Toward Basing Fully Homomorphic Encryption on Worst-Case Hardness

Craig Gentry

IBM T.J Watson Research Center
cbgentry@us.ibm.com

Abstract. Gentry proposed a fully homomorphic public key encryption scheme that uses ideal lattices. He based the security of his scheme on the hardness of two problems: an average-case decision problem over ideal lattices, and the sparse (or "low-weight") subset sum problem (SSSP).

We provide a key generation algorithm for Gentry's scheme that generates ideal lattices according to a "nice" average-case distribution. Then, we prove a worst-case / average-case connection that bases Gentry's scheme (in part) on the quantum hardness of the shortest independent vector problem (SIVP) over ideal lattices in the *worst-case*. (We cannot remove the need to assume that the SSSP is hard.) Our worst-case / average-case connection is the first where the average-case lattice is an ideal lattice, which seems to be necessary to support the security of Gentry's scheme.

1 Introduction

Recently, Gentry [10] presented a somewhat homomorphic encryption scheme that uses ideal lattices, and proved its security based on an average-case decision problem. In this paper, we focus on this somewhat homomorphic scheme and its security. Our main results are:

- Algorithms for his scheme – most importantly, a KeyGen algorithm for generating secret and public bases of an ideal lattice – that permit the scheme's semantic security to be based on a *search* problem over ideal lattices having a *nice* average-case distribution.
- A quantum worst-case / average-case reduction, which ultimately bases the security of Gentry's somewhat homomorphic scheme on the worst-case quantum hardness of the shortest independent vector problem (SIVP) over ideal lattices.

Gentry also showed that his somewhat homomorphic scheme, after some modifications, becomes "bootstrappable" and therefore can be used to construct a fully homomorphic encryption (FHE) scheme [31,10]. He proved that the FHE scheme is semantically secure if the original somewhat homomorphic scheme is semantically secure and the sparse (or "low-weight") subset sum problem (SSSP) [11,25] is hard. Those results are generic enough to work with our instantiation

T. Rabin (Ed.): CRYPTO 2010, LNCS 6223, pp. 116–137, 2010.

of KeyGen and the other algorithms. That is, we immediately obtain a FHE scheme whose security is based on two problems: the SSSP and worst-case quantum SIVP over ideal lattices.[1] Since the SSSP is an average-case problem, it remains an open problem to base FHE *entirely* on worst-case hardness. However, the more "troubling" of Gentry's two assumptions (in our opinion) is that the average-case decision problem over ideal lattices is hard. At least we can replace this assumption with one involving worst-case hardness.

1.1 Related Work

In 1996, Ajtai [1] found a surprising reduction of worst-case lattice problems to average-case ones. Unlike the random self-reduction of Diffie-Hellman, where the worst-case and average-case instances are over the same group G, Ajtai's worst-case problem is a completely general problem (over lattices) that is unconstrained by any parameters in the average-case problem. The average-case lattices in Ajtai's reduction are of a certain type: those generated by random parity-check matrices modulo an integer q.

Following Ajtai, improved worst-case / average-case connections were described in [8,22,23,28,24,26,20]. Also, various primitives have been based on worst-case hardness, including collision-resistant hash functions [1,8,22,17,24,27], public-key encryption [3,29,30,12,26,19], signatures [18,12], and (hierarchical) identity-based encryption [12,9,7]. Ajtai [2] showed how to generate his average-case lattices together with a short secret basis for the lattice that can be used as a decryption key in an encryption scheme [12]; Alwen and Peikert [4] tightened this result.

However, as far as we know, previous worst-case / average-case reductions cannot be used to base Gentry's somewhat homomorphic scheme on worst-case hardness. The essential problem is that Gentry's scheme [10] uses *ideal lattices* and relies heavily on the structure of these lattices as algebraic ideals in a ring to obtain homomorphism. However, in none of the previous reductions is the *average-case* lattice an ideal lattice.

Some previous work describes worst-case / average-case reductions where the worst-case lattice is an ideal lattice, and the average-case instances are *derived* from ideal lattices, in a fashion somewhat similar to how Ajtai's average-case lattices are derived from a worst-case instance. For example, for the ring $R = \mathbb{Z}[x]/(x^n - 1)$ and fixed $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R^m$, Micciancio [22,23] considered the lattice formed by solutions $\mathbf{v}_1, \ldots, \mathbf{v}_m \in R^m$ to $\sum_i \mathbf{a}_i \times \mathbf{v}_i = \mathbf{0}$, and showed that solving the bounded distance decoding problem (BDDP) or SIVP for such "quasi-cyclic" lattices in the average-case allows one to solve the BDDP or SIVP for "cyclic lattices" (ideal lattices in R) in the worst-case. While Micciancio's worst-case lattices are ideal lattices, the average-case lattices are not; they correspond to modules, rather than ideals. Peikert and Rosen [28] demonstrated a

[1] Technically, both in [10] and here, a "circular-security" assumption is also needed to obtain an FHE scheme whose public key size is independent of the circuit depth of the functions being homomorphically evaluated.

very tight worst-case / average-case reduction where the worst-case lattices are ideal lattices, and where the average-case lattices are derived from ideal lattices in a way similar to that used by Micciancio. Some other results in this line of work include [27,17,20].

However, again, previous work does not provide a worst-case / average-case "random self-reduction" where both average-case and worst-case lattices are ideal lattices of the same dimension in the same ring, which seems to be necessary to preserve the algebraic structure used by Gentry's scheme, and thus necessary to support the security of Gentry's somewhat homomorphic scheme. This suggests that we need an approach fundamentally different from Ajtai's and other previous work. We also need a KeyGen algorithm for Gentry's scheme that generates an ideal lattice, together with a secret basis of the lattice, according to the appropriate average-case distribution.

1.2 Our Worst-Case / Average-Case Self-reduction

We provide the first worst-case / average-case self-reduction where the average-case lattice is an ideal lattice. We focus on the reduction for BDDP over ideal lattices, but this reduction can be extended to other ideal lattice problems. Combining with other results presented here and in prior work, this reduction bases the security of Gentry's somewhat homomorphic scheme on worst-case hardness.

Our reduction makes heavy use of the algebraic properties of ideals. Interestingly, and quite unlike other worst-case / average-case reductions, our reduction uses an integer factoring oracle to factor ideals in the ring. This integer factoring oracle can be instantiated efficiently with quantum computation [32], and hence we get an efficient quantum reduction. The reduction is also meaningful in the classical setting, since there are known sub-exponential factoring algorithms for factoring (e.g., the number field sieve). If solving average-case problems over ideal lattices is easy, our reduction implies that there are surprising sub-exponential algorithms for solving worst-case problems over ideal lattices.

Since our worst-case and average-case instances involve ideal lattices of the same dimension within the same ring R, one may prefer to think of our reduction as a "random self-reduction". It is an "imperfect" self-reduction in that the approximation factor is larger in the worst-case problem than in the average-case problem by a poly(n) factor (for the rings R that we use). However, as far as we know, the BDDP is hard even for sub-exponential approximation factors – i.e., for factors much larger than our reduction's poly(n) lossiness.

Roughly speaking, the reduction works as follows. We are given the basis $\mathbf{B}_M$ of a worst-case ideal lattice M that corresponds to an ideal in the ring R, together with a vector $\mathbf{t} \in \mathbb{R}^n$ that is close to some vector $\mathbf{u} \in M$; the BDDP is to output $\mathbf{u}$. To generate an average-case instance, we first sample a "random" vector $\mathbf{v}$ from the inverse ideal M^{-1} according to a particular distribution. We multiply (in the ring R) each of the basis elements of $\mathbf{B}_M$ by $\mathbf{v}$ to obtain a basis $\mathbf{B}_L$ of the lattice for the ideal $L = M \cdot (\mathbf{v})$, and set $\mathbf{u}' \leftarrow \mathbf{v} \times \mathbf{u}$. L will be an ideal in R that is not divisible by M, since $\mathbf{v} \in M^{-1}$ and thus "cancels" M.

However, due to $\mathbf{v}$'s distribution, L's *geometry* will be very closely related to M; in particular, solving BDDP for $(L, \mathbf{u}')$ will help solve BDDP for $(M, \mathbf{u})$. Toward solving BDDP for $(L, \mathbf{u}')$, we use our factoring oracle to find a "suitable" ideal J that divides L (restarting if no suitable one exists), and output the instance $(J, \mathbf{u}')$ to our average-case BDDP solver. Note that L is a subset of J. As long as L is not an overly sparse subset, and for suitable parameters, the closest vector in J to $\mathbf{u}'$ will also be in L. Hence, a BDDP solution to average-case instance $(J, \mathbf{u}')$ leads to a BDDP solution to the worst-case instance $(M, \mathbf{u})$. We show that J comes from our desired average-case distribution – i.e., that it is uniformly random among regular prime ideals in R whose norms are in a prescribed interval. Of course, the target vector $\mathbf{u}'$'s distribution is not random – i.e., is not independent of the worst-case instance – but we also show how to randomize the target vector's distribution. See Section 3 for details and proofs.

1.3 How to Generate an Average Ideal Lattice, and Other Results

In [10], Gentry mentions some *ad hoc* ways of generating an ideal lattice, together with a secret basis for it. Here, we show how to generate ideal lattices (together with a secret basis) according to the average-case distribution used in our worst-case / average-case connection. Generating an ideal lattice according to this distribution is easy, but generating it together with a "good" secret basis is surprisingly difficult. Our solution to this problem is provided in Section 4.

Although the worst-case / average-case connection for BDDP over ideal lattices (Section 3) and the key generation algorithm (Section 4) are our main results, several other reductions are necessary to base our version of Gentry's somewhat homomorphic scheme on worst-case SIVP over ideal lattices. We summarize these reductions in Section 5.

