Approximate Trapdoors for Lattices and Smaller Hash-and-Sign Signatures

Yilei Chen[1], Nicholas Genise[2], and Pratyay Mukherjee[1]

[1] Visa Research, Palo Alto, USA
yilchen,pratmukh@visa.com
[2] University of California, San Diego, USA
ngenise@eng.ucsd.edu

Abstract. We study a relaxed notion of lattice trapdoor called *approximate trapdoor*, which is defined to be able to invert Ajtai's one-way function approximately instead of exactly. The primary motivation of our study is to improve the efficiency of the cryptosystems built from lattice trapdoors, including the hash-and-sign signatures.
Our main contribution is to construct an approximate trapdoor by modifying the gadget trapdoor proposed by Micciancio and Peikert [Eurocrypt 2012]. In particular, we show how to use the approximate gadget trapdoor to sample short preimages from a distribution that is simulatable without knowing the trapdoor. The analysis of the distribution uses a theorem (implicitly used in past works) regarding linear transformations of discrete Gaussians on lattices.
Our approximate gadget trapdoor can be used together with the existing optimization techniques to improve the concrete performance of the hash-and-sign signature in the random oracle model under (Ring-)LWE and (Ring-)SIS assumptions. Our implementation shows that the sizes of the public-key & signature can be reduced by half from those in schemes built from exact trapdoors.

1 Introduction

In the past two decades, lattice-based cryptography has emerged as one of the most active areas of research. It has enabled both advanced cryptographic capabilities, such as fully homomorphic encryption [29]; and practical post-quantum secure public-key encryptions and signatures, as observed in the ongoing NIST post-quantum cryptography (PQC) standardization procedure [4]. A large fraction of the lattice-based cryptosystems uses *lattice trapdoors*. Those cryptosystems include basic primitives like public-key encryption and signature schemes [33,39,38,31], as well as advanced primitives such as identity-based encryption [31,1,19], attribute-based encryption [34], and graded encodings [30].

In this work, we focus on the trapdoor for the lattice-based one-way function defined by Ajtai [2], and its application in digital signatures [31]. Given a wide, random matrix $\mathbf{A}$, and a target vector $\mathbf{y}$, the *inhomogeneous short integer solution* (ISIS) problem asks to find a short vector $\mathbf{x}$ as a preimage of $\mathbf{y}$, i.e.

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \pmod{q}.$$

Without a trapdoor for the matrix $\mathbf{A}$, finding a short preimage is proven to be as hard as solving certain lattice problems in the worst case [2]. A trapdoor for the matrix $\mathbf{A}$, on the other hand, allows its owner to efficiently produce a short preimage. An explicit construction of the trapdoor for Ajtai's function was first given in [3] and later simplified by [9,42].

Towards the proper use of lattice trapdoors in cryptography, what really gives the trapdoor a punch is the work of Gentry, Peikert and Vaikuntanathan [31]. They show how to sample a short preimage from a distribution that is simulatable without knowing the trapdoor, instead of a distribution which may leak information about the trapdoor (as observed by the attacks [32,46] on the initial attempts of building lattice-based signatures [33,38]). Such a preimage sampling algorithm allows [31] to securely build a hash-and-sign signature as follows. Let the matrix $\mathbf{A}$ be the public verification key, the trapdoor of $\mathbf{A}$ be the secret signing key. To sign a message m, first hash it to a vector $\mathbf{y}$, then use the trapdoor to sample a short preimage $\mathbf{x}$ as the signature. The secret signing key is guaranteed to be hidden from the signatures, since the signatures are simulatable without using the trapdoor.

Despite its elegant design, the hash-and-sign signature based on Ajtai's function suffers from practical inefficiency due to its large key size and signature size. Indeed, all the three lattice-based signature candidates that enter the second round of NIST PQC standardization [4] are built from two alternative approaches — Falcon [27] is based on the hash-and-sign paradigm over NTRU lattices; Dilithium [26] and qTESLA [8] are based on the rejection sampling approach [40,11]. The suggested parameters for the three candidates lead to competitive performance measures. For example, for 128-bit security, the sizes of the public keys & signatures for all the three candidates are below 5 kB & 4 kB (respectively). By contrast, for the hash-and-sign signature based on Ajtai's function, the sizes of the public keys & signatures are more than 35 kB & 25 kB according to the implementation benchmarks of [13,14,36].

1.1 Summary of our contributions

In this paper we develop new techniques to bring down the sizes of the public keys & signatures of the hash-and-sign signature based on Ajtai's one-way function. We define a relaxed notion of lattice trapdoor called *approximate trapdoor*, which can be used to solve the ISIS problem *approximately* instead of exactly. With a relaxation of the correctness requirement, it is possible to generate smaller public matrices, trapdoors, and preimages for Ajtai's function, which translate to smaller public-keys, secret-keys, and signatures for the hash-and-sign signature scheme.

Our main technical contribution is to show that the gadget trapdoor proposed by Micciancio and Peikert [42] can be modified to an approximate trapdoor. In particular, we show how to use the approximate gadget trapdoor to sample preimages from a distribution that is simulatable without knowing the trapdoor. The analysis of the distribution uses a theorem (implicitly used in past works) regarding linear transformations of discrete Gaussians on lattices.

Our approximate gadget trapdoor can be used together with all existing optimization techniques, such as using the Hermite normal form and using a bigger base in the gadget, to improve the concrete performance of the hash-and-sign signature in the random oracle model under RingLWE and RingSIS assumptions. Our proof-of-concept implementation shows that the sizes of the public-key & signature can be reduced to 5 kB & 4.45 kB for an estimation of 88-bit security, and 11.25 kB & 9.38 kB for an estimation of 184-bit security. Those are much closer to the sizes of the signatures based on the rejection sampling approach [40,11,26,8]. More details of the parameters are given in §1.3 and §5.2.

1.2 Technical overview

Given a public matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ where $m = O(n \log q)$, and a target $\mathbf{y}$, we call a vector $\mathbf{x} \in \mathbb{Z}^m$ an *approximate short preimage* of $\mathbf{y}$ if

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{y} + \mathbf{z} \pmod{q}$$

for some $\mathbf{z} \in \mathbb{Z}^n$, and both $\mathbf{x}$ and $\mathbf{z}$ are short. An *approximate trapdoor* for $\mathbf{A}$ is defined to be a string that allows its owner to efficiently find an approximate short preimage given a target $\mathbf{y}$.

Of course, to make sense of the word "trapdoor", we first need to argue that solving the approximate version of ISIS is hard without the trapdoor. Under proper settings of parameters, we show the approximate ISIS problem is as hard as the standard ISIS problem, or no easier than LWE. The reductions extensively use the Hermite normal form (HNF) and are pretty straightforward.

The approximate ISIS problem and the approximate trapdoor are natural generalizations of their exact variants. Indeed, both notions have been used in the literature, at least on an informal level. For example, the approximate ISIS problem was used in the work of Bai et al. [12] to improve the combinatorial algorithms of the exact ISIS problem.

It is well-known that an exact trapdoor of a public matrix in the HNF, say a trapdoor for $\mathbf{A} = [\mathbf{I}_n \mid \mathbf{A}']$, can be used as an approximate trapdoor for $\mathbf{A}'$. Such a method was often used in the implementation of signatures to decrease the sizes of the public key and the signature by a dimension of n. Our goal is thus to further reduce the sizes compared to the HNF approach, while preserving the quality of the trapdoor, i.e. at least not increasing the norm of the preimage.

Approximate gadget trapdoor. Our main contribution is to show that the gadget trapdoor (G-trapdoor) proposed by Micciancio and Peikert [42] can be modified to an approximate trapdoor, in a way that further reduces the sizes of the public matrix, the trapdoor, and the preimage.

Recall the core of the G-trapdoor is a specific "gadget" matrix of base b,

$$\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g}^t := \mathbf{I}_n \otimes (1, b, ..., b^{k-1}) \in \mathbb{Z}^{n \times (nk)},$$

where $k := \lceil \log_b q \rceil$. The base b is typically chosen to be 2 for simplicity, or a larger value in practical implementations.

Micciancio and Peikert [42] show how to generate a random matrix $\mathbf{A}$ together with a matrix $\mathbf{D}$ of small norm such that $\mathbf{A} \cdot \mathbf{D} = \mathbf{G} \pmod q$. In particular, $\mathbf{A}$ is designed to be

$$\mathbf{A} = [\bar{\mathbf{A}} | \mathbf{G} - \bar{\mathbf{A}}\mathbf{R}],$$

where $\mathbf{R}$ is a matrix with small entries and is the actual trapdoor. The matrix $\mathbf{D}$ is then equal to $\begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{nk} \end{bmatrix}$. Since the kernel of the $\mathbf{G}$ matrix has a public short basis, one can first solve the ISIS problem under the public matrix $\mathbf{G}$, then use $\mathbf{D}$ to solve the ISIS problem under the public matrix $\mathbf{A}$.

We observe that if we drop a few (say l) entries corresponding to the small powers of b from the gadget matrix $\mathbf{G}$, i.e. let the following $\mathbf{F}$ matrix be a modified gadget matrix

$$\mathbf{F} := \mathbf{I}_n \otimes \mathbf{f}^t := \mathbf{I}_n \otimes (b^l, ..., b^{k-1}) \in \mathbb{Z}^{n \times n(k-l)},$$

then we are still able to solve the ISIS problem w.r.t. the public matrix $\mathbf{F}$ up to a b^l-approximation of the solution (i.e., the norm of the error vector is proportional to b^l). Replacing $\mathbf{G}$ by $\mathbf{F}$ in $\mathbf{A}$ gives

$$\mathbf{A} = [\bar{\mathbf{A}} | \mathbf{F} - \bar{\mathbf{A}}\mathbf{R}]. \tag{1}$$

Then the dimensions of the trapdoor $\mathbf{R}$ and the public matrix $\mathbf{A}$ can be reduced.

Sampling from a distribution that is simulatable without knowing the trapdoor. Given a public matrix $\mathbf{A}$ together with its approximate G-trapdoor $\mathbf{R}$, finding an arbitrary approximate short preimage of a given target $\mathbf{u}$ is quite straightforward, but sampling the preimage from a distribution that is simulatable without knowing the trapdoor turns out to be non-trivial. As mentioned earlier, the ability to sample from such a distribution is fundamental to most of the trapdoor applications including digital signatures.

We provide an algorithm that samples an approximate short preimage from a distribution that is simulatable without knowing the trapdoor. The algorithm itself is a fairly simple generalization of the perturbation-based discrete Gaussian sampler from [42], but the analyses of the preimage distribution from [42] are not easy to generalize. Our analyses of the preimage distribution and the approximation error distribution extensively use a linear transformation theorem on lattice distributions (cf. Lemma 4, or Theorem 1, implicitly used in [42,43,15,25]).

The details of the analyses are quite technical. Here let us mention the difference in the way of obtaining the main result of ours compared to the ones from [31,42]. The approach taken by [31,42] is to first spell out the distributions of the preimages for *all* the target images $\mathbf{u} \in \mathbb{Z}_q^n$, then show the distributions are simulatable for *uniformly random* target images. For the approximate preimage sampling, we are only able to simulate the distributions of the preimages and the errors for *uniformly random* targets, without being able to spell out the meaningful distributions for *all* the targets an intermediate step. Still, simulating the preimages of uniform targets suffices for the application of digital signatures.

To briefly explain the reason behind the difference, let us point out that the methods we have tried to analyze the preimage distribution for *all* the target images require significant increases in the smoothing parameters of the lattice intersections required in the linear transformation theorem (Theorem 1). In other words, the norm of the resulting preimage increases significantly rendering the result meaningless.

1.3 Improvement in the efficiency compared to the exact trapdoor

We now explain the efficiency gain of using our approximate trapdoor compared to the exact trapdoor and the other existing optimization techniques, with a focus on the signature application. Our goal is to set the parameters to achieve the following "win-win-win" scenario:

1. Save on the size of the preimage (i.e., the signature).
2. Save on the size for the public matrix $\mathbf{A}$.
3. Retain, or even gain, concrete security, which is related to the discrete Gaussian width of the preimage and the norm of the error term.

Parameters	Exact G-trapdoor	Approximate G-trapdoor
m	$n(2+k)$	$n(2+(k-l))$
σ	$\sqrt{b^2+1}\cdot\omega(\sqrt{\log n})$	$\sqrt{b^2+1}\cdot\omega(\sqrt{\log n})$
s	$C\cdot\tau\cdot(\sqrt{m}+2\sqrt{n})\cdot\sigma$	$C\cdot\tau\cdot(\sqrt{m}+2\sqrt{n})\cdot\sigma$
ν	0	$b^l\cdot\sigma$

Fig. 1. A brief comparison of the parameters. The parameters in the table are derived under a fixed lattice dimension n, a fixed modulus $q \geq \sqrt{n}$, and a fixed base b. Let $k = \lceil \log_b q \rceil$. Let l denote the number of entries removed from $\mathbf{g}$ ($1 \leq l < k$). Then we list m as the dimension of the public matrix and the preimage; σ as the width of the gadget preimage distribution; s as the width of the final preimage distribution (where $C > 0$ is a universal constant); τ as the width, or subgaussian parameter, of the distribution of the entries in the trapdoor matrix $\mathbf{R}$; ν as the length bound of the error for each entry in the image.

