Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages

Zvika Brakerski[1] and Vinod Vaikuntanathan[2]

[1] Weizmann Institute of Science
zvika.brakerski@weizmann.ac.il
[2] Microsoft Research and University of Toronto
vinodv@cs.toronto.edu

Abstract. We present a somewhat homomorphic encryption scheme that is both very simple to describe and analyze, and whose security (quantumly) reduces to the worst-case hardness of problems on ideal lattices. We then transform it into a fully homomorphic encryption scheme using standard "squashing" and "bootstrapping" techniques introduced by Gentry (STOC 2009).

One of the obstacles in going from "somewhat" to full homomorphism is the requirement that the somewhat homomorphic scheme be circular secure, namely, the scheme can be used to securely encrypt its own secret key. For all known somewhat homomorphic encryption schemes, this requirement was not known to be achievable under any cryptographic assumption, and had to be explicitly assumed. We take a step forward towards removing this additional assumption by proving that our scheme is in fact secure when encrypting polynomial functions of the secret key.

Our scheme is based on the ring learning with errors (RLWE) assumption that was recently introduced by Lyubashevsky, Peikert and Regev (Eurocrypt 2010). The RLWE assumption is reducible to worst-case problems on ideal lattices, and allows us to completely abstract out the lattice interpretation, resulting in an extremely simple scheme. For example, our secret key is s, and our public key is $(a, b = as + 2e)$, where s, a, e are all degree $(n-1)$ integer polynomials whose coefficients are independently drawn from easy to sample distributions.

1 Introduction

Fully-homomorphic encryption is one of the most sought after goals of modern cryptography. In a nutshell, a fully homomorphic encryption scheme is an encryption scheme that allows evaluation of arbitrarily complex programs on encrypted data. The problem was first suggested by Rivest, Adleman and Dertouzos [36] back in 1978, yet the first plausible construction came thirty years later with the breakthrough work of Gentry in 2009 [14, 15] (although, there has been partial progress in the meanwhile; see, e.g., [21, 12, 30, 6]).

The cornerstone of Gentry's construction is the notion of a "somewhat homomorphic" encryption scheme – namely, an encryption scheme that allows evaluation of a class of functions below some complexity threshold. Specifically,

his construction of a somewhat homomorphic encryption scheme allows the homomorphic evaluation of any (arithmetic or Boolean) function whose polynomial representation has bounded degree. He then showed how to "bootstrap" from a sufficiently powerful somewhat homomorphic encryption scheme into a fully homomorphic encryption scheme. To construct a somewhat homomorphic encryption scheme, Gentry harnessed the power of *ideal lattices* – a sophisticated algebraic structure with many useful properties. Specifically, he was able to reduce the security of his *somewhat homomorphic encryption scheme* to the *worst-case hardness* of standard problems (such as the shortest vector problem) on ideal lattices [15].[3]

Gentry's construction is quite involved – the secret key, even in the private-key version of his scheme, is a short basis of a "random" ideal lattice. Generating pairs of public and secret bases with the right distributions appropriate for the worst-case to average-case reduction is technically quite complicated, and significant effort has been devoted recently to this issue [38, 16]. We will present a scheme where key generation is simply sampling a random degree-$(n-1)$ polynomial with coefficients in $\mathbb{Z}_q$. Furthermore, all parts of our scheme can be described in elementary terms, with no reference to ideals.

A parallel line of work that utilizes ideal lattices in cryptography dates back to the NTRU cryptosystem [22]. The focus of this line of work is to use ideal lattices for *efficient cryptographic constructions*. The added structure of ideal lattices, compared to ordinary lattices, makes their representation more succinct and enables fast computation. Starting with the work of Micciancio [28], there has been an ongoing effort [31, 23, 32, 25, 24] to come up with very efficient constructions of various cryptographic primitives whose security can formally be reduced to the hardness of short-vector problems in ideal lattices. A recent work along these lines, which serves as an essential stepping stone for this work, is that of Lyubashevsky, Peikert and Regev [26].

Lyubashevsky et al. [26] present the *ring learning with errors* (RLWE) assumption, which is the "ring counterpart" of Regev's learning with errors assumption [34]. Roughly speaking, the assumption is that given polynomially many samples over a certain ring of the form $(a_i, a_i s + e_i)$, where s is a random "secret ring element", a_i's are uniformly random in the ring, and e_i are "small" ring elements, an adversary cannot distinguish this sequence of samples from random pairs of ring elements. They show that this simple to state assumption can be (very efficiently) reduced to the worst case hardness of short-vector problems on ideal lattices. They also construct a very efficient ring counterpart to Regev's [34] public-key encryption scheme, as well as a counterpart to the identity based encryption scheme of [17] (using the basis sampling techniques of [39]). The description of the scheme is very elegant since, as explained above, RLWE is stated without directly referring to lattices (similarly to the LWE assumption and ordinary lattices).

[3] The specific variant of the (approximate) shortest vector problem, as well as the specific approximation factor, are irrelevant for the current discussion.

A natural question that comes out of these two lines of work is whether one can get the best of both worlds, namely the expressive functionality on the one hand, and the simplicity and efficiency on the other. We show that indeed this can be done – we construct a somewhat homomorphic encryption scheme based on RLWE and thus inherit the simplicity and efficiency, as well as the worst case relation to ideal lattices. Furthermore, our scheme enjoys *key dependent message security* (KDM security, also known as "circular security") – namely, the scheme can securely encrypt polynomial functions (over an appropriately defined ring) of its own secret key. This property, while interesting in its own right,[4] carries special significance in the context of homomorphic encryption as we explain next.

All known constructions of fully homomorphic encryption employ a "bootstrapping" technique, which enforces the public key of the scheme to grow linearly with the maximal depth of evaluated circuits. This is a major drawback with regards to the usability and the efficiency of the scheme. However, the size of the public key can be made independent of the circuit depth if the somewhat homomorphic scheme can securely encrypt its own secret key. Achieving circular secure somewhat homomorphic encryption has been, thus, an interesting open problem[5] which we resolve in this paper. Unfortunately, the circular security we can prove is with respect to the representation of the secret key as a ring element, where bootstrapping requires circular security with respect to the bitwise representation of the secret key (to be precise: the bitwise representation of the "squashed" secret key). However, since prior to this work it was not known whether somewhat homomorphism can co-exist with *any* form of circular security, we view this property as a significant first step towards removing the above assumption.

We also show how to transform this into a fully homomorphic encryption scheme, following Gentry's blueprint of "squashing" and "bootstrapping".[6] Alternatively, applying techniques from a followup work [10], "squashing" can be avoided at the cost of relying on a "sparse" version of RLWE that is not known to reduce to worst case problems.

Lastly, we remark that our scheme is (additive) key-homomorphic, a property which recently found applications to achieving security against related-key attacks [3].

We elaborate more on the properties of our scheme below.

---

[4] In some ranges of parameters, we improve upon the best known based on any lattice assumption, see Section 1.1.

[5] Of course, one can just *assume* that some scheme is circular secure and hope that it is correct. This has indeed been the solution so far.

[6] Although our somewhat homomorphic encryption scheme assumes only the hardness of ring LWE (which can be based on the worst-case hardness of ideal lattice problems), the squashing step adds another assumption, namely the hardness of the sparse subset sum problem. This is completely analogous to what happens in Gentry's work.

