Efficient Public Key Encryption Based on Ideal Lattices

Damien Stehlé[1,2], Ron Steinfeld[2], Keisuke Tanaka[3], and Keita Xagawa[3]

[1] CNRS/Department of Mathematics and Statistics (F07), University of Sydney NSW 2006, Australia.
damien.stehle@gmail.com – http://perso.ens-lyon.fr/damien.stehle
[2] Centre for Advanced Computing - Algorithms and Cryptography,
Department of Computing, Macquarie University, NSW 2109, Australia
rons@ics.mq.edu.au – http://www.ics.mq.edu.au/~rons/
[3] Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan
{keisuke,xagawa5}@is.titech.ac.jp – http://www.is.titech.ac.jp/~keisuke – http://xagawa.net

Abstract. The potential high efficiency of public-key encryption based on structured lattices was first indicated by the NTRU cryptosystem, which was proposed about 10 years ago. Unfortunately, the security of NTRU is only heuristic. Thus, it remained an important research challenge to construct an efficient encryption scheme based on structured lattices which admits a proof of security relative to a well established cryptographic assumption. We make progress in addressing the above challenge. We show how to construct a CPA-secure public-key encryption scheme with security provably based on the worst case hardness of the approximate Shortest Vector Problem in structured ideal lattices. Under the assumption that the latter is exponentially hard to solve even with a quantum computer, our scheme resists any subexponential attack and offers (quasi-)optimal asymptotic performance: if n is the security parameter, both keys are of bit-length $\widetilde{O}(n)$ and the amortized costs of both encryption and decryption are $\widetilde{O}(1)$ per message bit. Our construction adapts the trapdoor one-way function of Gentry, Peikert and Vaikuntanathan (STOC 2008), based on the Learning With Errors problem, to structured lattices. Our main technical tools are an adaptation of Ajtai's trapdoor key generation algorithm (ICALP 1999) to structured ideal lattices, and a re-interpretation of Regev's quantum reduction between the Closest Vector Problem and sampling short lattice vectors. We think these techniques are very likely to find further applications in the future.
Keywords: Public-Key Encryption, Lattices, Provable Security, Post-Quantum Cryptography

1 Introduction

Lattice-based cryptography has been rapidly developing in the last few years, inspired by the breakthrough theoretical result of Ajtai in 1996 [1], who constructed a one-way function with average-case security provably related to the worst-case complexity of hard lattice problems. The attractiveness of lattice-based cryptography stems from its provable security guarantees, well studied theoretical underpinnings, simplicity and potential efficiency (evaluating Ajtai's one-way function essentially consists of just a matrix-vector multiplication over a small finite field), and also the apparent security against quantum computing attacks. The main complexity assumption in lattice-based cryptography is the hardness of approximate versions of the Shortest Vector Problem (SVP). The $\gamma(n)$-SVP problem is to find, given a lattice of dimension n, a non-zero lattice vector of length at most $\gamma(n)$ times the length of the shortest non-zero lattice vector. The complexity of $\gamma(n)$-SVP increases with n, but decreases with $\gamma(n)$. Although this complexity is believed to be exponential in n for any polynomial $\gamma(n)$, minimizing the degree of $\gamma(n)$ is very important in practice, to allow the use of a practical dimension n for a given security level.

LATTICE-BASED PUBLIC-KEY ENCRYPTION. The first provably secure lattice-based cryptosystem was proposed by Ajtai and Dwork [4], and relies on the worst-case hardness of $\mathcal{P}oly(n)$-SVP in

arbitrary lattices. Several subsequent works focused on improving its efficiency [34, 32, 12, 29]. The current state of the art [12, 29] is a scheme with public/private key length $\widetilde{O}(n^2)$ and encryption/decryption throughput of $\widetilde{O}(n)$ bit operations per message bit. Its security against polynomially-bounded attacks relies on the quantum worst-case hardness of $\widetilde{O}(n^{1.5})$-SVP in arbitrary lattices. In parallel to the provably secure encryption schemes, there have also been heuristic proposals [14, 15]. In particular, unlike the above provably secure schemes which use unstructured random matrices, the NTRU encryption scheme [15] exploits the properties of *structured* matrices/lattices to achieve high efficiency with respect to both public/private key length ($\widetilde{O}(n)$ bits) and encryption/decryption cost ($\widetilde{O}(1)$ bit operation per message bit). Unfortunately, the security of NTRU remains heuristic and it was an important open challenge to provide a provably secure scheme with comparable efficiency.

Provably Secure Schemes from Ideal Lattices. Micciancio [21] introduced the class of structured *cyclic* lattices, which correspond to ideals in polynomial rings $\mathbb{Z}[x]/(x^n - 1)$, and presented the first Ajtai-type provably secure one-way function based on the worst-case hardness of the restriction of $\mathcal{P}oly(n)$-SVP to cyclic lattices. At the same time, thanks to its algebraic structure, this one-way function enjoys high efficiency comparable to the NTRU scheme ($\widetilde{O}(n)$ evaluation time and storage cost). Subsequently, Lyubashevsky and Micciancio [19] and independently Peikert and Rosen [30] showed how to modify Micciancio's function to construct an efficient and provably secure collision resistant hash function. For this, they introduced the more general class of *ideal* lattices, which correspond to ideals in polynomial rings $\mathbb{Z}[x]/f(x)$. The collision resistance of the hash function relies on the hardness of the restriction of $\mathcal{P}oly(n)$-SVP to ideal lattices (called $\mathcal{P}oly(n)$-Ideal-SVP). A variant (SWIFFT/SWIFFTX) has been implemented and submitted to the NIST SHA-3 hash competition [7]. The average-case collision-finding problem for this hash function is a natural computational problem called Ideal-SIS, which has been shown to be as hard as the worst-case instances of Ideal-SVP. Provably secure efficient signature schemes from ideal lattices have also been proposed [20, 17, 18, 16]. However, constructing efficient provably secure public key encryption from ideal lattices was an interesting open problem.

Our results. In this paper, we describe the first provably secure public key encryption scheme whose security relies on the hardness of the worst-case instances of $\widetilde{O}(n^2)$-Ideal-SVP against subexponentially bounded quantum attacks. It achieves asymptotically optimal efficiency: the public/private key length is $\widetilde{O}(n)$ bits and the amortized encryption/decryption costs are $\widetilde{O}(1)$ bit operations per message bit. We note that our security model rules out even subexponential time attacks, taking full advantage of the commonly believed exponential hardness of $\mathcal{P}oly(n)$-SVP, even against quantum attacks. Such a security model obsoletes most of modern public key cryptography and highlights the strength of lattice-based cryptography. Our main technical tools are an adaptation of Ajtai's trapdoor key generation algorithm [3] to structured ideal lattices, and a re-interpretation of Regev's quantum reduction [34] between the Closest Vector Problem and sampling short lattice vectors. We think these techniques are very likely to find further applications in the future. As examples, we also use our results to construct efficient provably secure trapdoor signatures and ID-based identification schemes.

All previous cryptosystems based on general lattices [34, 32, 12, 29] rely on the average case hardness of the *Learning With Errors* (LWE) problem introduced in [34]. Our scheme is similarly based on a structured variant of LWE, that we call Ideal-LWE. However, we introduce novel techniques to overcome two main difficulties that arise from the restriction to ideal lattices. Firstly, the previous cryptosystems based on unstructured matrices all make use of Regev's wost-case to average-case classical reduction [34] from a closest vector problem in a given lattice, to an LWE problem (this

is the *classical step* in the quantum reduction of [34] from SVP to LWE). This reduction seems to heavily rely on the unstructured-ness of the considered lattices, and does not seem to carry over to the structured lattices involved in Ideal-LWE. Secondly, the other ingredient used in previous cryptosystems, namely Regev's reduction [34] from the computational variant of LWE to its decisional variant, also fails for Ideal-LWE. We also remark that, unlike in the case of arbitrary lattices, the decisional variant of $\mathcal{P}oly(n)$-Ideal-SVP can be trivially solved in polynomial time [31]. This implies that Peikert's recent classical reduction [29] from the decisional variant of SVP to LWE is unlikely to be useful, even if it could be adapted to work for ideal lattices.

Our solution to the above difficulties avoids the *classical step* of the reduction from [34] altogether. Instead, we focus on the *quantum step* of the reduction from [34] and use it to construct a new quantum average case reduction from SIS (the unstructured variant of Ideal-SIS) to LWE. This reduction also works from Ideal-SIS to Ideal-LWE. Combined with the known reduction from worst case Ideal-SVP to average-case Ideal-SIS [19] (i.e., the proof of collision resistance for SWIFFT [7]), we obtain a quantum reduction from Ideal-SVP to Ideal-LWE. This shows the hardness of the computational variant of Ideal-LWE. Since we do not obtain the hardness of the decisional variant of Ideal-LWE, we use the Goldreich-Levin hardcore function [13, Sec. 2.5] to derive pseudorandom bits for encryption. But for decryption, we also need an Ideal-LWE inversion trapdoor. To this end, we show how to adapt to ideal lattices the trapdoor generation algorithm of Ajtai [3], recently improved by Alwen-Peikert [6].

The main idea of our new quantum reduction from Ideal-SIS to Ideal-LWE is a re-interpretation of Regev's quantum step in [34]. The latter was presented as a worst case quantum reduction from sampling short lattice vectors in a lattice L to solving a closest vector problem in the dual lattice $\hat{L}$. Our first observation is that this reduction is actually stronger: it is an average case reduction which works given an oracle for CVP in $\hat{L}$ with a normally distributed error vector. Our second observation is that LWE can be seen as a closest vector problem with a normally distributed error in a certain lattice whose dual is the SIS lattice (up to a scaling factor). Combining these observations leads to our SIS to LWE reduction. Finally we show how to apply it to reduce Ideal-SIS to Ideal-LWE – the main technical hurdle here involves a probabilistic lower bound for the minimum of the Ideal-LWE lattice (Lemma 2.5). We believe our new SIS to LWE reduction is of independent interest. For example, when choosing practical parameters for lattice-based encryption (see, e.g., [24]), it is impractical to rely on the worst case hardness of SVP. Instead, the practical average case hardness of LWE is evaluated based on the best known attack which consists in solving SIS. Our reduction justifies this heuristic in some sense by showing that it is indeed necessary to break SIS in order to solve LWE. Our reduction can also be seen as a unification of the security proofs of all lattice-based cryptography, since it allows to make all the worst case to average case reductions rely solely on SIS.

WORK IN PROGRESS. Our encryption scheme is IND-CPA secure. It is very likely that it can be modified to achieve IND-CCA2 security, by adapting the general techniques described in [29, 35]. Another question that we are currently investigating is how to choose practical parameters for our scheme.

ROAD-MAP. In Section 2, we provide the background that is necessary to the understanding of our results. Section 3 shows how to hide a trapdoor in the adaptation of SIS to ideal lattices. Section 4 contains the new reduction between SIS and LWE. Finally, in Section 5, we present our encryption scheme and briefly describe other cryptographic constructions.