2 Preliminaries

2.1 Ideal Lattices

By an *ideal lattice*, we mean an *ideal* in the *ring of integers* $R = \mathcal{O}_F$, where $f(x)$ is a monic, irreducible polynomial of degree n, and F is the field $\mathbb{Q}[x]/(f(x))$. A good example to keep in mind is $f(x) = x^n + 1$, where n is a power of 2. Then, the ring of integers is simply $\mathbb{Z}[x]/(f(x))$, integer polynomials modulo $f(x)$. In the full version, we address the general case $\mathbb{Z}[x]/(f(x)) \subseteq R \subseteq \mathcal{O}_F$.

Each element of R is associated to a coefficient vector in $\mathbb{Q}^n$ (in $\mathbb{Z}^n$ in our example). Since an ideal $I \subset R$ is additively closed, the coefficient vectors associated to elements of I form a *lattice*. The term "ideal lattice" emphasizes this object's dual nature as an algebraic ideal and a lattice.[2]

Ideals have additive structure as lattices, but they also have multiplicative structure. The *product* of two ideals I and J is $IJ = \{\sum \mathbf{v} \times \mathbf{w} : \mathbf{v} \in I, \mathbf{w} \in J\}$, where '$\times$' is ring multiplication. Let $F = \mathbb{Q}[x]/(f(x))$ be the field containing R.

[2] Alternative representations of an ideal lattice are possible – e.g., see [28,20].

The *inverse* of a ideal I is $I^{-1} = \{\mathbf{w} \in F : \forall \mathbf{v} \in I, \mathbf{v} \times \mathbf{w} \in R\}$. For example, the inverse of (2) is $(1/2) = \{\mathbf{r}/2 : \mathbf{r} \in R\}$. (The inverse of any *principal* ideal $(\mathbf{v})$ is given by $(\mathbf{v}^{-1})$, where the inverse $\mathbf{v}^{-1}$ is taken in F, but for a non-principal ideal the inverse is not always so simple.) We say that ideal I *divides* ideal J if $JI^{-1} \subset R$. I is a prime ideal if I dividing $A \cdot B$ implies I divides A or B. The ideal I^{-1} or JI^{-1} is sometimes called a fractional ideal, particularly when it is not a subset of R.

Ideals in R have many of the nice properties of integers, especially when R is the ring of integers. For example, in this case, ideals in R factor uniquely as a product of prime ideals. Also, all ideals in R are *invertible* – i.e., $I \cdot I^{-1} = R$. Furthermore, one can define the norm of a fractional ideal $\mathrm{Nm}(I)$ as the index $[R : I]$, and this map is multiplicative: $\mathrm{Nm}(IJ) = \mathrm{Nm}(I) \cdot \mathrm{Nm}(J)$.

Just as the prime number theorem states that the number of primes less than x is approximately $x/\ln x$, we have Landau's prime ideal theorem [15]:

Theorem 1 (Theorem 8.7.2 from [5]). *Let F be an algebraic number field of degree n. Let $\pi_F(x)$ denote the number of prime ideals in $\mathcal{O}_F$ whose norm is $\leq x$. Let $\lambda(x) = (\ln x)^{3/5}(\ln \ln x)^{-1/5}$. There is a $c > 0$ (depending on F) such that*

$$\pi_F(x) = x/\ln x + O(xe^{-c\lambda(x)})$$

With the Generalized Riemann Hypothesis, one can make a stronger statement.

Theorem 2 (Theorem 8.7.4 from [5]). *Assume GRH. Let F be an algebraic number field of degree n and discriminant Δ_F. For $x \geq 2$, we have*

$$|\pi_F(x) - x/\ln x| = O(\sqrt{x}(n \ln x + \ln |\Delta_F|))$$

The constant implied by the "O" symbol is absolute.

Regarding Theorem 2, Δ_F is upper-bounded by $\Delta(f)$, the discriminant of the polynomial f. Since $\Delta(f)$ is the determinant of the Sylvester matrix formed by $f(x)$ and its derivative $f'(x)$, it is upper bounded by $n^n \|f\|^{2n}$, where $\|f\|$ is the Euclidean length of the coefficient vector of $f(x)$ [33]. As in [10], we will always use $f(x)$ such that $\|f\| = \mathrm{poly}(n)$, which implies that $\ln |\Delta_F| = \mathrm{poly}(n)$.

We let γ_f denote the minimal value such that $\|\mathbf{u} \times \mathbf{v}\| \leq \gamma_f \cdot \|\mathbf{u}\| \cdot \|\mathbf{v}\|$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{Q}[x]/(f(x))$. For the values of irreducible $f(x)$ recommended in [10], we have $\gamma_f = \mathrm{poly}(n)$. A nice property of ideal lattices in such rings is that they are never too "oblong." In particular, trivially, $\lambda_n(I)/\lambda_1(I) \leq \gamma_f$, where $\lambda_k(I)$ is the k-th minimum of the ideal lattice I.

Again, a good choice for $f(x)$ is x^n+1, where n is a power of 2. This polynomial has the virtues of being irreducible, satisfying $R = \mathcal{O}_F = \mathbb{Z}[x]/(f(x))$, and having small values of $\Delta(f)$, $\|f\|$, and γ_f.

2.2 Gaussian Distributions and Other Preliminaries

For any real $s > 0$, define the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with parameter s as $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/s^2)$ for all $\mathbf{x} \in \mathbb{R}^n$. The associated *discrete* Gaussian distribution over lattice L is

$$\forall\ \mathbf{x} \in L, D_{L,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(L)}\ ,$$

where $\rho_{s,\mathbf{c}}(A)$ for set A denotes $\sum_{\mathbf{x}\in A} \rho_{s,\mathbf{c}}(\mathbf{x})$. In other words, the probability $D_{L,s,\mathbf{c}}(\mathbf{x})$ is simply proportional to $\rho_{s,\mathbf{c}}(\mathbf{x})$, the denominator being a normalization factor.

As in [24], for lattice L and real $\epsilon > 0$, we define the *smoothing parameter* $\eta_\epsilon(L)$ to be the smallest s such that $\rho_{1/s}(L^* \setminus \{\mathbf{0}\}) \leq \epsilon$. We say that s "exceeds the smoothing parameter" of L if $s \geq \eta_\epsilon(L)$ for negligible ϵ. In particular, this is true when $s \geq \lambda_n(L) \cdot \omega(\sqrt{\log n})$. Some useful lemmas are the following.

Lemma 1 (Lemma 4.4 of [24]). *For any n-dimensional lattice L, vector $\mathbf{c} \in \mathbb{R}^n$, and reals $0 < \epsilon < 1$, $s \geq \eta_\epsilon(L)$, we have*

$$\Pr_{\mathbf{x}\leftarrow D_{L,s,\mathbf{c}}}\{\|\mathbf{x} - \mathbf{c}\| > s\sqrt{n}\} \leq \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}$$

Lemma 2. *Let I, J be ideal lattices in R. Then for any $\epsilon \in (0, 1/2)$, and $s \geq \max\{\eta_\epsilon(I), \eta_\epsilon(J)\}$, and any $\mathbf{c} \in \mathbb{R}^n$, $\rho_{s,\mathbf{c}}(I)/\rho_{s,\mathbf{c}}(J)$ equals $\mathrm{Nm}(J)/\mathrm{Nm}(I)$, up to a multiplicative factor of between $(1+\epsilon)^2/(1-\epsilon)$ and its inverse.*

Proof. See full version.

We use $\mathbf{e}_i$ to refer to the vector $(0, \ldots, 0, 1, 0, \ldots, 0)$ with '1' in the ith position. We say that an equality $a \approx b$ holds "up to negligible error" if $a = (1 \pm \epsilon) \cdot b$ for some negligible ϵ.

3 Random Self-reduction of Ideal Lattice Problems

In this section, we present our worst-case / average-case "random self-reduction" for problems over ideal lattices, focusing on the bounded distance decoding problem (BDDP) [19,30]. We describe our average-case distribution, and specify our average-case and worst-case versions of BDDP. Then we show how to "randomize" worst-case ideal lattices into ideal lattices from our average-case distribution. In Section 4, we establish that the average-case distribution is suitable for KeyGen – i.e., we can efficiently (classically) sample an ideal lattice and a good basis for it according to this distribution.

3.1 Our Average-Case Distribution and Hard Problem

Our average-case distribution is simple: uniform over prime (non-fractional) ideals in R that have norms in some specified interval $[a, b]$.

Our average-case problem is really a "hybrid" of worst-case and average-case.

Definition 1 (Hybrid Bounded Distance Decoding Problem (HBDDP)). *Fix ring R and algorithm $\mathsf{IdealGen}$ that samples ideals in R, outputting the Hermite normal form basis of the sampled ideal lattice. Fix a positive real*

s_{HBDDP}. *The challenger sets* $\mathbf{B}_J \xleftarrow{R} \mathsf{IdealGen}(R)$. *The challenger sets* $\mathbf{x}$ *subject to the constraint that* $\|\mathbf{x}\| < s_{\mathsf{HBDDP}}$ *and sets* $\mathbf{t} \leftarrow \mathbf{x} \bmod \mathbf{B}_J$. *The problem is: given* $(\mathbf{B}_J, \mathbf{t})$ *(and the fixed values), output* $\mathbf{x}$.

The ideal lattice is generated according to an average-case distribution induced by an algorithm $\mathsf{IdealGen}$. However, the vector $\mathbf{t}$ is "worst-case", in that $\mathbf{t}$ is only required to be within a certain distance of the lattice; it need not be chosen according to any known (or even samplable) distribution.

The worst-case BDDP (WBDDP) is identical, except the ideal lattice is not necessarily chosen from an efficiently samplable distribution. For both of the BDDPs, we assume that the s parameter is chosen so that the solution is unique.