1.1 Our Results and Techniques

We present a public-key encryption scheme under the polynomial LWE (PLWE) assumption, which is a simplified version of the aforementioned RLWE. We show that our scheme is both somewhat homomorphic and circular secure. The former means that bounded complexity functions can be evaluated on encrypted data. The latter means that non-trivial functions of the secret key (including the secret key itself) can be securely encrypted by our scheme. Finally, we show how fully homomorphic encryption can be obtained by bootstrapping, using "Gentry-style" squashing. We also mention how squashing can be traded for a sparse variant of PLWE using techniques from a follow-up work. Details follow.

The Assumption. We formally define the *polynomial learning with errors* assumption (PLWE), which is a simplified version of [26]'s RLWE assumption. We emphasize that PLWE is implicit in [26] and we just make it explicit. In particular, using the results of [26], the hardness of PLWE can be based on the worst-case hardness of ideal lattice problems.

In the standard parameter setting, we consider the polynomial ring $R_q \doteq \mathbb{Z}_q[x]/\langle x^n+1\rangle$, where n is a power of 2, namely the ring of all integer polynomials of degree $(n-1)$ and coefficients in $\mathbb{Z}_q$. Addition and multiplication over this ring are defined modulo (x^n+1, q). The PLWE assumption in this setting is that it is hard to distinguish polynomially many samples from the distribution $(a_i, a_i s+e_i)$ and the same number of samples from the distribution (a_i, u_i), where s, the a_i's and the u_i's are uniform in R_q and the e_i's are "noise polynomials" whose coefficients are sampled (independently) from a narrow Gaussian (which we refer to as the noise distribution χ). An important observation is that the assumption still holds if s is sampled from the noise distribution χ rather than the uniform distribution (this is the "Hermite normal form" of the assumption).

The resemblance to standard learning with errors is apparent, especially when noticing that the additive group of R_q and $\mathbb{Z}_q^n$ are isomorphic (as vector spaces, obviously multiplication in the latter is undefined). The new aspect of PLWE is the use of multiplication in R_q in the place of inner product, which results in a larger amount of pseudo-randomness generated per sample. Rather than obtaining just one element in $\mathbb{Z}_q$, as in standard LWE, we here obtain n such elements.

The PLWE problem, in some parameter settings, is reducible to the worst case hardness of "short vector problems" in ideal lattices. This is a straightforward consequence of [26]. Specifically, we require that q is a sub-exponential prime and that there is a sub-exponential gap between the q and the standard deviation of the Gaussian error (used to sample the coefficients of the e_i's described above). These parameters translate to the worst-case hardness of approximating the shortest vector problem to within a (slightly) sub-exponential approximation factor, using (slightly) sub-exponential algorithms. We note that the best (ideal) lattice algorithms run in time roughly $2^{n/k}$ to come up with a 2^k-approximation of shortest vectors (where k is a "tunable" parameter of the algorithm).

The Basic Scheme. Our somewhat homomorphic encryption scheme is so simple that the best way to present it is to spell it out. We first present the symmetric-key variant of the scheme and then explain how to transform it into a public-key scheme.

To generate the (symmetric) key for our scheme, we sample $s \xleftarrow{\$} \chi$ (in fact, if we only care about homomorphism and not KDM security, sampling $s \xleftarrow{\$} R_q$ is sufficient). Encryption is performed by sampling $a \xleftarrow{\$} R_q$ and $e \xleftarrow{\$} \chi$ and outputting the ciphertext $\mathbf{c} = (c_0, c_1)$ where $c_1 = -a$ and $c_0 = as + 2e + m$. The message m resides in the ring of polynomials with binary coefficients $R_2 = \mathbb{Z}_2[x]/\langle x^n + 1\rangle$ (which is isomorphic to $\{0,1\}^n$ but, as one might guess, has additional structure that will be used for homomorphism). To decrypt the ciphertext $\mathbf{c} = (c_0, c_1)$, one computes $c_0 + c_1 s \pmod 2$. Note that we slightly deviate from the standard notation for LWE based schemes for reasons that will be apparent below.

The correctness of the scheme is apparent, and security follows from the PLWE assumption by noting that $(a, as + 2e)$ is indistinguishable from (a, u), where u is uniform (the additional factor of 2 is not a problem, since 2 is invertible in $\mathbb{Z}_q$). The ciphertext, thus, is indistinguishable from one that carries no information on the message.

A Public-Key Scheme. We obtain a public-key encryption scheme using a variant of the construction of [26]. Notice that in order to encrypt with our symmetric scheme, we only need the ability to generate pairs of the form $(a, as + 2e)$. We show that given one such pair, it is easy to re-randomize and generate as many of them as we want. Given $(a, b = as+2e)$, we consider the tuple $(a' = av+2e', b' = bv + 2e'')$, where $v, e' \xleftarrow{\$} \chi$, $e'' \xleftarrow{\$} \chi'$, where χ' is a noise distribution like χ, only with larger standard deviation. It holds that $b' = asv + 2(ev + e'') = a's + 2(ev + e'' - e's)$. If the standard deviation of e'' is sufficiently large, it holds that $b' \overset{s}{\approx} a's + 2e''$. In addition, a' is computationally indistinguishable from uniform (even for an adversary who knows s). Therefore (a', b') are computationally indistinguishable, even given the secret key, from an appropriately distributed pair that can be used for encryption. Security is not affected because the original pair (a, b) posted as the public key is computationally indistinguishable from being independent of s.

Somewhat Homomorphic Scheme. Achieving additive homomorphism is simple, via coordinate-wise addition: $\mathbf{c}_{\text{add}} = \mathbf{c} + \mathbf{c}' = (c_0 + c_0', c_1 + c_1') = ((a + a')s + 2(e+e') + (m+m'), -(a+a'))$, which decrypts properly so long as the error does not "blow up". It is multiplicative homomorphism that requires careful handling. The intuition is that multiplying together the c_0 elements of 2 ciphertexts should create an element that depends on the product of the messages. Writing it down, we indeed see that $c_0 \cdot c_0' = -aa's^2 + (c_0 a' + c_0' a)s + 2(2ee' + em' + e'm) + mm'$, which almost looks like a legitimate ciphertext, except for the term $-aa's^2$, which contains a high power of s. The key observation is that we can make this ciphertext decryptable at the expense of adding an element to the ciphertext.

Our new ciphertext will be $\mathbf{c}_{\text{mult}} = (c_{\text{mult},0}, c_{\text{mult},1}, c_{\text{mult},2})$, where $c_{\text{mult},2} = c_1 c'_1$, $c_{\text{mult},1} = c_0 c'_1 + c'_0 c_1$, $c_{\text{mult},0} = c_0 c'_0$. In other words, since we know that $m + 2e = c_0 + c_1 s$ and $m' + 2e' = c'_0 + c'_1 s$, then it holds that $(m + 2e) \cdot (m' + 2e') = (c_0 + c_1 s) \cdot (c'_0 + c'_1 s)$. We can open the parenthesis on the right hand side symbolically (without knowing s), and come up with $\mathbf{c}_{\text{mult}}$ such that $(c_0 + c_1 s) \cdot (c'_0 + c'_1 s) = c_{\text{mult},0} + c_{\text{mult},1} s + c_{\text{mult},2} s^2$. Note that all of the above can be computed from $\mathbf{c}, \mathbf{c}'$. To decrypt a 3 element ciphertext $\mathbf{c} = (c_0, c_1, c_2)$, the decryption process will be $c_0 + c_1 s + c_2 s^2 \pmod 2$. It is important to notice that ciphertexts of all lengths can be again added and multiplied, where addition results in a ciphertext of the maximal length of its operands and multiplication results in a ciphertext of length sum of operands minus 1.