NOTATION. Vectors will be denoted in bold. Most of our vectors are column vectors. If $\boldsymbol{x}, \boldsymbol{y}$ are two vectors, we denote their inner product by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. If $\boldsymbol{x} \in \mathbb{R}^n$, then $\|\boldsymbol{x}\|$ (resp. $\|\boldsymbol{x}\|_\infty$) denotes the Euclidean (resp. infinity) norm of $\boldsymbol{x}$. We denote by $\rho_s(\boldsymbol{x})$ (resp. ν_s) the standard n-dimensional Gaussian (resp. Gaussian distribution) with center $\mathbf{0}$ and variance s, i.e., $\rho_s(\boldsymbol{x}) = \exp(-\pi\|\boldsymbol{x}\|^2/s^2)$ (resp. $\nu_s(\boldsymbol{x}) = \rho_s(\boldsymbol{x})/s^n$). We make use of the standard Landau notations as well as the notations $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ (which hide poly-logarithmic factors). If X is a random variable, the probability of the event $X = x$ is denoted by $\Pr[X = x]$. If D_1 and D_2 are two probability distributions over a discrete domain E, their statistical distance is $\Delta(D_1, D_2) = \frac{1}{2}\sum_{x \in E} |D_1(x) - D_2(x)|$. If a function f over a countable domain E takes non-negative real values, its sum over an arbitrary $F \subseteq E$ will be denoted by $f(F)$. If q is a prime number, we denote by $\mathbb{Z}_q$ the field of integers modulo q. We denote by Ψ_s the reduction modulo q of ν_s.

2 Reminders and Background Results on Lattices

For a detailed introduction to the computational aspects of lattices, we refer to [22]. In the present section, we remind the reader very quickly some fundamental properties of lattices that we will need. We then introduce the so-called ideal lattices, and finally formally define some computational problems.

2.1 Euclidean lattices

An n-dimensional lattice L is the set of all integer linear relations of some linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^n$, i.e., $L = \sum \mathbb{Z}\boldsymbol{b}_i$. The $\boldsymbol{b}_i$'s are called a basis of L. The ith minimum $\lambda_i(L)$ is the smallest r such that L contains i linearly independent vectors of norms $\leq r$. We let $\lambda_1^\infty(L)$ denote the first minimum of L with respect to the infinity norm. If $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is a basis, we define its norm by $\|B\| = \max \|\boldsymbol{b}_i\|$ and its fundamental parallelepiped by $P(B) = \{\sum_i c_i \boldsymbol{b}_i : \boldsymbol{c} \in [0,1)^n\}$. Given a basis B for lattice L and a vector $\boldsymbol{c} \in \mathbb{R}^n$, we define $\boldsymbol{c} \bmod L$ as the unique vector in $P(B)$ such that $\boldsymbol{c} - (\boldsymbol{c} \bmod L) \in L$ (the basis being implicit). The Gram-Schmidt orthogonalization (GSO) is the tuple $(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*)$ defined as follows: the vector $\boldsymbol{b}_i^*$ is the component of $\boldsymbol{b}_i$ which is orthogonal to the linear span of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. The GSO norm is $\|B^*\| = \max \|\boldsymbol{b}_i^*\|$. For a given lattice L and a given parameter $s > 0$, the sum $\rho_s(L)$ is finite. We define the lattice Gaussian distribution by $D_{L,s}(\boldsymbol{b}) = \frac{\rho_s(\boldsymbol{b})}{\rho_s(L)}$, for any $\boldsymbol{b} \in L$. If L is a lattice, its dual lattice $\widehat{L}$ is the set of vectors $\widehat{\boldsymbol{b}}$ in $\mathbb{R}^n$ such that $\langle \widehat{\boldsymbol{b}}, \boldsymbol{b} \rangle \in \mathbb{Z}$ for all $\boldsymbol{b} \in L$. For $\varepsilon > 0$, we define the smoothing parameter $\eta_\varepsilon(L)$ as the smallest $s > 0$ such that $\rho_{1/s}(\widehat{L} \setminus \mathbf{0}) \leq \varepsilon$. We have $\eta_\varepsilon(L) \leq \sqrt{\ln(2n(1+1/\varepsilon))/\pi}/\lambda_1^\infty(\widehat{L})$ (see [28, Lemma 3.5]). We will use the following results.

Lemma 2.1 ([30, Lemma 2.11]). *For any $\boldsymbol{x}$ in an n-dimensional lattice L, $\varepsilon \in (0, 1/3)$ and $s \geq 2\eta_\varepsilon(L)$, we have $D_{L,s}(\boldsymbol{x}) \leq 2^{-n+1}$.*

Lemma 2.2 ([23, Lemma 2.10]). *Given an n-dimensional lattice L, we have $\Pr_{\boldsymbol{x} \sim D_{L,s}}[\|\boldsymbol{x}\| > s\sqrt{n}] \leq 2^{-n+1}$.*

2.2 Ideal lattices

Ideal lattices are a special subset of lattices that possess the computationally interesting property of being related to structured matrices and polynomials. The n-dimensional matrix-matrix and vector-

matrix products then respectively cost $\widetilde{O}(n^2)$ and $\widetilde{O}(n)$ arithmetic operations instead of $O(n^3)$ and $O(n^2)$.

Let $f \in \mathbb{Z}[x]$ a monic degree n polynomial. For any $g \in \mathbb{Q}[x]$, there is a unique pair (q, r) with $\deg(r) < n$ and $g = qf + r$. We denote r by $g \bmod f$ and identify r with the vector $\boldsymbol{r} \in \mathbb{Q}^n$ of its coefficients. We define $\mathrm{rot}_f(r) \in \mathbb{Q}^{n \times n}$ as the matrix whose rows are the $x^i r(x) \bmod f(x)$'s, for $i \in [0, n-1]$. We extend that notation to the matrices A over $\mathbb{Q}[x]/f$: by replacing each $A_{i,j}$ by $\mathrm{rot}_f(A_{i,j})$, one obtains $\mathrm{rot}_f(A)$. Note that $\mathrm{rot}_f(g_1)\mathrm{rot}_f(g_2) = \mathrm{rot}_f(g_1 g_2)$ for any $g_1, g_2 \in \mathbb{Q}[x]/f$. The strengths of our cryptographic constructions depend on the choice of f. Its quality is quantified by its expansion factor (we adapt the definition of [19] to the Euclidean norm):

$$\mathrm{EF}(f, k) = \max\left\{ \frac{\|g \bmod f\|}{\|g\|} \mid g \in \mathbb{Z}[x] \setminus 0 \ \text{ and } \ \deg(g) \leq k\,(\deg(f) - 1) \right\},$$

where we identified the polynomial $g \bmod f$ (resp. g) with the coefficients vector. Note that if $\deg(g) < n$, then $\|\mathrm{rot}_f(g)\| \leq \mathrm{EF}(f, 2) \cdot \|g\|$. We will concentrate on the polynomials $x^{2^k} + 1$, although most of our results are more general. We recall some basic properties of $x^{2^k} + 1$ (see [9] for the last one).

Lemma 2.3. *Let $k \geq 0$ and $n = 2^k$. Then $f(x) = x^n + 1$ is irreducible in $\mathbb{Q}[x]$. Its expansion factor $\mathrm{EF}(f, 2) \leq \sqrt{2}$. Also, for any $g = \sum_{i<n} g_i x^i \in \mathbb{Q}[x]/f$, we have $\mathrm{rot}_f(g)^T = \mathrm{rot}_f(\bar{g})$ where $\bar{g} = g_0 - \sum_{1 \leq i < n} g_{n-i} x^i$. Furthermore, if q is a prime such that $2n | (q-1)$, then $f(x)$ has n degree 1 factors in $\mathbb{Z}_q[x]$. Finally, if $k \geq 2$ and q is a prime with $q \equiv 3 \bmod 8$, then $f = f_1 f_2 \bmod q$ where, for any $i \in \{1, 2\}$, we have that f_i is irreducible in $\mathbb{Z}_q[x]$ and can be written $f_i = x^{n/2} + t_i x^{n/4} - 1$ with $t_i \in \mathbb{Z}_q$.*

Let I be an ideal of the ring $\mathbb{Z}[x]/f$, i.e., a subset of $\mathbb{Z}[x]/f$ that is closed under addition, and multiplication by arbitrary elements of $\mathbb{Z}[x]/f$. The ideal I corresponds to a sublattice of $\mathbb{Z}^n$. An *f-ideal lattice* is a sublattice of $\mathbb{Z}^n$ that corresponds to an ideal $I \subseteq \mathbb{Z}[x]/f$.

2.3 Hard lattice problems

The most famous lattice problem is SVP. Given a basis of a lattice L, it aims at finding a shortest vector in $L \setminus \mathbf{0}$. Its decision version is NP-hard under randomized reductions [2]. SVP can be relaxed by asking for a non-zero vector that is no longer than $\gamma(n)$ times a solution to SVP, for a prescribed function $\gamma(\cdot)$. The best polynomial time algorithm [5, 36] solves γ-SVP only for a slightly subexponential γ. When γ is polynomial in n, then the most efficient algorithm [5] has an exponential worst-case complexity both in time and space. If we restrict the set of input lattices to ideal lattices, we obtain the problem Ideal-SVP (resp. γ-Ideal-SVP), which is implicitly parameterized by a sequence of polynomials f of growing degrees. The computational hardness of Ideal-SVP is unknown, but to the best of our knowledge, no algorithm is known to perform non-negligibly better for Ideal-SVP than for SVP. It is believed that no subexponential quantum algorithm solves the computational variants of SVP or Ideal-SVP in the worst case. We are interested in the worst-case instances of these problems. Their worst-case hardness can be reduced to the average-case hardness of the following problems, introduced in [23] and [12].

Definition 2.1. *The Small Integer Solution problem with parameters $q(\cdot)$, $m(\cdot)$, $\beta(\cdot)$ ($\mathrm{SIS}_{q,m,\beta}$) is as follows: Given n and a matrix G sampled uniformly in $\mathbb{Z}_{q(n)}^{m(n) \times n}$, find $\boldsymbol{e} \in \mathbb{Z}^{m(n)} \setminus \mathbf{0}$ such that $\boldsymbol{e}^T G = \mathbf{0} \bmod q(n)$ (the modulus being taken component-wise) and $\|\boldsymbol{e}\| \leq \beta(n)$.*

*The Ideal Small Integer Solution problem with parameters $q(\cdot)$, $m(\cdot)$, $\beta(\cdot)$ and f (*Ideal-SIS$^f_{q,m,\beta}$*) is as follows: Given n and m polynomials $g_1, \ldots, g_m$ chosen uniformly and independently in $\mathbb{Z}_q[x]/f$, find $e_1, \ldots, e_m \in \mathbb{Z}[x]$ not all zero such that $\sum_{i \leq m} e_i g_i = 0$ in $\mathbb{Z}_q[x]/f$ and $\|\boldsymbol{e}\| \leq \beta$, where $\boldsymbol{e}$ is the vector obtained by concatenating the coefficients of the e_i's.*

These two average-case problems can be interpreted as lattice problems. Indeed, consider a matrix $G \in \mathbb{Z}_q^{m \times n}$. The set $G^\perp = \{\boldsymbol{b} \in \mathbb{Z}^m : \boldsymbol{b}^T G = \boldsymbol{0} \bmod q\}$ is an m-dimensional lattice. Solving SIS corresponds to finding a short non-zero vector in $G^\perp$. Similarly, Ideal-SIS corresponds to finding a small non-zero element in the $\mathbb{Z}[x]/f$-module $M^\perp(\boldsymbol{g}) = \{\boldsymbol{b} \in (\mathbb{Z}[x]/f)^m, \langle \boldsymbol{b}, \boldsymbol{g} \rangle = 0 \bmod q\}$, where $\boldsymbol{g} = (g_1, \ldots, g_m)$. Ideal-SIS can be seen as a lattice problem by applying the rot_f operator to the latter description. Note that the m of SIS is n times larger than the m of Ideal-SIS.