We base the security of our version of Gentry's scheme on HBDDP in the full version (and sketch this result in Section 5). As part of this result, we reduce HBDDP to a "purely" average-case BDDP where $\mathbf{t}$ is sampled according to a Gaussian distribution. In the full version, we also provide more reductions that (quantumly) reduce worst-case SIVP to WBDDP. We choose to focus on our techniques for randomizing the lattice since they are more interesting.

3.2 Statement of the Reduction

Our reduction is stated in the following theorem. It uses parameters that must satisfy certain $\mathsf{conditions}$ that we will specify momentarily.

Theorem 3. *Let* R *be the ring of integers for field* $F = \mathbb{Q}(x)/(f(x))$. *Let* M, N, s_{WBDDP}, t, a, *and* b *satisfy the* $\mathsf{conditions}$. *Suppose that there is an algorithm* $\mathcal{A}$ *that solves* s_{HBDDP}*-HBDDP with overwhelming probability (over the random coins chosen by* $\mathcal{A}$*) for a* ϵ *fraction of prime ideals* J *of* R *having norm in* $[a, b]$. *Then, there is an algorithm* $\mathcal{B}$, *which given access to a factoring oracle, solves with overwhelming probability the* s_{WBDDP}*-WBDDP for any (worst-case) ideal* M *of* R *with norm in* $[N, 2N]$ *when* $2t \cdot s_{\mathsf{WBDDP}} \leq s_{\mathsf{HBDDP}}$. *Regarding running times,* $time(\mathcal{B}) = time(\mathcal{A}) \cdot \mathrm{poly}(n)/\epsilon$.

The $\mathsf{conditions}$ are as follows (s refers to s_{WBDDP}):

- $\log N$ and $\log b$ are only polynomial in the lattice dimension n
- $s = \omega(\sqrt{\log n})$,
- $s = \gamma_f \cdot (b/N)^{1/n} \cdot \omega(\sqrt{\log n})$,
- $t \geq \gamma_f \cdot n^{1.5} \cdot s$,
- $|\mathcal{I}_{a,b}|/b$ is non-negligible, where $\mathcal{I}_{a,b}$ is the set of prime ideals with norm in $[a, b]$,
- a/b is non-negligible,
- $a^2 > 2N \cdot et_0^n$ where e is Euler's constant and $t_0 = t + s \cdot \sqrt{n}$.

Remark 1. Asymptotically, the requirement that $|\mathcal{I}_{a,b}|/b$ be non-negligible will be satisfied if $(b - a)/b$ is non-negligible. See Theorems 1 and 2.

To make the $\mathsf{conditions}$ more comprehensible, let us consider a concrete choice of parameters. Set $N = b = 2a$. Then, for any $g(n) = \omega(\sqrt{\log n})$, we can set

$s = \gamma_f \cdot g(n)$ and $t = {\gamma_f}^2 \cdot n^{1.5} \cdot g(n)$. The condition $a^2 = N^2/4 > 2N \cdot et_0^n$ is met when $N/8 > et_0^n \approx et^n = e \cdot {\gamma_f}^{2n} \cdot n^{1.5n} \cdot g(n)^n$. This is a very mild lower bound for N, considering that N is related to the norm of M. In particular, the condition $a^2 > 2N \cdot et_0^n$ can be met even when $\lambda_n(M)$ is small – e.g., polynomial in n.

A "deficiency" of the reduction is that, according to the conditions, the norm of the output average-case ideal is lower-bounded in terms of the norm of the worst-case ideal. It would be preferable to remove this constraint. In a reduction described in the full version, we show that ideals with "small" norms are the "hard case" when one is given access to a factoring oracle, and therefore our reductions ultimately apply even to average-case ideals with fairly small norms.

3.3 The RandomizeIdeal Algorithm

Toward proving Theorem 3, we present an algorithm RandomizeIdeal that, assuming the conditions are met, "randomizes" a worst-case lattice into our average-case distribution. In Section 3.4, we show that one can solve WBDDP by using RandomizeIdeal in combination with a HBDDP-solver.

RandomizeIdeal(R, M, N, s, t, a, b):

1. Outputs $\perp$ if the parameters do not satisfy the conditions.
2. Generates a vector $\mathbf{v}$ per the distribution $D_{M^{-1},s,t\cdot\mathbf{e}_1}$; sets $L \leftarrow M \cdot (\mathbf{v})$.
3. Uses a factoring oracle to compute lattice bases of the prime ideal divisors $\{\mathfrak{p}_i\}$ of L.
4. Sets J to be an ideal in $\{\mathfrak{p}_i\}$ with norm in $[a, b]$; if none exists, it aborts.
5. With probability $\mathrm{Nm}(J)/b$, outputs a basis $\mathbf{B}_J$ of J, along with the vector $\mathbf{v}$; otherwise, it aborts.

Regarding Step 2, one can sample from $D_{M^{-1},s,t\cdot\mathbf{e}_1}$ by using the GPV algorithm [12] with the independent set $\{\mathbf{e}_i\}$ in M^{-1}.

Regarding Step 3, let $R' = \mathbb{Z}[x]/(f(x))$ and consider the following theorem.

Theorem 4 (Kummer-Dedekind, as given in [33]). *Consider the factorization $f(x) = \prod_i g_i(x)^{e_i} \bmod p$ for prime integer p. The prime ideals $\mathfrak{p}_i \in \mathbb{Z}[x]/(f(x))$ of R' whose norms are powers of p are precisely*

$$\mathfrak{p}_i = (p, g_i(x))$$

There are polynomial time algorithms for factoring polynomials in $\mathbb{Z}_p[x]$ – e.g., by Kaltofen and Shoup [14]. Therefore, in R', if we have an integer factoring algorithm to factor $\mathrm{Nm}(L)$, we can efficiently discover all of the prime ideals that divide L. See [33] for details on how to extend this approach to rings $R \supset R'$. Note that since $R = \mathcal{O}_F$, the factorization in Step 3 is unique.

Regarding Step 4, there will be at most one ideal in $\{\mathfrak{p}_i\}$ with norm in $[a, b]$. If there were two such ideals $\mathfrak{p}_i, \mathfrak{p}_j$, the norm of their product would be at least $a^2 > 2N \cdot et_0^n$, where we will show the latter term exceeds the norm of L, a contradiction.

Before proving the reduction, we must establish that RandomizeIdeal outputs J according to our desired average-case distribution. We prove this in Lemma 6. Lemmas 3, 4 and 5 establish some preliminary facts.

Lemma 3. *Suppose the* conditions *are met. The probability that the ideal L has a divisor in $\mathcal{I}_{a,b}$ is non-negligible.*

Proof. See full version.

Lemma 4. *Suppose $\mathbf{v} = \mathbf{e}_1 + \mathbf{u}$ for $\|\mathbf{u}\| \leq 1/(2\gamma_f)$. Then, $e^{-2n\cdot\gamma_f\cdot\|\mathbf{u}\|} \leq \mathrm{Nm}((\mathbf{v})) \leq e^{n\cdot\gamma_f\cdot\|\mathbf{u}\|}$. In particular, when $\mathbf{v} \in t\cdot\mathbf{e}_1 + \mathcal{B}(s\sqrt{n})$, $\mathrm{Nm}((\mathbf{v})) \leq e\cdot t_0^n$.*

Proof. (Lemma 4) See full version.

Lemma 5. *Suppose the* conditions *are met.* RandomizeIdeal(R, M, N, s, t, a, b) *aborts with non-overwhelming probability.*

Proof. (Lemma 5) For Step 5, the probability of aborting is non-overwhelming, since a/b is non-negligible and $\mathrm{Nm}(J) \geq a$. Regarding Step 4, we use Lemma 3, which establishes that, for our choice of parameters, there is a non-negligible probability that $M\cdot(\mathbf{v})$ has a prime ideal divisor with norm in $[a, b]$ when $\mathbf{v}$ is sampled according to the above distribution. □

Lemma 6. *Suppose the* conditions *are met. Then,* RandomizeIdeal *samples J as a statistically uniform prime ideal (independent of M) subject to the constraint that $\mathrm{Nm}(J) \in [a, b]$.*

Proof. (Lemma 6) Consider the probability that a particular prime ideal J_0 with norm in $[a, b]$ is chosen as the ideal J in Step 4 in a single trial if there is no abort. (By Lemma 5, the probability of abort is non-overwhelming.) Assuming $\mathbf{v} \in t\cdot\mathbf{e}_1 + \mathcal{B}(s\cdot\sqrt{n})$ (which is indeed the case with overwhelming probability by Lemma 1), we claim that J_0 is chosen iff $\mathbf{v} \in J_0 M^{-1}$.

For the 'if' direction of our claim, if $\mathbf{v} \in J_0 M^{-1}$, then J_0 divides (is a super-lattice of) $L \leftarrow M\cdot(\mathbf{v})$. Since $\mathrm{Nm}((\mathbf{v})) \leq et_0^n$ when $\mathbf{v} \in t\cdot\mathbf{e}_1 + \mathcal{B}(s\cdot\sqrt{n})$ by Lemma 4, we have that $\mathrm{Nm}(L) = \mathrm{Nm}(M)\cdot\mathrm{Nm}((\mathbf{v})) \leq 2N\cdot et_0^n < a^2 \leq \mathrm{Nm}(J_0)^2$. Consequently, besides J_0, L cannot have any other prime ideal divisors with norm in $[a, b]$, and J_0 is chosen. For the 'only if' direction, that J_0 is chosen implies that J_0 divides (is a super-lattice of) $L = M\cdot(\mathbf{v})$. But then $J_0 M^{-1}$ is a super-lattice of $M^{-1}M\cdot(\mathbf{v}) = (\mathbf{v})$. Therefore, $(\mathbf{v})$ is contained in $J_0 M^{-1}$; in particular, $\mathbf{v} \in J_0 M^{-1}$.

