Ring Switching in BGV-Style Homomorphic Encryption

Craig Gentry Shai Halevi Nigel P. Smart

May 2, 2012

Abstract

BGV-style homomorphic encryption schemes over polynomial rings, rely for their security on rings of very large dimension. This large dimension is needed because of the large modulus-to-noise ratio in the key-switching matrices that are used for the top few levels of the evaluated circuit. However, larger noise (and hence smaller modulus-to-noise ratio) is used in lower levels of the circuit, so from a security standpoint it is permissible to switch to lower-dimension rings. Switching to a smaller ring, when possible, can help speeding up the homomorphic operations for the lower levels of the circuit. However, implementing such ring-switching is nontrivial, since these schemes rely on the ring algebraic structure for their homomorphic properties.

A basic ring-switching operation was introduced by Brakerski, Gentry and Vaikuntanathan, in the context of bootstrapping over polynomial rings of the form $\mathbb{Z}[X]/(X^{2^n}+1)$. In this work we first extend this technique to work over any cyclotomic ring. Then we build on the extended technique and show how it can be used not only for bootstrapping but also during the computation itself, in conjunction with the "packed ciphertext" techniques of Gentry, Halevi and Smart.

Acknowledgments

The first and second authors are supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20202. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

The third author is supported by the European Commission through the ICT Programme under Contract ICT-2007-216676 ECRYPT II and via an ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO, by EPSRC via grant COED–EP/I03126X, and by a Royal Society Wolfson Merit Award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the European Commission or EPSRC.

Contents

1 Introduction

The last year has seen a rapid advance in the state of fully homomorphic encryption; yet despite these advances the existing schemes are still too inefficient for most practical purposes. In this paper we make another step forward in making such schemes practical. In particular we present a technique to reduces the dimension of the ring needed for homomorphic computation of the lower levels of a circuit. Our techniques apply to homomorphic encryption schemes over polynomial rings, such as the scheme of Brakerski et al. [6, 7, 5], as well as the variants due to Lòpez-Alt et al. [15] and Brakerski [4].

The most efficient variants of all these schemes work over polynomial rings of the form $\mathbb{Z}[X]/F(X)$, and in all of them the ring dimension (which is the degree of $F(X)$) must be set high enough to ensure security. Specifically, to be able to handle depth-L circuits these schemes must use key-switching matrices with modulus-to-noise ratio of $2^{\tilde{\Omega}(L\cdot\mathrm{polylog}(\lambda))}$, hence the ring dimension must also be $\tilde{\Omega}(L\cdot\mathrm{polylog}(\lambda))$ (even if we assume that ring-LWE is hard to within fully exponential factors).[1] In practice, the ring dimension for moderately deep circuits can easily be many thousands. For example, to be able to evaluate AES homomorphically, Gentry et al. used in [14] circuits of depth $L \geq 50$, with corresponding ring-dimension of over 50000.

As homomorphic operations proceed, the noise in the ciphertext grows (or the modulus shrinks, if we use the modulus-switching technique from [7, 5]), hence reducing the modulus-to-noise ratio. Consequently, it becomes permissible to start using lower-dimension rings in order to speed up further homomorphic computation. However, in the middle of the computation we already have evaluated ciphertexts over the big ring, and so we need a method for transforming these into small-ring ciphertexts that encrypt the same thing. Such a "ring switching" procedure was described by Brakerski et al. [5], in the context of reducing the ciphertext-size prior to bootstrapping. The procedure in [5], however, is specific to polynomial rings of the form $R_{2^n} = \mathbb{Z}[X]/(X^{2^{n-1}}+1)$, and moreover it is not clear if and how it can be combined with the "packed evaluation" techniques of Gentry et al. [12]. Extending this procedure is the focus of this work.

1.1 Our Contribution

In this work we present two complimentary techniques:

- We first show how to extend the procedure from [5] to any cyclotomic ring $R_m = \mathbb{Z}[X]/\Phi_m(X)$ for a composite m. For $m = u\cdot w$, we show how to break a ciphertext over the big ring R_m into a collection of $u = m/w$ ciphertexts over the smaller ring $R_w = \mathbb{Z}[X]/\Phi_w(X)$, such that the plaintext-polynomial encrypted in the original big-ring ciphertext can be expressed as a simple function of the plaintext-polynomials encrypted in the smaller-ring ciphertexts.

- We then show (under some mild constraints on m, w) how to take a "packed" big-ring ciphertext that contains many plaintext elements in its plaintext slots, and distribute these plaintext elements among the plaintext slots of several small-ring ciphertexts. If the original big-ring ciphertext was "sparse" (i.e., if only few of its plaintext slots were used), then our technique yields just a small number of small-ring ciphertexts, only as many as needed to fit all the used plaintext slots.

This first technique on its own may be useful in the context of bootstrapping, but it is not enough to achieve our goal of reducing the computational overhead by switching to small-ring ciphertexts, since we

[1] The schemes from [5, 4] can replace large rings by using higher-dimension vectors over smaller rings. But their most efficient variants use big rings and low-dimension vectors, since the complexity of their key-switching step is quadratic in the the dimension of these vectors.

still need to show how to perform homomorphic operations on the resulting small-ring ciphertexts. This is achieved by utilizing the second technique.

To demonstrate the usefulness of the second technique, consider the application of homomorphic AES computation [14], where the original big-ring ciphertext contains only 16 plaintext elements (corresponding to the 16 bytes of the AES state). If the small-ring ciphertexts has 16 or more plaintext slots, then we can convert the original big-ring ciphertext into a single small-ring ciphertext containing the same 16 bytes in its slots, then continue the computation on this smaller ciphertext.