Let us start with an understanding of the dependency of the savings on the variable l, i.e, the number of entries dropped from the gadget $\mathbf{g}$. In Figure 1 we provide a comparison of the parameters between the exact G-trapdoor of [42] and the approximate G-trapdoor samplers in this paper. In both cases the public matrices are instantiated in the pseudorandom mode. For the approximate trapdoor, the dimension of the trapdoor decreases from nk to $n(k-l)$. The dimension m of the public matrix and the preimage decreases. The width s of the preimage distribution also decreases slightly following the decreasing of m. However, the norm of the error factor in the image grows with l. So in the concrete instantiation of the hash-and-sign signature discussed later, we need to

coordinate the value of l with the norms of the preimage and the error, which will determine the cost of the attacks together.

Our algorithm inherits the $O(\log q)$-space, $O(n \log q)$-time G-preimage sample subroutine from [42,28]. So the saving of space and time in the sampling of the perturbation is proportional to the saving in the dimension m.

Concrete parameters for the signatures. We give a proof-of-concept implementation of the hash-and-sign signature based on our approximate trapdoor. The security is analyzed in the random oracle model, assuming the hardness of RingLWE for the pseudorandomness of the public key and RingSIS for the unforgeability of the signature. Here we provide a short summary and leave more details in Section 5.2.

Let us first remark that different implementation results of the hash-and-sign signatures [13,14,36] possibly use different ways of measuring sizes and security, and not all the details behind the parameters are recoverable from these papers. So we also implementation the exact trapdoor as a reference. For an estimation of 88-bit security, our reference implementation for the exact trapdoor under the modulus $q \approx 2^{24}$ and base $b = 2$ matches the parameters reported in [13].

We also use smaller moduli and bigger bases to reduce the size and increase the security level. The parameters in Figure 2 suggest that for the 3 choices of q and b, using the approximate gadget trapdoor by setting $l = \lceil (\log_b q)/2 \rceil$ saves about half of the sizes in the public key and signatures comparing to using the exact trapdoor, with even a slight increase in the expected cost for the attacking algorithms. Let us mention that some schemes in the literature (like [23]) use an extremely large base of size $b \approx \sqrt{q}$ (the resulting gadget is $\mathbf{g} = [1, \sqrt{q}]$). However, for the small moduli like 2^{16} or 2^{18}, such large bases lead to Gaussian widths larger than the moduli. So we only use moderately large bases.

Params	Exact	Approx	Approx	Exact	Approx	Approx	Exact	Approx	Approx
n	512	512	512	512	512	512	512	512	512
$\lceil \log_2 q \rceil$	24	24	24	16	16	16	16	16	16
b	2	2	2	2	2	2	4	4	4
l	0	12	15	0	7	9	0	2	4
τ	40	40	40	2.6	2.6	2.6	2.6	2.6	2.6
s	38317.0	29615.3	26726.3	2170.7	1756.3	1618.2	3114.2	2833.3	2505.6
m	13312	7168	5632	9216	5632	4608	5120	4096	3072
$\|\mathbf{x}\|_2$	4441737.7	2521387.0	2035008.5	211100.9	133305.5	109339.1	223740.1	183004.9	138145.7
$\|\mathbf{z}\|_2$	0	374014.0	2118987.6	0	11897.9	46428.4	0	1402.3	19807.1
PK	37.50	19.50	15.00	17.00	10.00	8.00	9.00	7.00	5.00
Sig	25.68	13.53	10.51	13.16	7.83	6.30	7.62	5.94	4.45
LWE	100.0	100.0	100.0	104.7	104.7	104.7	104.7	104.7	104.7
AISIS	80.2	85.8	81.1	83.7	89.0	88.1	82.8	85.5	87.8

Fig. 2. Summary of the concrete parameters. The size of PK and Sig are measured in kB. $\|\mathbf{x}\|_2$, $\|\mathbf{z}\|_2$ are the upper-bounds of the norms of the preimage and the error term. LWE and AISIS refer to the estimations of security levels for the pseudorandomness of the PK and finding a short approximate preimage.

Our implementation shows that the sizes of the public-key & signature can be reduced to 5 kB & 4.45 kB for an estimation of 88-bit security, and 11.25 kB & 9.38 kB for an estimation of 184-bit security. Those are much closer to the sizes of the signatures based on the rejection sampling approach [40,11,26,8]. As a reference, the sizes of the public-key & signature for qTESLA [8] are 4.03 kB & 3.05 kB for an estimation of 128-bit security, and 8.03 kB & 6.03 kB for an estimation of 192-bit security. The sizes for Dilithium [26] are even smaller. Let us remark that our implementation has not adapted possible further optimizations used in Dilithium [26] and qTESLA [8]. So it is reasonable to expect we have more room to improve after adding making further optimizations. The parameters for Falcon [27] are the smallest due to the use of NTRU lattices, so they are rather incomparable with the ones based on RingLWE. As a side note, we do not know how to construct approximate trapdoors for NTRU lattices, and we leave it as an interesting question to investigate in future.

Using approximate trapdoors in the advanced lattice cryptosystems. Finally, let us briefly mention the possible applications of the approximate trapdoors in the cryptosystems built from the dual-Regev approach [31,1,19,34] and the GGH15 approach [30,17,18,35,52,21].

To use approximate trapdoors in the schemes based on the dual-Regev approach, we need to sample the LWE secret term with a small norm instead of from the uniform distribution to maintain the correctness of the schemes. For many of these schemes, the security analyses require the extensions of the Bonsai techniques in the approximate setting. We leave the extensions to future works.

For the schemes based on the GGH15-approach, the correctness of the schemes holds without any changes. The security also holds, except for the schemes in [21] which requires the extension of the Bonsai techniques. Let us remark that the saving in the dimension m is of significant importance to the applications built on the GGH15 graded encoding scheme (implemented in [37,20]). In those applications, the modulus q is proportional to m^d (where $d \in \mathbb{N}$ is the number of "levels" of the graded encodings; larger d supports richer functionalities). So reducing the dimension m would dramatically reduce the overall parameter.

Organizations. The rest of the paper is organized as follows. Section 2 provides the necessary background of lattices. Section 3 provides the definition and the hardness reductions of the approximate ISIS problem. Section 4 presents the approximate gadget trapdoors. Section 5 provides an instantiation of the hash-and-sign signature scheme under the approximate trapdoor, with concrete parameters.

2 Preliminaries

Notations and terminology. In cryptography, the security parameter (denoted as λ) is a variable that is used to parameterize the computational complexity of the cryptographic algorithm or protocol, and the adversary's probability of breaking

security. An algorithm is "efficient" if it runs in (probabilistic) polynomial time over λ.

When a variable v is drawn uniformly random from the set S we denote as $v \leftarrow U(S)$. We use $\approx_s$ and $\approx_c$ as the abbreviations for statistically close and computationally indistinguishable. For two distributions D_1, D_2 over the same support $\mathcal{X}$, we denote $D_1 \overset{\varepsilon}{\approx} D_2$ to denote that each $x \in \mathcal{X}$ has $D_1(x) \in [1 \pm \varepsilon] D_2(x)$ and $D_2(x) \in [1 \pm \varepsilon] D_1(x)$.

Let $\mathbb{R}, \mathbb{Z}, \mathbb{N}$ be the set of real numbers, integers and positive integers. Denote $\mathbb{Z}/q\mathbb{Z}$ by $\mathbb{Z}_q$. For $n \in \mathbb{N}$, $[n] := \{1, ..., n\}$. A vector in $\mathbb{R}^n$ (represented in column form by default) is written as a bold lower-case letter, e.g. $\mathbf{v}$. For a vector $\mathbf{v}$, the i^{th} component of $\mathbf{v}$ will be denoted by v_i. For an integer base $b > 1$, we call a positive integer's "b-ary" decomposition the vector $(q_0, q_1, \ldots, q_{k-1}) \in \{0, \ldots, b-1\}^k$ where $k := \lceil \log_b q \rceil$, and $q = \sum q_i b^i$.

A matrix is written as a bold capital letter, e.g. $\mathbf{A}$. The i^{th} column vector of $\mathbf{A}$ is denoted $\mathbf{a}_i$. The length of a vector is the ℓ_p-norm $\|\mathbf{v}\|_p := (\sum v_i^p)^{1/p}$, or the infinity norm given by its largest entry $\|\mathbf{v}\|_\infty := \max_i\{|v_i|\}$. The length of a matrix is the norm of its longest column: $\|\mathbf{A}\|_p := \max_i \|\mathbf{a}_i\|_p$. By default we use ℓ_2-norm unless explicitly mentioned. When a vector or matrix is called "small" or "short", we refer to its norm but not its dimension, unless explicitly mentioned. The thresholds of "small" or "short" will be precisely parameterized in the article when necessary.

2.1 Linear Algebra

Let $\{\mathbf{e}_i\}_{i=1}^n$ be the canonical basis for $\mathbb{R}^n$, with entries $\delta(j,k)$ where $\delta(j,k) = 1$ when $j = k$ and 0 otherwise. For any set $S \subseteq \mathbb{R}^n$, its span (denoted as $\text{span}(S)$) is the smallest subspace of $\mathbb{R}^n$ containing S. For a matrix, $\mathbf{M} \in \mathbb{R}^{n \times m}$, its span is the span of its column vectors, written as $\text{span}(\mathbf{M})$. We write matrix transpose as $\mathbf{M}^t$. Let $\tilde{\mathbf{B}}$ denote the Gram-Schmidt orthogonalization of $\mathbf{B}$. The GSO of an ordered basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k]$ is assumed to be from left to right, $\tilde{\mathbf{b}}_1 = \mathbf{b}_1$, unless stated otherwise.

Recall $\mathbf{M}$'s singular value decomposition (SVD), i.e. $\mathbf{M} = \mathbf{VDW} \in \mathbb{R}^{n \times m}$ where $\mathbf{V} \in \mathbb{R}^{n \times n}$ along with $\mathbf{W} \in \mathbb{R}^{m \times m}$ are unitary, and $\mathbf{D} \in \mathbb{R}^{n \times m}$ is a triangular matrix containing $\mathbf{M}$'s singular values. Further, let $q = \min\{n, m\}$ and $\mathbf{D}_q = \text{diag}(s_1, \ldots, s_q)$ be the diagonal matrix containing $\mathbf{M}$'s singular values $s_i = s_i(\mathbf{M})$. Throughout the paper, we are concerned with random, subgaussian [51] matrices $\mathbf{M}$ with $\{s_1 \geq \ldots \geq s_q > 0\}$. Then, $\mathbf{D} = \mathbf{D}_q$ when $n = m$, $\mathbf{D} = [\mathbf{D}_q \ \mathbf{0}]$ when $m > n$, and $\mathbf{D} = \begin{bmatrix} \mathbf{D}_q \\ \mathbf{0} \end{bmatrix}$ in the case $m < n$.

A symmetric matrix $\Sigma \in \mathbb{R}^{n \times n}$ is *positive semi-definite* if for all $\mathbf{x} \in \mathbb{R}^n$, we have $\mathbf{x}^t \Sigma \mathbf{x} \geq 0$. It is *positive definite*, $\Sigma > 0$, if it is positive semi-definite and $\mathbf{x}^t \Sigma \mathbf{x} = 0$ implies $\mathbf{x} = \mathbf{0}$. We say $\Sigma_1 > \Sigma_2$ ($\geq$) if $\Sigma_1 - \Sigma_2$ is positive-(semi)definite. This forms a partial ordering on the set of positive semi-definite matrices, and we denote $\Sigma \geq \alpha \mathbf{I}$ often as $\Sigma \geq \alpha$ for constants $\alpha \in \mathbb{R}^+$. For any positive semi-definite matrix Σ, we write $\sqrt{\Sigma}$ to be any full rank matrix $\mathbf{T}$ such

that $\Sigma = \mathbf{T}\mathbf{T}^t$. We say $\mathbf{T}$ is a *square root* of Σ. For two positive semi-definite matrices, Σ_1 and Σ_2, we denote the positive semi-definite matrix formed by their block diagonal concatenation as $\Sigma_1 \oplus \Sigma_2$. Let $\mathbf{M}^*$ denote Hermitian transpose. The *(Moore-Penrose) pseudoinverse* for matrix $\mathbf{M}$ with SVD $\mathbf{M} = \mathbf{VDW}$ is $\mathbf{M}^+ = \mathbf{WD}^+\mathbf{V}^*$ where $\mathbf{D}^+$ is given by transposing $\mathbf{D}$ and inverting $\mathbf{M}$'s nonzero singular values. For example, $\mathbf{T} = s\mathbf{I}$ and $\mathbf{T}^+ = s^{-1}\mathbf{I}$ for a covariance $\Sigma = s^2\mathbf{I}$. (An analogous $\mathbf{T}^+ = \mathbf{T}^{-1}$ is given for the non-spherical, full-rank case $\Sigma > 0$ using Σ's diagonalization.)