Given our claim, for fixed M, the probability that J_0 is chosen in Step 4 is:

$$\Pr[J_0] \approx \frac{\sum_{\mathbf{v}\in J_0 M^{-1}} \Pr[\mathbf{v}]}{\sum_{\mathbf{v}\in M^{-1}} \Pr[\mathbf{v}]} = \frac{\rho_{s,t\cdot\mathbf{e}_1}(J_0 M^{-1})}{\rho_{s,t\cdot\mathbf{e}_1}(M^{-1})}$$

(The approximate equality holds up to negligible error, since it relies on $\mathbf{v} \in t\cdot\mathbf{e}_1 + \mathcal{B}(s\cdot\sqrt{n})$.)

We claim that s exceeds the smoothing parameters of J_0M^{-1} and M^{-1}. Assuming this claim, Lemma 2 implies that

$$\frac{\rho_{s,t\cdot\mathbf{e}_1}(J_0M^{-1})}{\rho_{s,t\cdot\mathbf{e}_1}(M^{-1})} \approx \mathrm{Nm}(M^{-1})/\mathrm{Nm}(J_0M^{-1}) = 1/\mathrm{Nm}(J_0)$$

up to negligible error. Step 5 uses rejection to adjust this probability from $1/\mathrm{Nm}(J_0)$ to $1/b$, making the distribution statistically uniform (and statistically independent of M) over all prime ideals with norms in $[a, b]$.

It remains to prove our claim that s exceeds the smoothing parameters of J_0M^{-1} and M^{-1}. This is clearly true for M^{-1}, which contains $\mathbb{Z}^n$ as a sublattice. Regarding J_0M^{-1}, we have

$$\begin{aligned} s &= \gamma_f \cdot (b/N)^{1/n} \cdot \omega(\sqrt{\log n}) \\ &\geq \gamma_f \cdot \mathrm{Nm}(J_0)^{1/n}/\mathrm{Nm}(M)^{1/n} \cdot \omega(\sqrt{\log n}) \\ &\geq \gamma_f \cdot \mathrm{Nm}(J_0)^{1/n} \cdot \mathrm{Nm}(M^{-1})^{1/n} \cdot \omega(\sqrt{\log n}) \\ &\geq \gamma_f \cdot \mathrm{Nm}(J_0M^{-1})^{1/n} \cdot \omega(\sqrt{\log n}) \\ &\geq \gamma_f \cdot \lambda_1(J_0M^{-1}) \cdot \omega(\sqrt{\log n}) \\ &\geq \lambda_n(J_0M^{-1}) \cdot \omega(\sqrt{\log n}) \end{aligned}$$

and the claim follows. □

3.4 Proof of the Reduction

Finally, we prove Theorem 3, showing how to use the procedure RandomizeIdeal to reduce WBDDP to HBDDP.

Intuitively, RandomizeIdeal samples a vector $\mathbf{v}$ that is "nearly parallel" to $\mathbf{e}_1$ (since $t \gg s$), so that multiplying the basis vectors in $\mathbf{B}_M$ by $\mathbf{v}$ is similar (from a geometric perspective) to multiplying by t. Thus, L is geometrically similar to a simple scaling of M, and it is easy to see how a solution to a lattice problem over L (e.g., to BDDP or SIVP) yields a solution to a lattice problem over M. As long as L is not an overly sparse subset of J – e.g., suppose that $(\mathrm{Nm}(L)/\mathrm{Nm}(J))^{1/n}$ is $\mathrm{poly}(n)$ – then $\lambda_1(J)$ will be only $\mathrm{poly}(n)$ less than $\lambda_1(L)$, and the BDDP solution to $(L, \mathbf{u}')$ will be the same as to $(J, \mathbf{u}')$ as long as $\mathbf{u}'$ is sufficiently close to L.

Proof. (Theorem 3) $\mathcal{B}$ wants to solve the WBDDP instance $(M, \mathbf{u})$. It does the following:

1. Runs $(\mathbf{B}_J, \mathbf{v}) \xleftarrow{\mathrm{R}} \mathsf{RandomizeIdeal}(R, M, N, s, t, a, b)$.
2. Sets $\mathbf{u}' \leftarrow (\mathbf{u} \times \mathbf{v}) \bmod \mathbf{B}_J$.
3. Runs $\mathcal{A}$ on the instance $(J, \mathbf{u}')$, receiving back a vector $\mathbf{y}$ such that $\mathbf{u}' - \mathbf{y} \in J$. (If $\mathcal{A}$ does not solve this instance, restart.)
4. Outputs $\mathbf{x} \leftarrow \mathbf{y}/\mathbf{v}$.

First, we verify that $(J, \mathbf{u}')$ is a valid HBDDP instance that should be solvable by $\mathcal{A}$. By Lemma 6, RandomizeIdeal outputs the basis of an ideal J that is statistically uniform among invertible prime ideals with norm in $[a, b]$.

Now let us check that $\mathbf{u}'$ is also valid. By assumption, there exist $\mathbf{m} \in M$ and $\mathbf{z}$ with $\|\mathbf{z}\| \leq s_{\mathsf{WBDDP}}$ such that $\mathbf{u} = \mathbf{m} + \mathbf{z}$. So, $\mathbf{u}' = \mathbf{m}' + \mathbf{z}'$, where $\mathbf{m}' \in M \cdot (\mathbf{v})$ and $\mathbf{z}' = \mathbf{z} \times \mathbf{v}$. Assuming $\mathbf{v} \in t \cdot \mathbf{e}_1 + \mathcal{B}(s \cdot \sqrt{n})$, which occurs with overwhelming probability, we have

$$\|\mathbf{z}'\| = \|\mathbf{z} \times \mathbf{v}\| \leq t \cdot \|\mathbf{z}\| + \gamma_f \cdot s \cdot \sqrt{n} \cdot \|\mathbf{z}\| \leq 2t \cdot s_{\mathsf{WBDDP}} \leq s_{\mathsf{HBDDP}}$$

Since $M \cdot (\mathbf{v})$ is a sub-lattice of J, we have that $\mathbf{u}' = \mathbf{j} + \mathbf{z}'$ for some $\mathbf{j} \in J$.

By the analysis above, $\mathcal{A}$ should solve the instance $(J, \mathbf{u}')$ with probability at least ϵ. If $\mathcal{A}$ solves this instance – i.e., $\mathcal{B}$ receives from $\mathcal{A}$ the unique vector $\mathbf{y}$ with $\|\mathbf{y}\| < s_{\mathsf{HBDDP}}$ such that $\mathbf{u}' - \mathbf{y} \in J$. It must be that $\mathbf{y} = \mathbf{z}'$. Thus $\mathbf{x} = \mathbf{z}'/\mathbf{v} = \mathbf{z}$, and $\mathcal{B}$ solves its WBDDP instance.

The probability that RandomizeIdeal does not abort and $\mathcal{A}$ succeeds is at least $\epsilon/\text{poly}(n)$. These probabilities are independent over trials, and the claimed running time of $\mathcal{B}$ follows. □

4 KeyGen According to the Average-Case Distribution

4.1 Our Approach at a High Level

For KeyGen, we want an algorithm IdealGen that generates a random ideal J together with a short vector in $\mathbf{w} \in J^{-1}$ to be used as the secret key. Recall how decryption works in Gentry's somewhat homomorphic scheme, and suppose that $R = \mathbb{Z}[x]/(f(x))$ in this subsection for simplicity. A ciphertext is an integer vector of the form $\mathbf{c} = \mathbf{j} + \mathbf{e}$, where $\mathbf{j} \in J$ and $\mathbf{e}$ is a short noise vector containing the message. Decryption involves computing the fractional part $[\mathbf{w} \times \mathbf{c}]$, which equals $[\mathbf{w} \times \mathbf{e}]$ since $\mathbf{w} \times \mathbf{j}$ is in R and thus an integer vector. If $\mathbf{w}$ and $\mathbf{e}$ are short enough – in particular, if we have the guarantee that all of the coefficients of $\mathbf{w} \times \mathbf{e}$ have magnitude less than $1/2$ – then $[\mathbf{w} \times \mathbf{e}]$ equals $\mathbf{w} \times \mathbf{e}$ exactly. From $\mathbf{w} \times \mathbf{e}$, the decrypter can recover $\mathbf{e}$ and the message.

How short should $\mathbf{w}$ be? Since $\lambda_n(J^{-1})$ is at least $\text{Nm}(J)^{-1/n}$, we cannot expect $\mathbf{w}$ to be much shorter than this. (Recall that we choose R such that $\lambda_n(I)/\lambda_1(I)$ is polynomial in n.) So, we will consider $\mathbf{w}$ to be a "good" secret key with respect to ideal J if $\|\mathbf{w}\| \leq g(n) \cdot \text{Nm}(J)^{-1/n}$ for some small polynomial $g(n)$. Now, how do we generate a random ideal J together with a "good" $\mathbf{w} \in J^{-1}$?

Our first step is to generate a "small" random ideal K – "small" in the sense that its norm is in $[n^{cn}, 2n^{cn}]$ for some small constant c, which guarantees that $\lambda_n(K) = \text{poly}(n)$. Since the norm of K is so small, $\mathbf{e}_1 \in K^{-1}$ is trivially a good secret key for K according to our definition. K is not useful as the ideal in Gentry's scheme, since even very small errors $\mathbf{e}$ make ciphertexts indecipherable.

But suppose, as a thought experiment, that we simply set $J = K \cdot (\mathbf{v})$ where $\mathbf{v} = T \cdot \mathbf{e}_1$ for some large integer T. That is, J is simply a scaling of K. Then, $\mathbf{w} \leftarrow \mathbf{e}_1/T$ is a vector in J^{-1} that satisfies our definition of a good secret key. And J is "large" enough to handle larger error vectors.