1.2 An Overview of the Construction

Our starting point is (a slight adaptation of) the polynomial composition technique of Brakerski et al. [5]. When $m = u \cdot w$ then a polynomial of degree upto $m-1$, $a(X) = \sum_{i=0}^{m-1} a_i X^i$, can be broken into u polynomials of degree upto $w-1$ by splitting the coefficients of a according to their index modulo u. Namely, we denote by $a_{(0)}$ the polynomial of degree upto $(w-1)$ with coefficients $a_0, a_u, a_{2u}, \ldots$, by $a_{(1)}$ we denote the polynomial with coefficients $a_1, a_{1+u}, a_{1+2u}, \ldots$, and in general by $a_{(k)}$ the polynomial with coefficients $a_k, a_{k+u}, a_{k+2u}, \ldots$. Then we have

$$a(X) \;=\; \sum_{k=0}^{u-1}\sum_{j=0}^{w-1} a_{k+uj}X^{k+uj} \;=\; \sum_{k=0}^{u-1} X^k \sum_{j=0}^{w-1} a_{k+uj}X^{uj} \;=\; \sum_{k=0}^{u-1} X^k \sum_{j=0}^{w-1} a_{(k)}(X^u). \tag{1}$$

Following [5], our approach to ring-switching is to break a big-ring ciphertext encrypting a into a collection of small-ring ciphertexts encrypting the $a_{(k)}$'s, then use Equation (1) to express the original plaintext a as a linear combination of the $a_{(k)}$'s.

Using this approach in the non-power-of-two setting is far from trivial, however. Breaking a single big-ring ciphertext into many small-ring parts that look "syntactically correct" seems easy, but the main technical challenge is ensuring that these small-ring parts are indeed valid ciphertexts. To be a valid ciphertext, each part must have low noise, namely its inner product with the unknown secret key must be a low-norm polynomial. This is true of the original big-ring ciphertext, and the inner products over the small ring are related to the big-ring inner product by a formula similar to Equation (1), but these two facts are not enough to conclude that the small-ring inner products indeed yield low-norm polynomials. (In a nutshell, the problem is that we cannot conclude that some polynomials have low norm just from the fact that a certain linear combination of them has low norm.)

To address this issue, we use the "delayed reduction" technique from the full version of [12]. (Similar techniques are described in [9], in the context of proving hardness of ring-LWE in arbitrary cyclotomic rings.) Namely, we begin by lifting the original ciphertext from the big ring $\mathbb{Z}[X]/\Phi_m(X)$ into an even bigger ring $C_m = \mathbb{Z}[X]/(X^m - 1)$, then break it into u ciphertexts in the intermediate ring $C_w = \mathbb{Z}[X]/(X^w - 1)$, and finally reduce these ciphertexts into the small ring $R_w = \mathbb{Z}[X]/\Phi_w(X)$. The reason that this helps is that over the bigger ring C_m, the linear combination from Equation (1) is in fact a "direct sum", in the sense that every coefficient of the left-hand side comes from exactly one of the terms on the right. Thus if the result is a low-norm polynomial then all the summands must also be low-norm polynomials, which is what we need.[2]

A Key-Switching Optimization. One source of inefficiency in the ring-switching procedure of Brakerski et al. [5] is that initially the resulting u small-ring ciphertexts are all dimension-$(u+1)$ vectors over the

[2] In the power-of-two setting considered in [5], the same "direct sum" argument can be applied directly in the big ring R_{2^n}, hence they do not need the "lifting" technique.

small ring (whereas the original ciphertext is a dimension-2 vector over the big ring). Brakerski et al. point out that we can use key-switching/dimension-reduction to convert these high dimension ciphertexts into dimension-2 ciphertexts over the small ring, but processing u ciphertexts of dimension u requires work quadratic in u. Instead, here we describe an alternative procedure that saves a factor of u in running time: Roughly speaking, before breaking the ciphertext into pieces we use key-switching over the big ring to get a ciphertext with respect to a very sparse secret key, then we use the sparsity of this secret key to ensure that all the small-ring ciphertexts already have low dimension over the small ring. This is described in Section 3.1, where we prove that this procedure is secure if ring-LWE [16] is hard in the small ring.

Packed Ciphertexts. As sketched so far, the ring-switching procedure lets us convert a big-ring ciphertext encrypting a polynomial $a \in R_m$ into a collection of u small-ring ciphertexts encrypting the parts $a_{(k)} \in R_w$. However, coming in the middle of homomorphic evaluation, we may need to get small-ring ciphertexts encrypting things other than the $a_{(k)}$'s. Specifically, if the original polynomial a encodes several plaintext elements in its plaintext slots (as in [18, 12]), we may want to get encryption of small-ring polynomials that encode the same elements in their slots.

We note that the plaintext elements encoded in the polynomial $a \in R_m$ are the evaluations $a(\zeta_i)$ where the ζ_i's are primitive m-th roots of unity in some extension field $\mathbb{F}_{2^d}$ (see more details in Section 2.2). Similarly, the plaintext elements encoded in a polynomial $b \in R_w$ are the evaluations $b(\tau_j)$ with the τ_j's are primitive w-th roots of unity. Our goal, then, is to decompose a big-ring ciphertext encrypting a into small-ring ciphertexts encrypting some b_t's, such that for every i there are some t, j for which $b_t(\tau_j) = a(\zeta_i)$.

To that end, we interpret Equation (1) as expressing the value of a at an arbitrary point X as a linear combination of the values of the $a_{(k)}$'s at the point X^u (with coefficients $1, X, X^2, \ldots, X^{u-1}$). Observing that if ζ in an m-th root of unity then $\tau = \zeta^u$ is a w-th root of unity, we thus obtain a method of expressing the values of a in the m-th roots of unity as linear combinations of the values of the $a_{(k)}$'s in the w-th roots of unity. In Lemma 6 in Section 4 we show how to express, under some conditions on m and w, the coefficients of the linear combination from Equation (1) as (low norm) polynomials in the τ_j's. This allows us to compute the encryption of the b_t's that we seek as low-weight linear combination of the encryption of the $a_{(k)}$'s that we obtained before.

2 Notation and Preliminaries

Below we define the various algebraic structures that we need for this work. In this paper we will be utilizing various rings at different points, all will be associated to rings of roots of unity. Below let m, q be arbitrary positive integers. Let $\Phi_m(X)$ denote the m'th cyclotomic polynomial (i.e., $\Phi_m(X) = \prod_{i \in (\mathbb{Z}/mZ)^*}(X - \zeta_m^i)$, where ζ_m is the complex primitive m'th root of unity, $\zeta_m = e^{2\pi i/m}$). Recalling that Φ_m is an integer polynomial, we define the following rings:

$$\begin{array}{lllllll} R_m & = & \mathbb{Z}[X]/\Phi_m(X), & & C_m & = & \mathbb{Z}[X]/(X^m - 1) \\ R_{m,q} & = & \mathbb{Z}[X]/(\Phi_m(X), q), & & C_{m,q} & = & \mathbb{Z}[X]/(X^m - 1, q) \end{array}$$

We will be interested in cyclotomic rings for a composite $m = u \cdot w$.