Lyubashevsky and Micciancio [19] gave a reduction from the worst-case γ-Ideal-SVP to the average-case problem γ-Ideal-SIS. The approximation factors (and the problems) in [19] are given in terms of the infinity norm. For our purposes, it is more natural use the Euclidean norm. To avoid losing a $\sqrt{n}$ factor by simply applying the norm equivalence formula, we modify the proof of [19]. We also adapt it to handle the case where the given SIS$^f_{q,m,\beta}$ algorithm has a subexponential success probability, at the cost of an additional factor of $\widetilde{O}(\sqrt{n})$ in the SVP approximation factor γ (this involves using a larger Gaussian scaling parameter to improve the 'smoothing' precision). The details will be given in the full version.

Theorem 2.1. *Suppose that f is irreducible over $\mathbb{Q}$. We consider an integer $m = \mathcal{P}oly(n)$ and a prime $q = \widetilde{\Omega}(\mathrm{EF}(f,3)\beta m^2 n)$. A polynomial-time (resp. subexponential-time) algorithm solving* Ideal-SIS$^f_{q,m,\beta}$ *with probability $1/\mathcal{P}oly(n)$ (resp. $2^{-o(n)}$) can be used to solve γ-*Ideal-SVP *in polynomial-time (resp. subexponential-time) with $\gamma = \widetilde{O}(\mathrm{EF}^2(f,2)\beta m n^{1/2})$ (resp. $\gamma = \widetilde{O}(\mathrm{EF}^2(f,2)\beta m n)$).*

The LWE problem is dual to SIS in the sense that if $G \in \mathbb{Z}_q^{m \times n}$ is the SIS-matrix, it involves the dual of the lattice $G^\perp$. We have $\widehat{G^\perp} = \frac{1}{q} L(G)$ where $L(G) = \{\boldsymbol{b} \in \mathbb{Z}^m : \exists \boldsymbol{s} \in \mathbb{Z}_q^n, G\boldsymbol{s} = \boldsymbol{b} \bmod q\}$.

Definition 2.2. *The Learning With Errors problem with parameters $q(\cdot), m(\cdot)$ and a distribution $\chi(\cdot)$ on $\mathbb{R}/[1, q(\cdot)]$ (*LWE$_{q,m;\chi}$*) is as follows: Given n, a matrix $G \in \mathbb{Z}_{q(n)}^{m(n) \times n}$ sampled uniformly at random and $G\boldsymbol{s} + \boldsymbol{e} \in (\mathbb{R}/[1, q(n)])^n$, where $\boldsymbol{s} \in \mathbb{Z}_{q(n)}^n$ is chosen uniformly at random and the coordinates of $\boldsymbol{e} \in (\mathbb{R}/q(n))^{m(n)}$ are independently sampled from $\chi(n)$, find $\boldsymbol{s}$.*

*The Ideal Learning With Errors problem with parameters $q(\cdot), m(\cdot)$, a distribution $\chi(\cdot)$ on $\mathbb{R}/[1, q(\cdot)]$ and f (*Ideal-LWE$^f_{m,q;\chi}$*) is the same as above, except that G is of the form $G = \mathrm{rot}_f(\boldsymbol{g})$ with $\boldsymbol{g}$ chosen uniformly in $(\mathbb{Z}_q[x]/f)^m$.*

Contrary to Ideal-SIS, problem Ideal-LWE does not have a natural polynomial interpretation for a general f, as we are multiplying $\mathrm{rot}_f(\boldsymbol{g})$ on the right. However, if one uses $f = x^n + 1$, then Lemma 2.3 allows to interpret Ideal-LWE as m polynomial multiplications. We will use the following results to bound the smoothing parameters of lattices $G^\perp$. The second one is proved in appendix.

Lemma 2.4 ([12, Lemma 5.3]). *Let n, m and q be integers with q prime and $m \geq 2n \log q$. Then for all but a fraction $\leq q^{-n}$ of the G's in $\mathbb{Z}_q^{m \times n}$, we have $\lambda_1^\infty(L(G)) \geq q/4$.*

Lemma 2.5. *Let n, m and q be integers with $q = 3 \bmod 4$ prime and $m \geq 9 \log q$ and $n = 2^k$ for some $k \geq 2$. Then for all but a fraction $\leq q^{-n}$ of the $\boldsymbol{g}$'s in $(\mathbb{Z}_q[x]/f)^m$, we have $\lambda_1^\infty(L(\mathrm{rot}_f(\boldsymbol{g}))) \geq q/4$.*

3 Hiding a Trapdoor in Ideal-SIS

The purpose of the present section is to demonstrate how to hide a trapdoor in the problem Ideal-SIS. Ajtai [3] showed how to simultaneously generate a (SIS) matrix $G \in \mathbb{Z}_q^{m \times n}$ and a (trapdoor) basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) \in \mathbb{Z}^{m \times m}$ of the lattice $G^\perp = \{\boldsymbol{b} \in \mathbb{Z}^m : \boldsymbol{b}^T G = \boldsymbol{0} \bmod q\}$, with the following properties:

1. The distribution of G is statistically close to the uniform distribution over $\mathbb{Z}_q^{m \times n}$.
2. The basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ are small.

Recently, Alwen and Peikert [6] improved Ajtai's construction in the sense that the created basis has shorter vectors: $\|B\| = \widetilde{O}(\sqrt{m})$, where $m = \widetilde{\Omega}(n)$. We modify Ajtai's construction to obtain a trapdoor generation algorithm for the problem Ideal-SIS, with a resulting full-rank set of linearly independent lattice vectors whose lengths are as small as those of the basis vectors of Alwen and Peikert. By using a standard technique [22, Lemma 7.1], these vectors can be transformed into a basis whose GSO norm is bounded by the same quantity and whose length is $O(\sqrt{mn})$ longer. This is thus weaker than [6], but for our applications only the GSO norm and the length of the linearly independent lattice vectors matter. We do not know how to adapt the construction of [6] from SIS to Ideal-SIS, the difficulty stemming from the lack of a Hermite Normal Form for the considered rings.

Theorem 3.1. *There exists a probabilistic polynomial time algorithm with the following properties. It takes as inputs $n, \sigma, r > 0$, an odd prime q, and an integer $m \geq (\lceil \log q \rceil + 1)(\sigma + r)$. It also takes as input a degree n polynomial $f \in \mathbb{Z}[x]$. We let $f = \prod_{i \leq t} f_i$ be its factorization over $\mathbb{Z}_q$. The algorithm succeeds with probability $\geq 1 - p$ with $p = 1 - \prod_{i \leq t}(1 - q^{-\deg f_i \cdot \sigma})$. When it does, it returns $\boldsymbol{g} \in (\mathbb{Z}_q[x]/f)^m$ and a full-rank set S of linearly independent vectors in the lattice $\mathrm{rot}_f(\boldsymbol{g})^\perp$, such that:*

1. *The distance to uniformity of $\boldsymbol{g}$ is at most $p + \frac{m}{2}\left(\prod_{i \leq t}\left(1 + \left(\frac{q}{3^r}\right)^{\deg f_i}\right) - 1\right)^{1/2}$.*
2. *We have $\|S\| \leq \mathrm{EF}(f, 2) \cdot \sqrt{(9r + \sigma)} \cdot \sqrt{n}$.*

3.1 Regularity

In the Ajtai and Alwen-Peikert constructions, a few uniformly distributed random vectors $(\boldsymbol{g}_i)_{i \leq \sigma}$ are first generated and then are used as a seed to produce other random vectors $(\boldsymbol{g}_i)_{i > \sigma}$ that are seemingly uniformly distributed. The derived vectors are small random integer combinations of the seed vectors, thus creating short vectors in the lattice $G^\perp$ as by-products. The soundness of the procedure is guaranteed by a so-called regularity lemma. In the case of ring elements instead of vectors, such a regularity lemma was given by Micciancio in [21]. The precise result dealt with the particular ring $\mathbb{Z}_q[x]/(x^n - 1)$ but its proof can be easily adapted to obtain the following.

Theorem 3.2 (Adapted from [21, Th. 4.2]). *Let $\mathbb{F}$ be a finite field and $f \in \mathbb{F}[x]$ be monic and of degree $n > 0$. Let R be the ring $\mathbb{F}[x]/f$. Let $D \subseteq \mathbb{F}$ and $r > 0$. For $g_1, \ldots, g_r \in R$, we denote by $H(g_1, \ldots, g_r)$ the random variable $\sum_{i \leq r} b_i g_i \in R$ where the b_i's are degree $< n$ polynomials with coefficients chosen independently and uniformly in D. If $U_1, \ldots, U_r$ are independent uniform random*

variables in R, then the statistical distance to uniformity of $(U_1, \ldots, U_r, H(U_1, \ldots, U_r))$ is below:

$$\frac{1}{2}\sqrt{\prod_{i \leq t}\left(1 + \left(\frac{|\mathbb{F}|}{|D|^r}\right)^{\deg f_i}\right) - 1},$$

where $f = \prod_{i \leq t} f_i$ is the factorization of f over $\mathbb{F}$.

From now on, we will choose $\mathbb{F} = \mathbb{Z}_q$ with q an odd prime, and $D = \{-1, 0, 1\}$. Let us comment on two opposite situations. If the polynomial f splits completely over $\mathbb{F}$, then the distance to uniformity can be made negligible with respect to n by taking $r = \omega(\log(nq))$. However, to ensure that it is exponentially small, one needs to take $r = \Omega(n + \log q)$. Now, if f is irreducible over $\mathbb{F}$, then choosing $r \geq 1 + \log_3 q$ suffices to provide an exponentially small statistical distance to uniformity.

3.2 A trapdoor for Ideal-SIS

We now construct the trapdoor for Ideal-SIS. More precisely, we want to simultaneously construct a uniform $\boldsymbol{g} \in \mathcal{R}^m$ with $\mathcal{R} = \mathbb{Z}_q[x]/f$, and a full-rank set S of small linearly independent vectors belonging to the lattice $G^\perp$ where $G = \mathrm{rot}_f(\boldsymbol{g})$. For this, it suffices to find a rank m set of $\mathcal{R}_0$-linearly independent short vectors in the module $M^\perp(\boldsymbol{g}) = \{\boldsymbol{a} \in \mathcal{R}_0^m \mid \langle \boldsymbol{a}, \boldsymbol{g} \rangle \equiv 0 \bmod q\}$, with $\mathcal{R}_0 = \mathbb{Z}[x]/f$.