However, the simple scaling approach is obviously unsatisfactory for a few reasons. First, it does not generate J according to our desired average-case distribution. Also, it may not even be secure: all of the coefficients of J's vectors are divisible by T, and thus a ciphertext $\mathbf{c}$ leaks the value of $\mathbf{e} \bmod T$. Obviously, we want to avoid these deficiencies.

Instead, as our second step, we sample $\mathbf{v} \leftarrow D_{K^{-1},S,T\cdot\mathbf{e}_1}$ where $T/S = \text{poly}(n)$. Then, as before, we set $J = K \cdot (\mathbf{v})$, and $\mathbf{w} \leftarrow \mathbf{e}_1/\mathbf{v}$. That is, we do the same thing as in the simple scaling approach, except that we sample $\mathbf{v}$ from K^{-1} rather than from R, and we choose it to be very close to $T \cdot \mathbf{e}_1$ rather than being exactly equal. It turns out that, if $\mathbf{v}$ is very close to $T \cdot \mathbf{e}_1$, then $1/\mathbf{v}$ is very close to $\mathbf{e}_1/T$. In particular, $\mathbf{w}$ will be a good secret key for J. Fortunately, this approach avoids the deficiencies of simple scaling. We can prove that, by including a couple of rejection steps – to output J only if it is prime, to fine-tune the output distribution, etc. – the J sampled using this approach has the correct average-case distribution.

Intuitively, why does this approach induce a random distribution on J? At a very high level, we can ask: is J random *geometrically* (e.g., when one considers the "shape" of the parallelepiped formed by J's shortest independent set), and is J random *algebraically* (e.g., when one considers J's norm)? Geometrically, J inherits K's shape, since (up to some perturbation in the sampling of $\mathbf{v}$) it is a simple scaling of K. We choose K from a large enough space so that its shape, and hence J's shape, is quite "random". Algebraically, the fact that $\mathbf{v}$ is sampled from K^{-1} "randomizes" J algebraically – in particular, J is not divisible by K. But these are only intuitions. Before providing a more precise explanation, we need to describe our IdealGen algorithm in more detail.

4.2 IdealGen: The Details

IdealGen uses parameters $s = \omega(\sqrt{\log n})$, t such that $t \geq 42 \cdot \gamma_f \cdot s \cdot n^{1.5}$ and $t > 8 \cdot \gamma_f \cdot s \cdot n^{1.5} \cdot \|f\|^2$, and $\alpha \geq 1$; let $S = s \cdot \alpha$ and $T = t \cdot \alpha$. It invokes an algorithm $\mathsf{TempIdeal}(R, i, j)$, described in Section 4.3, that outputs a uniformly random ideal K with norm in $[i, j]$ (but not a nontrivial "good" key for K). IdealGen ultimately outputs a uniformly random prime ideal J with norm in $[2, 3] \cdot t^{2n} T^n$.

IdealGen:

1. Runs $\mathbf{B}_K \xleftarrow{R} \mathsf{TempIdeal}(R, t^{2n}, 4t^{2n})$.
2. Samples $\mathbf{v} \xleftarrow{R} D_{K^{-1},S,T\cdot\mathbf{e}_1}$ and sets $\mathbf{w} \leftarrow 1/\mathbf{v}$; aborts if $\mathbf{v} \notin T\cdot\mathbf{e}_1 + \mathcal{B}(2S\sqrt{n})$.
3. Sets $J \leftarrow K \cdot (\mathbf{v})$; aborts if J is not prime or $\text{Nm}(J) \notin [2, 3] \cdot t^{2n} T^n$.
4. Continues to Step 5 with probability $\text{Nm}(K)/4t^{2n}$; otherwise, aborts.
5. Continues to Step 6 with probability $\beta \cdot \frac{\rho_{S/T^2,(1/T)\cdot\mathbf{e}_1}(\mathbf{w})}{\rho_{S,T\cdot\mathbf{e}_1}(\mathbf{v})}$, where β will be defined later; otherwise, aborts.
6. With probability $2t^{2n}T^n/\text{Nm}(J)$, outputs $\mathbf{w}$ and the Hermite normal form of J; otherwise, aborts.

Remark 2. IdealGen is precisely what we outlined above, aside from the probability of aborting in Steps 2-6. We will show that the probability of aborting is non-overwhelming, and that these steps fine-tune the distribution so that J is a uniformly random prime ideal with norm in the prescribed interval. The algorithm can be re-run until it completes successfully.

Remark 3. In Step 2, one can sample from the distribution $D_{K^{-1},S,T\cdot\mathbf{e}_1}$ by using the GPV algorithm [12] with the independent set $\{\mathbf{e}_i\}$ in K^{-1}.

Remark 4. By Lemma 1, the vector $\mathbf{v}$ is in $T\cdot\mathbf{e}_1+\mathcal{B}(S\sqrt{n})$ with overwhelming probability. Note that we only abort in Step 2 if $\mathbf{v}\notin T\cdot\mathbf{e}_1+\mathcal{B}(2S\sqrt{n})$. We use a ball of radius $2S\sqrt{n}$ instead of $S\sqrt{n}$ in Step 2 for technical reasons – specifically, Corollary 2 below and its use in the proof of Theorem 7.

Remark 5. Regarding Step 5, we must ensure that the "probability" is a number in $[0,1]$. We show that $\rho_{S/T^2,(1/T)\cdot\mathbf{e}_1}(\mathbf{w})/\rho_{S,T\cdot\mathbf{e}_1}(\mathbf{v})\in[e^{-6\pi\sqrt{1/n}},e^{6\pi\sqrt{1/n}}]$. (See Lemma 10.) Therefore, we can take $\beta\leftarrow e^{-6\pi\sqrt{1/n}}$.

To begin analyzing our IdealGen algorithm, we state some useful lemmas about the vector $\mathbf{v}$ sampled in Step 2. Omitted proofs can be found in the full version. The theme of these lemmas is that since $\mathbf{v}$ is very close to $T\cdot\mathbf{e}_1$, it behaves in many respects like $T\cdot\mathbf{e}_1$.

Lemma 7. *If* $\mathbf{v}\in T\cdot\mathbf{e}_1+\mathcal{B}(2S\sqrt{n})$*, then* $\mathrm{Nm}((\mathbf{v}))\in[T^n/1.1,1.1\cdot T^n]$.

Lemma 8. *If* $\mathbf{v}\in T\cdot\mathbf{e}_1+\mathcal{B}(2S\sqrt{n})$*, then it is the only vector in* $(\mathbf{v})$ *inside that ball.*

Lemma 9. *If* $\|\mathbf{u}\|<1/\gamma_f$*, then*

$$\mathbf{e}_1/(\mathbf{e}_1-\mathbf{u})=\mathbf{e}_1+\mathbf{u}+\mathbf{x}\quad\text{for}\quad\|\mathbf{x}\|\le\frac{\gamma_f\cdot\|\mathbf{u}\|^2}{1-\gamma_f\cdot\|\mathbf{u}\|}$$

Corollary 1. *If* $\mathbf{v}\in T\cdot\mathbf{e}_1+\mathcal{B}(2S\sqrt{n})$*, then* $\mathbf{w}\in\mathbf{e}_1/T+\mathcal{B}(4S\sqrt{n}/T^2)$.

Corollary 2. *If* $\mathbf{w}\in\mathbf{e}_1/T+\mathcal{B}(S\sqrt{n}/T^2)$*, then* $\mathbf{v}\in T\cdot\mathbf{e}_1+\mathcal{B}(2S\sqrt{n})$.

Lemma 10. *If* $\mathbf{v}\in T\cdot\mathbf{e}_1+\mathcal{B}(2S\sqrt{n})$*, then*

$$\rho_{S,T\cdot\mathbf{e}_1}(\mathbf{v})/\rho_{S/T^2,(1/T)\cdot\mathbf{e}_1}(\mathbf{w})\in[e^{-6\pi\sqrt{1/n}},e^{6\pi\sqrt{1/n}}]$$

Our main results about IdealGen are captured in Theorems 5, 6, and 7 – namely, that it outputs a good secret key for J, it does not abort very often (and therefore can be efficiently re-run until it outputs a result), and it outputs J according to the desired average-case distribution.

Theorem 5. *The vector* $\mathbf{w}$ *output by* IdealGen *is a "good" key for* J*. Specifically,* $\|\mathbf{w}\|<6t^2\cdot\mathrm{Nm}(J)^{-1/n}$.

Proof. (Theorem 5) By Corollary 1, $\mathbf{w} \in \mathbf{e}_1/T + \mathcal{B}(4S\sqrt{n}/T^2)$. So, clearly, $\|\mathbf{w}\| < 2/T$. On the other hand, $\mathrm{Nm}(J)^{-1/n} \geq 1/(3^{1/n}t^2T)$. The result follows. □

Theorem 6. *The probability of aborting in Steps 2-6 is non-overwhelming.*

Proof. (Theorem 6) For Steps 4 and 6, the claim is clearly true. For Step 2, it follows from Lemma 1.

For Step 5, we invoke Lemma 10, which implies we can set $\beta \leftarrow e^{-6\pi\sqrt{1/n}}$, and the algorithm will continue to Step 6 with at least (non-negligible) probability $e^{-12\pi\sqrt{1/n}}$.

For Step 3, an abort occurs if J is not prime or $\mathrm{Nm}(J) \notin [2,3] \cdot t^{2n}T^n$. Asymptotically, Theorems 1 and 2 imply that, for an interval $[cx, x]$ with constant $c < 1$, prime ideals are a $O(1/\log x)$ fraction of ideals. Given that $\mathrm{Nm}(J) = \mathrm{Nm}(K) \cdot \mathrm{Nm}((\mathbf{v}))$ and $\mathrm{Nm}((\mathbf{v})) \in [T^n/1.1, 1.1 \cdot T^n]$ (by Lemma 7), $\mathrm{Nm}(J)$ falls outside the interval only if $\mathrm{Nm}(K)$ falls outside of $[2 \cdot 1.1, 3/1.1] \cdot t^{2n}$. By the distribution of ideals (see Theorems 1 and 2) and the claimed distribution of $\mathsf{TempIdeal}$, this occurs only with only constant probability, in which case the probability of aborting in Step 2 is a constant. □

Before getting to the last theorem, we state one more lemma.