The size of polynomials. Throughout this work we frequently refer to "low norm polynomials". The norm that we use to measure the size of polynomials is the l_2 norm of their coefficient vectors, i.e. for a polynomial f we set $\mathsf{norm}(f) = \sqrt{\sum f_i^2}$. (Most of our treatment is not very sensitive to the choice of the

particular norm function.) We informally say that a polynomial in $R_{m,q}$ or $C_{m,q}$ has low norm when its norm is much smaller then the parameter q.

The ring constant c_m. We sometime need to switch back and forth between $R_{m,q}$ and $C_{m,q}$ while maintaining "low norm" polynomials. For every integer m there exists a constant c_m that bounds the increase in norm due to reduction modulo $\Phi_m(X)$. Namely, for every polynomial f of degree upto $m-1$ is holds that $\mathsf{norm}(f \bmod \Phi_m) \leq c_m \cdot \mathsf{norm}(f)$.

Heuristically, the constants c_m for the parameters m that we work with is rather small (ranging between 2 and 50 for typical values). But in principle for very smooth m's the constant c_m can be super-polynomial in m. For the rest of the paper we always assume that our parameters are chosen so that $q \gg c_m$, so that we can take "low norm" polynomials in C_m and reduce them modulo Φ_m without increasing the norm too much (relative to q). Note that ring constant c_m is different, but related to, the associated ring constant from [8, 12].

2.1 RLWE-based BGV Cryptosystems

Below and throughout this work we denote by $[z]_q$ the reduction of the integer z modulo the positive integer q into the symmetric interval $(-q/2, q/2)$. In our initial ring-LWE-based BGV cryptosystem, secret keys and ciphertexts are 2-vectors over $R_{m,q}$ for some odd system parameter q, and moreover the secret key has the form $\mathbf{s} = (1, \mathfrak{s})$ where $\mathfrak{s} \in R_m$ is a low-norm polynomial (e.g., with coefficients in $\{-1, 0, 1\}$). The native plaintext space for our initial BGV scheme will be $R_{m,2}$, namely binary polynomials modulo $\Phi_m(X)$. A valid ciphertext $\mathbf{c} = (c_0, c_1) \in (R_{m,q})^2$ that encrypts the plaintext polynomial $a \in R_{m,2}$ with respect to $\mathbf{s} = (1, \mathfrak{s})$ satisfies the equality (over R_m)

$$[\langle \mathbf{c}, \mathbf{s} \rangle]_q = [c_0 + \mathfrak{s} \cdot c_1]_q = a + 2e, \tag{2}$$

for some low-norm polynomial $e \in R_m$. Note that by $[c_0 + \mathfrak{s} \cdot c_1]_q$ we mean reducing each of the coefficients of the polynomial $c_0 + \mathfrak{s} \cdot c_1 \in R_m$ into the interval $(-q/2, q/2)$. Decryption is then just computing $[c_0 + \mathfrak{s} \cdot c_1]_q$, then reducing modulo 2 to recover the plaintext polynomial a.

Throughout the paper we will switch back and forth between different rings. We will maintain the invariant that valid ciphertexts always satisfy Equation (2), but the ring over which this equation is evaluated (specifically the meaning of $\mathfrak{s} \cdot c_1$) will vary. In the input to the ring-switching procedure we will have a ciphertext where that equality holds over R_m, at the end we will have the output ciphertexts for which the equality holds over R_w, and in various intermediate points we will have that equality holding over C_m or C_w.

2.2 Plaintext Arithmetic

Following [18, 5, 12, 13, 14] we consider plaintext polynomials $a \in R_{m,2}$ as encoding vectors of plaintext elements from some finite field $\mathbb{F}_{2^d}$, where d is the order of 2 in the group $(\mathbb{Z}/m\mathbb{Z})^*$. (This implies that d divides $\phi(m)$, and also that $\mathbb{F}_{2^d}$ contains primitive m-th roots of unity.) Denoting $\ell = \phi(m)/d$, we can identify polynomials in $R_{m,2}$ with ℓ-vectors of elements from $\mathbb{F}_{2^d}$. The specific mapping between polynomials and vectors that we use is as follows:

Consider the quotient group $(\mathbb{Z}/m\mathbb{Z})^* / \langle 2 \rangle$ (which has exactly ℓ elements), and fix a specific set of representatives for this quotient group, $T_m = \{t_1, t_2, \ldots, t_\ell\} \subseteq (\mathbb{Z}/m\mathbb{Z})^*$, containing exactly one element from every conjugacy class in $(\mathbb{Z}/m\mathbb{Z})^* / \langle 2 \rangle$. [3] Also fix a specific primitive m-th root of unity $\zeta \in \mathbb{F}_{2^d}$, and

[3] In other words, the sets $T_m, 2T_m, 4T_m, \ldots 2^{d-1}T_m$ are all disjoint, and their union is the entire group $(\mathbb{Z}/m\mathbb{Z})^*$.

we identify each polynomial $a \in R_{m,2}$ with the ℓ-vector consisting of $a(\zeta^t)$ for all $t \in T_m$:

$$a \in R_{m,2} \;\longleftrightarrow\; \left\langle a(\zeta^{t_1}), \ldots, a(\zeta^{t_\ell}) \right\rangle \in (\mathbb{F}_{2^d})^\ell.$$

Showing that this is indeed a one-to-one mapping is a standard exercise. In one direction clearly from a we can compute all the values $a(\zeta^{t_i})$. In the other direction we use the fact that since the coefficients of a are all in the base field $\mathbb{F}_2$ then $a(X^2) = a(X)^2$ for any $X \in \mathbb{F}_{2^d}$. In particular from $a(\zeta^{t_i})$ we can compute $a(\zeta^{2t_i})$, $a(\zeta^{4t_i})$, $a(\zeta^{8t_i})$, and so on. Since T_m is a complete set of representatives for the quotient group $(\mathbb{Z}/m\mathbb{Z})^*/\langle 2\rangle$, then we can get this way the evaluations of $a(\zeta^j)$ for all the indexes $j \in (\mathbb{Z}/m\mathbb{Z})^*$. This gives us the evaluation of a in $\phi(m)$ different points, from which we can interpolate a itself.

