$ is bounded by $2B(2m \log q + 2\lambda)$ in magnitude. The partial information $\mathbf{E} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}} \in \mathbb{Z}^{2m}$ about $\mathbf{r}_{\mathbf{x}|^{j+1}}$ thus lives in a set of size $(4B(m \log q + \lambda))^{2m}$.

If $\mathbf{r}_{\mathbf{x}|^{j+1}}$ is sampled from $U(\{-1,1\}^{2m \log q + 2\lambda})$, conditionally on $\mathbf{E} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}$, the distribution of $\mathbf{S} \cdot \mathbf{r}_{\mathbf{x}|^{j+1}} \in \mathbb{Z}_q^m$ is within statistical distance

$$\frac{1}{2} \cdot \sqrt{\frac{(4B(m \log q + \lambda))^{2m} \cdot q^m}{q^{2m} \cdot 2^{2\lambda}}} = \frac{1}{2} \cdot \Big(\frac{4B(m \log q + \lambda)}{q^{1/2}}\Big)^m \cdot \frac{1}{2^\lambda} < \ 2^{-\lambda}$$

from the distribution $U(\mathbb{Z}_q^m)$. This is because $B = \alpha q \sqrt{\lambda} \leq \sqrt{q}/(4(m \log q + \lambda))$, by assumption on $\alpha$. Taking a union bound over all queries, we obtain $|\Pr[W_{j.2}] - \Pr[W_{j.1}]| \leq Q \cdot 2^{-\lambda}$.

**Game** $j.3$ $(1 \leq j \leq \kappa)$**:** In this game, we change again the distribution of the vectors $\mathbf{t}_{\mathbf{x}|^j}$ of (23) and remove the terms $\mathbf{E}_0 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}$ and $\mathbf{E}_1 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}$ from the right-hand-side member of (23). The vectors $\mathbf{t}_{\mathbf{x}|^j}$ of (23) are thus changed into

$$\mathbf{t}_{\mathbf{x}|^j} = \left[\frac{\mathbf{A}_0 \cdot \mathbf{k}'_{\mathbf{x}|^{j+1}}}{\mathbf{A}_1 \cdot \mathbf{k}'_{\mathbf{x}|^{j+1}}}\right]. \tag{24}$$

We show that, except with negligible probability, this modification has no impact on $\mathcal{A}$'s view since it does not affect the rounding operation in (21). In other words, with overwhelming probability, the vector $\mathbf{t}_{\mathbf{x}|^j}$ of (24) leads to the same value of $F^{(j)}(\mathbf{x})$ as the one of (23).

To see this, let us call $\mathsf{bad}_{j.3}$ the event that Game $j.3$ deviates from Game $j.2$. To bound $\Pr[\mathsf{bad}_{j.3}]$, we first note that, since $\chi$ is a $B$-bounded distribution, each entry of $\mathbf{E}_0 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}$ and $\mathbf{E}_1 \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}$ is smaller than $4B(m \log q + \lambda)$ in magnitude. If we define

$$\mathbf{w}_{b,\mathbf{x}} = \prod_{i=1}^{j-1} \mathbf{A}_{\mathbf{x}[i]} \cdot (\mathbf{E}_b \cdot \mathbf{r}_{\mathbf{x}|^{j+1}}) \in \mathbb{Z}^m$$

for $b \in \{0,1\}$, we have $\|\mathbf{w}_{b,\mathbf{x}}\| \leq B_{\max} = 4m^{\kappa-1}\sqrt{m}B(m \log q + \lambda)$. Defining

$$\mathbf{y}_{\mathbf{x}} = \prod_{i=1}^{j} \mathbf{A}_{\mathbf{x}[i]} \cdot \mathbf{k}'_{\mathbf{x}|^{j+1}}, \tag{25}$$

we see that $\mathsf{bad}_{j.3}$ happens when $\lfloor \mathbf{y}_{\mathbf{x}} + \mathbf{w}_{\mathbf{x}[j],\mathbf{x}} \rfloor_p \neq \lfloor \mathbf{y}_{\mathbf{x}} \rfloor_p$, which implies that at least one of the coordinates of $\mathbf{y}_{\mathbf{x}}$ is within distance $B_{\max}$ from the nearest multiple of $q/p$. For a given query $\mathbf{x}$, we call this event $\mathsf{bad}_{\mathbf{x},j.3}$.

For a given query $\mathbf{x}$, we know that $\mathbf{k}'_{\mathbf{x}|^{j+1}}$ is uniformly distributed over $\mathbb{Z}_q^m$. Since $\mathbf{A}_0, \mathbf{A}_1 \in \{0,1\}^{m \times m}$ are $\mathbb{Z}_q$-invertible, so is any multi-product of these matrices and the vector $\mathbf{y}_{\mathbf{x}}$ of

(25) is thus uniformly distributed over $\mathbb{Z}_q^m$. The probability that any given coordinate of $\mathbf{y}_\mathbf{x}$ lands within distance $B_{\max}$ from a multiple of $q/p$ is at most $8m^{\kappa-1}\sqrt{m}B(m\log q+\lambda)\cdot(p/q)$. A union bound over the $m$ coordinates yields the inequality

$$\Pr[\mathsf{bad}_{\mathbf{x},j.3}] \leq 8m^{\kappa}\sqrt{m}B(m\log q+\lambda)\cdot\frac{p}{q}.$$

Taking a union bound over the number $Q<2^\kappa$ of queries, we get

$$\begin{aligned}|\Pr[W_{j.3}]-\Pr[W_{j.2}]| &\leq \Pr[\mathsf{bad}_{j.3}]\\ &\leq 2^{\kappa+3}\cdot m^{\kappa}\sqrt{m}B(m\log q+\lambda)\cdot\frac{p}{q}.\end{aligned}$$

Now, we observe that Game $j.3$ is identical to Game $j+1$ as the hybrid function that $\mathcal{A}$ interacts with can be written

$$F(\mathbf{x}) = \left\lfloor \prod_{i=1}^{j} \mathbf{A}_{\mathbf{x}[i]}\cdot\mathbf{k}'_{\mathbf{x}|^{j+1}} \right\rfloor_p.$$

Hence, $\Pr[W_{j.3}]=\Pr[W_{j+1}]$.

When putting the above altogether, the triangle inequality implies the claimed upper bound (19) for the distance $|\Pr[W_0]-\Pr[W_{\kappa+1}]|$. □