The limiting factor on the number of homomorphic operations is the growth of the error term. We start with a sub-exponential ratio between the modulus q and the elements of e. In order to decrypt correctly, we need this ratio to be more than 2. Making the calculations (see Section 3), the total degree of the evaluated function (represented as a polynomial) needs to be less than n^ϵ for some constant ϵ (additions are relatively negligible).

The restriction on the error also limits the total length of a ciphertext: A decryptable ciphertext can have no more than n^ϵ elements (recall that the number of elements in a ciphertext and the degree of homomorphic operations are closely related). We denote this limit on the maximal degree by D.

KDM *Security.* The KDM properties of our scheme take after ideas from the work of Applebaum, Cash, Peikert and Sahai [2], who showed KDM security (w.r.t. linear functions) for Regev's LWE based scheme, and from the work of Malkin, Teranishi and Yung [27], who showed KDM security w.r.t. polynomials of the secret key (treated as integer) based on the decisional composite residuosity assumption.

To see that our scheme can encrypt non-trivial functions of its own secret key, consider the ciphertext $\mathbf{c} = (as + 2e + s, -a)$ which "looks like" an encryption of the secret key s.[7] If we define $a' = a + 1$, however, we have that $\mathbf{c} = (a's + 2e, -a' + 1)$. We notice that $(a', a's + 2e)$ is exactly a PLWE instance, so it is computationally indistinguishable from (a', u'), where u' is uniform. We have that $c \overset{c}{\approx} (u', -a' + 1)$, which is a completely uniform pair. This methodology is easy to extend to any linear function of s and for any polynomial number of ciphertexts.

We now revisit our previous claims that the aforementioned $\mathbf{c}$ "looks like" an encryption of s. In fact, s, drawn from χ, does not necessarily lie in the message space of the scheme, R_2, so the above statement is possibly meaningless! There are two ways to resolve this difficulty. One is to observe that choosing the parameters correctly, our c is statistically indistinguishable from an encryption of $s \pmod 2$, which is a non-trivial function of the secret key. Alternatively, and perhaps more satisfactory, is replacing the coefficient 2 in our scheme by a

[7] In fact, there is an important discrepancy between c and a legal encryption of s, but we will ignore it for this part of the discussion and return to it later.

larger prime t, such that with all but negligible probability, $s \in R_t$. This enables achieving KDM security w.r.t. linear functions over the ring R_t.

To obtain KDM security for higher degree polynomials, we use a technique similar to that of [27]. We focus on quadratic functions for the sake of concreteness: We change our encryption algorithm so that encryption of a message m is performed in 2 stages: First, we compute a ciphertext as in our previous scheme $(a_1 s + 2e_1 + m, -a_1)$, but then, rather than sending $-a_1$ as a part of the ciphertext, we encrypt it too and obtain $(a_2 s + 2e_2 - a_1, -a_2)$. The final ciphertext will be $\mathbf{c} = (c_0, c_1, c_2) = (a_1 s + 2e_1 + m, a_2 s + 2e_2 - a_1, -a_2)$. To decrypt, we first extract $-a_1$ (plus some noise) from c_2, c_1, and then use this noisy $-a_1$ to decrypt c_0 and extract m. The decryption process here involves more noise than our standard scheme, but an appropriate choice of parameters enables correct decryption. Now let us consider $\mathbf{c} = (c_0, c_1, c_2) = (a_1 s + 2e_1 + s^2, a_2 s + 2e_2 - a_1, -a_2)$. Defining $a'_1 = a_1 + s$, $a'_2 = a_2 + 1$, we have that $\mathbf{c} = (a'_1 s + 2e_1, a'_2 s + 2e_2 - a'_1, -a'_2 + 1)$. Applying PLWE twice, we have that $\mathbf{c}$ is computationally indistinguishable from a tuple of uniform ring elements.

By repeating the above process, we can securely encrypt degree D polynomials using ciphertexts of length D. We notice that similar considerations apply in this case and in the case of the somewhat homomorphic scheme defined above, and indeed the decryption process in the two cases is very similar.

We further remark that our scheme can be proven to be $\mathrm{KDM}^{(\nu)}$ secure w.r.t. the same class of polynomial functions. Namely, even in the case where there is a polynomial number, ν, of users, encrypting functions of each other's secret keys, our scheme remains secure. However, this property is less relevant for homomorphism and we refer the reader to Section 4.1 for details. Achieving $\mathrm{KDM}^{(\nu)}$ security w.r.t. super-constant degree polynomials of the secret key was not known under any lattice assumption.

Full Homomorphism Using Squashing. One way to obtain a fully homomorphic scheme is to use Gentry's "bootstrapping" and "squashing". First, we notice that, as in all previously known somewhat homomorphic encryption schemes, the decryption circuit of our basic scheme has higher degree than can be homomorphically evaluated. Thus we use the by now established technique of posting, along with the public key, a sequence of elements that "hide" the secret key as a sparse subset sum. This enables reducing the complexity of decryption as we describe below.

We consider the vector $\mathbf{s} = (1, s, \ldots, s^D) \in R_q^{D+1}$ which contains all powers of the secret key s that are relevant for decryption. Note that to decrypt a ciphertext vector $\mathbf{c} = (c_0, \ldots, c_D)$, one needs to compute $\sum_{i=0}^{D} c_i s^i = \langle \mathbf{c}, \mathbf{s} \rangle$, where the "inner product" is over the ring R_q, and then take the result mod 2.

We post, along with the public key, a sequence of vectors $\mathbf{z}_1, \ldots, \mathbf{z}_m \in R_q^{D+1}$ that are uniformly sampled conditioned on the existence of a small set $L \subseteq [m]$ s.t. $|L| = n^\delta$ and $\sum_{\ell \in L} \mathbf{z}_i = \mathbf{s}$. The new secret key, therefore, is the set L (represented as a binary incidence vector over $\{0,1\}^m$). The encryptor then, along with the ciphertext vector $\mathbf{c}$, also posts $\tau_j = \langle \mathbf{c}, \mathbf{z}_\ell \rangle$, for all $\ell \in [m]$. The decryption task reduces to computing $\sum_{\ell \in L} \tau_\ell$ and taking the result modulo

2. As can be verified, this process, expressed as a polynomial over the bits of the incidence vector, has degree $\sim n^{\delta}$. If we choose $\delta < \epsilon$ (recall that n^{ϵ} is the maximal degree that can be evaluated), we have that the decryption circuit is shallow enough to be homomorphically evaluated. This is sufficient for the "bootstrapping" procedure a la Gentry.

As explained above, in order to use the bootstrapping method to obtain a public key whose size does not depend on the evaluated circuit, it is necessary to provide the evaluator with an encryption of the secret key of the somewhat homomorphic scheme. This requirement refers to the secret key of the *squashed scheme*, namely to the bits of the incidence vector of L. As we explained above, our KDM security proof does not extend to this case.

*Full Homomorphism Using Sparse-*PLWE. We mention a different way to achieve full homomorphism, as an alternative to squashing. In a followup of this paper, [10] introduced a "re-linearization" technique which they use to construct a fully homomorphic scheme based on the standard LWE assumption. One can verify that the re-linearization technique can be applied to PLWE as well – namely, a ciphertext $\mathbf{c} = (c_0, \ldots, c_D)$ can be "re-linearized" to a ciphertext $\mathbf{c} = (c_0, c_1)$ that is, in turn, decrypted in the same way. The decryption circuit thus becomes $c_0 + c_1 s \pmod 2$ and its complexity depends on the number of non-zero coefficients in the polynomial $s \in R_q$. If we sampled s from a distribution over n^{δ}*-sparse* polynomials, namely ones that have at most n^{δ} non-zero coefficients, then the decryption complexity will reduce in a similar manner to squashing. For this method to work, one needs to explicitly assume that PLWE is secure even when using such sparse s. Although the hardness of such assumption has not been thoroughly explored, we are not aware of an approach for breaking it either.