Suppose first that we generate (uniformly and independently) k elements in $\mathcal{R}$. Using Theorem 3.2, they can be used to sample k' other elements of $\mathcal{R}$ which are (almost) uniformly distributed. In total, we get $k + k'$ elements in $\mathcal{R}$, but we only have k' small relations between them. However, we need $k + k'$ linearly independent small relations, and it is not clear how to find another k. Also, notice that our relations hold over $\mathcal{R}_0$, i.e., they do not make use of the modulus q. Ajtai's workaround in [3] consists in using an intermediate set of ring elements (or rather, in his case, vectors). The principle is the following:

- First generate some seed elements $g_1, \ldots, g_\sigma$ in $\mathcal{R}$. They will be the first elements of the Ideal-SIS instantiation. They need sufficient yield so that arbitrary elements of $\mathcal{R}$ can be written up as linear combinations of them.
- From the seed, derive some intermediate ring elements $h_{\sigma+1}, \ldots, h_m$. Among them, there should be sufficiently many (say the last r ones) that are uniformly distributed in $\mathcal{R}$, so that we may use Theorem 3.2. The first h_i's have the technical purpose of allowing to have small relations between $g_1, \ldots, g_\sigma, h_{\sigma+1}, \ldots, h_m$. More precisely, we will have:

$$[A|B](g_1, \ldots, g_\sigma, h_{\sigma+1}, \ldots, h_m)^T = (0, \ldots, 0)^T, \tag{1}$$

  where $A \in \mathcal{R}_0^{(m-\sigma)\times\sigma}$ and $B \in \mathcal{R}_0^{(m-\sigma)\times(m-\sigma)}$ have small entries and B is lower triangular with 1's on the diagonal.
- Finally, since the last r of polynomials $h_{m-r+1}, \ldots, h_m$ are uniformly distributed in $\mathcal{R}$, we take them as the last g_i's, and we construct the missing g_i's by:

$$g_i = \begin{cases} h_i + H(h_{m-r+1}, \ldots, h_m), & \forall i \in [\sigma + 1, m - r], \\ h_i, & \forall i \in [m - r + 1, m]. \end{cases} \tag{2}$$

  Thanks to Theorem 3.2, we see that whatever were the first h_i's, the uniformity of the last ones provides a close to uniform distribution of $(g_1, \ldots, g_m)$. We have that, for all $i \in [\sigma + 1, m - r]$,

$g_i = h_i + \sum_{j=m-r+1}^{m} y_{i,j} h_j$, where each $y_{i,j}$ is a degree $< n$ polynomial with coefficients chosen independently and uniformly in D. We define $C \in \mathcal{R}_0^{(m-\sigma)\times(m-\sigma)}$ as follows:

$$C = \left[\begin{array}{c|c} Id_{m-\sigma-r} & (y_{i,j}) \\ \hline 0 & Id_r \end{array}\right].$$

Equation (2) implies that

$$C \cdot (h_{\sigma+1}, \ldots, h_m)^T = (g_{\sigma+1}, \ldots, g_m)^T \text{ and } [A \mid B \cdot C] \cdot (g_1, \ldots, g_m)^T = (0, \ldots, 0)^T.$$

So far, we have showed how to create the g_i's. What is less clear is that we can have m linearly independent short vectors in M. Thanks to Equations (1) and (2), we have $m-\sigma$ linearly independent small relations. Again, as in the first attempt, some relations are missing and all those we have do not involve the modulus q. But the shape of the $m-\sigma$ relations we have allows us do build extra-relations that we can prove linearly independent. For instance, among the h_i's, say the σ before the regularity ones (i.e., $h_{m-r-\sigma+1}, \ldots, h_{m-r}$), we create traps. Recall that we can have arbitrary h_i's as long as the last ones are uniformly distributed and Equation (1) holds. In the same spirit as [3], we define $h_{m-r-\sigma+i} = 2^{-1} g_i$, for $i \leq \sigma$, where the inverse is taken modulo q. This gives us σ additional relations, since (for $i \leq \sigma$):

$$2g_{m-r-\sigma+i} = 2h_{m-r-\sigma+i} + 2H(h_{m-r+1}, \ldots, h_m) = g_i + 2H(g_{m-r+1}, \ldots, g_m) \bmod q. \qquad (3)$$

Let $2g_{m-r-\sigma+i} = g_i + 2\sum_{j=m-r+1}^{m} z_{i,j} g_j$ over $\mathbb{Z}$, where each $z_{i,j}$ is a degree $< n$ polynomial which has coefficients chosen uniformly and independently from D. We define the $\sigma \times (m-\sigma)$ matrix K:

$$K = \left[\begin{array}{c|c|c} 0 & -2Id_\sigma & 2(z_{i,j}) \end{array}\right].$$

From Equation (3), we have that $[Id_\sigma \mid K] \cdot (g_1, \ldots, g_m)^T \equiv (0, \ldots, 0)^T \bmod q$. Overall, we obtain (close to) uniformly distributed g_i's and relations of the type (over $\mathcal{R}$):

$$\left[\begin{array}{c|c} A & B \cdot C \\ \hline Id_\sigma & K \end{array}\right] \cdot \begin{bmatrix} g_1 \\ \vdots \\ g_m \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix},$$

where A and B are those of Equation (1), $C \in \mathcal{R}_0^{(m-\sigma)\times(m-\sigma)}$ is upper triangular with 1's on its diagonal, and $K \in \mathcal{R}_0^{\sigma\times(m-\sigma)}$ is divisible by 2. All these matrices have small entries. To see that the relations are linearly independent over $\mathcal{R}$, notice that the determinant of the above block-matrix modulo 2 is 1: the determinant of B and C are both 1, and $K = 0 \bmod 2$.

It only remains to explain how the matrices A and B are built. We start by generating $h_{m-r+1}, \ldots, h_m$ uniformly and independently in $\mathcal{R}$. Additionally, we set $h_{m-r-\sigma+i} = 2^{-1} g_i$ for $i \leq \sigma$. If we need other h_i's, these can be arbitrarily chosen. For any $i \in [m-r-\sigma+1, m]$, we write h_i as a linear combination (over $\mathcal{R}$) of the seeds: $h_i = \sum_{j \leq \sigma} x_{i,j} g_j$. Remember that σ is chosen so that this is actually possible. The $x_{i,j}$'s are polynomials with coefficients modulo q. We use the binary decompositions of these polynomials to write: $h_i = \sum_{k=0}^{\kappa-1} \sum_{j \leq \sigma} x_{i,j,k} g_j 2^k$, with $\kappa = \lceil \log q \rceil$. The $x_{i,j,k}$'s are polynomials with $\{0,1\}$-coefficients.

We use the $x_{i,j,k}$'s in A. The latter starts with $k+\sigma$ blocks of $R_0^{\kappa\times\sigma}$ (and then zeros). The ith block is made of the $x_{i,j,k}$'s (for $j \leq s$ and $0 \leq k \leq \kappa - 1$) by decreasing value of k. The matrix B is as follows:

$$B = \left[\begin{array}{ccc|c} T & & & \\ & \ddots & & \\ & & T & \\ \hline & E & & Id \end{array}\right],$$

where there are $r+\sigma$ blocks T and $E[i,j] = -1$ if $i = j\kappa$ and 0 otherwise.

3.3 Analysis of the trapdoor construction and comments

The correctness of the construction has already been established. It can also be run in polynomial time. What remains to be studied is the quality (in terms of dimension and norm) of the built trapdoor.

We choose σ large enough so that the seed generates R with high probability. The failure probability is bounded by $1 - \prod_{i\leq t}(1 - q^{-\deg f_i \cdot \sigma})$, where $f = \prod_{i\leq t} f_i$ is the factorization of f modulo q. Note that even in the totally splitting case, we can choose $\sigma = 1$ and reject the randomly generated seed element as long as it is not invertible. We do not get a uniformly distributed instantiation of Ideal-SIS (since the distribution of g_1 is skewed). However, we get a non-negligible proportion of Ideal-SIS instances, which suffices for Theorem 2.1 (in the reduction, run the sampling step until the obtained Ideal-SIS instance has its first vector that is invertible). The parameter r is determined by Theorem 3.2. The construction thus requires $m \geq (r+\sigma)(1+\kappa)$. Note that the lower bound on m can be further decreased by using a base $\beta > 2$ decomposition of the $x_{i,j}$'s instead of a base 2 decomposition. However, this makes the norms of the module vectors grow. Finally, the largest possible vector norms come from the first $m - \sigma$ module vectors. The A part has norm $\leq \sqrt{\sigma n}$ while the components of the right handside are degree $< n$ polynomials with coefficients of magnitudes below 3.

4 From LWE to SIS

We show that if an efficient algorithm solves LWE with some non-negligible probability, then it may be used by a quantum machine to efficiently solve SIS with non-negligible probability. A crucial property of the reduction is that the matrix underlying the SIS and LWE instances is preserved: if the initial algorithm solves LWE for a given matrix $G \in \mathbb{Z}_q^{m\times n}$, then the resulting algorithm solves SIS for the same matrix G. That property allows the reduction to remain valid while working on Ideal-SIS and Ideal-LWE.

Theorem 4.1. *Let q, m, n be integers, and $\alpha \in (0,1)$ with $\mathcal{P}oly(n) \geq m \geq 2n\log q$ and $\alpha < \frac{1}{16\sqrt{m}}$. Suppose that there exists an algorithm that solves $\mathrm{LWE}_{m,q;\Psi_{\alpha q}}$ in time $T = 2^{o(n)}$ and with probability $\varepsilon = 2^{-o(n)}$. Then there exists a quantum algorithm that solves $\mathrm{SIS}_{m,q;\frac{\sqrt{n}}{2\alpha}}$ in time $\mathcal{P}oly(T,n)$ and with probability $\frac{\varepsilon^3}{16} - O(\varepsilon^5)$. The result still holds when replacing $\mathrm{LWE}_{m,q;\Psi_{\alpha q}}$ by $\text{Ideal-LWE}^f_{m,q;\Psi_{\alpha q}}$ and $\mathrm{SIS}_{m,q;\frac{\sqrt{n}}{2\alpha}}$ by $\text{Ideal-SIS}^f_{m,q;\frac{\sqrt{n}}{2\alpha}}$, for $f = x^n + 1$ with $n = 2^k$ and $q \equiv 3 \bmod 8$.*

The reduction is made of two components. First, we argue that an algorithm solving LWE provides an algorithm that solves a certain closest vector problem, where the difference to the lattice is distributed according to an n-dimensional Gaussian. In a second step, we show that Regev's quantum algorithm [33, Lemma 3.14] can use such an algorithm to construct small solutions to SIS.

4.1 From LWE to CVP

An algorithm solving LWE allows us to solve, for certain lattices, a variation of the Closest Vector Problem. In that variation of CVP, the error vector is sampled according to a specified distribution.

Definition 4.1. *The problem* CVP_χ *with parameter distribution* $\chi(\cdot)$ *is as follows: Given an* n*-dimensional lattice* L *and a vector* $\boldsymbol{t} = \boldsymbol{b} + \boldsymbol{e}$ *where* $\boldsymbol{b} \in L$ *and* $\boldsymbol{e}$ *is distributed according to* $\chi(n)$*, the goal is to find* $\boldsymbol{b}$*.*

We show that if we have an algorithm that solves $\mathrm{LWE}_{m,q;\Psi_{\alpha q}}$, then we can contruct an algorithm solving $\mathrm{CVP}_{\nu_{\alpha q}}$ for some lattices.