2.2 Lattices background

An n-dimensional lattice Λ of rank $k \leq n$ is a discrete additive subgroup of $\mathbb{R}^n$. Given k linearly independent basis vectors $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_k \in \mathbb{R}^n\}$, the lattice generated by $\mathbf{B}$ is

$$\Lambda(\mathbf{B}) = \Lambda(\mathbf{b}_1, ..., \mathbf{b}_k) = \{\sum_{i=1}^{k} x_i \cdot \mathbf{b}_i, x_i \in \mathbb{Z}\}.$$

Given $n, m \in \mathbb{N}$ and a modulus $q \geq 2$, we often use q-ary lattices and their cosets, denoted as

$$\text{for } \mathbf{A} \in \mathbb{Z}_q^{n \times m}, \text{ denote } \Lambda^\perp(\mathbf{A}) \text{ or } \Lambda_q^\perp(\mathbf{A}) \text{ as } \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \pmod q\};$$

$$\text{for } \mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{w} \in \mathbb{Z}_q^n, \quad \text{denote } \Lambda_{\mathbf{w}}^\perp(\mathbf{A}) \text{ as } \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{w} \pmod q\}.$$

Gaussians on lattices. For any $s > 0$ define the Gaussian function on $\mathbb{R}^n$ with parameter s:

$$\forall \mathbf{x} \in \mathbb{R}^n, \ \rho_s(\mathbf{x}) = e^{-\pi \|\mathbf{x}\|^2 / s^2}.$$

For any $\mathbf{c} \in \mathbb{R}^n$, real $s > 0$, and n-dimensional lattice Λ, define the discrete Gaussian distribution $D_{\Lambda + \mathbf{c}, s}$ as:

$$\forall \mathbf{x} \in \Lambda + \mathbf{c}, \ D_{\Lambda + \mathbf{c}, s}(\mathbf{x}) = \frac{\rho_s(\mathbf{x})}{\rho_s(\Lambda + \mathbf{c})}.$$

The subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$ (respectively) when omitted.

For any positive semidefinite $\Sigma = \mathbf{T} \cdot \mathbf{T}^t$, define the non-spherical Gaussian function as

$$\forall \mathbf{x} \in \text{span}(\mathbf{T}) = \text{span}(\Sigma), \ \rho_{\mathbf{T}}(\mathbf{x}) = e^{-\pi \mathbf{x}^t \Sigma^+ \mathbf{x}},$$

and $\rho_{\mathbf{T}}(\mathbf{x}) = 0$ for all $\mathbf{x} \notin \text{span}(\Sigma)$. Note that $\rho_{\mathbf{T}}(\cdot)$ only depends on Σ but not the specific choice of the $\mathbf{T}$, so we may write $\rho_{\mathbf{T}}(\cdot)$ as $\rho_{\sqrt{\Sigma}}(\cdot)$.

For any $\mathbf{c} \in \mathbb{R}^n$, any positive semidefinite Σ, and n-dimensional lattice Λ such that $(\Lambda + \mathbf{c}) \cap \text{span}(\Sigma)$ is non-empty, define the discrete Gaussian distribution $D_{\Lambda + \mathbf{c}, \sqrt{\Sigma}}$ as:

$$\forall \mathbf{x} \in \Lambda + \mathbf{c}, \ D_{\Lambda + \mathbf{c}, \sqrt{\Sigma}}(\mathbf{x}) = \frac{\rho_{\sqrt{\Sigma}}(\mathbf{x})}{\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{c})}.$$

Smoothing parameter. We recall the definition of smoothing parameter and some useful facts.

Definition 1 (Smoothing parameter [44]). *For any lattice Λ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest real $s > 0$ such that $\rho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$.*

Notice that for two lattices of the same rank $\Lambda_1 \subseteq \Lambda_2$, the denser lattice always has the smaller smoothing parameter, i.e. $\eta_\epsilon(\Lambda_2) \leq \eta_\epsilon(\Lambda_1)$.

We will need a generalization of the smoothing parameter to the non-spherical Gaussian.

Definition 2. *For a positive semi-definite $\Sigma = \mathbf{T}\mathbf{T}^t$, an $\epsilon > 0$, and a lattice Λ with $span(\Lambda) \subseteq span(\Sigma)$, we say $\eta_\epsilon(\Lambda) \leq \sqrt{\Sigma}$ if $\eta_\epsilon(\mathbf{T}^+\Lambda) \leq 1$.*

When the covariance matrix $\Sigma > 0$ and the lattice Λ are full-rank, $\sqrt{\Sigma} \geq \eta_\varepsilon(\Lambda)$ is equivalent to the minimum eigenvalue of Σ, $\lambda_{min}(\Sigma)$, being at least $\eta_\varepsilon^2(\Lambda)$.

Lemma 1 ([44]). *For any n-dimensional lattice Λ of rank k, and any real $\epsilon > 0$,*

$$\eta_\epsilon(\Lambda) \leq \lambda_k(\Lambda) \cdot \sqrt{\log(2k(1+1/\epsilon))/\pi}.$$

Lemma 2 ([44]). *Let Λ be a lattice, $\mathbf{c} \in span(\Lambda)$. For any $\Sigma \geq 0$, if $\sqrt{\Sigma} \geq \eta_\epsilon(\Lambda)$ for some $\epsilon > 0$, then*

$$\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{c}) \in \left[\frac{1-\epsilon}{1+\epsilon}, 1\right] \cdot \rho_{\sqrt{\Sigma}}(\Lambda)$$

The following is a generalization of [31, Corollary 2.8] for non-spherical Gaussian.

Corollary 1 (Smooth over the cosets). *Let Λ, Λ' be n-dimensional lattices s.t. $\Lambda' \subseteq \Lambda$. Then for any $\epsilon > 0$, $\sqrt{\Sigma} \geq \eta_\epsilon(\Lambda')$, and $\mathbf{c} \in span(\Lambda)$, we have*

$$\Delta(D_{\Lambda+\mathbf{c},\sqrt{\Sigma}} \bmod \Lambda',\ U(\Lambda \bmod \Lambda')) < 2\epsilon$$

Lemma 3 ([49,44]). *Let $\mathbf{B}$ be a basis of an n-dimensional lattice Λ, and let $s \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\log n)$, then $\Pr_{\mathbf{x} \leftarrow D_{\Lambda,s}}[\|\mathbf{x}\| \geq s \cdot \sqrt{n} \vee \mathbf{x} = \mathbf{0}] \leq \mathsf{negl}(n)$.*

Linear Transformations of Discrete Gaussians. We will use the following general theorem, implicitly used in [42,43,15], regarding the linear transformation, $\mathbf{T}$, of a discrete Gaussian. It states that as long as the original discrete Gaussian over a lattice Λ is smooth enough in the lattice intersect the kernel of $\mathbf{T}$ ($\Lambda \cap \ker(\mathbf{T})$), then the distribution transformed by $\mathbf{T}$ is statistically close to another discrete Gaussian.

Theorem 1 ([41]). *For any positive definite Σ, vector $\mathbf{c}$, lattice coset $A := \Lambda + \mathbf{a} \subset \mathbf{c} + span(\Sigma)$, and linear transformation $\mathbf{T}$, if the lattice $\Lambda_\mathbf{T} = \Lambda \cap ker(\mathbf{T})$ satisfies $span(\Lambda_\mathbf{T}) = ker(\mathbf{T})$ and $\eta_\epsilon(\Lambda_\mathbf{T}) \leq \sqrt{\Sigma}$, then*

$$\mathbf{T}(D_{A,\mathbf{c},\sqrt{\Sigma}}) \overset{\bar{\epsilon}}{\approx} D_{\mathbf{T}A,\mathbf{T}\mathbf{c},\mathbf{T}\sqrt{\Sigma}}$$

where $\bar{\epsilon} = 2\epsilon/(1-\epsilon)$.

We remark that if $\mathbf{T}$ is injective (i.e. $\ker(\mathbf{T})$ is trivial), then $\mathbf{T}(D_{A,\mathbf{c},\sqrt{\Sigma}}) = D_{\mathbf{T}A,\mathbf{Tc},\mathbf{T}\sqrt{\Sigma}}$.

Let us also remark that at the time of writing this article, the following lemma (which is a special case of Theorem 1) has already been proven in [25]. This lemma is suitable for all of our proofs using a non-injective linear transformation of a discrete gaussian.

In what follows, the max-log distance between two distributions with the same support S is $\Delta_{ML}(\mathcal{X},\mathcal{Y}) = \max_{s\in S} |\log \mathcal{X}(s) - \log \mathcal{Y}(s)|$ [45].

Lemma 4 (Lemma 3, [25]). *Let $\mathbf{T} \in \mathbb{Z}^{n\times m}$ such that $\mathbf{T}\mathbb{Z}^m = \mathbb{Z}^n$ and $\Lambda^\perp(\mathbf{T}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{Tx} = \mathbf{0} \in \mathbb{Z}^n\}$. Let $\Sigma = \mathbf{T}\mathbf{T}^t$. For $\epsilon \in (0, 1/2)$, $\hat{\epsilon} = \epsilon + O(\epsilon^2)$, $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{T}))$, the max-log distance between $\mathbf{T}\cdot D_{\mathbb{Z}^m,r}$ and $D_{\mathbb{Z}^n,r\sqrt{\Sigma}}$ is at most $4\hat{\epsilon}$.*

2.3 Gadgets, or G-Lattices

Let $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^t \in \mathbb{Z}_q^{n\times nk}$ with $\mathbf{g}^t = (1, b, \ldots, b^{k-1})$, $k = \lceil \log_b q \rceil$. $\mathbf{G}$ is commonly referred to the gadget matrix. The gadget matrix's q-ary lattice, $\Lambda_q^\perp(\mathbf{G})$, is the direct sum of n copies of the lattice $\Lambda_q^\perp(\mathbf{g}^t)$. Further, $\Lambda_q^\perp(\mathbf{g}^t)$ has a simple basis,

$$\mathbf{B}_q = \begin{bmatrix} b & & & q_0 \\ -1 & \ddots & & \vdots \\ & \ddots & b & q_{k-2} \\ & & -1 & q_{k-1} \end{bmatrix}$$

where $(q_0, \ldots, q_{k-1}) \in \{0, 1, \ldots, b-1\}^k$ is the b-ary decomposition of the modulus, q. When $q = b^k$, we cheat by having $q_0 = q_1 = \ldots = q_{k-2} = 0$ and $q_{k-1} = b$. Either way, the integer cosets of $\Lambda_q^\perp(\mathbf{g}^t)$ can be viewed as the syndromes of $\mathbf{g}^t$ as a check matrix, in the terminology of coding theory. These cosets are expressed as $\Lambda_u^\perp(\mathbf{g}^t) = \{\mathbf{x} \in \mathbb{Z}^k : \mathbf{g}^t\mathbf{x} = u \mod q\} = \Lambda_q^\perp(\mathbf{g}^t) + \mathbf{u}$ where $\mathbf{u}$ can be any coset representative. A simple coset representative of $\Lambda_u^\perp(\mathbf{g}^t)$ is the b-ary decomposition of u. The integer cosets of $\Lambda_q^\perp(\mathbf{G})$ are expressed through the direct-sum construction, $\Lambda_{\mathbf{u}}^\perp(\mathbf{G}) = \Lambda_{u_1}^\perp(\mathbf{g}^t) \oplus \ldots \oplus \Lambda_{u_n}^\perp(\mathbf{g}^t)$ where $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{Z}_q^n$. We call $\mathbf{G}$ a gadget matrix since the following problems, SIS and LWE, are easily solved on the matrix $\mathbf{G}$ [42].

2.4 SIS, LWE, and the trapdoor

We first recall the short integer solution (SIS) problem.

Definition 3 (SIS [2]). *For any $n, m, q \in \mathbb{Z}$ and $\beta \in \mathbb{R}$, define the short integer solution problem $\mathsf{SIS}_{n,m,q,\beta}$ as follows: Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\| \leq \beta$, and*

$$\mathbf{Ax} = \mathbf{0} \bmod q.$$

Definition 4 (ISIS). *For any $n, m, q \in \mathbb{Z}$ and $\beta \in \mathbb{R}$, define the inhomogeneous short integer solution problem $\mathsf{ISIS}_{n,m,q,\beta}$ as follows: Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{y} \in \mathbb{Z}_q^n$, find $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\| \leq \beta$, and*

$$\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q.$$

Lemma 5 (Hardness of (I)SIS based on the lattice problems in the worst case [2,44,31]). *For any $m = \mathsf{poly}(n)$, any $\beta > 0$, and any sufficiently large $q \geq \beta \cdot \mathsf{poly}(n)$, solving $\mathsf{SIS}_{n,m,q,\beta}$ or $\mathsf{ISIS}_{n,m,q,\beta}$ (where $\mathbf{y}$ is sampled uniformly from $\mathbb{Z}_q^n$) with non-negligible probability is as hard as solving GapSVP_γ and SIVP_γ on arbitrary n-dimensional lattices with overwhelming probability, for some approximation factor $\gamma = \beta \cdot \mathsf{poly}(n)$.*

All the (I)SIS problems and their variants admit the Hermite normal form (HNF), where the public matrix $\mathbf{A}$ is of the form $[\mathbf{I}_n \mid \mathbf{A}']$ where $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$. The HNF variant of (I)SIS is as hard as the standard (I)SIS. This can be seen by rewriting $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ as $\mathbf{A} =: [\mathbf{A}_1 \mid \mathbf{A}_2] = \mathbf{A}_1 \cdot [\mathbf{I}_n \mid \mathbf{A}_1^{-1} \cdot \mathbf{A}_2]$ (we always work with n, q such that $\mathbf{A}_1 \leftarrow U(\mathbb{Z}_q^{n \times n})$ is invertible with non-negligible probability).

Learning with errors. We recall the decisional learning with errors (LWE) problem.

Definition 5 (Decisional learning with errors [50]). *For $n, m \in \mathbb{N}$ and modulus $q \geq 2$, distributions for secret vectors, public matrices, and error vectors $\theta, \pi, \chi \subseteq \mathbb{Z}_q$. An LWE sample is obtained from sampling $\mathbf{s} \leftarrow \theta^n$, $\mathbf{A} \leftarrow \pi^{n \times m}$, $\mathbf{e} \leftarrow \chi^m$, and outputting $(\mathbf{A}, \mathbf{y}^t := \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \mod q)$.*

We say that an algorithm solves $\mathsf{LWE}_{n,m,q,\theta,\pi,\chi}$ if it distinguishes the LWE sample from a random sample distributed as $\pi^{n \times m} \times U(\mathbb{Z}_q^m)$ with probability greater than $1/2$ plus non-negligible.