An Application: Private Information Retrieval. Our somewhat homomorphic encryption scheme can be used to construct a very efficient private information retrieval protocol [11, 18] with almost logarithmic communication complexity, and security under worst-case hardness assumptions. While any (appropriate) somewhat homomorphic encryption scheme can be used to construct a PIR protocol (in particular, the scheme of [14]), our construction from ring LWE results in a particularly efficient and elegant PIR scheme. Due to space constraints, we do not provide a detailed explanation in this extended abstract.

1.2 Other Related Works

The only known candidate for fully homomorphic encryption, aside from Gentry's aforementioned scheme (and a variant thereof [38]), was presented by van Dijk, Gentry, Halevi and Vaikuntanathan [13]. Their scheme works over the integers and relies on a new assumption which roughly states that finding the greatest common divisor of many "noisy" multiples of a number is computationally hard. They cannot, however, reduce their assumption to worst case hardness.

The efficiency of implementing Gentry's scheme also gained much attention. Smart and Vercauteren [38], as well as Gentry and Halevi [16] conduct a study

on reducing the complexity of implementing the scheme, specifically the key generation process. We note that the key generation process in this work is simpler and does not require generating lattice bases.

Candidate KDM secure encryption schemes in the standard model (i.e. without random oracles) started with the work of Boneh, Halevi, Hamburg and Ostrovsky [7] who presented a scheme based on the decisional Diffie-Hellman assumption that can securely encrypt linear combinations of the bits of its secret key. The aforementioned work of [2] showed a similar result based on LWE, where now the linear functions were over the components of the secret key, that reside in the space $\mathbb{Z}_p$ for some prime p. Later, Brakerski and Goldwasser [8] showed KDM security for linear functions based on a class of assumptions they refer to as "subgroup indistinguishability assumptions", which includes quadratic residuosity and decisional composite residuosity. The domain of the functions varied by the assumption. A number of works [4, 9, 1] showed that KDM w.r.t. linear functions can be extended to more complex functions. The work of [27] takes a different path by treating the secret key as an element in a ring (integers modulo a composite, in their case), an approach we adopt here as well.

In a followup work, [10] showed that fully homomorphic encryption can be achieved from the classical LWE assumption, without referring to ideal lattices and without squashing. Some of their techniques can also be applied to PLWE (e.g. to achieve full homomorphism via sparse polynomials as we describe above).

1.3 Notation

Let $\mathcal{D}$ denote a distribution over some finite set S. Then, $d \xleftarrow{\$} \mathcal{D}$ is used to denote the fact that d is chosen from the distribution $\mathcal{D}$. When we say $d \xleftarrow{\$} S$, we simply mean that d is chosen from the uniform distribution over S.

The ring of polynomials over the integers (i.e. symbolic polynomials with integer coefficients) is denoted $\mathbb{Z}[x]$. Given a degree n polynomial $f(x)$, the ring $\mathbb{Z}[x]/\langle f(x)\rangle$ is the ring of all polynomials modulo $f(x)$. The ring of polynomials with coefficients in $\mathbb{Z}_q$ is denoted $\mathbb{Z}_q[x]$ and $\mathbb{Z}_q[x]/\langle f(x)\rangle$ is defined analogously to above. For additional background in algebraic number theory, we refer the reader to to [40].

We denote scalars in plain (e.g. x) and vectors in bold (e.g. $\mathbf{v}$). A norm of a vector is denoted by $\|\mathbf{v}\|$ and always refers to ℓ_∞: $\|\mathbf{v}\| = \max_i |v_i|$. The norm of a polynomial $\|p(x)\|$ is the norm of its coefficient vector. More generally, we use the standard isomorphism between degree $(n-1)$ polynomials in $\mathbb{Z}[x]$ and vectors in $\mathbb{Z}^n$, given by the vector of coefficients, that allows to treat the two objects interchangeably: the vector $\mathbf{p}$ will indicate the vector of coefficients of $p(x)$. We explicitly mention when we use this isomorphism.

We let the distribution $D_{\mathbb{Z}^n,r}$ to indicate the *n-dimensional discrete Gaussian distribution*. To sample a vector $\mathbf{x} \in \mathbb{Z}^n$ from this distribution, sample $y_i \in \mathbb{R}$ from the Gaussian of standard deviation r and set $x_i := \lfloor y_i \rceil$, where $\lfloor \cdot \rceil$ represents rounding to the nearest integer. Using the isomorphism mentioned above, we treat $D_{\mathbb{Z}^n,r}$ as a distribution over integer degree n polynomials. Note that in

this work we only need *spherical* Gaussian distributions, in which the standard deviation over each dimension is the same.

2 The Ring LWE Problem, and Variants

In this section, we describe a variant of the "ring learning with errors" (RLWE) assumption of Lyubaskevsky, Peikert and Regev [26], that we call *polynomial* LWE (or, PLWE). This assumption is in fact implicit in [26], and can be thought of as a special case of their general RLWE assumption. Fortunately, for the parameters of interest to us, it follows from [26] that breaking the PLWE assumption leads to an algorithm to solve worst-case ideal lattice problems. Our motivation in working with the PLWE assumption is due in part to our desire to keep the exposition *elementary*, but is also dictated by the particular choice of the message encoding in our encryption schemes. See Section 2.1 for a detailed comparison, and the statement of the worst-case to average-case reduction for PLWE.

The PLWE assumption is analogous to the (by now standard) "learning with errors" (LWE) assumption, defined by Regev [34, 35] (generalizing the learning parity with noise assumption of Blum et al. [5]). In the PLWE assumption, we consider rings $R \doteq \mathbb{Z}[x]/\langle f(x)\rangle$ and $R_q \doteq R/qR$ for some degree n integer polynomial $f(x) \in \mathbb{Z}[x]$ and a prime integer $q \in \mathbb{Z}$. Note that $R_q \equiv \mathbb{Z}_q[x]/\langle f(x)\rangle$, i.e. the ring of degree $(n-1)$ polynomials with coefficients in $\mathbb{Z}_q$. Addition in these rings is done component-wise in their coefficients (thus, their additive group is isomorphic to $\mathbb{Z}^n$ and $\mathbb{Z}_q^n$ respectively). Multiplication is simply polynomial multiplication modulo $f(x)$ (and also q, in the case of the ring R_q).

Thus an element in R (or R_q) can be viewed as a degree $(n-1)$ polynomial over $\mathbb{Z}$ (or $\mathbb{Z}_q$). As we mentioned in Section 1.3, we represent such an element using the vector of its coefficients. For an element $a(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} \in R$, we let $\|a\| = \max |a_i|$ denote its ℓ_∞ norm.

The $\mathsf{PLWE}_{f,q,\chi}$ assumption is parameterized by an integer polynomial $f(x) \in \mathbb{Z}[x]$ of degree n (which defines the ring $R = \mathbb{Z}[x]/\langle f(x)\rangle$), a prime integer $q \in \mathbb{Z}$ and an error distribution χ over R.[8] We require that χ is efficiently sampleable in our representation, namely that it is efficient to sample the coefficients of the polynomial representing the sampled element.