Lemma 4.1. *Let* q, m, n *be integers and* $\alpha \in (0,1)$*, with* $m, \log q = \mathcal{P}oly(n)$ *and* $\alpha < \frac{1}{16\sqrt{m}}$*. Suppose that there exists an algorithm* $\mathcal{A}$ *that solves* $\mathrm{LWE}_{m,q;\Psi_{\alpha q}}$ *in time* $T = 2^{o(n)}$ *and with probability* $\varepsilon = 2^{-o(n)}$*. Then there exists* $\mathcal{S} \subseteq \mathbb{Z}_q^{m \times n}$ *of proportion* $\geq \varepsilon/2$ *and an algorithm* $\mathcal{A}'$ *such that if* $G \in \mathcal{S}$*, algorithm* $\mathcal{A}'$ *solves* $\mathrm{CVP}_{\nu_{\alpha q}}$ *for* $L(G)$ *in time* $T + \mathcal{P}oly(n)$ *and with probability* $\geq \varepsilon/2 - 2^{-\Omega(n)}$*.*

Proof. If $G \in \mathbb{Z}_q^{m \times n}$ and $\boldsymbol{s} \in \mathbb{Z}_q^n$ are sampled uniformly and if the coordinates of $\boldsymbol{e}$ are sampled according to $\Psi_{\alpha q}$, then $\mathcal{A}$ finds $\boldsymbol{s}$ with probability $\geq \varepsilon$ over the choices of $G, \boldsymbol{s}$ and $\boldsymbol{e}$ and a string w of internal random bits. This implies that there exists a subset $\mathcal{S}$ of the G's of proportion $\geq \varepsilon/2$ such that for any $G \in \mathcal{S}$, algorithm $\mathcal{A}$ succeeds with probability $\geq \varepsilon/2$ over the choices of $\boldsymbol{s}$, $\boldsymbol{e}$ and w. For any $G \in \mathcal{S}$:

$$\Pr_{\boldsymbol{s},\boldsymbol{e},w}[\mathcal{A}(\boldsymbol{y}, w) = \boldsymbol{s}] \geq \varepsilon/2, \text{ where } \boldsymbol{y} = G\boldsymbol{s} + \boldsymbol{e}.$$

On input $\boldsymbol{t} = \boldsymbol{b} + \boldsymbol{e}$, algorithm $\mathcal{A}'$ works as follows: it samples $\boldsymbol{s} \in \mathbb{Z}^n$ such that it is uniformly distributed modulo q; it computes $\boldsymbol{t}' = \boldsymbol{t} + A\boldsymbol{s}$, which is of the form $\boldsymbol{t}' = G\boldsymbol{s}' + q\boldsymbol{k} + \boldsymbol{e}$, where $\boldsymbol{k} \in \mathbb{Z}^m$; it calls $\mathcal{A}$ on $\boldsymbol{t}'$ mod q and finds $\boldsymbol{s}'$ (with probability $\geq \varepsilon/2$); it then computes $\boldsymbol{e}' = \boldsymbol{t}' - G\boldsymbol{s}'$ mod q and returns $\boldsymbol{t} - \boldsymbol{e}'$. Suppose that $\mathcal{A}$ succeeds, i.e., we have $\boldsymbol{s} = \boldsymbol{s}'$. Then $\boldsymbol{e}' = \boldsymbol{e}$ mod q. Using the standard tail bound on the continuous Gaussian and the upper bound on α we obtain that with probability $\geq 1 - \exp(-\pi/(2\alpha)^2) = 1 - 2^{-\Omega(n)}$, we have $\|\boldsymbol{e}\|_\infty < q/2$. We thus have $\Pr_{\boldsymbol{e}}[\boldsymbol{e}' = \boldsymbol{e} | \boldsymbol{s}' = \boldsymbol{s}] \geq 1 - 2^{-\Omega(n)}$. □

We now show that an algorithm solving $\mathrm{CVP}_{\nu_{\alpha q}}$ can be used to solve a quantized version of the latter. This quantization is required for the quantum part of our reduction. The intuition of the proof is that the discretization grid is so fine (the parameter R can chosen extremely large) that at the level of the grid the distribution ν_s looks constant. The proof is given in appendix.

Lemma 4.2. *Suppose that there exists an algorithm* $\mathcal{A}$*, a parameter* $s > 0$ *and an* n*-dimensional lattice* L *such that* $\mathcal{A}$ *solves* CVP_{ν_s} *for* L *within time* $T = 2^{o(n)}$ *and with probability* $\varepsilon = 2^{-o(n)}$*. Then there exists an* R*, whose bit-length is polynomial in* $T, n, \log s$ *and the bit-size of the given basis of* L*, and an algorithm* $\mathcal{A}'$ *that solves* $\mathrm{CVP}_{D_{\frac{L}{R},s}}$ *with probability* $\geq \varepsilon - 2^{-\Omega(n)}$ *and within a time polynomial in* $\log R$*.*

At this point, we have an R of subexponential bit-length and a subexponential time algorithm $\mathcal{B}$ that solves $\mathrm{CVP}_{D_{\frac{L(G)}{R},\alpha q}}$, for any G in a subset $\mathcal{S} \subseteq \mathbb{Z}_q^{m\times n}$ of proportion $\geq \varepsilon/2$, with probability $\geq \varepsilon/2 - 2^{-\Omega(n)}$ over the random choices of $\boldsymbol{e}$ and the internal randomness w. We implement algorithm $\mathcal{B}$ quantumly as follows: the quantum algorithm $\mathcal{B}_Q$ maps the state $|\boldsymbol{b}\rangle\,|\boldsymbol{b}+\boldsymbol{e}\rangle\,|w\rangle$ to the state $|\boldsymbol{b}-\mathcal{B}(\boldsymbol{b}+\boldsymbol{e},w)\rangle\,|\boldsymbol{b}+\boldsymbol{e}\rangle\,|w\rangle$.

4.2 A new interpretation of Regev's quantum reduction

We first recall Regev's quantum reduction [33, Lemma 3.14]. It uses an oracle that solves CVP on a given lattice L as long as the target vector is within a prescribed distance $d < \frac{\lambda_1(L)}{2}$ of L. It returns a sample from the distribution $D_{\widehat{L},\frac{\sqrt{n}}{\sqrt{2}d}}$.

1. Set R to be a large constant and build a quantum state which is within ℓ_2 distance $2^{-\Omega(n)}$ of the normalized state corresponding to $\sum_{\boldsymbol{x}\in\frac{L}{R},\|\boldsymbol{x}\|<d} \rho_{\frac{d}{\sqrt{n}}}(\boldsymbol{x})\,|\boldsymbol{x}, \boldsymbol{x} \bmod L\rangle$.
2. Using the CVP oracle, remove the entanglement to obtain a state which is within ℓ_2 distance $2^{-\Omega(n)}$ of the normalized state corresponding to $\sum_{\boldsymbol{x}\in\frac{L}{R},\|\boldsymbol{x}\|<d} \rho_{\frac{d}{\sqrt{n}}}(\boldsymbol{x})\,|\boldsymbol{x} \bmod L\rangle$.
3. Apply the quantum Fourier transform over $\mathbb{Z}_R^n$ to obtain a state that is within ℓ_2 distance $2^{-\Omega(n)}$ of the normalized state corresponding to $\sum_{\boldsymbol{x}\in\widehat{L},\|\boldsymbol{x}\|<\frac{n}{d}} \rho_{\frac{\sqrt{n}}{d}}(\boldsymbol{x})\left|\boldsymbol{x} \bmod (R\cdot\widehat{L})\right\rangle$.
4. Measure the latter to obtain a vector $\widehat{\boldsymbol{b}} \bmod R\cdot\widehat{L}$. Using Babai's algorithm [8], recover $\widehat{\boldsymbol{b}}$ and output it. Its distribution is within statistical distance $2^{-\Omega(n)}$ of $D_{\widehat{L},\frac{\sqrt{n}}{\sqrt{2}d}}$.

First, one should notice that Regev's CVP oracle in [33] is in fact a randomized algorithm, so at Step 2 the quantum oracle is also given a string $|w\rangle$ of random qubits. The oracle succeeds with probability that is exponentially close to 1 (over the choice of w). Furthermore, one can notice that Regev's algorithm does not require a worst-case CVP solver: it suffices that the oracle solves $\mathrm{CVP}_{D_{\frac{L}{R},\frac{d}{\sqrt{2n}}}}$. Finally, we show below that the $\mathrm{CVP}_{D_{\frac{L}{R},\frac{d}{\sqrt{2n}}}}$ solver does not even need to succeed with overwhelming probability, but only with non-negligible probability.

Lemma 4.3. *Suppose that we are given an n-dimensional lattice L, a parameter $R > 2^{2n}\lambda_n(L)$, a parameter $s < \frac{\lambda_1(L)}{2\sqrt{2n}}$ and a probabilistic oracle that solves $\mathrm{CVP}_{D_{\frac{L}{R},s}}$ with probability $\varepsilon = 2^{-o(n)}$. If one replaces the worst-case* CVP *solver by latter probabilistic oracle, then Regev's algorithm (with parameter $d = \sqrt{2n}s$) outputs a vector $\widehat{\boldsymbol{b}} \in \widehat{L}$ whose distribution is within distance $1-\varepsilon^2/2+O(\varepsilon^4)$ of $D_{\widehat{L},\frac{1}{2s}}$. It finishes in time polynomial in* $\log R$.

Proof. We just saw that if the $\mathrm{CVP}_{D_{\frac{L}{R},s}}$ oracle was succeeding with probability $1-2^{-\Omega(n)}$, then the output vector $\widehat{\boldsymbol{b}}$ would follow a distribution whose statistical distance to $D_{\widehat{L},\frac{1}{2s}}$ would be $2^{-\Omega(n)}$. To work around the requirement that the oracle succeeds with overwhelming probability, we use the notion of trace distance between two quantum states, which is an adaptation of the statistical distance (see [26, Ch. 9]). The trace distance between two (pure) quantum states $|t_1\rangle$ and $|t_2\rangle$ is $\delta(|t_1\rangle, |t_2\rangle) = \sqrt{1-|\langle t_1|t_2\rangle|^2}$. Its most important property is that for any generalized measurement (POVM), if D_1 (resp. D_2) is the resulting probability distribution when starting from $|t_1\rangle$ (resp. $|t_2\rangle$) then $\Delta(D_1,D_2) \leq \delta(|t_1\rangle, |t_2\rangle)$. Now, if we call $|t_1\rangle$ the ideal state at the end of Step 2 of Regev's algorithm, and if $|t_2\rangle$ is the state that we obtain by using our imperfect oracle, then we

have $|\langle t_1|t_2\rangle| \geq \varepsilon - 2^{-\Omega(n)}$. Therefore, we have that the output distribution is within statistical distance $\sqrt{1-\varepsilon^2+2^{-\Omega(n)}}$ of $D_{\widehat{L},\frac{1}{2s}}$. □

To prove Theorem 4.1, we apply Lemma 4.3 to the lattices $L(G)$ for $G \in \mathcal{S}$, with algorithm $\mathcal{B}$. For that, we need to ensure that the hypothesis $\alpha q < \frac{\lambda_1(L(G))}{2\sqrt{2m}}$ is satisfied. From Lemma 2.4 (and Lemma 2.5 in the case of Ideal-LWE), we know that with probability $1-2^{-\Omega(n)}$ over the choice of G in $\mathbb{Z}_q^{m\times n}$, we have $\lambda_1^\infty(L(G)) \geq \frac{q}{4}$. We consider the set $\mathcal{S}'$ of the G's in $\mathcal{S}$ for which that condition is satisfied. The set $\mathcal{S}'$ represents a proportion $\geq \varepsilon/2 - 2^{-\Omega(n)}$ of $\mathbb{Z}_q^{m\times n}$. Suppose now that $G \in \mathcal{S}'$. Lemma 4.3 shows that we can find a vector $\boldsymbol{s} \in G^\perp = q\widehat{L(G)}$ that follows a distribution whose distance to $D_{G^\perp,\frac{1}{2\alpha}}$ is $1-\varepsilon^2/8+O(\varepsilon^4)$. Thanks to Lemmas 2.1 and 2.2 (since $G \in \mathcal{S}'$, we have $\frac{1}{2\alpha} \geq \eta_{2^{-n}}(G^\perp)$), we have that with probability $1-2^{-\Omega(n)}$ the returned $\boldsymbol{s}$ is a non-zero vector of $G^\perp$ whose norm is $\leq \frac{\sqrt{n}}{2\alpha}$.