Lemma 6 (Hardness of LWE based on the lattice problems in the worst case [50,47,16,48]). *Given $n \in \mathbb{N}$, for any $m = \mathsf{poly}(n)$, $q \leq 2^{\mathsf{poly}(n)}$. Let $\theta = \pi = U(\mathbb{Z}_q)$, $\chi = D_{\mathbb{Z},s}$ where $s \geq 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that breaks $\mathsf{LWE}_{n,m,q,\theta,\pi,\chi}$, then there exists an efficient (possibly quantum) algorithm for solving GapSVP_γ and SIVP_γ on arbitrary n-dimensional lattices with overwhelming probability, for some approximation factor $\gamma = \tilde{O}(nq/s)$.*

The next lemma shows that LWE with the secret sampled from the error distribution is as hard as the standard LWE.

Lemma 7 ([10,16]). *For n, m, q, s chosen as was in Lemma 6, $\mathsf{LWE}_{n,m',q,D_{\mathbb{Z},s},U(\mathbb{Z}_q),D_{\mathbb{Z},s}}$ is as hard as $\mathsf{LWE}_{n,m,q,U(\mathbb{Z}_q),U(\mathbb{Z}_q),D_{\mathbb{Z},s}}$ for $m' \leq m - (16n + 4\log\log q)$.*

Trapdoor. A *trapdoor* for a public matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is a string that allows its owner to efficiently solve both the (I)SIS and LWE problems w.r.t. $\mathbf{A}$.

3 The Approximate Trapdoor for Ajtai's Function

Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, define an *approximate trapdoor* of $\mathbf{A}$ as anything that allows us to efficiently solve the approximate version of the ISIS problem w.r.t. $\mathbf{A}$. We first define the approximate ISIS problem.

Definition 6 (Approximate ISIS). *For any $n, m, q \in \mathbb{N}$ and $\alpha, \beta \in \mathbb{R}$, define the approximate inhomogeneous short integer solution problem $\mathsf{Approx.ISIS}_{n,m,q,\alpha,\beta}$ as follows: Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{y} \in \mathbb{Z}_q^n$, find a vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\| \leq \beta$, and there is a vector $\mathbf{z} \in \mathbb{Z}^n$ satisfying*

$$\|\mathbf{z}\| \leq \alpha \qquad \text{and} \qquad \mathbf{A}\mathbf{x} = \mathbf{y} + \mathbf{z} \pmod{q}.$$

Let us remark that the approximate ISIS is only non-trivial when the bounds α, β are relatively small compared to the modulus q. Also, our definition chooses to allow the zero vector to be a valid solution, which means when $\|\mathbf{y}\| \leq \alpha$, the zero vector is trivially a solution. Such a choice in the definition does not cause a problem in the application, since the interesting case in the application is to handle all the $\mathbf{y} \in \mathbb{Z}_q^n$, or $\mathbf{y}$ sampled uniformly random from $\mathbb{Z}_q^n$.

Definition 7 (Approximate trapdoor). *A string τ is called an (α, β)-approximate trapdoor for a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ if there is a probabilistic polynomial time algorithm (in n, m, $\log q$) that given τ, $\mathbf{A}$ and any $\mathbf{y} \in \mathbb{Z}_q^n$, outputs a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\| \leq \beta$, and there is a vector $\mathbf{z} \in \mathbb{Z}^n$ satisfying*

$$\|\mathbf{z}\| \leq \alpha \qquad \text{and} \qquad \mathbf{A}\mathbf{x} = \mathbf{y} + \mathbf{z} \pmod{q}.$$

3.1 Hardness of the approximate ISIS problem

To make sense of the approximate trapdoor, we argue that for those who do not have the trapdoor, the approximate ISIS problem is a candidate one-way function under proper settings of parameters.

First, we observe a rather obvious reduction that bases the hardness of solving approximate ISIS (given an arbitrary target) on the hardness of decisional LWE with low-norm secret (e.g. when the secret is sampled from the error distribution). In the theorem statement below, when the norm symbol is applied on a distribution D, i.e. $\|D\|$, it denotes the lowest value $v \in \mathbb{R}^+$ such that $\Pr_{d \leftarrow D}[\|d\| < v] > 1 - \mathsf{negl}(\lambda)$.

Theorem 2. *For $n, m, q \in \mathbb{Z}$, $\alpha, \beta \in \mathbb{R}^+$, θ, χ be distributions over $\mathbb{Z}$ such that $q > 4(\|\theta\| \cdot (\alpha + 1) + \|\theta^n\| \cdot \alpha \cdot \sqrt{n} + \|\chi^m\| \cdot \beta \cdot \sqrt{m})$. Then $\mathsf{LWE}_{n,m,q,\theta,U(\mathbb{Z}_q),\chi} \leq_p \mathsf{Approx.ISIS}_{n,m,q,\alpha,\beta}$.*

Proof. Suppose there is a polynomial time adversary A that breaks $\mathsf{Approx.ISIS}_{n,m,q,\alpha,\beta}$, we build a polynomial time adversary B that breaks decisional LWE.

Let $r = \lfloor \alpha \rceil + 1$. Given an LWE challenge $(\mathbf{A}, \mathbf{w}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$, where $\mathbf{w}$ is either an LWE sample or sampled uniformly from $\mathbb{Z}_q^m$. B picks a vector $\mathbf{y} := (r, 0, ..., 0)^t \in \mathbb{Z}_q^n$, sends $\mathbf{A}$ and $\mathbf{y}$ to the adversary A as an approximate ISIS challenge. A replies with $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\| \leq \beta$, and there is a vector $\mathbf{z} \in \mathbb{Z}^n$ satisfying

$$\|\mathbf{z}\| \leq \alpha \quad \text{and} \quad \mathbf{A}\mathbf{x} = \mathbf{y} + \mathbf{z} \pmod{q}.$$

Note that $\mathbf{x} \neq \mathbf{0}$ since $\|\mathbf{y}\| > \alpha$.

B then computes $v := \langle \mathbf{w}, \mathbf{x} \rangle$. If $\mathbf{w}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$ for $\mathbf{s} \leftarrow \theta^n$, $\mathbf{e} \leftarrow \chi^m$, then

$$v = (\mathbf{s}^t\mathbf{A} + \mathbf{e}^t)\mathbf{x} = \mathbf{s}^t(\mathbf{y} + \mathbf{z}) + \mathbf{e}^t\mathbf{x} \quad \Rightarrow$$
$$\|v\| \leq \|\theta\| \cdot r + \|\theta^n\| \cdot \alpha \cdot \sqrt{n} + \|\chi^m\| \cdot \beta \cdot \sqrt{m} < q/4.$$

Otherwise v distributes uniformly random over $\mathbb{Z}_q$. So B can compare v with the threshold value and wins the decisional LWE challenge with probability $1/2$ plus non-negligible.

Alternatively, we can also prove that the approximate ISIS problem is as hard as the standard ISIS. The reductions go through the HNFs of the ISIS and the approximate ISIS problems. All the reductions in the following theorem works for uniformly random target vectors.

Theorem 3. $\mathsf{ISIS}_{n,n+m,q,\beta} \geq_p \mathsf{Approx.ISIS}_{n,m,q,\alpha+\beta,\beta}$; $\mathsf{ISIS}_{n,n+m,q,\alpha+\beta} \leq_p \mathsf{Approx.ISIS}_{n,m,q,\alpha,\beta}$.

Proof. We will show $\mathsf{ISIS} = \mathsf{HNF.ISIS} = \mathsf{HNF.Approx.ISIS} = \mathsf{Approx.ISIS}$ under proper settings of parameters.

Recall that $\mathsf{ISIS}_{n,m,q,\beta} = \mathsf{HNF.ISIS}_{n,m,q,\beta}$ as explained in the preliminary. Also, $\mathsf{HNF.ISIS}_{n,m,q,\beta} \geq_p \mathsf{HNF.Approx.ISIS}_{n,m,q,\alpha,\beta}$ for any $\alpha \geq 0$ by definition. It remains to show the rest of the connections.

Lemma 8. $\mathsf{HNF.ISIS}_{n,m,q,\alpha+\beta} \leq_p \mathsf{HNF.Approx.ISIS}_{n,m,q,\alpha,\beta}$.

Proof. Suppose there is a polynomial time algorithm A that solves $\mathsf{HNF.Approx.ISIS}_{n,m,q,\alpha,\beta}$, we build a polynomial time algorithm B that solves $\mathsf{HNF.ISIS}_{n,m,q,\alpha+\beta}$. Given an $\mathsf{HNF.ISIS}$ instance $[\mathbf{I}_n \mid \mathbf{A}] \in \mathbb{Z}_q^{n \times m}$, $\mathbf{y}$, B passes the same instance to A, gets back a vector $\mathbf{x}$ such that

$$[\mathbf{I}_n \mid \mathbf{A}] \cdot \mathbf{x} = \mathbf{y} + \mathbf{z} \pmod{q}.$$

where $\|\mathbf{x}\| \leq \beta$, $\|\mathbf{z}\| \leq \alpha$. Now write $\mathbf{x} =: [\mathbf{x}_1^t \mid \mathbf{x}_2^t]^t$ where $\mathbf{x}_1 \in \mathbb{Z}^n$, $\mathbf{x}_2 \in \mathbb{Z}^m$. Then $\mathbf{x}' := [(\mathbf{x}_1 - \mathbf{z})^t \mid \mathbf{x}_2^t]^t$ satisfies

$$[\mathbf{I}_n \mid \mathbf{A}] \cdot \mathbf{x}' = \mathbf{y} \pmod{q},$$

and $\|\mathbf{x}'\| \leq \alpha + \beta$. So $\mathbf{x}'$ is a valid solution to $\mathsf{HNF.ISIS}$.

Lemma 9. $\mathsf{HNF.Approx.ISIS}_{n,n+m,q,\alpha,\beta} \leq_p \mathsf{Approx.ISIS}_{n,m,q,\alpha,\beta}$.

Proof. Suppose there is a polynomial time algorithm A that solves $\mathsf{Approx.ISIS}_{n,m,q,\alpha,\beta}$, we build a polynomial time algorithm B that solves $\mathsf{HNF.Approx.ISIS}_{n,n+m,q,\alpha,\beta}$. Given $[\mathbf{I}_n \mid \mathbf{A}] \in \mathbb{Z}_q^{n\times(n+m)}$, $\mathbf{y} \in \mathbb{Z}_q^n$ as an $\mathsf{HNF.Approx.ISIS}$ instance, B passes $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{y}$ to A, gets back a short vector $\mathbf{x} \in \mathbb{Z}^m$. Then $[\mathbf{0}_n^t \mid \mathbf{x}^t]^t$ is a valid solution to the $\mathsf{HNF.Approx.ISIS}$ instance.

Lemma 10. $\mathsf{HNF.Approx.ISIS}_{n,n+m,q,\alpha,\beta} \geq_p \mathsf{Approx.ISIS}_{n,m,q,\alpha+\beta,\beta}$.

Proof. Suppose there is a polynomial time algorithm A that solves $\mathsf{HNF.Approx.ISIS}_{n,n+m,q,\alpha,\beta}$, we build a polynomial time algorithm B that solves $\mathsf{Approx.ISIS}_{n,m,q,\alpha+\beta,\beta}$. Given an $\mathsf{Approx.ISIS}$ instance $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{y} \in \mathbb{Z}^n$, B passes $[\mathbf{I}_n \mid \mathbf{A}] \in \mathbb{Z}_q^{n\times(n+m)}$, $\mathbf{y}$ as an $\mathsf{HNF.Approx.ISIS}$ instance to A, gets back an answer $\mathbf{x} \in \mathbb{Z}^{m+n}$ such that

$$[\mathbf{I}_n \mid \mathbf{A}] \cdot \mathbf{x} = \mathbf{y} + \mathbf{z} \pmod{q}, \tag{2}$$

where $\|\mathbf{x}\| \leq \beta$, $\|\mathbf{z}\| \leq \alpha$.

Now write $\mathbf{x} =: [\mathbf{x}_1^t \mid \mathbf{x}_2^t]^t$ where $\mathbf{x}_1 \in \mathbb{Z}^n$, $\mathbf{x}_2 \in \mathbb{Z}^m$. Rewriting Eqn. (2) gives

$$\mathbf{A} \cdot \mathbf{x}_2 = \mathbf{y} + \mathbf{z} - \mathbf{x}_1 \pmod{q},$$

so $\mathbf{x}_2$ is a valid solution to $\mathsf{Approx.ISIS}_{n,m,q,\alpha+\beta,\beta}$.

Theorem 3 then follows the lemmas above.

The following statement immediately follows the proof of Lemma 10.

Corollary 2. *An (α, β)-approximate trapdoor for $[\mathbf{I} \mid \mathbf{A}]$ is an $(\alpha + \beta, \beta)$-approximate trapdoor for $\mathbf{A}$.*

4 Approximate Gadget Trapdoor

We present an instantiation of an approximate trapdoor based on the gadget-based trapdoor generation and preimage sampling algorithms of Micciancio and Peikert [42] (without the tag matrices). In short, we show how to generate a pseudorandom $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ along with an approximate trapdoor $\mathbf{R}$ with small integer entries.

In the rest of this section, we first recall the exact G-trapdoor from [42], then present the approximate trapdoor generation algorithm and the approximate preimage sampling algorithm. Finally we show that the preimage and the error distributions for uniformly random targets are simulatable.