Let $s \xleftarrow{\$} R_q$ be a uniformly random ring element. The assumption is that given any polynomial number of samples of the form $(a_i, b_i = a_i \cdot s + e_i) \in (R_q)^2$, where a_i is uniformly random in R_q and e_i is drawn from the error distribution χ, the b_i's are computationally indistinguishable from uniform in R_q. If the number of samples that the distinguisher obtains is limited by $\ell = \ell(\kappa)$, then we call this assumption $\mathsf{PLWE}^{(\ell)}_{f,q,\chi}$. Our formal definition below presents the *hermite normal form* of the assumption, where the secret s is sampled from the noise distribution χ rather than being uniform in R_q. This presentation is more useful for the purposes of this paper and it turns out that to be equivalent to the original one, up to obtaining one additional sample [2, 26].

[8] To be precise, $n(\kappa), q(\kappa)$ are functions of the security parameter κ and $\{f_\kappa(x)\}$ and $\{\chi_\kappa\}$ are ensembles of polynomials and distributions respectively.

Definition 1 (The PLWE Assumption - Hermite Normal Form). *For all $\kappa \in \mathbb{N}$, let $f(x) = f_\kappa(x) \in \mathbb{Z}[x]$ be a polynomial of degree $n = n(\kappa)$, let $q = q(\kappa) \in \mathbb{Z}$ be a prime integer, let the ring $R \doteq \mathbb{Z}[x]/\langle f(x)\rangle$ and $R_q \doteq R/qR$, and let χ denote a distribution over the ring R.*

The polynomial LWE assumption $\mathsf{PLWE}_{f,q,\chi}$ *states that for any $\ell = \text{poly}(\kappa)$ it holds that*

$$\{(a_i, a_i \cdot s + e_i)\}_{i\in[\ell]} \overset{c}{\approx} \{(a_i, u_i)\}_{i\in[\ell]} \ ,$$

where s is sampled from the noise distribution χ, a_i are uniform in R_q, the "error polynomials" e_i are sampled from the error distribution χ, and finally, the ring elements u_i are uniformly random over R_q.

When we require the indistinguishability to hold given only ℓ samples (for some $\ell = \text{poly}(\kappa)$), we denote the assumption by $\mathsf{PLWE}^{(\ell)}_{f,q,\chi}$.

Note that we define the PLWE assumption as a *decisional* assumption. One could also define the search assumption which requires an adversary to find $s \in R_q$, given any polynomial number of samples $(a_i, a_i \cdot s + e_i)$. The search and decisional assumptions are equivalent for some range of parameters, as shown by [26]. We focus here on the decisional assumption since that is the most natural for cryptographic applications.

Scaling the noise. It is very useful in our schemes to generate the PLWE samples as $(a_i, a_i \cdot s + t \cdot e_i)$, where a_i, s, e_i are as above and $t \in \mathbb{Z}_q^*$. This variant is equivalent to PLWE just by virtue of q and t being relatively prime as stated below. The proof is straightforward and is omitted.

Proposition 1. *Let $f(x), q$ and χ be as in Definition 1. Let $t = t(\kappa) \in \mathbb{Z}_q^*$ (thus t and q are relatively prime). Then for any $\ell = \text{poly}(\kappa)$, the $\mathsf{PLWE}^{(\ell)}_{f,q,\chi}$ assumption implies that,*

$$\{(a_i, a_i \cdot s + t \cdot e_i)\}_{i\in[\ell]} \overset{c}{\approx} \{(a_i, u_i)\}_{i\in[\ell]} \ .$$

where a_i, s, e_i and u_i are as in Definition 1.

2.1 Choice of Parameters

Our results rely on a specific choices of the polynomial $f(x)$, the modulus q, and the error distribution χ. The parameter choices are dictated by the search-to-decision reduction of [26], as well as our choice of message encoding in the encryption scheme (which seems necessary to achieve homomorphic properties). In particular, setting κ as our security parameter, we assume that:

- We set $f(x) = x^n + 1$, where $n = 2^{\lfloor \log \kappa \rfloor - 1}$ (this polynomial is also denoted $\Phi_m(x)$, where $m = 2n$; see below). Since $n \in (\kappa/4, \kappa]$, all asymptotics can be stated in terms of n. For the knowledgable reader we mention that $f(x) = \Phi_m(x)$ is the m^{th} cyclotomic polynomial.
  In addition, the fact that $f(x) = x^n + 1$ means that multiplication of ring elements does not increase their norm by too much (see lemmas below).
- The error distribution χ is the discrete Gaussian distribution $D_{\mathbb{Z}^n, r}$ for some $r > 0$. A sample from this distribution defines a polynomial $e(x) \in R$.

Some Useful Facts. We present some elementary facts about the Gaussian error distribution, and multiplication over the ring $\mathbb{Z}[x]/\langle x^n+1\rangle$. The first fact bounds the (Euclidean and therefore, the ℓ_∞) length of a vector drawn from a discrete Gaussian of standard deviation r by $r\sqrt{n}$. The second says that the statistical distance between two Gaussian distributions with the same standard deviation r (but different centers) is proportional to Δ/r, where Δ is the distance between their centers. The third and final fact says that multiplication in the ring $\mathbb{Z}[x]/\langle\Phi_m(x)\rangle$ increases the norm of the constituent elements only by a modest amount.

Lemma 1 (see [29], Theorem 4.4). *Let $n \in \mathbb{N}$. For any real number $r = \omega(\sqrt{\log n})$, we have $\Pr_{x\leftarrow D_{\mathbb{Z}^n,r}}[||x|| > r\sqrt{n}] \leq 2^{-n+1}$.*

Lemma 2 (see [20], Lemma 3). *Let $n \in \mathbb{N}$. For any real number $r = \omega(\sqrt{\log n})$, and any $\mathbf{c} \in \mathbb{Z}^n$, the statistical distance between the distributions $D_{\mathbb{Z}^n,r}$ and $D_{\mathbb{Z}^n,r,\mathbf{c}}$ is at most $||\mathbf{c}||/r$.*

Lemma 3 (see [23, 14]). *Let $n \in \mathbb{N}$, $m = 2n$, and let $f(x) = \Phi_m(x) = x^n + 1$ and let $R = \mathbb{Z}[x]/\langle\Phi_m(x)\rangle$. For any $s, t \in R$, $||s\cdot t \pmod{\Phi_m(x)}|| \leq \sqrt{n}\cdot||s||\cdot||t||$, and $||s \cdot t \pmod{\Phi_m(x)}||_\infty \leq n \cdot ||s||_\infty \cdot ||t||_\infty$.*

The Worst-case to Average-case Connection. We state a worst-case to average-case reduction from the shortest vector problem on ideal lattices to the PLWE problem for our setting of parameters. The reduction stated below is a special case of the results of [26].