We thus view the evaluation of the plaintext polynomial in ζ^{t_j} as the j'th "plaintext slot", and note that arithmetic operations in the ring $R_{m,2}$ act on the plaintext slots in a SIMD manner, namely point-wise adding or multiplying the elements in the slots.

2.3 Breaking Polynomials in Parts

As sketched in the introduction, our approach is rooted at the technique for assembling a high-degree polynomial from low-degree parts by interleaving the coefficients of the parts. Alternatively, we can view this as breaking a high-degree polynomial into small-degree parts. Recall that for a polynomial a of degree upto $m-1$, and for any integer $u < m$, we can break a into u parts of degree less than $w = \lceil m/u \rceil$, denoted $a_{(0)}, \ldots, a_{(u-1)}$, by splitting the coefficients of a according to their index mod u, thus obtaining

$$a(X) \;=\; \sum_{k=0}^{u-1}\sum_{j=0}^{w-1} a_{k+uj}\cdot X^{k+uj} \;=\; \sum_{k=0}^{u-1} X^k \cdot \left(\sum_{j=0}^{w-1} a_{k+uj}\cdot X^{uj}\right) \;=\; \sum_{k=0}^{u-1} X^k \cdot \left(\sum_{j=0}^{w-1} a_{(k)}(X^u)\right).$$

Of particular interest to us will be the case where $m = u \cdot w$, where working with w-degree polynomials that are evaluated at X^u allows us to move between big rings and small rings. The following lemma (whose proof is in Appendix A) will be useful later in the paper.

Lemma 1. *Let m, w be positive integers such that w divides m, and let $u = m/w$. Also let $\Phi_m(X)$, $\Phi_w(X)$ be the m-th and w-th cyclotomic polynomials, respectively.*

a. Consider three polynomials $f(X), g(X), h(X)$ of degree at most $\phi(w) - 1$. If $h(X) \equiv f(X)\cdot g(X) \pmod{\Phi_w(X)}$ then $h(X^u) \equiv f(X^u)\cdot g(X^u) \pmod{\Phi_m(X)}$.

b. Consider three polynomials $f(X), g(X), h(X)$ of degree at most $w - 1$. If $h(X) \equiv f(X)\cdot g(X) \pmod{X^w - 1}$ then $h(X^u) \equiv f(X^u)\cdot g(X^u) \pmod{X^m - 1}$.

3 The Basic Ring-Switching Procedure

Given a big-ring ciphertext $\mathbf{c} \in (R_{m,q})^2$, encrypting a plaintext polynomial $a \in R_{m,2}$ relative to a big-ring secret key $\mathfrak{s} \in R_m$, our goal is roughly to come up with u small-ring ciphertexts $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{u-1} \in (R_{w,q})^2$ with $\mathbf{c}_i$ encrypting the part $a_{(i)} \in R_{w,2}$, all relative to some small ring secret key $\mathfrak{s}' \in R_w$. The basic procedure consists of the following steps:

1. **Key-switch.** We use the BGV key-switching method from [5] to switch into a "low-dimension" secret key, still over the big ring $R_{m,q}$. The "low-dimension" key is $\mathfrak{s}'' \in R_m$, where $\mathfrak{s}''$ has nonzero coefficients only for powers X^i where $i \equiv 0 \pmod{u}$. That is, we have $\mathfrak{s}''_{(0)} = \mathfrak{s}'$ and $\mathfrak{s}''_{(i)} = 0$ for all $i > 0$ (in other words $\mathfrak{s}''(X) = \mathfrak{s}'(X^u)$).

2. **Lift.** Next we lift the resulting ciphertext from the big ring $R_{m,q}$ to the even bigger ring $C_{m,q}$, using the delayed-reduction technique of Gentry et al. [12]. As described in Section 3.2, the new ciphertext encrypts over the bigger ring $C_{m,q}$ a plaintext polynomial a' related to a, still relative to the big-ring secret key $\mathfrak{s}''$.

3. **Break.** Now we can break the bigger-ring ciphertext into a collection of u intermediate-ring ciphertexts (i.e., pairs over $C_{w,q}$), such that the k'th ciphertext is a valid encryption of the k'th part of a' (i.e., $a'_{(k)} \in C_{w,2}$). All these ciphertexts are valid (over $C_{w,q}$) with respect to the small-ring secret key $\mathfrak{s}'$.

4. **Reduce.** Finally we reduce all the intermediate ring ciphertexts modulo $(\Phi_w(X), q)$, thereby getting small ring ciphertexts over $R_{w,q}$, valid relative to $\mathfrak{s}'$.

We observe that the small ring ciphertext that we get this way may not encrypt the parts $a_{(k)}$ of the original polynomial a. Rather, we will show that they encrypt some other polynomials $\tilde{a}_k$, which are defined as $\tilde{a}_k = a'_{(k)} \bmod (\Phi_w, 2)$. We will show, however, that these plaintext polynomials $\tilde{a}_k$ satisfy the same relation to the original plaintext polynomial, namely $a(X) \equiv \sum_k X^k \cdot \tilde{a}_k(X^u) \pmod{\Phi_m, 2}$, which is all we need for our application.

3.1 Switching to a Low-Dimension Key

To enable this transformation, we include in the public key a "key switching matrix", essentially encrypting the old key $\mathfrak{s}$ under the new low-dimension key $\mathfrak{s}''$. Note that using such a low-dimension secret key has security implications (since it severely reduces the dimension of the underlying LWE problem). In our case, however, the whole point of switching to a smaller ring is to get lower dimension, so we do not sacrifice anything new. Indeed, we show below that assuming the hardness of the decision-ring-LWE problem [16] over the ring $R_{w,q}$, the key-switching matrix in the public key is indistinguishable from a uniformly random matrix over $R_{m,q}$ (even for a distinguisher that knows the old secret key $\mathfrak{s}$).