4.1 Recall the G-trapdoor from [42]

Let $b \geq 2$ be the base for the G-lattice. Let q be the modulus, $k = \lceil \log_b q \rceil$. b is typically chosen to be 2 for simplicity, but often a higher base b is used for efficiency trade-offs in lattice-based schemes.

Recall the gadget-lattice trapdoor technique from [42]: the public matrix is

$$\mathbf{A} = [\bar{\mathbf{A}} | \mathbf{G} - \bar{\mathbf{A}}\mathbf{R}]$$

where $\mathbf{G}$ is the commonly used gadget matrix, $\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g}_k^t$, $\mathbf{g}_k^t := (1, b, \ldots, b^{k-1})$, and $\mathbf{R}$ is a secret, trapdoor matrix with small, random entries. $\mathbf{A}$ is either statistically close to uniformly random or pseudorandom, depending on the structure of $\bar{\mathbf{A}}$ and the choice of χ (in the pseudorandom case $\chi \subseteq \mathbb{Z}$ is chosen to be a distribution such that $\mathsf{LWE}_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$ is hard). In this paper we focus on the pseudorandom case since the resulting public matrix $\mathbf{A}$ and preimage have smaller dimensions.

In order to sample a short element in $\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A})$, we use the trapdoor to map short coset representatives of $\Lambda_q^{\perp}(\mathbf{G})$ to short coset representatives of $\Lambda_q^{\perp}(\mathbf{A})$ by the relation

$$\mathbf{A} \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{G}.$$

Using the trapdoor as a linear transformation alone leaks information about the trapdoor. Therefore, we perturb the sample to statistically hide the trapdoor. Let Σ_p be a positive definite matrix defined as $\Sigma_p := s^2\mathbf{I} - \sigma^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^t & \mathbf{R}^t \\ \mathbf{R} & \mathbf{I} \end{bmatrix}$ where σ is at least $\eta_\varepsilon(\Lambda_q^{\perp}(\mathbf{G}))$. The perturbation can be computed offline as $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma_p}}$. We then sample a G-lattice vector in a coset dependent on $\mathbf{p}$ as $\mathbf{z} \leftarrow D_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}),\sigma}$ and $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p} \in \mathbb{Z}_q^n$. Finally, the preimage is set to be

$$\mathbf{y} := \mathbf{p} + \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z}.$$

4.2 The algorithms of the approximate G-trapdoor

As mentioned in the introduction, the main idea of obtaining an approximate trapdoor is to adapt the algorithms from [42] with a gadget matrix without the lower-order entries. Let $0 < l < k$ be the number of lower-order entries dropped from the gadget vector $\mathbf{g} \in \mathbb{Z}_q^k$. Define the resulting approximate gadget vector as $\mathbf{f} := (b^l, b^{l+1}, ..., b^{k-1})^t \in \mathbb{Z}_q^{(k-l)}$. Let $w = n(k-l)$ be the number of columns of the approximate gadget $\mathbf{F} := \mathbf{I}_n \otimes \mathbf{f}^t \in \mathbb{Z}^{n \times w}$. Then the number of columns of $\mathbf{A}$ will be $m := 2n + w$.

Once we replace the gadget matrix $\mathbf{G}$ with its truncated version, $\mathbf{F}$, our approximate trapdoor generation and approximate preimage sampling algorithms match the original gadget-based algorithms. The generation and preimage algorithms are given as Algorithms 2 and 3, respectively. Algorithm 1 represents our approximate $\mathbf{F}$-sampling algorithm. It simply runs the G-lattice preimage sampling algorithm and drops the first l entries from the preimage. The covariance of the perturbation in Algorithm 3 is chosen as

$$\Sigma_p := s^2\mathbf{I}_m - \sigma^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^t & \mathbf{R} \\ \mathbf{R}^t & \mathbf{I} \end{bmatrix}.$$

Algorithm 1: GSAMP.CUT(v, σ)

Input: $v \in \mathbb{Z}_q$, $\sigma \in \mathbb{R}^+$
Output: $\mathbf{z} \in \mathbb{Z}^{k-l}$

Sample $\mathbf{x} \in \mathbb{Z}^k$ from $D_{\Lambda_v^\perp(\mathbf{g}^t),\sigma}$
Let $\mathbf{z}$ be the last $k-l$ entries of $\mathbf{x}$
return $\mathbf{z}$.

Algorithm 2: APPROX.TRAPGEN$_\chi$

Input: Security parameter λ
Output: matrix-approximate trapdoor pair $(\mathbf{A}, \mathbf{R})$.

Sample a uniformly random $\hat{\mathbf{A}} \leftarrow U(\mathbb{Z}_q^{n \times n})$.
Let $\bar{\mathbf{A}} := [\mathbf{I}_n, \hat{\mathbf{A}}]$.
Sample the approximate trapdoor $\mathbf{R} \leftarrow \chi^{2n \times w}$.
Form $\mathbf{A} := [\bar{\mathbf{A}} | \mathbf{F} - \bar{\mathbf{A}}\mathbf{R}] \in \mathbb{Z}_q^{n \times m}$.
return $(\mathbf{A}, \mathbf{R})$.

Algorithm 3: APPROX.SAMPLEPRE.

Input: $(\mathbf{A}, \mathbf{R}, \mathbf{u}, s)$ as in Thm. 4.
Output: An approximate preimage of $\mathbf{u}$ for $\mathbf{A}$, $\mathbf{y} \in \mathbb{Z}^m$.

Sample a perturbation $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma_p}}$.
Form $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p} \in \mathbb{Z}_q^n$.
Sample the approximate gadget preimage $\mathbf{z} \in \mathbb{Z}^{n(k-l)}$ as $\mathbf{z} \leftarrow$ GSAMP.CUT$(\mathbf{v}, \sigma)$.
Form $\mathbf{y} := \mathbf{p} + \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z} \in \mathbb{Z}^m$.
return $\mathbf{y}$.

Fig. 3. Pseudocode for the approximate trapdoor sampling algorithm in Subsection 4.3. We abuse notation and let GSAMP.CUT$(\mathbf{v}, \sigma)$ denote n independent calls to Algorithm 1 on each entries of $\mathbf{v} \in \mathbb{Z}_q^n$, and then concatenate the output vectors. The distribution $\chi \subseteq \mathbb{Z}$ is chosen so that $\mathsf{LWE}_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$ is hard.

The results of this section are summarized in the following theorem.

Theorem 4. *There exists probabilistic, polynomial time algorithms* APPROX.TRAPGEN$(\cdot)$ *and* APPROX.SAMPLEPRE$(\cdot,\cdot,\cdot,\cdot)$ *satisfying the following.*

1. APPROX.TRAPGEN(n) *takes as input a security parameter* n *and returns a matrix-approximate trapdoor pair* $(\mathbf{A}, \mathbf{R}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{2n \times n(k-l)}$.
2. *Let* $\mathbf{A}$ *be generated with an approximate trapdoor as above and let* APPROX.$\mathbf{A}^{-1}(\cdot)$ *denote the approximate preimage sampling algorithm,* APPROX.SAMPLEPRE$(\mathbf{A}, \mathbf{R}, s, \cdot)$. *The following two distributions are statistically indistinguishable:*

$$\{(\mathbf{A}, \mathbf{y}, \mathbf{u}, \mathbf{e}) : \quad \mathbf{u} \leftarrow U(\mathbb{Z}_q^n), \quad \mathbf{y} \leftarrow \text{APPROX.}\mathbf{A}^{-1}(\mathbf{u}), \quad \mathbf{e} = \mathbf{u} - \mathbf{A}\mathbf{y} \mod q\}$$

and

$$\{(\mathbf{A}, \mathbf{y}, \mathbf{u}, \mathbf{e}) : \mathbf{y} \leftarrow D_{\mathbb{Z}^m, s}, \mathbf{e} \leftarrow D_{\mathbb{Z}^n, \sigma\sqrt{(b^{2l}-1)/(b^2-1)}} \mod q, \mathbf{u} = \mathbf{A}\mathbf{y} + \mathbf{e} \mod q\}$$

for any $\sigma \geq \sqrt{b^2+1} \cdot \omega(\sqrt{\log n})$ *and* $s \gtrsim \sqrt{b^2+1}\frac{s_1^2(\mathbf{R})}{s_{2n}(\mathbf{R})}\eta_\epsilon(\mathbb{Z}^{nk})$[3]. *Furthermore, in the second distribution,* $\mathbf{A}$ *is computationally indistinguishable from random assuming* $\mathsf{LWE}_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$.

4.3 Simulate the preimage and error distributions

This subsection is dedicated to proving Theorem 4. For the convenience of explanation, in this subsection we redefine the gadget $\mathbf{G}$ by permuting the columns so that the columns of smaller entries are all on the left, i.e.

$$\mathbf{G} := [\mathbf{M}|\mathbf{F}] := [\mathbf{I}_n \otimes (1, b, \ldots, b^{l-1})|\mathbf{F}]$$

Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{Z}^{nl} \times \mathbb{Z}^{n(k-l)}$ denote the short preimage of $\mathbf{v} := \mathbf{u} - \mathbf{A}\mathbf{p} \pmod q$ under the full gadget matrix $\mathbf{G}$, i.e. $\mathbf{G}\mathbf{x} = \mathbf{v} \pmod q$.

The first attempt of proving Theorem 4 is to first show that the joint distribution of $(\mathbf{p}, \mathbf{x})$ produced in Algorithm 3 is statistically close to $D_{\Lambda^\perp_{\mathbf{u}}[\mathbf{A},\mathbf{G}],\sqrt{\Sigma_p \oplus \sigma^2 \mathbf{I}_{nk}}}$ for *any* $\mathbf{u} \in \mathbb{Z}_q^n$, then apply the linear transformation theorem on $(\mathbf{p}, \mathbf{x})$ to obtain the distributions of the preimage $\mathbf{y}$ and the error term $\mathbf{e}$. However, applying the linear transformation theorem directly on the lattice coset $\Lambda^\perp_{\mathbf{u}}[\mathbf{A}, \mathbf{G}]$ leads to a technical problem. That is, the intermediate lattice intersections $\Lambda_{\mathbf{T}}$ required in Theorem 1 have large smoothing parameters, which means even if we go through that route, the Gaussian width of the resulting preimage would blow up significantly.

Instead, we work only with a uniformly random target $\mathbf{u}$ instead of an arbitrary target, and directly construct the simulation algorithm. We show that if the simulation algorithm produces $(\mathbf{p}, \mathbf{x}) \leftarrow D_{\mathbb{Z}^{m+nk},\sqrt{\Sigma_p \oplus \sigma^2 \mathbf{I}_{nk}}}$, then it is able to simulate the distributions of $\mathbf{y}$ and $\mathbf{e}$ correctly without using the trapdoor. Now the support of $(\mathbf{p}, \mathbf{x})$ is the integer lattice $\mathbb{Z}^{m+nk}$. Working with the integer lattice is important for two reasons. First, it allows us to treat $\mathbf{x}_1$ and $\mathbf{x}_2$ as statistically independent samples; and second, it gives us short vectors in the kernels summoned when using Lemma 4 or Theorem 1.

Formally, let $\varepsilon = \mathrm{negl}(\lambda) > 0$. We first prove three lemmas.

Lemma 11. *For any* $\sigma \geq \eta_\varepsilon(\Lambda^\perp(\mathbf{G}))$*, the following two distributions are statistically close.*

1. *First sample* $\mathbf{v} \leftarrow U(\mathbb{Z}_q^n)$*, then sample* $\mathbf{x} \leftarrow D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),\sigma}$*, output* $(\mathbf{x}, \mathbf{v})$*;*
2. *First sample* $\mathbf{x} \leftarrow D_{\mathbb{Z}^{nk},\sigma}$*, then compute* $\mathbf{v} = \mathbf{G}\mathbf{x} \pmod q$*, output* $(\mathbf{x}, \mathbf{v})$*.*

Proof. The proof follows directly from $\det(\Lambda^\perp_q(\mathbf{G})) = q^n$ and Corollary 1. Alternatively, one can use two applications of the fact $\rho_r(\Gamma + \mathbf{c}) \in (1 \pm \varepsilon)\sigma^n/\det(\Gamma)$ for any $r \geq \eta_\varepsilon(\Gamma)$. The latter yields $\Pr\{\text{Process returns } \mathbf{x}\} \in \left(\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right) \cdot D_{\mathbb{Z}^{nk},\sigma}(\mathbf{x})$.

[3] We remark that the ratio $\frac{s_1(\mathbf{R})}{s_{2n}(\mathbf{R})}$ is a small constant for commonly-used subgaussian distributions for $\mathbf{R}$'s entries [51].

Lemma 12. *The following random processes are statistically close for any* $\sigma \geq \sqrt{b^2+1}\cdot\omega(\sqrt{\log n}) \geq \eta_\varepsilon(\mathbf{g}^t)$*: sample* $\mathbf{x}_1 \leftarrow D_{\mathbb{Z}^l,\sigma}$ *and return* $e = [1, b, \ldots, b^{l-1}]\mathbf{x}_1$*; or, return* $e \leftarrow D_{\mathbb{Z},\sigma\sqrt{(b^{2l}-1)/(b^2-1)}}$.