5 Cryptographic Applications

We now use the results of Sections 3 and 4 to construct efficient cryptographic primitives based on ideal lattices. This includes the first lattice-based public-key encryption scheme with asymptotically optimal encryption and decryption computation costs of $\widetilde{O}(1)$ bit operations per message bit, as well as trapdoor signatures and ID-based identification schemes.

5.1 Efficient public-key encryption scheme

Our public key encryption scheme is constructed in two steps. First, we use the LWE mapping $(\boldsymbol{s},\boldsymbol{e}) \to G\cdot\boldsymbol{s}+\boldsymbol{e} \bmod q$ as an injective trapdoor one-way function, with the trapdoor being the full-dimensional set of vectors in $G^\perp$ from Section 3, and the one-wayness being as hard as Ideal-SIS (and hence Ideal-SVP) by Theorem 4.1. This trapdoor function is an efficient ideal lattice analogue of some trapdoor functions presented in [12, 29] for arbitrary lattices. In the second step, we apply the Goldreich-Levin hard core function based on Toeplitz matrices [13, Sec. 2.5] to our trapdoor function, and XOR the message with the hard core bits to obtain a semantically secure encryption.

Our trapdoor function family $\mathsf{Id}-\mathsf{Trap}$ is defined in Figure 1. For security parameter $n = 2^k$, we fix $f(x) = x^n+1$ and $q = O(\mathcal{P}oly(n))$ a prime satisfying $q \equiv 3 \bmod 8$. From Lemma 2.3, it follows that f splits modulo q into exactly two irreducible factors of degree $n/2$. We set $\sigma = 1$, $r = 1+\log_3 q = \widetilde{O}(1)$ and $m = (\lceil \log q \rceil + 1)\sigma + r = \widetilde{O}(1)$. We let $\mathcal{R}$ denote the ring $\mathbb{Z}_q[x]/f$. The following lemma (whose proof is in appendix) ensures the correctness of the scheme (this is essentially identical to [29, Sec. 4.1]) and asserts that the evaluation and inversion functions can be implemented efficiently.

Lemma 5.1. *Let $q > 2\sqrt{mn}L$ and $\alpha < \frac{1}{\omega(\sqrt{\log n})L}$. Then for any $s \in \mathcal{R}$ and for $\boldsymbol{e}$ sampled from $\overline{\Psi}_{\alpha q}$, the inversion algorithm recovers $(\boldsymbol{s},\boldsymbol{e})$ with probability $1-n^{-\omega(1)}$ over the choice of $\boldsymbol{e}$. Furthermore, the evaluation and inversion algorithms for $h_{\boldsymbol{g}}$ can be implemented with run-time $\widetilde{O}(n)$.*

The one-wayness of $\mathsf{Id}-\mathsf{Trap}$ is equivalent to $\mathrm{LWE}_{m,q,\overline{\Psi}_{\alpha q}}$. Furthermore, an instance $(G, \boldsymbol{y} = G\boldsymbol{s}+\boldsymbol{e} \bmod q)$ of $\mathrm{LWE}_{m,q,\Psi_{\alpha q}}$ can be efficiently converted by rounding to an instance of $\mathrm{LWE}_{m,q,\overline{\Psi}_{\alpha q}}$, namely $(G, \lceil \boldsymbol{y} \rfloor) = (G, G\boldsymbol{s} + \lceil \boldsymbol{e} \rfloor \bmod q)$. This proves Lemma 5.2.

- **Generating a function with trapdoor.** Run the algorithm from Theorem 3.1, using $f = x^n+1, n, q, r, \sigma$ and m as inputs. Suppose it succeeds. It gives a vector $\boldsymbol{g} \in (\mathbb{Z}_q[x]/f)^m$ (function index) and a trapdoor full-rank set S of linearly independent vectors in $\mathrm{rot}_f(\boldsymbol{g})^\perp \subseteq \mathbb{Z}_q^{mn \times mn}$ with $\|S\| \leq 2\sqrt{(9r+\sigma)} \cdot \sqrt{n} =: L$ (we have $L = \widetilde{O}(\sqrt{n})$).
- **Function evaluation.** Given function index $\boldsymbol{g}$, we define the trapdoor function $h_{\boldsymbol{g}} : \mathbb{Z}_q^n \times \mathbb{Z}_q^{mn}$ as follows. On input $\boldsymbol{s}$ uniformly random in $\mathbb{Z}_q^n$ and $\boldsymbol{e} \in \mathbb{Z}_q^{mn}$ sampled from $\overline{\Psi}_{\alpha q}$ (defined as $\Psi_{\alpha q}$ after rounding to the closest integer vector), we compute and return:

$$\boldsymbol{c} = h_{\boldsymbol{g}}(\boldsymbol{s}, \boldsymbol{e}) := \mathrm{rot}_f(\boldsymbol{g}) \cdot \boldsymbol{s} + \boldsymbol{e} \bmod q.$$

- **Function inversion.** Given $\boldsymbol{c} = h_{\boldsymbol{g}}(\boldsymbol{s}, \boldsymbol{e})$ and trapdoor S, compute $\boldsymbol{d} = S^T \cdot \boldsymbol{c} \bmod q$ and $\boldsymbol{e}' = S^{-T} \cdot \boldsymbol{d}$ (in $\mathbb{Q}$). Compute $\boldsymbol{u} = \boldsymbol{c} - \boldsymbol{e}' \bmod q$ and $\boldsymbol{s}' = (\mathrm{rot}_f(g_1))^{-1} \cdot \boldsymbol{u}_1 \bmod q$, where $\boldsymbol{u}_1$ consists of the first n coordinates of $\boldsymbol{u}$. Return $(\boldsymbol{s}', \boldsymbol{e}')$.

Fig. 1. The trapdoor function family $\mathsf{Id-Trap}$.

Lemma 5.2. *Suppose that there exists an attacker against the one-wayness of* $\mathsf{Id-Trap}$ *(with parameters m, α, q) with run-time T and success probability ε. Then there exists an algorithm that solves* $\mathrm{LWE}_{m,q;\Psi_{\alpha q}}$ *with run-time T and success probability ε.*

By combining our trapdoor function with the GL hard core function [13, Sec. 2.5] we get the encryption scheme of Figure 2. The semantic security (in the sense of IND-CPA) follows from the one-wayness of $\mathsf{Id-Trap}$ (i.e., from the hardness of Ideal-LWE) and from the GL hard core function.

- **Key generation.** For security parameter n, run the key generation algorithm of $\mathsf{Id-Trap}$ to get an index $\boldsymbol{g}$ for $h_{\boldsymbol{g}} : \mathbb{Z}_q^n \times (\mathbb{Z}_q)^{nm}$ and a trapdoor S. We can view the domain of $h_{\boldsymbol{g}}$ as a subset of $\mathbb{Z}_2^{\ell_i}$ for $\ell_i = O((n+1)m \log q) = \widetilde{O}(n)$. The public key is $\boldsymbol{g}$ and the secret key is S.
- **Encryption.** Given ℓ_m-bit message m and public key $\boldsymbol{g}$, sample $(\boldsymbol{s}, \boldsymbol{e})$ with $\boldsymbol{s}$ uniform in $\mathbb{Z}_q^n$ and $\boldsymbol{e}$ sampled from $\overline{\Psi}_{\alpha q}$, and evaluate $c_1 = h_{\boldsymbol{g}}(\boldsymbol{s}, \boldsymbol{e})$. Generate a uniformly random vector $\boldsymbol{r} \in \mathbb{Z}_2^{\ell_m + \ell_i}$. Compute $c_2 = m \oplus (M_{GL} \cdot (\boldsymbol{s}, \boldsymbol{e}))$, where the product $M_{GL} \cdot (\boldsymbol{s}, \boldsymbol{e})$ is computed over $\mathbb{Z}_2$, and $(\boldsymbol{s}, \boldsymbol{e})$ is viewed as a string over $\mathbb{Z}_2^{\ell_i}$. The GL matrix $M_{GL} \in \mathbb{Z}_2^{\ell_m \times \ell_i}$ is Toeplitz (thus allowing fast multiplication [27]) and its ith row is $[r_i, \ldots, r_{\ell_i + i - 1}]$. Return the ciphertext $(c_1, c_2, \boldsymbol{r})$.
- **Decryption.** Given ciphertext $(c_1, c_2, \boldsymbol{r})$ and secret key S, invert c_1 to compute $(\boldsymbol{s}, \boldsymbol{e})$ such that $h_{\boldsymbol{g}}(\boldsymbol{s}, \boldsymbol{e}) = c_1$, and return $m = c_2 \oplus (M_{GL} \cdot (\boldsymbol{s}, \boldsymbol{e}))$.

Fig. 2. The semantically secure encryption scheme $\mathsf{Id-Enc}$.

Theorem 5.1. *Suppose that there exists an IND-CPA attacker against* $\mathsf{Id-Enc}$ *running in time T and having success probability $1/2 + \varepsilon$. Then there exists an algorithm for* $\text{Ideal-LWE}^{\tilde{f}}_{m,q;\Psi_{\alpha q}}$ *with run-time $T' = O(2^{3\ell_m} n^3 \varepsilon^{-3} \cdot T)$ and success probability $\varepsilon' = \Omega(2^{-\ell_m} n^{-1} \cdot \varepsilon)$.*

Proof. The given IND-CPA attacker can be converted to a GL hard core function distinguisher that, given $c_1 = h_{\boldsymbol{g}}(\boldsymbol{s}, \boldsymbol{e})$, M_{GL}, and ℓ_m bit string z, for $\boldsymbol{s}$ sampled uniformly in $\mathbb{Z}_q^n$, $\boldsymbol{e}$ sampled from $\overline{\Psi}_{\alpha q}$, and M_{GL} constructed as in the key generation procedure, distinguishes whether z is uniformly random (independent of $\boldsymbol{s}$ and $\boldsymbol{e}$) or $z = M_{GL} \cdot (\boldsymbol{s}, \boldsymbol{e})$. It has run-time T and advantage ε. The result follows by applying to the function $h_{\boldsymbol{g}}$ the Lemma 2.5.8, Proposition 2.5.7 and Proposition 2.5.3 in [13]. □

By using Lemma 5.1 and Theorems 5.1, 4.1, and 2.1, we get our main result.