The ring-LWE problem in $R_{w,q}$. We denote the secret-key and error-distributions prescribed in the ring-LWE problem in $R_{w,q}$ by $\mathcal{S}_w$ and $\mathcal{E}_w$, respectively. (E.g., these could be low-variance Gaussians in R_w rounded modulo q, or some distributions involving the dual as in [16].) We also denote the uniform distribution on $R_{w,q}$ by $\mathcal{U}_w$. For a fixed random secret $\mathfrak{s}' \leftarrow \mathcal{S}_w$, the ring-LWE problem in $R_{w,q}$ is given many pairs (γ_i, δ_i) with $\gamma_i \leftarrow \mathcal{U}_w$, to distinguish the cases where the δ_i's are chosen as $\delta_i = \mathfrak{s}' \cdot \gamma_i + \eta_i$ with η_i from the case where they are chosen uniformly at random $\delta_i \leftarrow \mathcal{U}_w$.

The key-switching matrix. Let $\mathfrak{s} \in R_m$ be the old big-ring secret key, and $\mathfrak{s}' \in R_w$ be the small-ring secret-key that we want to switch into (where $\mathfrak{s}'$ was chosen from the secret-key distribution $\mathcal{S}_w$). Define the new big-ring low-dimension key $\mathfrak{s}'' \in R_m$ as the unique polynomial of degree less than m such that $\mathfrak{s}''_{(0)} = \mathfrak{s}'$ and $\mathfrak{s}''_{(k)} = 0$ for all $k > 0$. In other words, $\mathfrak{s}''(X) = \tilde{\mathfrak{s}}(X^u)$, i.e., the coefficients $\mathfrak{s}''_0, \mathfrak{s}''_u, \mathfrak{s}''_{2u}, \ldots$ are exactly the coefficients of $\mathfrak{s}'$, and all the other coefficients of $\mathfrak{s}''$ are zero.

For our key-switching matrix we use the following distribution of "error vectors" in $R_{w,q}$: We first draw independently at random u low-norm polynomials from the ring-LWE error distribution, $\eta_{(k)} \leftarrow \mathcal{E}_w$, then assemble from the $\eta_{(k)}$'s a single error polynomial $\epsilon'(X) = \sum_{k=0}^{u-1} X^k \cdot \eta_{(k)}(X^u)$, and output $\epsilon =$

$\epsilon' \bmod (\Phi_w, q)$. That is, we have the distribution

$$\mathcal{E}_m \;=\; \left\{ \eta_{(0)},\ldots,\eta_{(u-1)} \leftarrow \mathcal{E}_w,\;\; \mathsf{output} \sum_{k=0}^{u-1} X^k \cdot \eta_{(k)}(X^u) \bmod (\Phi_w, q) \right\}$$

Note that ϵ' before the reduction mod (Φ_m, q) has degree smaller than $\phi(w) \cdot u < m$, and its norm-squared is the sum of norm-squared of the $\epsilon_{(k)}$'s. Hence ϵ' is a low-norm polynomial, and the norm of ϵ after the reduction is larger by at most a factor of c_m (c_m is the ring constant for R_m), so ϵ too is a low norm polynomial.[4]

Given the old key $\mathfrak{s} \in R_{m,q}$ and the new $\mathfrak{s}' \in R_{w,q}$, we draw at random $l = \lceil \log q \rceil$ elements from the error distribution $\epsilon_0,\ldots,\epsilon_{l-1} \leftarrow \mathcal{E}_m$, and the columns of our key-switching matrix are the pairs

$$\left\{ (\beta_i, \alpha_i)^t : \alpha_i \leftarrow \mathcal{U}_m,\;\; \beta_i = 2^i\mathfrak{s} - (\mathfrak{s}'' \cdot \alpha_i + 2\epsilon_i) \bmod (\Phi_m, q) \right\},$$

where $\mathcal{U}_m$ is the uniform distribution over the big ring $R_{m,q}$. (Note that even if the secret-key and error distributions over the small ring involce the "dual lattice" as in [16], the β's are still going to be in the big ring, because all their parts $\beta_{(k)}$ are in the small ring.)[5]

Since the errors ϵ_i have low-norm, this is a functional key-switching matrix, as described in [7]. Given an $\mathfrak{s}$-ciphertext $\mathbf{c} = (c_0, c_1)$ we decompose c_1 into its bit representation, thus getting an l-vector of polynomials with 0-1 coefficients. Multiplying that vector by the key-switching matrix and adding c_0 to the first coordinate we get a new ciphertext $\mathbf{c}' = (c'_0, c'_1)$ with respect to the new low-dimension big-ring key $\mathfrak{s}''$. As for security, we prove the following lemma, whose proof is in Appendix A.

Lemma 2. *If the decision ring-LWE problem over the ring $R_{w,q}$ is hard, then the key-switching matrix above is indistinguishable from a uniformly random $2 \times l$ matrix with all the entries drawn independently from $\mathcal{U}_m$. The indistinguishability holds even if the distinguisher gets as input the old secret key $\mathfrak{s} \in R_m$.*

3.2 Lifting to the Bigger Ring $C_{m,q}$

To lift the ciphertexts from the big ring $R_{m,q}$ to the bigger ring $C_{m,q}$, we use the "delayed reduction" technique of Gentry et al. (from the full version of [12]), which builds on the following lemma:

Lemma 3. *([12, Lemma 12]) For any integer m there is an integer polynomial G_m of degree $\leq m-1$, such that $G_m(\alpha) = m$ for every complex primitive m-th root of unity α, and $G_m(\beta) = 0$ for every complex non-primitive m-th root of unity β. Moreover the Euclidean norm of G_m's coefficient vector is $\sqrt{m \cdot \phi(m)}$.*

Denoting $Q_m(X) = (X^m - 1)/\Phi_m(X)$, then $G_m(X) \equiv m \pmod{\Phi_m}$ and $G_m(X) \equiv 0 \pmod{Q_m}$. We can use polynomial Chinese remaindering to construct G_m from its remainders modulo $\Phi_m(X)$ and $Q_m(X)$. Since $G_m(X) \equiv 0 \pmod{Q_m}$ then we can use G_m to "lift" any equality modulo Φ_m to an equality modulo $X^m - 1$. Namely, if we have $f \equiv g \pmod{\Phi_m}$ then we also have $G \cdot f \equiv G \cdot g \pmod{X^m - 1}$. Specifically for the decryption formula, we start from a valid big-ring ciphertext that satisfies the formula $c_0 + c_1 \cdot \mathfrak{s}'' \equiv a + 2e + q\kappa \pmod{\Phi_m}$ (for some low-norm polynomial e and a quotient polynomial κ), then multiply both sides by G_m to obtain

$$(G_m \cdot c_0) + (G_m \cdot c_1) \cdot \mathfrak{s}'' \equiv 2(G_m \cdot e) + (G_m \cdot a) + q(G_m \cdot \kappa) \pmod{X^m - 1}.$$

[4] This argument can be refined to eliminate the dependence on the "smallness" of c_m, see Remark 1 at the end of the section.