Theorem 1 (A special case of [26]). *Let κ be the security parameter. Let $k \in \mathbb{N}$ and let $m = 2^{\lfloor \log \kappa \rceil}$ be a power of two. Let $\Phi_m(x) = x^n + 1$ be the m^{th} cyclotomic polynomial of degree $n = \varphi(m) = m/2$. Let $r \geq \omega(\sqrt{\log n})$ be a real number, and let $q \equiv 1 \pmod m$ be a prime integer. Let $R = \mathbb{Z}[x]/\langle\Phi_m(x)\rangle$. Then:*

- *There is a randomized reduction from $2^{\omega(\log n)} \cdot (q/r)$-approximate R-SVP to $\mathsf{PLWE}_{\Phi_m,q,\chi}$ where $\chi = D_{\mathbb{Z}^n,r}$ is the discrete Gaussian distribution. The reduction runs in time $\mathrm{poly}(n, q)$.*
- *There is a randomized reduction from $(n^2q/r) \cdot (n(\ell+1)/\log(n(\ell+1)))^{1/4}$-approximate R-SVP to $\mathsf{PLWE}^{(\ell)}_{\Phi_m,q,\chi}$ where $\chi = D_{\mathbb{Z}^n,r}$ is the discrete Gaussian distribution. The reduction runs in time $\mathrm{poly}(n, q, \ell)$.*[9]

3 A Somewhat Homomorphic Encryption Scheme

We present a somewhat homomorphic encryption scheme with message space $R_t = \mathbb{Z}_t[x]/\langle f(x)\rangle$ for some integer $t = t(\kappa)$. The homomorphism will be over this ring. For the sake of concreteness, we advise the reader to think of $t = 2$

[9] For the interested reader, we remark that the term $(\ell + 1)$ replaces the original ℓ of [26] due to our choice to define PLWE in hermite normal form.

as a running example. We describe our scheme in the symmetric case in Section 3.1 and then describe the public-key variant in Section 3.2. The transition to full homomorphism via squashing and bootstrapping is fairly standard and is omitted due to space limitations.

3.1 The Symmetric Scheme

Let κ denote the security parameter. Our scheme is parameterized by a prime number q and a prime $t \in \mathbb{Z}_q^*$, a degree n polynomial $f(x) \in \mathbb{Z}[x]$, and an error distribution χ over the ring $R_q \doteq \mathbb{Z}_q[x]/\langle f(x)\rangle$. The parameters n, f, q and χ are public and we assume that given κ, there are polynomial-time algorithms that output f and q, and sample from the error distribution χ. An additional parameter of the scheme is an integer $D \in \mathbb{N}$ that is related to the maximal degree of homomorphism allowed (and to the maximal ciphertext length). This is not a "free" parameter, and is determined by f, q, χ in the analysis of the scheme.

- $\mathsf{SH.Keygen}(1^\kappa)$: Sample a ring element $s \xleftarrow{\$} \chi$ and set the secret key $sk{:=}s$. Define the *secret key vector* as $\mathbf{s}{:=}(1, s, s^2, \ldots, s^D) \in R_q^{D+1}$, which is efficiently computable given s and will be used in the decryption process.
- $\mathsf{SH.Enc}(sk, m)$: Recall that our message space is R_t. Namely, we encode our message as a degree n polynomial with coefficients in $\mathbb{Z}_t$.
  To encrypt, sample $(a, b = as{+}te) \in R_q^2$, where $a \xleftarrow{\$} R_q$ and $e \xleftarrow{\$} \chi$. Compute

  $$c_0{:=}b + m \in R_q \qquad \text{and} \qquad c_1{:=} -a$$

  and output the ciphertext $\mathbf{c}{:=}(c_0, c_1) \in R_q^2$.
  An important note is that the encryptor only uses the key s in order to sample (a, b). This will be important when we present our public-key scheme, where we will show that the public key enables sampling from this distribution without direct access to s.
  While the encryption algorithm only generates ciphertexts $\mathbf{c} \in R_q^2$, homomorphic operations (described below) might add more elements to the ciphertext. Thus the most generic form of a decryptable ciphertext in our scheme is $\mathbf{c} = (c_0, \ldots, c_d)$ for $d \leq D$. We remark that, as we will show below, "padding with zeros" does not effect the ciphertext. Namely $(c_0, \ldots, c_d) \equiv (c_0, \ldots, c_d, 0, \ldots, 0)$.
- $\mathsf{SH.Eval}(p(\xi_1, \ldots, \xi_\ell), (\mathbf{c}_1, \ldots, \mathbf{c}_\ell))$: We show how to evaluate an ℓ-variate polynomial $p : R_t^\ell \to R_t$. To this end, we show how to homomorphically add and multiply two elements in R_t.
  - Given two ciphertexts $\mathbf{c} = (c_0, \ldots, c_d)$ and $\mathbf{c}' = (c'_0, \ldots, c'_d)$ (we assume w.l.o.g that they have the same length, e.g. by padding), output the ciphertext $\mathbf{c}_{\text{add}} = \mathbf{c} + \mathbf{c}' = (c_0 + c'_0, \ldots, c_d + c'_d) \in R_q^{d+1}$, as an encryption of the *sum* of the underlying messages. Namely, addition is done by coordinate-wise vector addition of the ciphertext vectors. Note that addition does not increase the number of elements in the ciphertext vectors.

- Given two ciphertexts $\mathbf{c} = (c_0, \ldots, c_d)$ and $\mathbf{c}' = (c'_0, \ldots, c'_{d'})$ (here we do not pad with zeros), an encryption of their *product* is computed as follows.
  Let v be a *symbolic* variable and consider the expression $\left(\sum_{i=0}^{d} c_i v^i\right) \cdot \left(\sum_{i=0}^{d'} c'_i v^i\right)$ (over R_q). We can (symbolically, treating v as an unknown variable) open the parenthesis to compute $\hat{c}_0, \ldots, \hat{c}_{d+d'} \in R_q$ such that for all $v \in R_q$

$$\left(\sum_{i=0}^{d} c_i v^i\right) \cdot \left(\sum_{i=0}^{d'} c'_i v^i\right) \equiv \sum_{i=0}^{d+d'} \hat{c}_i v^i \ .$$

  The output ciphertext is $\mathbf{c}_{\text{mult}} = (\hat{c}_0, \ldots, \hat{c}_{d+d'})$.

  We claim that if $\mathbf{c}$ is an encryption of a message $m \in R_t$ and $\mathbf{c}'$ is an encryption of $m' \in R_t$, then these two operations generate ciphertexts that decrypt to $m + m'$ and mm', respectively, where arithmetics is over R_t.

– $\mathsf{SH.Dec}(sk, \mathbf{c})$: Recall that the general form of a decryptable ciphertext is w.l.o.g $\mathbf{c} = (c_0, c_1, \ldots, c_D) \in R_q^{D+1}$, e.g. by padding.
  To decrypt, we first compute $\langle \mathbf{c}, \mathbf{s} \rangle \doteq \sum_{i=0}^{D} c_i s^i \in R_q$, which can be interpreted as inner product over R_q^{D+1}, and output $m = \langle \mathbf{c}, \mathbf{s} \rangle \pmod{t}$ as the message.
  Note that the condition for correct decryption is that the ℓ_∞ norm of the polynomial $\langle \mathbf{c}, \mathbf{s} \rangle$ is smaller than $q/2$.

We note that for the sake of the symmetric key somewhat homomorphic encryption scheme, the secret key s can be chosen uniformly at random. Choosing s from the error distribution is important both in the public-key variant as well as for KDM security.

We state the correctness and security below. Proofs are omitted from this extended abstract.

Theorem 2. *Let $\chi = D_{\mathbb{Z}^n, r}$ be the discrete Gaussian noise distribution with standard deviation r. The scheme described above is a somewhat homomorphic encryption scheme capable of evaluating ℓ-variate degree-D polynomials over R_t, as long as $M \cdot (trn^{1.5})^D < q/2$, where M is the ℓ_∞ norm of the polynomial (i.e. its maximal coefficient).*

Theorem 3. *Let $n, q, f(x)$ be as in the scheme, let $r = \text{poly}(n)$ and $q = 2^{n^\epsilon}$ for some $0 < \epsilon < 1$. Then, the scheme allows evaluation of degree-$O(n^\epsilon / \log n)$ polynomials with at most $2^{O(n^\epsilon / \log n)}$ terms, and is secure under the worst-case hardness of approximating shortest vectors on ideal lattices to within a factor of $O(2^{n^\epsilon})$.*

3.2 Public-Key Encryption

There is a number of ways to go from symmetric-key to public-key somewhat homomorphism. The work of Rothblum [37] provides a generic though inefficient

way to go from homomorphic symmetric to public key encryption. Alternatively, one can use re-randomization via the leftover hash lemma (as used in Regev's LWE based scheme). However, the greatest efficiency is achieved using a method that appears in the full version of [26] and due to space limitations will be the only one discussed here.