Corollary 5.1. *Suppose that there exists an IND-CPA attacker against the encryption scheme* $\mathsf{Id}-\mathsf{Enc}$ *with message length* $\ell_m = n/\log n = o(n)$*, running in time* $2^{o(n)}$ *and having success probability* $1/2 + 2^{-o(n)}$*. Then there exists a quantum algorithm for* $\widetilde{O}(n^2)$-Ideal-SVP *with* $f = x^n + 1$ *and* $n = 2^k$*. It has run-time* $2^{o(n)}$ *and overwhelming success probability. Furthermore, the scheme* $\mathsf{Id}-\mathsf{Enc}$ *has encryption and decryption costs* $\widetilde{O}(1)$ *bit operations per encrypted message bit, and keys of* $\widetilde{O}(n)$ *bits.*

5.2 Trapdoor signature scheme

Gentry *et. al.* [12] showed how to derive a probabilistic full domain hash trapdoor signature construction PFDHS with key generation, signing and verification algorithms, from any Trapdoor Collision Resistant Hash Family with Preimage Sampling (TCRPS). Informally, a TCRPS f_a is a collision resistant function (family) for which there is a (secret) trapdoor t that allows to efficiently sample a 'random' preimage under f_a of a given target y. The trapdoor t is the signer's secret key, and the signature on a message m has the form $(f_a^{-1}(H(m,r)), r)$, for a hash function H and random 'salt' r. It is shown in [12] how to perform such preimage sampling for the function $f_G(\boldsymbol{x}) = \boldsymbol{x}^T G$, where $G \in \mathbb{Z}_q^{m\times n}$, using a trapdoor basis of $G^\perp$ with small GSO norm. By applying this preimage sampling algorithm to a structured $G = \mathrm{rot}_f(\boldsymbol{g})$ and using the trapdoor generation algorithm from Section 3, we obtain a TCRPS (whose collision resistance relies on Ideal-SIS and hence Ideal-SVP) and thus a structured variant of the trapdoor signature scheme of [12], with $\widetilde{O}(n)$ verification time and signature length. More details will be given in the full version.

5.3 Identity-based identification schemes

From lattice-based signatures, we derive ID-based identification (IBI) and ID-based signature (IBS) schemes based on lattice problems. We first observe that the Micciancio-Vadhan (MV) protocol [25] for $O(\sqrt{n})$-CVP together with the lattice-based hash functions [1, 19] yield concurrently secure identification schemes. Briefly speaking, in the schemes, the secret key is a short vector $\boldsymbol{e} \in \mathbb{Z}^m$ and the public key is a tuple of $G \in \mathbb{Z}_q^{m\times n}$ and $\boldsymbol{y} = \boldsymbol{e}^T G \bmod q$. The public key is interpreted by both the verifier and the prover as a pair of a lattice $G^\perp$ and a vector $\boldsymbol{t}$ such that $\boldsymbol{y} = \boldsymbol{t}^T G \bmod q$. The prover shows that it has the lattice vector $\boldsymbol{s} = \boldsymbol{t} - \boldsymbol{e} \in G^\perp$. The hardness of SIS (resp. Ideal-SIS) yields the concurrent security.

Applying the standard strategy, we construct lattice-based IBI schemes as follows: The master generates a key pair of a lattice-based signature scheme, say (G, S); Each user obtains from the master a short vector $\boldsymbol{e}$ such that $\boldsymbol{e}^T G = H(id)$, where H is a random oracle; The prover proves to the verifier that he/she has a short vector $\boldsymbol{e}$ through the MV protocol. This combination yields concurrently secure IBI schemes based on $\widetilde{O}(n^2)$-SVP and $\widetilde{O}(n^2)$-Ideal-SVP in the random oracle model. As the MV protocol is witness indistinguishable, we can use the Fiat-Shamir heuristic [11] and obtain IBS schemes based on lattice problems. The details and rigorous proofs will appear in the full paper.

References

1. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the 28th Symposium on the Theory of Computing (STOC 1996)*, pages 99–108. ACM Press, 1996.
2. M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the 30th Symposium on the Theory of Computing (STOC 1998)*, pages 284–293. ACM Press, 1998.

3. M. Ajtai. Generating hard instances of the short basis problem. In *Proceedings of the 1999 International Colloquium on Automata, Languages and Programming (ICALP 1999)*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer-Verlag, 1999.
4. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th Symposium on the Theory of Computing (STOC 1997)*, pages 284–293. ACM Press, 1997.
5. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd Symposium on the Theory of Computing (STOC 2001)*, pages 601–610. ACM Press, 2001.
6. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *Proceedings of the annual symposium on theoretical aspects of computer science (STACS 2009)*, Lecture Notes in Computer Science. Springer-Verlag, 2009.
7. Y. Arbitman, G. Dogon, V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFTX: a proposal for the SHA-3 standard. Submission to NIST, 2008.
8. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
9. I. F. Blake, S. Gao, and R. C. Mullin. Explicit factorization of $x^{2^k} + 1$ over f_p with prime $p \equiv 3 \bmod 4$. *App. Alg. in Eng., Comm. and Comp*, 4:89–94, 1992.
10. L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.
11. A. Fiat and A. Shamir. How to prove yourself – practical solutions to identification and signature problems. In *Proceedings of Crypto 1986*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1986.
12. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Symposium on the Theory of Computing (STOC 2008)*, pages 197–206. ACM Press, 2008.
13. O. Goldreich. *Foundations of Cryptography*, volume II – Basic Applications. Cambridge University Press, 2001.
14. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Proceedings of Crypto 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer-Verlag, 1997.
15. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring based public key cryptosystem. In *Proceedings of the 3rd Algorithmic Number Theory Symposium (ANTS III)*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.
16. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Proceedings of Asiacrypt 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer-Verlag, 2008.
17. V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Proceedings of PKC 2008*, pages 162–179, 2008.
18. V. Lyubashevsky. *Towards Practical Lattice-Based Cryptography*. PhD thesis, University of California, San Diego, 2008.
19. V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *Proceedings of the 2006 International Colloquium on Automata, Languages and Programming (ICALP (2))*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer-Verlag, 2006.
20. V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *Proceedings of the 2008 Theory of Cryptography Conference (TCC)*, pages 37–54, 2008.
21. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.
22. D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective*. Kluwer Academic Press, 2002.
23. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.
24. D. Micciancio and O. Regev. *Post-Quantum Cryptography*, chapter Lattice-based Cryptography. Springer-Verlag, 2008.
25. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Proceedings of Crypto 2003*, pages 282–298, 2003.
26. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
27. V. Y. Pan. *Structured matrices and polynomials, unified superfast algorithms*. Springer-Verlag and Birkhäuser, 2001.
28. C. Peikert. Limits on the hardness of lattice problems in ℓ_p norms. *Computational Complexity*, 2(17):300–351, 2008.

29. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem, 2009. To appear in the proceedings of STOC'09. Available at `http://people.csail.mit.edu/cpeikert/`.
30. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proceedings of the 2006 Theory of Cryptography Conference (TCC)*, pages 145–166, 2006.
31. C. Peikert and A. Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *Proceedings of the 39th Symposium on the Theory of Computing STOC 2007*, pages 478–487. ACM Press, 2007.
32. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Proceedings of Crypto 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer-Verlag, 2008.
33. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. Extended version of [34] dated May 2, 2009. Available at the URL `http://www.cs.tau.ac.il/~odedr/`.
34. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Symposium on the Theory of Computing (STOC 2005)*, pages 84–93. ACM Press, 2005.
35. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *Proceedings of TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*, pages 419–436. Springer-Verlag, 2009.
36. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theor. Comput. Sci*, 53:201–224, 1987.

Proof of Lemma 2.5. By using Lemma 2.3, we know that $f = f_1 f_2 \bmod q$ where, for each $i \in \{1,2\}$, the polynomial f_i is irreducible in $\mathbb{Z}_q[x]$ and can be written $f_i = x^{n/2} + t_i x^{n/4} - 1$ for some $t_i \in \mathbb{Z}_q$.

Let $s \in \mathbb{Z}_q^n$ and $\boldsymbol{v} \in \mathbb{Z}_q^{mn}$. We want to bound the probability that $\mathrm{rot}_f(\boldsymbol{g})\boldsymbol{s} = \boldsymbol{v}$ when $\boldsymbol{g}$ is taken uniformly in $(\mathbb{Z}_q[x]/f)^m$. We write $\boldsymbol{g} = (g_1, \ldots, g_m)^T$, and we call $s, v_1, \ldots, v_m$ the elements of $\mathbb{Z}_q[x]/f$ that correspond to $\boldsymbol{s}$ and $\boldsymbol{v}$ (identifying vector coordinates and polynomial coefficients). We define the map ϕ_s from $\mathbb{Z}_q[x]/f$ to itself that maps g to $g \cdot s$. Using the $g \mapsto \bar{g}$ isomorphism (see Lemma 2.3), we see that our goal is exactly the same as bounding $\prod_{j \le m} \Pr_g[\phi_s(g) = v_j]$.

We consider two cases, depending whether s and f are coprime or not. If s and f are coprime, then ϕ_s is a bijection, and $\Pr_g[\phi_s(g) = v_j] = q^{-n}$. Suppose now that $s = f_i s'$ for some $i \in \{1,2\}$ and $s' \in \mathbb{Z}_q[x]$ of degree $< n/2$. If v_j is not of the form $f_i v_j'$ for some $v_j' \in \mathbb{Z}_q[x]$ of degree $< n/2$, then $\Pr_g[\phi_s(g) = v_j] = 0$. Otherwise, since the kernel of ϕ_s is of cardinality $q^{n/2}$, we have $\Pr_g[\phi_s(g) = v_j] = q^{-n/2}$.

We now take a union bound over all the vectors $\boldsymbol{s} \in \mathbb{Z}_q^n \setminus \boldsymbol{0}$ and the vectors $\boldsymbol{v} \in \mathbb{Z}_q^m$ such that $\|\boldsymbol{v}\|_\infty < q/4$. The probability (over the $\boldsymbol{g}$) that we have $\lambda_1^\infty(L(\mathrm{rot}_f(\boldsymbol{g}))) < q/4$ is upper bounded by:

$$\sum_{\substack{\boldsymbol{s} \in \mathbb{Z}_q^n \\ \gcd(s,f)=1}} \sum_{\substack{\boldsymbol{v} \in \mathbb{Z}_q^m \\ \|\boldsymbol{v}\|_\infty < q/4}} \prod_{j \le m} \Pr_g[\phi_s(g) = v_i] + 2 \sum_{\substack{\boldsymbol{s} \in \mathbb{Z}_q^n \\ f_1 | s}} \sum_{\substack{\boldsymbol{v} \in \mathbb{Z}_q^m \\ \|\boldsymbol{v}\|_\infty < q/4}} \prod_{j \le m} \Pr_g[\phi_s(g) = v_i].$$

The first term is $\le q^n (q/2)^{mn} q^{-mn}$. The second term is $\le 2q^{n/2} N^m q^{-mn/2}$, where N is the number of $v \in \mathbb{Z}_q[x]/f$ such that $\|\boldsymbol{v}\| < q/4$ and $v = f_1 v'$ for some $v' \in \mathbb{Z}_q[x]$ of degree $< n/2$. Thanks to the shape of f_1, the latter conditions imply that $\|\boldsymbol{v}''\|_\infty < q/4$ where $\boldsymbol{v}'' \in \mathbb{Z}_q[x]$ is the vector made of the $n/4$ lower degree coefficients of v'. As a consequence, we have $N \le q^{n/2}/2^{n/4}$. A straightforward computation provides the result. □

Proof of Lemma 4.2. Since $\mathcal{A}$ runs in time T, it cannot look at more than $B \le T$ bits of the samples from ν_s. So it actually solves CVP_{D_1} with $D_1 = 2^{-B} \lfloor 2^B \nu_s \rfloor$. We have, for any $\boldsymbol{x} \in \mathbb{Z}^n$: $D_1(2^{-B}\boldsymbol{x}) = \int_{\boldsymbol{t} \in 2^{-B}(\boldsymbol{x} + [0,1]^n)} \nu_s(\boldsymbol{t}) d\boldsymbol{t}$. We choose $R = 16 \cdot 2^{2nB+2n} n^{3n} \cdot \max(s^n, 1/s^n) \cdot \max_i \|\boldsymbol{b}_i^*\|$, where the $\boldsymbol{b}_i^*$'s are the GSO of the given basis.