Lemma 11. *Let J be an ideal such that $\mathrm{Nm}(J) \in [2,3] \cdot t^{2n}T^n$. Then S/T^2 exceeds the smoothing parameter of J^{-1}.*

Proof. (Lemma 11) We have

$$\frac{S}{T^2} = \frac{s}{tT} \geq \frac{s \cdot \gamma_f}{2^{1/n}t^2T} \geq \frac{s \cdot \gamma_f}{\mathrm{Nm}(J)^{1/n}} \geq s \cdot \gamma_f \cdot \lambda_1(J^{-1}) \geq s \cdot \lambda_n(J^{-1}) \ .$$

Since $s = \omega(\sqrt{\log n})$, the result follows. □

Theorem 7. *For any $\alpha \geq 1$, $\mathsf{IdealGen}$ with parameter α efficiently outputs a prime ideal J that is statistically uniform subject to the constraint that $\mathrm{Nm}(J) \in [2,3] \cdot t^{3n}\alpha^n$.*

Proof. (Theorem 7) Let $\mathcal{K}$ be the sets of ideals with norms in $[1,4] \cdot t^{2n}$, and let $\mathcal{J}$ be the sets of prime ideals with norms in $[2,3] \cdot t^{2n}T^n$. For convenience, we define some sets of ideals associated to $J \in \mathcal{J}$. Let

$$\begin{aligned}
\mathcal{S}_J &= \{K \in \mathcal{K} : \exists \mathbf{v} \text{ s.t. } J = K \cdot (\mathbf{v}) \text{ and } \mathbf{v} \in T \cdot \mathbf{e}_1 + \mathcal{B}(2S\sqrt{n})\} \\
\mathcal{V}_J &= \{\mathbf{w} : J \cdot (\mathbf{w}) \in \mathcal{K} \text{ and } 1/\mathbf{w} \in T \cdot \mathbf{e}_1 + \mathcal{B}(2S\sqrt{n})\} \\
\mathcal{W}_J &= \{\mathbf{w} : J \cdot (\mathbf{w}) \in \mathcal{K} \text{ and } \mathbf{w} \in (1/T) \cdot \mathbf{e}_1 + \mathcal{B}(S\sqrt{n}/T^2)\}
\end{aligned}$$

Define $\mathcal{S}'_J$ identically to $\mathcal{S}_J$, except they include only those K for which there is *exactly* one such $\mathbf{v}$. Lemma 8 implies that $\mathcal{S}_J = \mathcal{S}'_J$.

Consider the probability $\Pr[J_0]$ that a particular ideal J_0 is chosen as J in Step 3. We have

$$\Pr[J_0] = \sum_{K \in \mathcal{S}_{J_0}} \Pr[J_0 \wedge K] = c_1 \cdot \sum_{K \in \mathcal{S}_{J_0}} \Pr[J_0|K] = c_1 \cdot \sum_{K \in \mathcal{S}'_{J_0}} \Pr[J_0|K],$$

for some universal constant c_1, where the second inequality follows from the fact that K is chosen uniformly by TempIdeal.

For a particular candidate pair (K_0, J_0) with $K_0 \in S'_{J_0}$, let $\mathbf{v}_0$ be the unique vector in $J_0 K_0^{-1} \cap (T \cdot \mathbf{e}_1 + \mathcal{B}(2S\sqrt{n}))$. We claim that, at Step 3,

$$\Pr[J_0|K_0] = \rho_{S,T\cdot\mathbf{e}_1}(\mathbf{v}_0)/\rho_{S,T\cdot\mathbf{e}_1}(K_0^{-1})$$

This follows because the latter quantity is $\Pr[\mathbf{v}_0|K_0]$, and from the fact that J_0 and $\mathbf{v}_0$ determine each other once K_0 is fixed.

Now, consider the denominator $\rho_{S,T\cdot\mathbf{e}_1}(K_0^{-1})$; we claim that, for fixed (S,T), this sum is proportional to $\mathrm{Nm}(K_0)$, up to negligible error. This follows from Lemma 2, and the fact that S exceeds the smoothing parameter of K_0^{-1} (since $\mathbb{Z}^n$ is a sub-lattice of K_0^{-1}). So, after Step 3, we have

$$\Pr[J_0|K_0] = c_2 \cdot \rho_{S,T\cdot\mathbf{e}_1}(\mathbf{v}_0)/\mathrm{Nm}(K_0)$$

up to negligible error for some universal constant c_2. After Steps 4 and 5, we have

$$\Pr[J_0|K_0] = c_3 \cdot \rho_{S/T^2,(1/T)\cdot\mathbf{e}_1}(\mathbf{w}_0)$$

up to negligible error for some universal constant c_3, where $\mathbf{w}_0 = 1/\mathbf{v}_0$ and thus

$$\Pr[J_0] = c_4 \cdot \sum_{K_0 \in \mathcal{S}'_{J_0}} \rho_{S/T^2,(1/T)\cdot\mathbf{e}_1}(\mathbf{w}_0)$$

We claim that

$$\sum_{K_0 \in \mathcal{S}'_{J_0}} \rho_{\frac{S}{T^2},\frac{\mathbf{e}_1}{T}}(\mathbf{w}_0) = \sum_{\mathbf{w}_0 \in \mathcal{V}_{J_0}} \rho_{\frac{S}{T^2},\frac{\mathbf{e}_1}{T}}(\mathbf{w}_0) = \rho_{\frac{S}{T^2},\frac{\mathbf{e}_1}{T}}(J_0^{-1}) = c_5 \cdot \mathrm{Nm}(J_0) \quad (1)$$

up to negligible error for some universal constant c_5. This claim lets us complete the proof. The abort in Step 6 adjusts this probability so that it becomes $c_5 \cdot 2t^{2n}T^n$, independent of J_0, and thus makes $\Pr[J_0]$ statistically uniform across all $J_0 \in \mathcal{J}$.

In Equation 1, the second sum is just a syntactic rewriting of the first sum.

To prove the second equality in Equation 1, first note that $\mathcal{W}_{J_0} \subset \mathcal{V}_{J_0} \subset J_0^{-1}$. The first inclusion follows from the fact that, by Lemma 2, for every $\mathbf{w}_0 \in (1/T) \cdot \mathbf{e}_1 + \mathcal{B}(S\sqrt{n}/T^2)$, it is the case that $1/\mathbf{w}_0 \in T \cdot \mathbf{e}_1 + \mathcal{B}(2S\sqrt{n})$. The second inclusion follow from the fact that each $\mathbf{w}_0$ satisfies $(\mathbf{w}_0) = J^{-1}K$ for some $K \in \mathcal{K}$; in particular, $\mathbf{w}_0 \in J_0^{-1}$. Now, we claim that

$$\sum_{\mathbf{w}_0 \in \mathcal{W}_{J_0}} \rho_{\frac{S}{T^2},\frac{\mathbf{e}_1}{T}}(\mathbf{w}_0) = \rho_{\frac{S}{T^2},\frac{\mathbf{e}_1}{T}}(J_0^{-1})$$

up to negligible error, which would establish the second equality (up to negligible error). This equality holds because $\mathcal{W}_{J_0}$ contains all of the $\mathbf{w}_0$'s in J_0^{-1} that

contribute substantially to the sum. Specifically, since S/T^2 exceeds the smoothing parameter of J_0^{-1} (by Lemma 11), the sum $\rho_{\frac{S}{T^2},\frac{\mathbf{e}_1}{T}}(J_0^{-1})$ is only negligibly affected when restricted to the set $J_0^{-1} \cap ((1/T)\cdot\mathbf{e}_1 + \mathcal{B}(S\sqrt{n}/T^2))$ (Lemma 1). However, this set is contained in $\mathcal{W}_J$, since if we set $K_0 \leftarrow J_0 \cdot (\mathbf{w}_0)$, then K_0 is indeed in $\mathcal{K}$, since $\mathrm{Nm}(K_0) = \mathrm{Nm}(J_0)\cdot\mathrm{Nm}((\mathbf{w}_0))$, which is in the interval $[2/1.1, 3\cdot 1.1]\cdot t^{2n} \subset [1,4]\cdot t^{2n}$.

The third equality in Equation 1 follows from Lemma 11 and Lemma 2. □

One aspect of the proof may seem a bit mysterious. Why did we use Step 5 to convert $\Pr[J_0]$ from a sum of $\rho(\mathbf{v})$'s to a sum of $\rho(1/\mathbf{v})$'s? Note that $\mathbf{v} \in J_0K^{-1}$ for some K, and $\mathbf{w} = 1/\mathbf{v} \in J_0^{-1}K$. Summing over $\rho(\mathbf{w})$'s is more natural, since all of the points are in a single ideal – namely, J_0^{-1}. In contrast, summing over vectors in J_0K^{-1} for different K's is not a sum we know how to evaluate.

4.3 The TempIdeal Algorithm

Here, we construct an efficient algorithm $\mathsf{TempIdeal}(R, i, j)$ that outputs a uniformly random ideal $K \subset R$ with norm in $[i, j]$. TempIdeal only needs to output *some* basis of K, not necessarily a "good" basis. Let us begin at a high level by considering some possible approaches.

Suppose we sample random $\mathbf{v}$ from R, and set $K \leftarrow (\mathbf{v})$, re-sampling if $\mathrm{Nm}(K) \notin [i, j]$. Then, K is a principal ideal, and unfortunately the probability that a "random" ideal from R is principal is typically negligible in n. (More accurately, the field $F = \mathbb{Q}(x)/(f(x))$ has an associated *class group*, where each member of the group consists of an equivalence class of ideals. The set of principal ideals is only one class, whereas the class group size is typically exponential in n.) Clearly, this approach does not sample a "random" ideal.