[5] We could alternatively use the key-switching variant from [14] where the "matrix" consists of a single column $(\beta, \alpha)^t$), but with respect to a largest modulus $Q \approx q^2 \cdot m$. The proof of security would then depend on the hardness of ring-LWE in $R_{w,Q}$ rather than in $R_{w,q}$.

Assuming that $q \gg m$, the products $G_m \cdot e \bmod (X^m - 1)$ and $G_m \cdot a \bmod (X^m - 1)$ are both low-norm. Thus, denoting $c'_0 = G_m \cdot c_0 \bmod (X^m - 1)$ and $c'_1 = G_m \cdot c_1 \bmod (X^m - 1)$, we get that the ciphertext (c'_0, c'_1) is a valid encryption over the bigger ring C_m of $a' = G_m \cdot a \bmod (X^m - 1, 2)$, relative to the secret key $\mathfrak{s}''$. (We note that upon decryption, one can recover the original plaintext polynomial a, simply by reducing a' modulo $(\Phi_m(X), 2)$, this yields $[m \cdot a]_2 = a$, because $G_m(X) \equiv m \pmod{\Phi_m}$ and m is odd.)

3.3 Breaking The Ciphertext into Parts

After the transformation of the previous step, our ciphertext consists of a pair (c, d) of polynomials in the bigger ring $C_{m,q} = \mathbb{Z}[X]/((X^m - 1), q)$. This ciphertext is valid with respect to the low-dimension secret key $\mathfrak{s}''$ of degree smaller than $\phi(m)$, satisfying $\mathfrak{s}''_{(0)} = \mathfrak{s}' \in R_{w,q}$ and $\mathfrak{s}''_{(1)} = \mathfrak{s}''_{(2)} = \cdots = \mathfrak{s}''_{(u-1)} = 0$, in other words $\mathfrak{s}''(X) = \mathfrak{s}'(X^u)$. Breaking c, d into their parts $c_{(k)}, d_{(k)}$, we then have the following lemma, whose proof is in Appendix A.

Lemma 4. *The polynomials $c_{(k)}$ and $d_{(k)}$ are such that the following equality holds over $\mathbb{Z}[X]$:*

$$\big[c + d \cdot \mathfrak{s}'' \bmod (X^m - 1, q)\big](X) \;=\; \sum_{k=0}^{u-1} X^k \cdot \big[c_{(k)} + d_{(k)} \cdot \mathfrak{s}' \bmod (X^w - 1, q)\big](X^u).$$

(In the above equality, we have on both sides polynomials that are reduced to a lower degree and have their coefficients reduced modulo q, then evaluated at X or X^u.)

Size of Polynomials. Importantly, the sum on the right-hand side of the last equality is a "direct sum", in the sense that the k'th summand has non-zero coefficients only in powers X^i such that $i = k \pmod u$. This means that each coefficient in the sum comes from exactly one of the summands. This, in turn, implies that the norm-squared of the the left-hand side is the sum of norm-squared of the terms on the right-hand side. Hence if the left-hand side has low norm, then also *every summand on the right* must have low norm.

We stress that this "direct sum" argument is the reason why we lift our ciphertext to the bigger ring $C_{m,q}$. This argument does not apply when working modulo Φ_m, thus without lifting we could not have used the fact that the left-hand side has low norm to argue that all the terms on the right have low norm.

Ciphertexts in the intermediate ring $C_{w,q}$. Consider now the u intermediate-ring ciphertexts over $C_{w,q}$:

$$\mathbf{c}_0 = (c_{(0)}, d_{(0)}), \;\; \mathbf{c}_1 = (c_{(1)}, d_{(1)}), \;\; \ldots, \;\; \mathbf{c}_{u-1} = (c_{(u-1)}, d_{(u-1)}).$$

Since the bigger-ring ciphertext (c, d) was a valid encryption of $a' = G_m \cdot a \bmod (X^m - 1, 2)$ over $C_{m,q}$ with respect to secret key $\mathfrak{s}''$, we know that we have $[c + d \cdot \mathfrak{s}'' \bmod (X^m - 1, q)] = 2e' + a'$ for some low-norm error e'. Let us denote $b' = 2e' + a'$. From the equalities above (and the "direct sum" argument), we know that the k'th part of b', namely $b'_{(k)} = 2e'_{(k)} + a'_{(k)}$, is obtained as $b'_{(k)} = [c_{(k)} + d_{(k)} \cdot \mathfrak{s}' \bmod (X^w - 1, q)]$. As $e'_{(k)}$ is a low-norm error term, we conclude that the vectors $\mathbf{c}_k$ are valid encryption of the parts $a'_{(k)}$ over $C_{w,q}$ with respect to secret key $\mathfrak{s}'$. Thus we have shown that valid ciphertexts encrypting the parts $a'_{(k)}$ of a' (over the intermediate ring $C_{w,q}$ with respect to $\mathfrak{s}'$) can be obtained simply by breaking the polynomials c, d into their parts.

3.4 Reducing to the Small Ring $R_{w,q}$

Now that we have valid ciphertext $(c_{(k)}, d_{(k)})$ encrypting the parts $a'_{(k)}$ over the intermediate ring $C_{w,q}$ relative to $\mathfrak{s}'$, it only remains to reduce them into the small ring $R_{w,q}$. We do this simply by reducing each of the element $(c_{(k)}, d_{(k)})$ modulo (Φ_w, q), i.e. we set $\tilde{c}_k = c_{(k)} \bmod (\Phi_w, q)$ and $\tilde{d}_k = d_{(k)} \bmod (\Phi_w, q)$.

Lemma 5. *The ciphertext $(\tilde{c}_k, \tilde{d}_k)$ is an encryption (over $R_{w,q}$) of the plaintext $\tilde{a}_k = a'_{(k)} \bmod (\Phi_w, 2) \in R_{w,2}$.*

Again see Appendix A for the proof.