Algorithm $\mathcal{A}'$ works as follows: it takes $\boldsymbol{t} = \boldsymbol{b}+\boldsymbol{e}$ as input, where $\boldsymbol{b} \in L$ and $\boldsymbol{e}$ is distributed according to $D_{\frac{L}{R},s}$; it rounds $\boldsymbol{t}$ to $2^{-B}\mathbb{Z}^n$ and calls algorithm $\mathcal{A}$; it returns the output of $\mathcal{A}$. For the result to hold, it suffices to show that $\Delta(D_1, D_2) = 2^{-\Omega(n)}$, where $D_2 = 2^{-B}\lfloor 2^B D_{\frac{L}{R},s}\rfloor$. From [33, Lemma 3.2], we know that $\Delta(D_{\frac{L}{R},s}, \nu_s - (\nu_s \bmod \frac{L}{R})) = 2^{-\Omega(n)}$. It thus suffices to show that $\Delta(D_1, D_3) = 2^{-\Omega(n)}$, where $D_3 = 2^{-B}\lfloor 2^B(\nu_s - (\nu_s \bmod \frac{L}{R}))\rfloor$. We have, for all $\boldsymbol{x} \in \mathbb{Z}^n$:

$$D_3(2^{-B}\boldsymbol{x}) = \sum_{\frac{\boldsymbol{b}}{R} \in \frac{L}{R} \cap 2^{-B}(\boldsymbol{x}+[0,1]^n)} \int_{\boldsymbol{t} \in \frac{1}{R}(\boldsymbol{b}+P(L))} \nu_s(\boldsymbol{t})d\boldsymbol{t} = \int_{\boldsymbol{t} \in T(\boldsymbol{x})} \nu_s(\boldsymbol{t})d\boldsymbol{t},$$

where $T(\boldsymbol{x}) = \{\boldsymbol{t} : \boldsymbol{t} - (\boldsymbol{t} \bmod \frac{L}{R}) \in 2^{-B}(\boldsymbol{x} + [0,1]^n)\}$. As a consequence, the quantity $|D_1(2^{-B}\boldsymbol{x}) - D_3(2^{-B}\boldsymbol{x})|$ is smaller than s^{-n} times the volume of the difference between the sets $2^{-B}(\boldsymbol{x} + [0,1]^n)$ and $T(\boldsymbol{x})$. A standard argument shows that $2^{-B}(\boldsymbol{x} + [0,1]^n)$ contains $T(\boldsymbol{x})$ (resp. is contained in $T(\boldsymbol{x})$) once the latter has been shrinked (resp. enlarged) by a factor $1 \pm \frac{2^B\sqrt{n}\max\|\boldsymbol{b}_i^*\|}{R}$. Therefore, we have $|D_1(2^{-B}\boldsymbol{x}) - D_3(2^{-B}\boldsymbol{x})| \leq s^{-n}2^{-nB}\left[\left(1 + \frac{2^B\sqrt{n}\max\|\boldsymbol{b}_i^*\|}{R}\right)^n - \left(1 - \frac{2^B\sqrt{n}\max\|\boldsymbol{b}_i^*\|}{R}\right)^{-n}\right] =: C$.

Notice now that for both D_1 and D_3, we have $\Pr[\|2^{-B}\boldsymbol{x}\| > 2s\sqrt{n}] = 2^{-\Omega(n)}$. As a consequence:

$$\Delta(D_1, D_3) \leq 2^{-\Omega(n)} + \sum_{\boldsymbol{x},\ \|2^{-B}\boldsymbol{x}\| \leq 2s\sqrt{n}} |D_1(2^{-B}\boldsymbol{x}) - D_3(2^{-B}\boldsymbol{x})| \leq 2^{-\Omega(n)} + (2^{B+2}s\sqrt{n})^n C \leq 2^{-\Omega(n)}.$$

This completes the proof. □

Proof of Lemma 5.1. Let $G = \mathrm{rot}_f(\boldsymbol{g})$. By definition of the trapdoor S, we have $S^T \cdot G = 0$. It follows that $\boldsymbol{d} = S^T\boldsymbol{e} \bmod q$. Hence if we show that each entry of $S^T\boldsymbol{e}$ (computed over $\mathbb{Z}[x]/f$ without reduction modulo q) has magnitude $< q/2$, then we will have $S^T\boldsymbol{e} = S^T\boldsymbol{e} \bmod q$. But each entry of $S^T\boldsymbol{e}$ is an inner product (over $\mathbb{R}$) of the form $\langle \boldsymbol{s}_i, \boldsymbol{e}\rangle$, where $\boldsymbol{s}_i$ is the ith column of S. Now, by construction, we have $\boldsymbol{e} = \boldsymbol{e}_c + \boldsymbol{w}$, where $\boldsymbol{e}_c$ is sampled from the continuous Gaussian $\nu_{\alpha q}$ and $\boldsymbol{w}$ is the rounding error, with $\|\boldsymbol{w}\|_\infty \leq 1/2$. The inner product $\langle \boldsymbol{s}_i, \boldsymbol{e}_c\rangle$ is distributed according to ν_r with $r = \|\boldsymbol{s}_i\| \cdot \alpha q \leq L\alpha q$, so, by the Gaussian tail inequality, we have $|\langle \boldsymbol{s}_i, \boldsymbol{e}_c\rangle| \leq \omega(\sqrt{\log n}) \cdot L\alpha q$, with probability $1 - n^{-\omega(1)}$. Also, by the Schwarz inequality, we have $|\langle \boldsymbol{s}_i, \boldsymbol{w}\rangle| \leq L\sqrt{mn}/2$. Thanks to the assumptions on q and α, it follows that $|\langle \boldsymbol{s}_i, \boldsymbol{e}\rangle| < q/2$ and hence $\boldsymbol{d} = S^t \cdot \boldsymbol{e}$ (using the union bound) with probability $1 - n^{-\omega(1)}$. As a consequence, we have that $\boldsymbol{e}' = \boldsymbol{e}$, $\boldsymbol{u} = G \cdot \boldsymbol{s} \bmod q$ and $\boldsymbol{s}' = \boldsymbol{s}$, with probability $1 - n^{-\omega(1)}$. We just used the fact that $\mathrm{rot}_f(g_1)$ is full rank over $\mathbb{Z}_q$ with probability $\geq (1 - q^{-n/2})^2 > 1 - 2q^{-n/2}$ over the choice of $g_1 \in \mathcal{R}$, since f splits into two irreducible factors modulo q of degree $n/2$ each.

For the evaluation of $h_{\boldsymbol{g}}$, the matrix-vector product $G \cdot \boldsymbol{s}$ involves $m = \widetilde{O}(1)$ multiplications $g_i \cdot s \bmod q$ in $\mathcal{R}$. We compute $g_i \cdot s \in \mathcal{R}$ by first computing $g_i \cdot s$ in $\mathbb{Z}[x]/f$ (without reduction modulo q) and then reducing modulo q. The computation of $g_i \cdot s$ can be done in $\widetilde{O}(n)$ time by using a fast Fourier transform over $\mathbb{Z}_{\bar{q}}$ for some prime $\bar{q}$ such that $2n|\bar{q} - 1$ (see Lemma 2.3) and $\bar{q} > 2nq^2$ (so that $\|g_i \cdot s\|_\infty < \bar{q}/2$ and hence $g_i \cdot s \bmod \bar{q} = g_i \cdot s$). Since $\bar{q} = \mathcal{P}oly(n)$, it follows that the computation takes $\widetilde{O}(n)$ bit operations, as required. We also need to be able to sample efficiently from the one-dimensional distribution $\overline{\Psi}_{\alpha q}$ in time $\widetilde{O}(1)$. Since $\alpha q = O(n)$, this can be done with standard techniques [10].

For the inversion of $h_{\boldsymbol{g}}$, we recall that the matrix S consists of m^2 blocks $\mathrm{rot}_f(s_{i,j})$ with $s_{i,j} \in \mathbb{Z}[x]/f$. Hence the matrix S^T consists of m^2 blocks $\mathrm{rot}_f(\overline{s_{i,j}})$ (see Lemma 2.3). Therefore, the

multiplication $S^T \cdot \boldsymbol{c} \bmod q$ reduces to $m^2 = \widetilde{O}(1)$ polynomial multiplications in $\mathcal{R}$, which can be done in time $\widetilde{O}(n)$, as explained above. For the multiplication $S^{-T} \cdot \boldsymbol{d}$ in $\mathbb{Q}$ we proceed as follows. We have $S^{-T} = \mathrm{rot}_f(\bar{S}^{-1})$, where $\bar{S} = (\overline{s_{i,j}})_{i,j}$ and the inverse is over the field $\mathbb{Q}[x]/f$. We precompute (during the key generation) the matrix $\bar{S}^{-1}$ over $\mathbb{Q}$. During the key generation, we also choose a prime q' which is (pairwise) coprime with all the denominators appearing in the rational matrix S^{-T}, and larger than the entries of $\boldsymbol{e}$ (the result of $S^{-T} \cdot \boldsymbol{d}$), which has entries $< q$ with probability $1 - n^{-\omega(1)}$. We compute (and store as part of the trapdoor) the matrix $T = \bar{S}^{-T} \bmod q'$. During the inversion of $h_{\boldsymbol{g}}$, we implement the product $S^{-T} \cdot \boldsymbol{d}$ over $\mathbb{Q}$ by computing $S^{-T} \cdot \boldsymbol{d} \bmod q' = \mathrm{rot}_f(T) \cdot \boldsymbol{d} \bmod q'$. The product $\mathrm{rot}_f(T) \cdot \boldsymbol{d} \bmod q'$ can be computed using $m^2 = \widetilde{O}(1)$ multiplications in $\mathbb{Z}_{q'}[x]/f$. Each of the latter polynomial multiplications takes time $\widetilde{O}(n)$, by using an FFT method as above, as long as $q' = \mathcal{P}oly(n)$.

It remains to show that we can find a $q' = \mathcal{P}oly(n)$ which is (pairwise) coprime with all the denominators in S^{-T}. For this we note that, thanks to Cramer's rule and Hadamard's inequality, the denominators of the rational entries of S^{-T} have absolute values $\leq L^{mn}$ (using the fact that $\|S\| \leq L$). Hence there are $\leq (nm)^2 \cdot \log(L^{mn}) = \widetilde{O}(n^3)$ distinct primes dividing the denominators of S^{-T} and we can take $q' = \widetilde{O}(n^3)$, as required. $\square$