A more promising approach is to use Kummer-Dedekind (Theorem 4), which *can* actually be used to sample a uniformly random *prime* ideal, as follows. Sample a uniform prime power $p^e \in [i, j]$, and use Kaltofen and Shoup [14] to (efficiently) compute the factorization $f(x) = \prod_i g_i(x)^{e_i} \bmod p$. Kummer-Dedekind tells us that all prime ideals of $\mathbb{Z}[x]/(f(x))$ having norm p^e are of the form $(p, g_i(x))$, where $g_i(x)$ is an irreducible degree-e factor of $f(x)$ modulo p. There can be at most n ideals of norm p^e. If there are $r \le n$ such factors $g_i(x)$, restart with probability $1 - r/n$. Otherwise, sample one of these $g_i(x)$'s uniformly and output $K \leftarrow (p, g_i(x))$. (It it is straightforward to extend this method recover all prime ideals with norm p^e in rings $\mathbb{Z}[x]/(f(x)) \subset R \subseteq \mathcal{O}_F$ [33].) This works, but unfortunately we require TempIdeal to sample K from all ideals with norm in $[i, j]$, not just from prime ideals.

Consider the following modification to the above approach: sample a uniform (possibly composite) integer $N \in [i, j]$, and compute the factorization $f(x) = \prod_i g_i(x)^{e_i} \bmod N$, etc. But computing this factorization is hard in general when N is composite. In fact, we do not see a way to generate a random ideal K without knowing the factorization of its norm.

These considerations lead us to construct an algorithm for generating a random *factored* ideal whose norm is in the prescribed interval, even though, in

principle, we do not need the factorization. For this task, a good place to start is to look at existing algorithms for generating a random factored integer – especially Kalai's elegantly simple algorithm [13].

Kalai's Algorithm for Generating a Random Factored Number:

Input: Integer $b > 0$.

Output: A uniformly random number $1 \leq N \leq b$, with its factorization.

1. Generate a sequence $b \geq s_1 > s_2 > \cdots > s_\ell = 1$ by uniformly choosing $s_{i+1} \in \{1, \ldots, s_i - 1\}$. (Use b as s_0.) Put all prime s_i's in a list L.
2. For each $s_i \in L$, put s_i into L at least k additional times with probability $1/s_i^k$.
3. Let N be the product of the numbers in L (with repetition).
4. If $N > b$, restart.
5. Output N and the prime s_i's with probability N/b; otherwise, restart.

Remark 6. Kalai presents his algorithm somewhat differently.

As Kalai highlights, the reason this algorithm works is because a prime $p \leq b$ is in the sequence independently with probability exactly $1/p$, since it occurs iff it is chosen before any number in $\{1, \ldots, p-1\}$. That is, we could replace the first step of Kalai's algorithm with this alternative step without affecting the output distribution:

1. For each prime number $s_i \in [1, b]$, put s_i in a list L with probability $1/s_i$.

Of course, the algorithm with this alternative step is grossly inefficient; Kalai's main insight is a way to obtain the same output efficiently. After this insight, the remainder of the analysis is relatively straightforward. The prime p appears at least e times in L independently with probability $1/p^e$ through Step 2, and thus the probability that a b-smooth number N is selected in Step 3 is proportional to $1/N$. The final two rejection steps ensure uniformity across numbers in $[1, b]$. By Mertens' theorem, the algorithm will *not* restart in Step 4 with probability $\theta(1/\log b)$. See Kalai's one page paper for more details.

Our TempIdeal algorithm is a modification of Kalai's algorithm that accounts for the fact that there could be up to n prime ideals that are "tied" with the same norm. To each integer s, we associate n ideals $\{I_{s,j}\}$. Specifically, if there are $r \leq n$ distinct prime ideals of norm s, we let $I_{s,1}, \ldots, I_{s,r}$ be these ideals, and set $I_{s,r+1} = \cdots = I_{s,n} = 1$.

TempIdeal(R, a, b):

1. Generate a sequence $b \geq s_1 > s_2 > \cdots > s_\ell = 1$ by uniformly choosing $s_{i+1,j} \in \{1, \ldots, s_i - 1\}$ for all $j \in \{1, \ldots, n\}$ and setting $s_{i+1} \leftarrow \max_j \{s_{i+1,j}\}$. (Use b as s_0.) Put each s_i that is a norm of a prime ideal in a list L.
2. For each $s_i \in L$, do the following. First, generate $j \in [1, n]$ uniformly and put the ideal $I_{s_i,j}$ into multiset M. Then, for each j, insert at least k additional instances of $I_{s_i,j}$ into M with probability $1/s_i^k$.

3. Remove those ideals in M that are equal to 1.
4. Let K be the product of the ideals remaining in M (with repetition).
5. If $\mathrm{Nm}(K) \notin [a, b]$, restart.
6. Output a basis for K with probability $\mathrm{Nm}(K)/b$; otherwise, restart.

Remark 7. Obviously, in Step 2, we could have avoided putting any ideals that equal 1 in to M in the first place, since we remove them in Step 3. But we leave this in, since it will make the analysis a bit simpler.

Theorem 8. TempIdeal *uniformly samples an ideal $K \subset R$ with norm in $[a, b]$. The algorithm takes time $b/(a-b) \cdot \mathrm{poly}(n, \log b)$.*

To simplify the proof of Theorem 8, we define a "slow" version of the above algorithm – SlowTempIdeal – which is analogous to the "slow" version of Kalai's algorithm with the alternative first step.

SlowTempIdeal(R, a, b):

1. For each $s_i \in [1, b]$ that is the norm of a prime ideal, for each $j \in [1, n]$, put at least k instances of $I_{s_i,j}$ into multiset M' with probability $1/s_i^k$. If there is some ideal $I_{s_i,j}$ in M', put s_i into L.
2. Run Steps 2-6 of TempIdeal(R, a, b).

Now, Theorem 8 follows from Lemmas 12, 14, and 15.

Lemma 12. *The distribution of L is the same in* TempIdeal *and* SlowTempIdeal*, and hence the two algorithms have the same output distribution.*

Proof. (Lemma 12) Consider the probability that a fixed s is in L. For TempIdeal, this equals the probability that s is in the sequence. If $s_i > s$, the probability that $s_{i+1} \in [1, s]$ is $s^n/(s_i - 1)^n$, whereas the probability that $s_{i+1} \in [1, s-1]$ is $(s-1)^n/(s_i-1)^n$. Thus, when sampling s_i, the probability that s_{i+1} is in $[1, s-1]$ given that it is in $[1, s]$ is $(s-1)^n/s^n$. Consequently, since s_{i+1} must eventually be in $[1, s]$ for some i, the probability that s is in the sequence is $1 - (s-1)^n/s^n$. This probability is independent of whether or not other values s' are in L. For SlowTempIdeal, the probability that none of the n ideals $I_{s,j}$ is in M' is $(s-1)^n/s^n$. So, the probability that some ideal $I_{s,j}$ is in M', and hence $s \in L$, is the same as in TempIdeal: $1 - (s-1)^n/s^n$. □

Lemma 13. *Through Step 4 of* SlowTempIdeal*, the probability that a fixed ideal K_0 with prime ideal factors in $[1, b]$ is selected is*

$$\frac{1}{\mathrm{Nm}(K_0)} \cdot \prod_{\mathrm{Nm}(\mathfrak{p}) \le b} \frac{\mathrm{Nm}(\mathfrak{p}) - 1}{\mathrm{Nm}(\mathfrak{p})}$$

where the product is over prime ideals.

Proof. (Lemma 13) It is clear that the multisets M and M' have exactly the same distribution conditioned on the list L. That is, if $s_i \notin L$, neither multiset contains an ideal $I_{s_i,j}$. If $s_i \in L$, then both M and M' contain a random non-empty multiset S with elements from $\{I_{s_i,1}, \ldots, I_{s_i,n}\}$, where $\Pr[S]$ is proportional to $1/s_i^{|S|}$. Therefore, we could have used M' instead of M beginning in Step 3 of SlowTempIdeal without affecting the output distribution.

Remove the primes that equal 1 from M'. A (nontrivial) ideal $I_{s_i,j}$ is in M' at least k times independently with probability $1/s_i^k = 1/\mathrm{Nm}(I_{s_i,j})^k$, and therefore *exactly* k times independently with probability $(\mathrm{Nm}(I_{s_i,j})-1)/\mathrm{Nm}(I_{s_i,j})^{k+1}$. By the independence of these probabilities, and by multiplicativity of the norm map over ideals, the result follows. □

Lemma 14. SlowTempIdeal *uniformly samples an ideal $K \subset R$ with norm in $[a, b]$.*

Proof. (Lemma 14) Given Lemma 13 – i.e., the fact that through Step 4 the probability that some K_0 is chosen equals $1/\mathrm{Nm}(K_0)$ times some universal constant that is independent of K_0 – it is clear that the final two rejection sampling steps ensure that K is uniform among ideals with norm in $[a, b]$. □

Lemma 15. TempIdeal *takes time $b/(a-b) \cdot \mathrm{poly}(n, \log b)$.*

Proof. (Lemma 15) Let us consider the probability that a restart occurs.

Regarding Step 5, by Merten's theorem for number fields, we have

$$\prod_{\mathrm{Nm}(\mathfrak{p}) \leq b} (1 - 1/\mathrm{Nm}(\mathfrak{p})) = \frac{e^{-\gamma}}{a_K} \frac{1}{\log b} + O(\frac{1}{\log^2 b})$$

where a_K is the residue of $\zeta_K(s)$, the Dedekind zeta-function, at $s = 1$, and γ denotes Euler's constant 0.577.... Denote the above term by α. By Lemma 13, the probability that some K with norm at most b is selected in Step 4 is

$$\alpha \cdot \sum_{\mathrm{Nm}(K) \leq b} 1/\mathrm{Nm}(K)$$

There are $\theta(b)$ ideals of norm at most b (this follows from Theorems 1 and 2), and thus the above sum is $\Omega(1/\log(b))$.