Proof. We use Lemma 4 or Theorem 1 where $[1, b, \ldots, b^{l-1}]$ is the linear transformation. Notice that the kernel of $[1, b, \ldots, b^{l-1}]$ is the linear span of $[\mathbf{b}_1, \ldots, \mathbf{b}_{l-1}]$ where

$$\mathbf{b}_1 = (b, -1, 0, \ldots, 0), \mathbf{b}_2 = (0, b, -1, 0, \ldots, 0), \ldots, \mathbf{b}_{l-1} = (0, \ldots, 0, b, -1) \in \mathbb{Z}^l.$$

The support of $\mathbf{x}_1$, $\mathbb{Z}^l$, contains the $(l-1)$-dimensional lattice, $\Gamma = \mathbb{Z}^l \cap \mathrm{Ker}([1, b, \ldots, b^{l-1}])$, spanned by $[\mathbf{b}_1, \ldots, \mathbf{b}_{l-1}]$. Further, $\sigma \geq \eta_\epsilon(\mathbf{g}^t)$ implies σ is larger than the smoothing parameter of Γ since $\|\mathbf{b}_i\| \leq \sqrt{b^2+1}$ for $i = 1, \ldots, l-1$. Finally by routine calculation on the Gaussian width (and support), we have $e = [1, b, \ldots, b^{l-1}]\mathbf{x}_1 \approx_s D_{\mathbb{Z},\sigma\sqrt{(b^{2l}-1)/(b^2-1)}}$.

Let $\mathbf{R}' := \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{n(k-l)} \end{bmatrix}$. Next, we analyze the distribution given by the linear transformation representing the convolution step:

$$\mathbf{y} = \mathbf{p} + \mathbf{R}'\mathbf{x}_2 = [\mathbf{I}_m | \mathbf{R}'] \begin{pmatrix} \mathbf{p} \\ \mathbf{x}_2 \end{pmatrix}$$

for $(\mathbf{p}, \mathbf{x}_2) \leftarrow D_{\mathbb{Z}^{m+n(k-l)},\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}}}$. Let $\mathbf{L} := [\mathbf{I}_m | \mathbf{R}']$ in Lemma 13 and its proof below.

Lemma 13. *For* $\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}} \geq \eta_\varepsilon\left(\Lambda\begin{pmatrix} \mathbf{R}' \\ -\mathbf{I}_{n(k-l)} \end{pmatrix}\right)$, $\mathbf{L}D_{\mathbb{Z}^{m+n(k-l)},\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}}}$ *is statistically close to* $D_{\mathbb{Z}^m,s}$*. Further,* $\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}} \geq \eta_\varepsilon\left(\Lambda\begin{pmatrix} \mathbf{R}' \\ -\mathbf{I}_{n(k-l)} \end{pmatrix}\right)$ *is satisfied when* $s \gtrsim \sqrt{b^2+1}\frac{s_1^2(\mathbf{R})}{s_{2n}(\mathbf{R})}\eta_\epsilon(\mathbb{Z}^{nk})$.

Proof. The range and covariance are immediate. Next, we use Theorem 1. The kernel of $\mathbf{L}$ is given by all vectors $(\mathbf{a}, \mathbf{b})$ where $\mathbf{b} \in \mathbb{R}^{n(k-l)}$ and $\mathbf{a} = -\mathbf{R}'\mathbf{b}$. The integer lattice $\mathbb{Z}^{m+n(k-l)}$ contains all such integer vectors so $\Lambda_\mathbf{L} := \mathbb{Z}^{m+n(k-l)} \cap \ker(\mathbf{L})$ spans $\mathbf{L}$'s kernel. So $\begin{pmatrix} \mathbf{R}' \\ -\mathbf{I}_{n(k-l)} \end{pmatrix}$ is a basis of $\Lambda_\mathbf{L}$. Given that $\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}} \geq \eta_\varepsilon\left(\Lambda\begin{pmatrix} \mathbf{R}' \\ -\mathbf{I}_{n(k-l)} \end{pmatrix}\right)$, the lemma follows Theorem 1. Lastly, the implication that $\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}} \geq \eta_\varepsilon\left(\Lambda\begin{pmatrix} \mathbf{R}' \\ -\mathbf{I}_{n(k-l)} \end{pmatrix}\right)$ whenever $s \gtrsim \sqrt{b^2+1}\frac{s_1^2(\mathbf{R})}{s_{2n}(\mathbf{R})}\eta_\epsilon(\mathbb{Z}^{nk})$ is proved in Appendix A.

We are now ready to prove Theorem 4.

Proof. (of Theorem 4) The proof's overview is given via the following. Let

- $\mathbf{p} \leftarrow D_{\mathbb{Z}^m,\sqrt{\Sigma_p}}$ be a perturbation,

- $\mathbf{u} \in \mathbb{Z}_q^n$ be the input target coset,
- $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p} \in \mathbb{Z}_q^n$ be the G-lattice coset,
- $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \leftarrow D_{\mathbb{Z}^{nk},\sigma}$ (G-lattice randomized over uniform coset $\mathbf{v}$ and $\sigma \geq \eta_\epsilon(\mathbf{g}^t)$, Lemma 11)
- $\mathbf{e} \leftarrow D_{\mathbb{Z}^n,\sigma\sqrt{(b^{2l}-1)/(b^2-1)}}$ be the concatenation of the errors, e, in Lemma 12,
- and $\mathbf{y} \leftarrow D_{\mathbb{Z}^m,s}$ as in Lemma 13.

The proof is best summarized via the sequence of hybrids below:

$$\begin{aligned}
\mathbf{u} &= \mathbf{v} + \mathbf{A}\mathbf{p} \\
&\approx_s \mathbf{G}\mathbf{x} + \mathbf{A}\mathbf{p} \\
&= \mathbf{M}\mathbf{x}_1 + \mathbf{F}\mathbf{x}_2 + \mathbf{A}\mathbf{p} \\
&\approx_s \mathbf{e} + \mathbf{F}\mathbf{x}_2 + \mathbf{A}\mathbf{p} \\
&= \mathbf{e} + \mathbf{A}\mathbf{R}'\mathbf{x}_2 + \mathbf{A}\mathbf{p} \\
&= \mathbf{e} + \mathbf{A}\mathbf{L}\begin{pmatrix}\mathbf{p}\\ \mathbf{x}_2\end{pmatrix} \\
&\approx_s \mathbf{e} + \mathbf{A}\mathbf{y}.
\end{aligned}$$

The first $\approx_s$ is through swapping the order of sampling $\mathbf{u}$ and $\mathbf{v}$ uniformly at random, then using the fact that $\sigma \geq \eta_\epsilon(\mathbf{G})$ (Lemma 11). The next $\approx_s$ is given by Lemma 12. Finally, the last $\approx_s$ is given by concatenating $(\mathbf{p}, \mathbf{x}_2) \leftarrow D_{\mathbb{Z}^{m+n(k-l)},\sqrt{\Sigma_p \oplus \sigma^2 \mathbf{I}_{n(k-l)}}}$ and using Lemma 13.

We remark that the key in the equivalences above is that we can separate $\mathbf{x}$ into two statistically independent samples, $\mathbf{x}_1$ and $\mathbf{x}_2$, concatenate $\mathbf{p}$ and $\mathbf{x}_2$, then perform two instances of Theorem 1 (Lemma 4) on the statistically independent samples $\mathbf{L}(\mathbf{p}, \mathbf{x}_2)$ and $\mathbf{M}\mathbf{x}_1$. The statistical independence of $\mathbf{x}_1$ and $\mathbf{x}_2$ is due to the orthogonality of $\mathbb{Z}^{nk}$ and the same cannot be said if $\mathbf{x} \sim D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),\sigma}$ for a fixed $\mathbf{v}$ (via a fixed $\mathbf{u}$). This difference highlights why we must argue security for a uniformly random input coset $\mathbf{u}$ (and $\mathbf{v}$).

Real distribution: The real distribution of $\{(\mathbf{A}, \mathbf{y}, \mathbf{u}, \mathbf{e})\}$ is:
$\mathbf{A}$, $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{p} \leftarrow D_{\mathbb{Z}^m,\sqrt{\Sigma_p}}$, $\mathbf{v} := \mathbf{u} - \mathbf{A}\mathbf{p}$, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \leftarrow D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),\sigma}$, $\mathbf{e} = \mathbf{M}\mathbf{x}_1$, and $\mathbf{y} = \mathbf{L}(\mathbf{p}, \mathbf{x}_2)$.

Hybrid 1: Here we swap the order of sampling $\mathbf{u}$ and $\mathbf{v}$. Let $\mathbf{v} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{p} \leftarrow D_{\mathbb{Z}^m,\sqrt{\Sigma_p}}$, $\mathbf{u} = \mathbf{v} + \mathbf{A}\mathbf{p}$. We keep $\mathbf{x}, \mathbf{e}$, and $\mathbf{y}$ unchanged: $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \leftarrow D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),\sigma}$, $\mathbf{e} = \mathbf{M}\mathbf{x}_1$, and $\mathbf{y} = \mathbf{L}(\mathbf{p}, \mathbf{x}_2)$. Then, the real distribution and Hybrid 1 are the same.

Hybrid 2: Instead of sampling a uniform $\mathbf{v} \in \mathbb{Z}_q^n$ and a G-lattice sample $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \leftarrow D_{\Lambda^\perp_{\mathbf{v}}(\mathbf{G}),\sigma}$, we sample $\mathbf{x} \leftarrow D_{\mathbb{Z}^{nk},\sigma}$ and let $\mathbf{v} = \mathbf{G}\mathbf{x} \in \mathbb{Z}_q^n$. The rest remains the same:
$\mathbf{A}$, $\mathbf{x} \leftarrow D_{\mathbb{Z}^{nk},\sigma}$, $\mathbf{v} = \mathbf{G}\mathbf{x}$, $\mathbf{p} \leftarrow D_{\mathbb{Z}^m,\sqrt{\Sigma_p}}$, $\mathbf{u} = \mathbf{v} + \mathbf{A}\mathbf{p}$, $\mathbf{e} = \mathbf{M}\mathbf{x}_1$, and $\mathbf{y} = \mathbf{L}(\mathbf{p}, \mathbf{x}_2)$. Lemma 11 implies Hybrid 1 and Hybrid 2 are statistically close.

Hybrid 3: We combine $\mathbf{p}, \mathbf{x}_2$ into the joint distribution $(\mathbf{p}, \mathbf{x}_2) \leftarrow D_{\mathbb{Z}^{m+n(k-l)}, \sqrt{\Sigma_p \oplus \sigma^2 \mathbf{I}}}$: $\mathbf{A}$, $(\mathbf{p}, \mathbf{x}_2) \leftarrow D_{\mathbb{Z}^{m+n(k-l)}, \sqrt{\Sigma_p \oplus \sigma^2 \mathbf{I}}}$, $\mathbf{e} = \mathbf{M}\mathbf{x}_1$, $\mathbf{y} = \mathbf{L}(\mathbf{p}, \mathbf{x}_2)$, $\mathbf{v} = \mathbf{G}\mathbf{x}$, and $\mathbf{u} = \mathbf{v} + \mathbf{A}\mathbf{p}$.

Hybrid 4: Here we apply the linear transformation theorem on $\mathbf{L}$ and $\mathbf{M}$. $\mathbf{A}$, $\mathbf{e} \leftarrow D_{\mathbb{Z}^{nl}, \sigma\sqrt{(b^{2l}-1)/(b^2-1)}}$, $\mathbf{y} \leftarrow D_{\mathbb{Z}^m, s}$, $\mathbf{v} = \mathbf{A}\mathbf{y} + \mathbf{e}$.
Lemmas 12 and 13 imply Hybrids 3 and 4 are statistically close.

Final distribution: Sample $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n \times m})$ and keep the rest of the vectors from the same distribution as Hybrid 4 (notice that the trapdoor $\mathbf{R}$ of $\mathbf{A}$ is not used to sample $\mathbf{p}$, $\mathbf{x}$, $\mathbf{e}$ and $\mathbf{y}$). The final distribution is computationally indistinguishable from Hybrid 4 assuming $\mathsf{LWE}_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$.

5 Hash-and-Sign Signature Instantiated with the Approximate Trapdoor

We spell out the details of the hash-and-sign signature scheme from [31] instantiated with the approximate G-trapdoor instead of an exact trapdoor.

Recall the parameters from the last section. We set $k = \lceil \log_b q \rceil$, set l to be the number of entries dropped from the G-trapdoor such that $1 \leq l < k$ and $m = n(2 + (k - l))$. Let $\sigma, s \in \mathbb{R}^+$ be the discrete Gaussian widths of the distributions over the cosets of $\Lambda_q^\perp(\mathbf{G})$ and $\Lambda_q^\perp(\mathbf{A})$ respectively. Let χ be the distribution of the entries of the trapdoor $\mathbf{R}$ chosen so that $\mathsf{LWE}_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$ is hard.

Construction 5 *Given an approximate trapdoor sampler from Theorem 4, a hash function $H = \{H_\lambda : \{0,1\}^* \to R_\lambda\}$ modeled as a random oracle, we build a signature scheme as follows.*

- $\mathsf{Gen}(1^\lambda)$*: The key-generation algorithm samples $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with its (α, β)-approximate trapdoor $\mathbf{R}$ from* $\textsc{Approx.TrapGen}(1^\lambda)$*. Let the range R_λ of H be $\mathbb{Z}_q^n$. It outputs $\mathbf{A}$ as the verification key, keeps $\mathbf{R}$ as the secret signing key.*
- $\mathsf{Sig}(\mathbf{R}, m)$*: The signing algorithm checks if the message-signature pair $(m, \mathbf{x}_m)$ has been produced before. If so, it outputs $\mathbf{x}_m$ as the signature of m; if not, computes $\mathbf{u} = H(m)$, and samples an approximate preimage $\mathbf{x}_m \leftarrow$* $\textsc{Approx.SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{u}, s)$*. It outputs $\mathbf{x}_m$ as the signature and stores $(m, \mathbf{x}_m)$ in the list.*
- $\mathsf{Ver}(\mathbf{A}, m, \mathbf{x})$*: The verification algorithm checks if $\|\mathbf{x}\| \leq \beta$ and $\|\mathbf{A} \cdot \mathbf{x} - H(m)\| \leq \alpha$. If so, it outputs accept; otherwise, it outputs reject.*

5.1 Security analysis

In the security analysis we use the following property on the distributions produced by $\textsc{Approx.SamplePre}$ proven in Theorem 4. That is, the preimage and

error term for a random target can be simulated from distributions denoted by D_{pre} and D_{err}. Both of them are independent of the public key $\mathbf{A}$ and the secret key $\mathbf{R}$.