What are the $\tilde{a}_k$'s? At this point we are done converting the original big-ring ciphertext encrypting $a \in R_{m,2}$ into a collection of valid small-ring ciphertexts encrypting the $\tilde{a}_k$'s. But how are these $\tilde{a}_k$'s related to the original plaintext polynomial a? Ideally we would have liked the $\tilde{a}_k$ to be the parts of a (i.e. $\tilde{a}_k = a_{(k)}$), but this is not necessarily what we get. Still, we show that we can recover the original polynomial a from the $\tilde{a}_k$'s via the same assembly formula,

$$a(X) \;=\; \sum_{k=0}^{u-1} X^k \cdot \tilde{a}_k(X^u) \bmod (\Phi_m, 2).$$

To show that we first observe that on both sides of the equation are 0-1 polynomials of degree less than $\phi(m)$, so to demonstrate equality it is enough to show that they agree when evaluated at $\phi(m)$ different points (from any field of our choice). In particular, we now show that they agree on all the primitive m'th roots of unity over the finite field $\mathbb{F}_{2^d}$. For this we recall the following basic facts:

1. The field $\mathbb{F}_{2^d}$ contains primitive m'th roots of unity, and if $\zeta \in \mathbb{F}_{2^d}$ is a primitive m'th roots of unity then ζ^u is a primitive w'th root of unity.
2. Since $G_m \equiv m \equiv 1 \pmod{\Phi_m, 2}$, then $[G_m \bmod 2](\zeta) = 1$ for every primitive m'th root of unity $\zeta \in \mathbb{F}_{2^d}$. Since $a' = G_m \cdot a \bmod (X^m - 1, 2)$, it then follows that $a'(\zeta) = a(\zeta)$ for every primitive m'th root of unity $\zeta \in \mathbb{F}_{2^d}$.
3. Since $\tilde{a}_k = a'_{(k)} \bmod (\Phi_w, 2)$, then $\tilde{a}_k(\tau) = a'_{(k)}(\tau)$ for every primitive w'th root of unity $\tau \in \mathbb{F}_{2^d}$.

Putting all of these facts together, and using the assembly formula for a' from the parts $a'_{(k)}$, we get for every primitive m'th root of unity $\zeta \in \mathbb{F}_{2^d}$:

$$a(\zeta) \overset{\text{Fact 2}}{=} a'(\zeta) \;=\; \sum_{k=0}^{u-1} \zeta^k \cdot a'_{(k)}(\zeta^u) \overset{\text{Facts 1,3}}{=} \sum_{k=0}^{u-1} \zeta^k \cdot \tilde{a}_k(\zeta^u)$$

Remark 1. *If we use the delayed reduction technique from [12, Appendix E] then we can keep everything relative to $X^m - 1$ and $X^w - 1$ and we do not need to rely on the smallness of the ring constants c_m, c_w. The key-switching matrices will remain modulo Φ_m, however.*

4 Homomorphic Computation in the Small Ring

So far we have shown how to break a big-ring ciphertext, encrypting some big-ring polynomial $a \in R_{m,2}$, into a collection of u small-ring ciphertexts encrypting small-ring polynomials $\tilde{a}_0, \tilde{a}_1, \ldots, \tilde{a}_{u-1} \in R_{w,2}$, that are "related" to the original plaintext polynomial a. Namely a can be constructed as a particular big-ring linear combination of the $\tilde{a}_k$'s, $a(X) = \sum_k X^k \cdot \tilde{a}_k(X^u) \bmod (\Phi_m, 2)$.

This, however, still falls short of our goal of speeding-up homomorphic computation by switching to small-ring ciphertexts. Indeed we have not shown how to use the encryption of the $\tilde{a}_k$'s for further homomorphic computation. Following the narrative of SIMD homomorphic computation from [18, 12, 13, 14], we view the big-ring plaintext polynomial a as an encoding in the big ring of several plaintext elements from the extension field $\mathbb{F}_{2^d}$ (with d the order of 2 in $(\mathbb{Z}/m\mathbb{Z})^*$). We therefore wish to obtain small-ring ciphertexts encrypting small-ring polynomials that encode of the same underlying $\mathbb{F}_{2^d}$ elements.

One potential "algebraic issue" with this goal, is that it may not always be possible to embed $\mathbb{F}_{2^d}$ elements inside small-ring polynomials from $R_{w,2}$. Recall that the extension degree d is determined by the the order of 2 in $(\mathbb{Z}/m\mathbb{Z})^*$. But the order of 2 in $(\mathbb{Z}/w\mathbb{Z})^*$ may be smaller than d, in general it will be some d' that divides d. If $d' < d$ then we can only embed elements of the sub-field $\mathbb{F}_{2^{d'}}$ in small-ring polynomials from $R_{w,2}$, and not the $\mathbb{F}_{2^d}$ elements that we have encoded in the big-ring polynomial a. For most of this section we only consider the special case where the order of 2 in both $(\mathbb{Z}/m\mathbb{Z})^*$ and $(\mathbb{Z}/w\mathbb{Z})^*$ is the same d. We discuss possible extensions to the general case at the end of the section.

Even for the special case where the order of 2 in $(\mathbb{Z}/m\mathbb{Z})^*$ and $(\mathbb{Z}/w\mathbb{Z})^*$ is the same (and hence the "plaintext slots" in the small ring contain elements from the same extension field as those in the big ring), we still need to tackle the issue that big ring polynomials have more plaintext slots than small ring polynomials. Specifically, big-ring polynomials have $\ell_m = \phi(m)/d$ slots, whereas small-ring polynomials only have $\ell_w = \phi(w)/d$ slots. The solution here is simple: we just partition the slots in the original big-ring polynomial a into $\ell_m/\ell_w = \phi(m)/\phi(w)$ groups, each consisting of ℓ_w slots. For each group we then construct a small-ring ciphertext, encrypting a small-ring polynomial that encodes the plaintext slots from that group.

One advantage of this approach is that if the original plaintext polynomial a was "sparsely populated", holding only a few plaintext elements in its slots, then we can reduce the number of small ring ciphertexts that we generate to the bear minimum number needed to hold these few plaintext slots. A good example for this scenario is the computation of the AES circuit in [14]: Since there are only 16 bytes in the AES state, we only use 16 slots in the plaintext polynomial a. In this case, as long as we have at least 16 slots in small-ring polynomials, we can continue working with a single small-ring ciphertext (as opposed to the u ciphertexts that the technique of the previous section gives us).