Regarding Step 6, among K's with norm at most b, approximately a $(b-a)/b$ fraction of them have norm at least a. (Again this follows from Theorems 1 and 2.) The result follows. □

5 Basing Gentry's Somewhat Homomorphic Scheme on SIVP over Ideal Lattices

We showed how to reduce WBDDP to HBDDP for our average-case distribution. It remains to base our variant of Gentry's scheme on HBDDP, and to reduce SIVP to WBDDP. We sketch these results here. Details are in the full version.

First, we specify some details of our variant. As in [10], the public key includes ideals I and J, and a short independent set $\mathbf{B}_I$ of I – e.g., where $\|\mathbf{B}_I\| = \text{poly}(n)$. J is output by our new $\mathsf{IdealGen}$ algorithm. The cosets of I form the plaintext space. Regarding I, we have a new requirement: that $\text{Nm}(I)$ is prime and very small – i.e., $\text{poly}(n)$. To find such an I, we can either construct $f(x)$ to ensure that the associated ring of integers has an ideal of small prime norm, or we can apply Kummer-Dedekind (Theorem 4) to primes of size $\text{poly}(n)$. By Theorem 8.7.7. of [5], for appropriate values of $f(x)$ and assuming GRH, applying Kummer-Dedeking will eventually give us the basis a prime ideal $\mathfrak{p}$ in R having $\text{poly}(n)$-norm. From this basis, we can compute an independent set of $\mathfrak{p}$ of length at most $\text{Nm}(\mathfrak{p})$. We set $I \leftarrow \mathfrak{p}$ and $\mathbf{B}_I$ to be this independent set. We sample ciphertexts per a Gaussian distribution: $c \leftarrow c' \bmod \mathbf{B}_J$ where $c' \leftarrow D_{m+I,s,0}$ for some s.

To reduce HBDDP to the semantic security of this scheme, we first reduce HBDDP to a decision problem that we call the inner ideal membership problem (IIMP): (roughly) given $(\mathbf{B}_J, \mathbf{t})$ where $\mathbf{B}_J \xleftarrow{R} \mathsf{IdealGen}(R)$ and $\mathbf{t} \leftarrow \mathbf{x} \bmod \mathbf{B}_J$ for some $\mathbf{x} \in R$ with $\|\mathbf{x}\| < s_{\mathsf{IIMP}}$, decide whether or not $\mathbf{x} \in I$. Essentially, a HBBDP-solver can use a IIMP-solver to find out which coset of I that $\mathbf{x}$ is in. (For this search to be efficient, $\text{Nm}(I)$ must be $\text{poly}(n)$.) Using "Hensel lifting", the HBDDP-solver can recover $\mathbf{x}$ modulo I^k for large k – large enough that $\mathbf{x}$ becomes the shortest vector in $\mathbf{x}+I^k$ by such a large margin that is efficient to recover $\mathbf{x}$ using Babai's nearest plane algorithm. To reduce the IIMP to the semantic security of the scheme, we sample a uniform coset of I, set $\mathbf{u} \in R$ to be a short vector in that coset, and set the challenge ciphertext as follows: $c^* \leftarrow c' \bmod \mathbf{B}_J$ where $c' \leftarrow m_b + \mathbf{t} \times \mathbf{u} + D_{I,s,0}$. When $\mathbf{x} \in I$, $c' \in m_b + I$, and the ciphertext has the correct distribution. (This is not quite true: but we can smooth out the discrepancy by choosing s large enough – in particular, so that $s/s_{\mathsf{IIMP}} = \text{poly}(n)/\epsilon$.) When $\mathbf{x} \notin I$, c' is in a random coset of I that conveys no information about m_b. Overall, for some polynomial $g(n)$, if there is an algorithm $\mathcal{A}$ that breaks the semantic security of the scheme in time t with probability ϵ for parameter s, then there is an algorithm that, for a $O(\epsilon)$ fraction of bases output by $\mathsf{IdealGen}$, solves HBDDP for parameter $s_{\mathsf{HBDDP}} \leq s \cdot \epsilon / g(n)$ with overwhelming probability in time $O(t \cdot \text{Nm}(I)/\epsilon)$. This reduction is entirely classical (non-quantum).

To reduce SIVP to WBDDP (quantumly), the heavy lifting has already been done by Regev [30]. He provided a quantum reduction of SIVP over the dual lattice L^* to BDDP over the lattice L. A bit more work is necessary to turn his result into a quantum reduction of SIVP over an inverse ideal lattice I^{-1} to BDDP over the ideal lattice I (the inverse of an ideal lattice is not the same as its dual), and then to extend this result to SIVP over (non-inverse) ideals of R.

6 Conclusions and Open Problems

We showed that ideal lattice problems within some fixed rings are, in a sense, random self-reducible. However, the reduction uses a factoring oracle. One open problem is to find a random self-reduction that is efficient in the classical setting – in particular, to find a reduction that does not use factorization.

We presented a KeyGen algorithm that generates ideals according to our average-case distribution, together with a secret key. However, this algorithm is rather complicated, and one wonders whether there is a simpler approach.

While we are able to base Gentry's *somewhat* homomorphic encryption scheme on worst-case hardness, his FHE scheme requires an additional computational assumption – namely, that the (average-case) SSSP is hard. Currently, we do not have a worst-case / average-case reduction for the SSSP that would allow his FHE scheme to be based entirely on worst-case hardness.

Acknowledgments. We thank Dan Boneh, Shai Halevi, Vadim Lyubashevsky, Chris Peikert, Oded Regev, Vinod Vaikuntanathan, and the anonymous reviewers for helpful comments and discussions.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC 1996, pp. 99–108 (1996)
2. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999)
3. Ajtai, M., Dwork, C.: A public key cryptosystem with worst-case / average-case equivalence. In: STOC 1997, pp. 284–293 (1997)
4. Alwen, J., Peikert, C.: Generating Shorter Bases for Hard Random Lattices. In: STACS 2009, pp. 75–86 (2009)
5. Bach, E., Shallit, J.: Algorithmic Number Theory, vol. 1 (1996)
6. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen 296(4), 625–635 (1993)
7. Boyen, X.: Of Lettuces of Lattices: a Framework for Short Signatures and IBE with Full Security. PKC 2010 (to appear 2010)
8. Cai, J.-Y., Nerurkar, A.P.: An Improved Worst-Case to Average-Case Connection for Lattice Problems (extended abstract). In: FOCS 1997, pp. 468–477. IEEE, Los Alamitos (1997)
9. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
10. Gentry, C.: Fully Homomorphic Encryption Using Ideal Lattices. In: STOC 2009, pp. 169–178 (2009)
11. Gentry, C.: A Fully Homomorphic Encryption Scheme. Ph.D. thesis, Stanford University (2009), http://crypto.stanford.edu/craig
12. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for Hard Lattices and New Cryptographic Constructions. In: STOC 2008, pp. 197–206 (2008)
13. Kalai, A.: Generating Random Factored Numbers. Easily. J. Cryptology 16(4), 287–289 (2003); Preliminary version in SODA 2002 (2002)
14. Kaltofen, E., Shoup, V.: Subquadratic-time factoring of polynomials over finite fields. In: STOC 1995, pp. 398–406. ACM, New York (1995)
15. Landau, E.: Neuer Beweis des Primzahlsatzes und Beweis des Primidealsatzes. Mathematische Annalen 56, 645–670
16. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. 261(4), 515–534 (1982)

17. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
18. Lyubashevky, V., Micciancio, D.: Asymptotically efficient lattice-based digital signatures. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 37–54. Springer, Heidelberg (2008)
19. Lyubashevky, V., Micciancio, D.: On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 577–594. Springer, Heidelberg (2009)
20. Lyubashevky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
21. Micciancio, D.: Improving Lattice Based Cryptosystems Using the Hermite Normal Form. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 126–145. Springer, Heidelberg (2001)
22. Micciancio, D.: Improved cryptographic hash functions with worst-case / average-case connection. In: STOC 2002, pp. 609–618 (2002); Full version: Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. SIAM Journal on Computing, 34(1):118–169 (2004)
23. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: FOCS 2002, pp. 356–365 (2002)
24. Micciancio, D., Regev, O.: Worst-Case to Average-Case Reductions Based on Gaussian Measures. In: FOCS 2004, pp. 372–381 (2004); Full version: SIAM J. Comput., 37(1), 267–302 (2007)
25. Nguyen, P.Q., Stern, J.: Adapting Density Attacks to Low-Weight Knapsacks. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 41–58. Springer, Heidelberg (2005)
26. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009, pp. 333–342. ACM, New York (2009)
27. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006)
28. Peikert, C., Rosen, A.: Lattices that Admit Logarithmic Worst-Case to Average-Case Connection Factors. In: Proc. of STOC 2007, pp. 478–487 (2007)
29. Regev, O.: New lattice-based cryptographic constructions. Journal of the ACM 51(6), 899–942 (2004); Extended abstract in STOC 2003 (2003)
30. Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In: Proc. of STOC 2005, pp. 84–93 (2005)
31. Rivest, R., Adleman, L., Dertouzos, M.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation, pp. 169–180 (1978)
32. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing 26(5), 1484–1509 (1997); Extended abstract in FOCS 1994 (1994)
33. Stevenhagen, P.: The Arithmetic of Number Rings. In: Algorithmic Number Theory, vol. 44. MSRI Publications (2008); See also Stevenhagen's course notes Number Rings