Recall that in order to encrypt with our symmetric scheme, we only need the ability to generate pairs of the form $(a, as + te)$. We show that given one such pair (with smaller noise parameter), it is easy to re-randomize and generate as many of them as we want.

Concretely, we show that given one sample $(a, b = as+te)$, where the noise e comes from a distribution χ, we can generate as many additional samples as we would like, without knowing s, but with noise coming from a distribution χ' of greater standard deviation. To be even more precise, we will generate samples that are computationally indistinguishable from the desired distribution, even given the secret key s, under the $\mathsf{PLWE}^{(1)}_{f,q,\chi}$ assumption. This is sufficient for all of our purposes since in all scenarios we consider (including KDM security) the randomness used to generate these samples is not revealed to any entity (including the decryptor). The re-randomization lemma follows.

Lemma 4. *Let $f, q, \chi = D_{\mathbb{Z}^n,r}$ be parameters for PLWE and let t be co-prime to q. Let $\chi' = D_{\mathbb{Z}^n,r'}$, with $r' \geq 2^{\omega(\log n)} \cdot r$. Let $s, v, e, e' \xleftarrow{\$} \chi$, $a, a' \xleftarrow{\$} R_q$, $e'' \xleftarrow{\$} \chi'$, $b \doteq as + te$, then under the $\mathsf{PLWE}^{(1)}_{f,q,\chi}$ assumption,*

$$(s, (a, b), (av + te', bv + te'')) \overset{c}{\approx} (s, (a, b), (a', a's + te'')) \ .$$

Proof. Denote $\alpha \doteq av + te'$ and $\beta \doteq bv + te''$. Then it holds that $\beta = (as + te)v + te'' = \alpha s + t(e'' + ev - e's)$. By Lemma 1 and Lemma 2, it holds that $e'' + ev - e's \overset{s}{\approx} e''$. Namely

$$(s, (a, b), (av + te', bv + te'')) \overset{s}{\approx} (s, (a, b), (\alpha, \alpha s + te'')) \ .$$

However, $(s, a, \alpha) = (s, a, av + te') \overset{c}{\approx} (s, a, a')$ by $\mathsf{PLWE}^{(1)}_{f,q,\chi}$ and the result follows.

Therefore, to achieve a public key scheme, the following changes need to be made in our scheme. Let χ, χ' be as above.

1. In the key generation, in addition to the secret key $sk = s \xleftarrow{\$} \chi$, a public key $pk \doteq (a_0, b_0 = a_0 s + te_0)$ is output. Where $a_0 \xleftarrow{\$} R_q$, $e_0 \xleftarrow{\$} \chi$.
2. In the encryption algorithm, instead of using $(a, as + te)$, the encryptor will use $(a_0 v + te', b_0 v + te'')$, where $v, e' \xleftarrow{\$} \chi$ and $e'' \xleftarrow{\$} \chi'$.

As a side note we remark that for all applications except KDM security, it is sufficient to generate $(a, as + te)$ where the error e is not distributed according to the correct error distribution. Generating such "skewed" samples is easier and improves the parameters of the scheme. We omit the details.

4 Key Dependent Message (Circular) Security

We show that our somewhat homomorphic scheme from Section 3 (using a sufficiently large parameter t) is KDM secure w.r.t. linear functions of the secret key, over the ring R_t. We further show that changing just the encryption algorithm, allows for KDM security w.r.t. degree-d polynomials. An interesting interplay between KDM security and somewhat homomorphism allows the key generation and even the decryption circuit to stay unchanged. We note that even though we change the encryption algorithm, the ciphertexts of the resulting scheme can still undergo homomorphic operations (although a little fewer than before). Let us elaborate a bit more about the connection between homomorphism and KDM security.

Assume that we can prove that our basic scheme, as is, is secure w.r.t. linear functions of the secret key, then somewhat homomorphism implies that we can generate encryptions of degree $d \leq D$ polynomials of the secret key, by taking encryptions of the secret key and e.g. multiplying them together to generate quadratic polynomials, multiply by constants, add more terms etc. The above implies a very weak form of KDM: that it is possible to generate secure ciphertexts that decrypt to the right function of the secret key. To show full KDM, we need to present an encryption algorithm that produces indistinguishable ciphertexts whether it encrypts functions of the secret key or the constant message 0. Not surprisingly, we accomplish this by modifying the encryption algorithm to generate ciphertexts that look a lot like the one generated by homomorphism. Specifically, to be secure against degree-d polynomials, our encryption algorithm will generate $(d+1)$-element ciphertexts (contrast this with the encryption algorithm of the somewhat homomorphic encryption scheme in Section 3, that generates ciphertexts with just two non-zero ring elements). Our techniques borrow from a recent work of Malkin et al. [27]. We describe our scheme in the symmetric key setting only, noting that the public key variant applies to here as well. Due to space limitations, we do not provide proofs in this section. The formal definition of KDM security is omitted as well (see e.g. [7]).

We define $\mathcal{P}_d = \mathcal{P}_d[R_t]$ to be the class of all degree d polynomials over R_t, i.e. all functions of the form $p(z) = \sum_{i=0}^{d} \alpha_i z^i$, where $\alpha_i \in R_t$ and arithmetics is over R_t as well.

The Scheme. Let κ denote the security parameter. Our scheme is parameterized by the same parameters as our somewhat homomorphic scheme from Section 3: The primes $q, t \in \mathbb{Z}_q^*$, a degree n polynomial $f(x) \in \mathbb{Z}[x]$, and an error distribution χ over the ring $R_q \doteq \mathbb{Z}_q[x]/\langle f(x) \rangle$. As before, there is the maximal degree parameter D. An additional parameter $d \leq D$ determines the class of functions for which KDM security holds. The message space is R_t.

- $\mathsf{KDM.Keygen}(1^\kappa)$: As we explained above, the key generation is identical to that of our basic scheme $\mathsf{SH.Keygen}$: The secret key is generated as $s \xleftarrow{\$} \chi$. (Note that we will set χ to be a Gaussian distribution with small enough parameter $r \ll t$ such that with all but negligible probability $s \in R_t$)

– $\mathsf{KDM.Enc}(sk, m)$: To encrypt a message $m \in R_t$, we generate the ciphertext $c = (c_0, \ldots, c_d) \in R_q^{d+1}$ as follows.