To prove that the signature satisfies the strong EU-CMA security, we need an additional "near-collision-resistance" property for Ajtai's function, which can be based on the standard SIS assumption. Let us remark that without this property, we can still prove the signature scheme satisfies static security based on the hardness of the approximate ISIS problem, which is tighter by a factor of two according to Theorem 3.

Lemma 14 (The near-collision-resistance of Ajtai's function). *For any $n, m, q \in \mathbb{N}$ and $\alpha, \beta \in \mathbb{R}$. If there is an efficient adversary A that given $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$, finds $\mathbf{x}_1 \neq \mathbf{x}_2 \in \mathbb{Z}^m$ such that*

$$\|\mathbf{x}_1\| \leq \beta \quad \text{and} \quad \|\mathbf{x}_2\| \leq \beta \quad \text{and} \quad \|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2 \pmod q\| \leq 2\alpha$$

Then there is an efficient adversary B that solves $\mathsf{SIS}_{n,n+m,q,2(\alpha+\beta)}$.

Proof. Suppose B gets an $\mathsf{HNF.SIS}_{n,n+m,q,2(\alpha+\beta)}$ challenge (which is as hard as $\mathsf{SIS}_{n,n+m,q,2(\alpha+\beta)}$) with the public matrix $[\mathbf{I}_n \mid \mathbf{A}]$, B sends $\mathbf{A}$ to A, gets back $\mathbf{x}_1 \neq \mathbf{x}_2 \in \mathbb{Z}^m$ such that

$$\|\mathbf{x}_1\| \leq \beta \quad \text{and} \quad \|\mathbf{x}_2\| \leq \beta \quad \text{and} \quad \|\mathbf{y} := \mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2 \pmod q\| \leq 2\alpha$$

B then sets $\mathbf{z} := [-\mathbf{y}^t \mid (\mathbf{x}_1 - \mathbf{x}_2)^t]^t$ as the solution. $\mathbf{z}$ is then non-zero and satisfies $\|\mathbf{z}\| \leq 2(\alpha+\beta)$ and $[\mathbf{I}_n \mid \mathbf{A}]\mathbf{z} = \mathbf{0} \pmod q$.

Theorem 6. *Construction 5 is strongly existentially unforgeable under a chosen-message attack in the random oracle model assuming the hardness of $\mathsf{SIS}_{n,n+m,q,2(\alpha+\beta)}$ and $\mathsf{LWE}_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$.*

Proof. Suppose there is a polynomial time adversary A that breaks the strong EU-CMA of the signature scheme, we construct a polynomial time adversary B that breaks the near-collision-resistance of Ajtai's function, which is as hard as $\mathsf{SIS}_{n,n+m,q,2(\alpha+\beta)}$ due to Lemma 14.

To start, B sends Ajtai's function $\mathbf{A}$ to A as the public key for the signature scheme. Once A makes a random oracle query w.r.t. a message m, B samples $\mathbf{x} \leftarrow D_{\text{pre}}$, computes $\mathbf{u} := \mathbf{A}\mathbf{x} + D_{\text{err}} \pmod q$ as the random oracle response on m. B then replies $\mathbf{u}$ to A and stores $(m, \mathbf{u})$ in the random oracle storage, $(m, \mathbf{x})$ in the message-signature pair storage. Once A makes a signing query on the message m (wlog assume m has been queried to the random oracle before, since if not B can query it now), B finds $(m, \mathbf{x})$ in the storage and reply $\mathbf{x}$ as the signature. The signatures and the hash outputs produced by B are indistinguishable from the real ones due to the properties of the distributions D_{pre} and D_{err}, and the assumption that a real public key is indistinguishable from random under $\mathsf{LWE}_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$.

Without loss of generality, assume that before A tries to forge a signature on m^*, A has queried H on m^*. Denote the pair that B prepares and stores

in the random oracle storage as $(m^*, \mathbf{u}^*)$, and the pair in the signature storage as $(m^*, \mathbf{x}^*)$. Finally A outputs $\mathbf{x}$ as the forged signature on m^*. So we have $\|\mathbf{A}(\mathbf{x} - \mathbf{x}^*) \pmod q\| \leq 2\alpha$. It remains to prove that $\mathbf{x} \neq \mathbf{x}^*$ so as to use them as a near-collision-pair. If m^* has been queried to the signing oracle before, then $\mathbf{x} \neq \mathbf{x}^*$ by the definition of a successful forgery; if m^* has not been queried to the signing oracle before, then $\mathbf{x}^*$ is with high min-entropy by the settings of the parameter, so $\mathbf{x} \neq \mathbf{x}^*$ with overwhelming probability.

5.2 Concrete parameters

We provide a proof-of-concept implementation of the signature. Experiments are performed over several groups of parameters using different dimensions n, moduli q, bases b, targeting different security level (mainly around 80 to 90-bit and 170 to 185-bit security). In each group of parameters, we use fixed n, q, b, and compare the use of exact trapdoor (under our reference implementation) versus approximate trapdoor. In Figures 4 and 5 we list 6 groups of parameters.

Params	Exact	Approx	Approx	Exact	Approx	Approx	Exact	Approx	Approx
n	512	512	512	512	512	512	512	512	512
$\lceil \log_2 q \rceil$	24	24	24	20	20	20	16	16	16
b	2	2	2	2	2	2	2	2	2
l	0	12	15	0	10	12	0	7	9
τ	40	40	40	10	10	10	2.6	2.6	2.6
s	38317.0	29615.3	26726.3	8946.4	6919.8	6416.4	2170.7	1756.3	1618.2
m	13312	7168	5632	11264	6144	5120	9216	5632	4608
$\|\mathbf{x}\|_2$	4441737.7	2521387.0	2035008.5	956758.1	545470.5	464022.0	211100.9	133305.5	109339.1
$\|\mathbf{x}\|_\infty$	184653	111909	94559	38507	25275	24762	8848	6853	6334
$\|\mathbf{z}\|_2$	0	374014.0	2118987.6	0	94916.6	343682.9	0	11897.9	46428.4
$\|\mathbf{z}\|_\infty$	0	46895	346439	0	13265	52789	0	1439	7213
PK	37.50	19.50	15.00	26.25	13.75	11.25	17.00	10.00	8.00
Sig	25.68	13.53	10.51	18.87	10.01	8.29	13.16	7.83	6.30
LWE	100.0	100.0	100.0	102.8	102.8	102.8	104.7	104.7	104.7
AISIS	80.2	85.8	81.1	82.0	87.5	84.3	83.7	89.0	88.1
δ	1.00685	1.00643	1.00678	1.00670	1.00631	1.00653	1.00658	1.00621	1.00628
k	174	193	177	180	199	188	186	204	201

Fig. 4. Summary of the concrete parameters, with base $b = 2$, aiming at around 80 to 90-bit security. The sizes of PK and Sig are measured in kB. τ is the Gaussian width of the secret matrix $\mathbf{R}$. s is the Gaussian width of the preimage. "LWE" refers to the security level of the pseudorandomness of the PK. "AISIS" refers to the security level of breaking approximate ISIS. δ and k are the variables used in the AISIS security estimation.

Methods for security estimation. Let us first explain how we make the security estimations. The concrete security estimation of lattice-based cryptographic primitive is a highly active research area and more sophisticated methods are proposed recently. Here we use relatively simple methods to estimate the pseudorandomness of the public-key (henceforth "LWE security"), and the hardness of

Params	Exact	Approx	Approx	Exact	Approx	Approx	Exact	Approx	Approx
n	512	512	512	1024	1024	1024	1024	1024	1024
$\lceil \log_2 q \rceil$	16	16	16	18	18	18	18	18	18
b	4	4	4	8	8	8	4	4	4
l	0	2	4	0	2	3	0	4	5
τ	2.6	2.6	2.6	2.8	2.8	2.8	2.8	2.8	2.8
s	3114.2	2833.3	2505.6	8861.1	7824.8	7227.9	5118.8	4297.8	4015.5
m	5120	4096	3072	8192	6144	5120	11264	7168	6144
$\|\mathbf{x}\|_2$	223740.1	183004.9	138145.7	805772.9	604711.5	516446.3	552713.4	369981.2	311153.9
$\|\mathbf{x}\|_\infty$	13320	11868	8948	35348	28823	30435	19274	18283	14927
$\|\mathbf{z}\|_2$	0	1402.3	19807.1	0	7316.5	54379.8	0	29958.0	115616.4
$\|\mathbf{z}\|_\infty$	0	174	2448	0	905	6680	0	3025	12070
PK	9.00	7.00	5.00	15.75	11.25	9.00	22.50	13.50	11.25
Sig	7.62	5.94	4.45	13.70	10.14	8.36	18.74	11.09	9.38
LWE	104.7	104.7	104.7	192.7	192.7	192.7	192.7	192.7	192.7
AISIS	82.8	85.5	87.8	165.3	172.9	174.9	175.8	185.7	183.7
δ	1.00664	1.00645	1.00629	1.0036	1.00347	1.00343	1.00342	1.00326	1.00329
k	183	192	200	462	488	495	498	532	525

Fig. 5. Summary of the concrete parameters, with base $b \geq 4$, aiming at around 80 to 90-bit and 170 to 184-bit security.

breaking approximate ISIS (henceforth "AISIS security"). Let us remark that our estimations may not reflect the state-of-art, but at least provide a fair comparison of the parameters for the exact trapdoor versus the approximate trapdoor.

LWE security depends on the choices of q, n, and the Gaussian width τ of the trapdoor $\mathbf{R}$. The estimation of LWE security was done with the online LWE bit security estimator with BKZ as the reduction model[4] [5].

For the approximate ISIS problem, the only direct cryptanalysis result we are aware of is the work of Bai et al. [12], but it is not clearly applicable to the parameters we are interested. Instead we estimate AISIS through $\mathsf{ISIS}_{n,m,q,\alpha+\beta}$ following the reduction in Lemma 8, where α and β are the upper-bounds of l_2 norm of the error $\mathbf{z}$ and preimage $\mathbf{x}$. We estimate the security level of $\mathsf{ISIS}_{n,m,q,\alpha+\beta}$ based on how many operations BKZ would take to find a vector in the lattice $\Lambda_q^\perp(\mathbf{A})$ of length $\alpha+\beta$. Further, we can throw away columns in $\mathbf{A}$. We choose to only use $2n$ columns of $\mathbf{A}$ as done in [14], denoted $\mathbf{A}_{2n}$, since Minkowski's theorem[5] tells us $\Lambda_q^\perp(\mathbf{A}_{2n})$ has a short enough vector. Following [7,5], we use sieving as the SVP oracle with time complexity $2^{.292k+16.4}$ in the block size, k. BKZ is expected to return a vector of length $\delta^{2n}\det^{1/2n}$ for a lattice of dimension $2n$. Hence, we found the smallest block size k achieving the needed δ corresponding to forging a signature, $\frac{\alpha+\beta}{\sqrt{q}} = \delta^{2n}$. Finally, we used the heuristic $\delta \approx (\frac{k}{2\pi e}(\pi k)^{1/k})^{1/2(k-1)}$ to determine the relation between k and δ, and we set the total time complexity of BKZ with block-size k, dimension $2n$ as $8 \cdot 2n \cdot \mathrm{time}(SVP) = 8 \cdot 2n \cdot 2^{.292k+16.4}$ [22,7]. Here we use the "magic eight tour number" for BKZ to keep consistency with the LWE online estimator. We have

[4] https://bitbucket.org/malb/lwe-estimator

[5] For any lattice $\mathbf{L}$, $\lambda_1 \leq \sqrt{r}\det(\mathbf{L})^{1/r}$ where r is the rank of the lattice.

not incorporated the more recent developments in [24] and [6] in the security estimation.

The comparison. For an estimation of 80-bit[6] security, our reference implementation for the exact trapdoor under the modulus $q \approx 2^{24}$ and base $b = 2$ matches the parameters reported in [13] (the parameters in the other implementation [14,36] are possibly measured in different ways). We also use smaller moduli and bigger bases to reduce the size and increase the security level. The parameters in Figures 4 and 5 suggest that for all the choices of q and b, using the approximate gadget trapdoor by setting $l = \lceil (\log_b q)/2 \rceil$ saves about half of the sizes in the public key and signatures comparing to using the exact trapdoor, with even a slight increase in the security estimation.

Our implementation shows that the sizes of the public-key & signature can be reduced to 5 kB & 4.45 kB for an estimation of 88-bit security, and 11.25 kB & 9.38 kB for an estimation of 184-bit security. Those are still larger than, but much closer to the sizes for the signatures based on the rejection sampling approach [40,11,26,8]. As a reference, the sizes of the public-key & signature for qTESLA [8] are 4.03 kB & 3.05 kB for an estimation of 128-bit security, and 8.03 kB & 6.03 kB for an estimation of 192-bit security.