We generate d pairs $\{(a_i, b_i = a_i s + te_i)\}_{i\in[d]}$, where $a_i \xleftarrow{\$} R_q$, $e_i \xleftarrow{\$} \chi'$ (χ' will be set with noise parameter much larger than χ) and set:

$$c_0 = b_1 + m \ ; \qquad \forall_{i\in[d-1]}.\ c_i = b_{i+1} - a_i \ ; \qquad c_d = -a_d \ .$$

This encryption algorithm coincides with our basic scheme when $d = 1$. While we analyze the correctness of the scheme separately, let us justify our encryption algorithm by noting that for our ciphertext it holds that

$$\begin{aligned}\sum_{i=0}^{d} c_i s^i &= b_1 + m + \sum_{i\in[d-1]} (b_{i+1} - a_i)\, s^i - a_d s^d \\ &= m + \sum_{i\in[d]} b_i s^{i-1} - \sum_{i\in[d]} a_i s^i \\ &= m + \sum_{i\in[d]} (b_i - a_i s) s^{i-1} \\ &= m + t \cdot \sum_{i\in[d]} e_i s^{i-1} \ . \end{aligned} \tag{1}$$

– $\mathsf{KDM.Dec}(sk, \mathbf{c})$: The decryption algorithm is identical to the basic scheme and in fact is able to decrypt ciphertexts that were originally generated by the encryption algorithm of our KDM scheme above, and then underwent somewhat homomorphic operations. Recall that the general form of a decryptable ciphertext is, w.l.o.g, $\mathbf{c} = (c_0, c_1, \ldots, c_D) \in R_q^{D+1}$. To decrypt, we first compute $\langle \mathbf{c}, \mathbf{s} \rangle = \sum_{i=0}^{D} c_i s^i \in R_q$, and output $m = \langle \mathbf{c}, \mathbf{s} \rangle \pmod t$ as the message.

Parameter Setting. Let us now describe a plausible parameter setting for our scheme. As usual, we will set $f(x) = \Phi_m(x)$ (where $n = 2^{\lfloor \log \kappa \rceil - 1}$, and $m = 2n$) and our error distributions will be Gaussian $\chi = D_{\mathbb{Z}^n, r}$, $\chi' = D_{\mathbb{Z}^n, r'}$ for r, r' defined next. Our q needs to be super-polynomial as implied by the selection below.

We set $r = 2^{\omega(\log n)}$ to be a super-polynomial function, and $r' = 2^{\omega(\log n)} \cdot r^d$. We note that the r^d factor is so that χ' can "swallow" degree d polynomials over χ.

The parameter t is chosen so that $t > r\sqrt{n}$ (i.e. a sample from χ resides in R_t with all but negligible probability) and on the other hand, for correctness, $t < 2^{-\omega(\log n)} \cdot r^{-d} \cdot q$.

We conclude that in our parameter setting, the scheme supports KDM functions of degree $d \approx (\log q - \log t - O(1))/\omega(\log n) = \log q / \omega(\log n)$, for some super-logarithmic $\omega(\log n)$. Security will be based on $\mathsf{PLWE}_{\Phi_m, q, r}$.

We next state correctness in light of this parameter setting. (We remark that we can somewhat improve efficiency with a more aggressive parameter setting; we choose to present the concrete setting above for simplicity).

Lemma 5. *Consider the parameters of our scheme as defined above. Then for all $m \in R_t$ it holds that*

$$\Pr[\mathsf{KDM.Dec}(s, \mathsf{KDM.Enc}(s, m)) \neq m] = \mathrm{negl}(\kappa) \ ,$$

where the probability is taken over the choice of s and over the randomness of KDM.Enc.

The $\mathrm{KDM}^{(1)}$-security of our scheme is stated below.

Theorem 4. *Our scheme is $\mathrm{KDM}^{(1)}_{\mathcal{P}_d}$-secure under the $\mathsf{PLWE}_{\Phi_m,q,\chi}$ assumption.*

4.1 $\mathrm{KDM}^{(\nu)}$ Security

We proceed to show that our scheme is $\mathrm{KDM}^{(\nu)}$-secure for any polynomial ν. We use a methodology introduced by [7] and used by all following $\mathrm{KDM}^{(\nu)}$ constructions: The ν secret keys associated with the ν users are simulated by one "real" secret key. The secret key of each specific user is obtained by offsetting the "real" secret key by a known (to the challenger) amount. The offset can be done without knowing the real key and the offset keys look like appropriately generated keys. This enables using the same techniques as for $\mathrm{KDM}^{(1)}$. We present a variant of this argument where the offset is drawn from a distribution that "swallows" the real secret key. A formal statement follows, the proof is omitted.

Towards formally stating our scheme, we introduce an additional distribution and parameter. For the purpose of the proof, we will need to sample our keys from an even narrower distribution as before. We denote this distribution by $\chi^* = D_{\mathbb{Z}_q^n, r^*}$, and require that $r^* = 2^{-\omega(\log n)} \cdot r$. Namely that it is "swallowed" by our "normal" secret key distribution. One can verify that such r^* can be chosen without affecting the other parameters of the scheme.

Theorem 5. *Our scheme is $\mathrm{KDM}^{(\nu)}_{\mathcal{P}_d}$-secure under the $\mathsf{PLWE}_{\Phi_m,q,\chi^*}$ assumption, for any $\nu = \mathrm{poly}(\kappa)$.*

Acknowledgments. We thank Chris Peikert for providing us with a full version of [26], and Nigel Smart and the anonymous CRYPTO reviewers for numerous insightful comments on the draft.

References

1. B. Applebaum. Key-dependent message security: Generic amplification and completeness theorems. In *EUROCRYPT*, 2011. (To appear).
2. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.
3. B. Applebaum, D. Harnik, and Y. Ishai. Semantic security under related-key attacks and applications. To Appear in Innovations in Computer Science (ICS) 2011. http://eprint.iacr.org/2010/544.

4. B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In Gilbert [19], pages 423–444.
5. A. Blum, M. L. Furst, M. J. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In D. R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1993.
6. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Theory of Cryptography - TCC'05*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.
7. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision diffie-hellman. In D. Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2008.
8. Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Rabin [33], pages 1–20.
9. Z. Brakerski, S. Goldwasser, and Y. Kalai. Balck-box circular secure encryption beyond affine functions. In *TCC*, 2011. (To appear).
10. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from standard lwe. 2011. Manuscript.
11. C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *EUROCRYPT*, pages 402–414, 1999.
12. J. D. Cohen and M. J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, pages 372–382. IEEE, 1985.
13. M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Gilbert [19], pages 24–43. Full Version in http://eprint.iacr.org/2009/616.pdf.
14. C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.
15. C. Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In Rabin [33], pages 116–137.
16. C. Gentry and S. Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *EUROCRYPT*, 2011. (To appear).
17. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In C. Dwork, editor, *STOC*, pages 197–206. ACM, 2008.
18. C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 803–815. Springer, 2005.
19. H. Gilbert, editor. *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*. Springer, 2010.
20. S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In A. C.-C. Yao, editor, *ICS*, pages 230–240. Tsinghua University Press, 2010.
21. Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377. ACM, 1982.
22. J. Hoffstein, J. Pipher, and J. H. Silverman. Ntru: A ring-based public key cryptosystem. In J. Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

23. V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.
24. V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In R. Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2008.
25. V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. Swifft: A modest proposal for fft hashing. In K. Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 54–72. Springer, 2008.
26. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In Gilbert [19], pages 1–23. Draft of full version was provided by the authors.
27. T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with kdm security. In *EUROCRYPT*, 2011. (To appear).
28. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.
29. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.
30. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.
31. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In S. Halevi and T. Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 145–166. Springer, 2006.
32. C. Peikert and A. Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In D. S. Johnson and U. Feige, editors, *STOC*, pages 478–487. ACM, 2007.
33. T. Rabin, editor. *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*. Springer, 2010.
34. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *STOC*, pages 84–93. ACM, 2005.
35. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
36. R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–177. Academic Press, 1978.
37. R. Rothblum. Homomorphic encryption: From private-key to public-key. In *TCC*, pages 219–234, 2011.
38. N. P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.
39. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, 2009.
40. W. Stein. *A Brief Introduction to Classical and Adelic Algebraic Number Theory*. 2004.