Acknowledgments

We are grateful to Daniele Micciancio for valuable advice and his generous sharing of ideas on the subject of this work. We would also like to thank Léo Ducas, Steven Galbraith, Thomas Prest, Yang Yu, Chuang Gao, Eamonn Postlethwaite, Chris Peikert, and the anonymous reviewers for their helpful suggestions and comments.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.
2. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.
3. Miklós Ajtai. Generating hard instances of the short basis problem. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
4. Gorjan Alagic, Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process*. US Department of Commerce, National Institute of Standards and Technology, 2019.

[6] When one applies our security estimate methods to Table 1 of [13], one gets 82-bit security under the $\lambda = 97$, $n = 512$, $q = 2^{24}$ column.

5. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the {LWE, NTRU} schemes! In *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, pages 351–367, 2018.
6. Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In *EUROCRYPT (2)*, volume 11477 of *Lecture Notes in Computer Science*, pages 717–746. Springer, 2019.
7. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.
8. Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Patrick Longa, and Jefferson E. Ricardini. The lattice-based digital signature scheme qtesla. *IACR Cryptology ePrint Archive*, 2019:85, 2019.
9. Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory of Computing Systems*, 48(3):535–553, 2011.
10. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *LNCS*, pages 595–618. Springer, 2009.
11. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In *CT-RSA*, volume 8366 of *Lecture Notes in Computer Science*, pages 28–47. Springer, 2014.
12. Shi Bai, Steven D. Galbraith, Liangze Li, and Daniel Sheffield. Improved combinatorial algorithms for the inhomogeneous short integer solution problem. *J. Cryptology*, 32(1):35–83, 2019.
13. Rachid El Bansarkhani and Johannes A. Buchmann. Improvement and efficient implementation of a lattice-based signature scheme. In *Selected Areas in Cryptography*, volume 8282 of *Lecture Notes in Computer Science*, pages 48–67. Springer, 2013.
14. Pauline Bert, Pierre-Alain Fouque, Adeline Roux-Langlois, and Mohamed Sabt. Practical implementation of ring-sis/lwe based signature and IBE. In *PQCrypto*, volume 10786 of *Lecture Notes in Computer Science*, pages 271–291. Springer, 2018.
15. Florian Bourse, Rafaël Del Pino, Michele Minelli, and Hoeteck Wee. FHE circuit privacy almost for free. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 62–89. Springer, 2016.
16. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.
17. Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In *ITCS*, pages 147–156. ACM, 2016.
18. Ran Canetti and Yilei Chen. Constraint-hiding constrained prfs for nc^1 from LWE. In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 446–476, 2017.
19. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of cryptology*, 25(4):601–639, 2012.

20. Cheng Chen, Nicholas Genise, Daniele Micciancio, Yuriy Polyakov, and Kurt Rohloff. Implementing token-based obfuscation under (ring) LWE. *IACR Cryptology ePrint Archive*, 2018:1222, 2018.
21. Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In *CRYPTO (2)*, volume 10992 of *Lecture Notes in Computer Science*, pages 577–607. Springer, 2018.
22. Yuanmi Chen. *Réduction de réseau et sécurité concréte du chiffrement complétement homomorphe*. PhD thesis, Paris 7, 2013.
23. Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 574–591, 2018.
24. Léo Ducas. Shortest vector from lattice sieving: A few dimensions for free. In *EUROCRYPT (1)*, volume 10820 of *Lecture Notes in Computer Science*, pages 125–145. Springer, 2018.
25. Léo Ducas, Steven Galbraith, Thomas Prest, and Yang Yu. Integral matrix gram root and lattice gaussian sampling without floats. *IACR Cryptology ePrint Archive*, 2019:320, 2019.
26. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.
27. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru, 2018.
28. Nicholas Genise and Daniele Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, pages 174–203, 2018.
29. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
30. Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 498–527, 2015.
31. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.
32. Craig Gentry and Mike Szydlo. Cryptanalysis of the revised ntru signature scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 299–320. Springer, 2002.
33. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 112–131, 1997.
34. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554. ACM, 2013.
35. Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *FOCS*, pages 612–621. IEEE Computer Society, 2017.

36. Kamil Doruk Gür, Yuriy Polyakov, Kurt Rohloff, Gerard W Ryan, and Erkay Savas. Implementation and evaluation of improved gaussian sampling for lattice trapdoors. In *Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 61–71. ACM, 2018.
37. Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing bp-obfuscation using graph-induced encoding. In *ACM Conference on Computer and Communications Security*, pages 783–798. ACM, 2017.
38. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: digital signatures using the NTRU lattice. In *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 122–140. Springer, 2003.
39. Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *Algorithmic number theory*, pages 267–288. Springer, 1998.
40. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.
41. Daniele Micciancio. personal communication.
42. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 700–718, 2012.
43. Daniele Micciancio and Chris Peikert. Hardness of sis and lwe with small parameters. In *Advances in Cryptology–CRYPTO 2013*, pages 21–39. Springer, 2013.
44. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measure. *SIAM Journal on Computing*, 37(1):267–302, 2007.
45. Daniele Micciancio and Michael Walter. On the bit security of cryptographic primitives. In *EUROCRYPT (1)*, volume 10820 of *Lecture Notes in Computer Science*, pages 3–28. Springer, 2018.
46. Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 271–288, 2006.
47. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342, 2009.
48. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *STOC*, pages 461–473. ACM, 2017.
49. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography*, pages 145–166. Springer, 2006.
50. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
51. Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing*, pages 210–268. Cambridge University Press, 2012.
52. Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *FOCS*, pages 600–611. IEEE Computer Society, 2017.

A The Smoothing Parameter of $\Lambda_{\mathbf{L}}$

Recall the notations that $\mathbf{R}' = \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{n(k-l)} \end{bmatrix} \in \mathbb{Z}^{m \times (n(k-l))}$, $\Sigma_p := s^2\mathbf{I}_m - \mathbf{R}'(\mathbf{R}')^t$. Here we derive the conditions of s so that $\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}} \geq \eta_\epsilon(\Lambda_{\mathbf{L}})$ holds, where $\Lambda_{\mathbf{L}}$ is the lattice generated by

$$\mathbf{B} := \begin{bmatrix} -\mathbf{R}' \\ \mathbf{I}_{n(k-l)} \end{bmatrix}.$$

We do this in three steps: first we write out the dual basis of $\mathbf{B}$, then we reduce $\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}} \geq \eta_\epsilon(\Lambda_{\mathbf{L}})$ to a statement about the smoothing parameter of $\mathbb{Z}^{n(k-l)}$, and finally we find when $\sqrt{\Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}} \geq \eta_\epsilon(\Lambda_{\mathbf{L}})$ as a function of s.

Dual basis, $\mathbf{B}^*$: Let $\Sigma = \Sigma_p \oplus \sigma^2\mathbf{I}_{n(k-l)}$. By definition, we need $\rho(\sqrt{\Sigma}^t\Lambda^*_{\mathbf{L}}) \leq 1+\epsilon$. In general, the dual basis Λ^* is generated by the dual basis $\mathbf{B}(\mathbf{B}^t\mathbf{B})^{-1}$. In the case of $\Lambda_{\mathbf{L}}$, we can write the dual basis as

$$\mathbf{B}^* := \begin{bmatrix} -\mathbf{R}' \\ \mathbf{I}_{n(k-l)} \end{bmatrix} \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-1}.$$

Reducing to $\eta_\epsilon(\mathbb{Z}^{n(k-l)})$: Next, the gaussian sum $\rho(\sqrt{\Sigma}^t\Lambda^*_{\mathbf{L}})$ is equal to

$$\sum_{\mathbf{x} \in \mathbb{Z}^{n(k-l)}} \exp(-\pi \mathbf{x}^t(\mathbf{B}^*)^t\Sigma\mathbf{B}^*\mathbf{x}).$$

This reduces to showing $\sqrt{(\mathbf{B}^*)^t\Sigma\mathbf{B}^*} \geq \eta_\epsilon(\mathbb{Z}^{n(k-l)})$.

Now we write out the matrix product $(\mathbf{B}^*)^t\Sigma\mathbf{B}^*$,

$$\begin{aligned}(\mathbf{B}^*)^t\Sigma\mathbf{B}^* &= \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-t} \left[-(\mathbf{R}')^t\ \mathbf{I}\right] \begin{bmatrix} \Sigma_p & \mathbf{0} \\ \mathbf{0} & \sigma^2\mathbf{I} \end{bmatrix} \begin{bmatrix} -\mathbf{R}' \\ \mathbf{I} \end{bmatrix} \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-1} \\ &= \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-t} \left[(\mathbf{R}')^t\Sigma_p\mathbf{R}' + \sigma^2\mathbf{I}\right] \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-1}.\end{aligned}$$

Before we continue, we consider the structure of the middle matrix:

$$\begin{aligned}\Sigma_s := (\mathbf{R}')^t\Sigma_p\mathbf{R}' &= \left[\mathbf{R}^t\ \mathbf{I}\right] \left(s^2\mathbf{I} - \sigma^2 \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \left[\mathbf{R}^t\ \mathbf{I}\right]\right) \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \\ &= \left[\mathbf{R}^t\mathbf{R} + \mathbf{I}\right] \left(s^2\mathbf{I} - \sigma^2\left[\mathbf{R}^t\mathbf{R} + \mathbf{I}\right]\right).\end{aligned}$$

Derive the condition for s: Now we will derive the condition for s so that

$$\left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-t} \left[\Sigma_s + \sigma^2\mathbf{I}\right] \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-1} \geq \eta^2_\epsilon(\mathbb{Z}^{n(k-l)}).$$

Claim. All invertible matrices of the form $(\mathbf{R}^t\mathbf{R} + \alpha\mathbf{I})^i$ for $i \in \mathbb{Z}, \alpha \in \mathbb{R}$ commute.

Proof. Let $\mathbf{QSV}^t$ be $\mathbf{R}$'s singular value decomposition. Now, $\mathbf{R}^t\mathbf{R} + \alpha\mathbf{I} = \mathbf{VDV}^t + \mathbf{V}(\alpha\mathbf{I})\mathbf{V}^t$ where $\mathbf{D} = \mathbf{S}^t\mathbf{S} = \mathrm{diag}(s_i^2(\mathbf{R}))$ since $\mathbf{V}, \mathbf{Q}$ are orthogonal. Equivalently, we have $\mathbf{R}^t\mathbf{R} + \alpha\mathbf{I} = \mathbf{V}\mathbf{D}_\alpha\mathbf{V}^t$ where $\mathbf{D}_\alpha = \mathrm{diag}(s_i^2(\mathbf{R}) + \alpha) = \mathbf{S}^t\mathbf{S} + \alpha\mathbf{I}_{2n}$. By induction, we have $(\mathbf{R}^t\mathbf{R} + \alpha\mathbf{I})^i = \mathbf{V}\mathbf{D}^i_\alpha\mathbf{V}^t$, $i \in \mathbb{Z}$. Finally, $\mathbf{D}^i_\alpha$ is a diagonal matrix so $\mathbf{D}^i_\alpha$ and $\mathbf{D}^j_{\alpha'}$ commute for all α, α' since diagonal matrices commute. The result follows from the orthogonality of $\mathbf{V}$ ($\mathbf{V}^t\mathbf{V} = \mathbf{I}$).

Claim A allows us to lower-bound the smallest eigenvalue of

$$\begin{aligned}(\mathbf{B}^*)^t \Sigma \mathbf{B}^* &= \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-2} \left(\left[\mathbf{R}^t\mathbf{R} + \mathbf{I}\right]\left[s^2\mathbf{I} - \sigma^2\left[\mathbf{R}^t\mathbf{R} + \mathbf{I}\right]\right] + \sigma^2\mathbf{I}\right) \\ &= \left[\mathbf{R}^t\mathbf{R} + 2\mathbf{I}\right]^{-2} \left(s^2[\mathbf{R}^t\mathbf{R} + \mathbf{I}] - \sigma^2[2\mathbf{R}^t\mathbf{R} + (\mathbf{R}^t\mathbf{R})^2]\right).\end{aligned}$$

Viewing these matrices as their diagonal matrices of eigenvalues, we see $(\mathbf{B}^*)^t \Sigma \mathbf{B}^*$'s least eigenvalue is lower-bounded by

$$\lambda_{lb}(s, \mathbf{R}) := \frac{s^2(s_{2n}^2(\mathbf{R}) + 1) - \sigma^2(s_1^4(\mathbf{R}) + 2s_1^2(\mathbf{R}))}{(s_1^2(\mathbf{R}) + 2)^2}.$$

Next, we assume $\sigma = \sqrt{b^2+1}\eta_\epsilon(\mathbb{Z}^{nk}) \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{G}))$ and solve for s using $\lambda_{lb}(s, \mathbf{R}) \geq \eta_\epsilon^2(\mathbb{Z}^{n(k-l)})$,

$$s^2 \geq \frac{s_1^2(\mathbf{R}) + 1}{s_{2n}^2(\mathbf{R}) + 1}\eta_\epsilon^2(\mathbb{Z}^{n(k-l)}) + \frac{(b^2+1)(s_1^4(\mathbf{R}) + 2s_1^2(\mathbf{R}))}{s_{2n}^2(\mathbf{R}) + 1}\eta_\epsilon^2(\mathbb{Z}^{nk}).$$

This is

$$s \gtrsim \sqrt{b^2+1}\frac{s_1^2(\mathbf{R})}{s_{2n}(\mathbf{R})}\eta_\epsilon(\mathbb{Z}^{nk}).$$

We remark that the ratio $\frac{s_1(\mathbf{R})}{s_{2n}(\mathbf{R})}$ is a constant for commonly-used subgaussian distributions for $\mathbf{R}$'s entries [51].