4.1 Ring-Switching with Plaintext Encoding

Below we describe our method for converting the plaintext encoding between the different rings, for the special case where the order of 2 is the same in $(\mathbb{Z}/m\mathbb{Z})^*$ and $(\mathbb{Z}/w\mathbb{Z})^*$. As explained in Section 2.2, each plaintext slot in the big-ring polynomial is associated with a conjugacy class of 2 in $(\mathbb{Z}/m\mathbb{Z})^*$ (equivalently, an element in the quotient group $\mathcal{Q}_m = (\mathbb{Z}/m\mathbb{Z})^*/\langle 2\rangle$), and similar association holds between plaintext slots in small-ring polynomials and elements of the quotient group $\mathcal{Q}_w = (\mathbb{Z}/w\mathbb{Z})^*/\langle 2\rangle$. We thus begin by relating the structures and representations of these two quotient groups. Below let $T_w = \{t'_1, \ldots, t'_{\ell_w}\} \subseteq (\mathbb{Z}/w\mathbb{Z})^*$ be a representative set for $\mathcal{Q}_w$. i.e., a set containing exactly one element from each conjugacy class in $(\mathbb{Z}/w\mathbb{Z})^*$, ordered arbitrarily.

Clearly, since w divides m then $(\mathbb{Z}/m\mathbb{Z})^*$ consists of $\phi(m)/\phi(w)$ copies of $(\mathbb{Z}/w\mathbb{Z})^*$. That is, $(\mathbb{Z}/m\mathbb{Z})^*$ can be partitioned into $\phi(m)/\phi(w)$ disjoint sets, each of size $\phi(w)$, and each of them congruent modulo w

to $(\mathbb{Z}/w\mathbb{Z})^*$. Moreover, it is easy to see that when the order of 2 is the same in $(\mathbb{Z}/m\mathbb{Z})^*$ and $(\mathbb{Z}/w\mathbb{Z})^*$ then this partitioning can be made to respect the conjugacy classes of 2. Namely for any $t \in (\mathbb{Z}/w\mathbb{Z})^*$, we put $2t \bmod m$ in the same part as t. Such conjugation-respecting partition of $(\mathbb{Z}/m\mathbb{Z})^*$ can be constructed greedily, adding conjugacy classes from $(\mathbb{Z}/m\mathbb{Z})^*$ to the current part until we have a complete copy of $(\mathbb{Z}/w\mathbb{Z})^*$, then proceeding to the next part. Let $S_1, S_2, S_3, \ldots$ be this partition of $(\mathbb{Z}/m\mathbb{Z})^*$, so we have the properties:

- $S_i \cap S_j = \emptyset$ for all $i \neq j$, and $\cup_i S_i = (\mathbb{Z}/m\mathbb{Z})^*$;
- For all i we have $|S_i| = \phi(w)$, and also $S_i \bmod w = \{(s \bmod w) : s \in S_i\} = (\mathbb{Z}/w\mathbb{Z})^*$; and
- For all i we have $2S_i \bmod m = \{(2s \bmod m) \ : \ s \in S_i\} = S_i$.

Given the partition of $(\mathbb{Z}/m\mathbb{Z})^*$ to S_i's and the ordered representative set T_w for $\mathcal{Q}_w$, one way of getting an ordered representative set T_m for $\mathcal{Q}_m$ is to set

$$T_m \ = \ \left\{t \in (\mathbb{Z}/m\mathbb{Z})^* \ : \ \exists\, t' \in T_w \text{ s.t. } t \equiv t' \pmod{w}\right\},$$

obviously this set T_m has exactly one element from each conjugacy class in every part S_i. We can order it, $T_m = \{t_1, t_2, \ldots, t_{\ell_m}\}$, by taking all the elements from one part S_i before taking any of the elements from the next part S_{i+1}, and among the elements from the same part use the ordering of T_w.

Finally, fixing a specific primitive m'th root of unity $\zeta \in \mathbb{F}_{2^d}$ and the particular primitive w'th root of unity $\tau = \zeta^u$, we let the j'th plaintext slot encoded in $a \in R_{m,2}$ be the evaluation $a(\zeta^{t_j}) \in \mathbb{F}_{2^d}$, and similarly the j'th plaintext slot encoded in $a^* \in R_{w,2}$ is the evaluation $a^*(\tau^{t'_j})$. The following lemma, whose proof is in Appendix A, plays an important role in our transformation:

Lemma 6. *Let $m = u \cdot w$ for odd integers u, w, and denote by d the order of 2 in $(\mathbb{Z}/m\mathbb{Z})^*$. Let ζ be a primitive m'th root of unity in $\mathbb{F}_{2^d}$, and denote $\tau = \zeta^u$, so τ is a primitive w'th root of unity.*

Let $S \subset (\mathbb{Z}/w\mathbb{Z})^*$ be a subset satisfying (a) $|S| = \phi(w)$ and $S \bmod w = (\mathbb{Z}/w\mathbb{Z})^*$, and (b) S is closed under multiplication by 2, $S = 2S \bmod m$. Then there exists a polynomial $h \in R_{w,2}$ such that for all $j \in S$, it holds that $h(\tau^j) = \zeta^j$.

We are now ready to show how to convert a big-ring ciphertext $\mathbf{c}$, encrypting some polynomial $a \in R_{m,2}$ into *a single small-ring ciphertext* that encrypt some other $a^* \in R_{w,2}$, such that a^* encodes all the plaintext elements that were encoded in the plaintext slots corresponding to one of the S_i's (i.e., all the slots $T_m \cap S_i$ for some S_i).

We begin by using the transformation from the previous section to construct from $\mathbf{c}$ the collection of u small-ring ciphertexts $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{u-1}$ that encrypt the polynomials $\tilde{a}_0, \tilde{a}_1, \ldots, \tilde{a}_{u-1} \in R_{w,2}$, respectively, where the $\tilde{a}_k$'s are related to the original a via the assembly formula $a(X) = \sum_k X^k \cdot \tilde{a}_k(X^u) \bmod (\Phi_m, 2)$. Considering all of these 0-1 polynomials as members of $\mathbb{F}_{2^d}[X]$, and letting $\zeta \in \mathbb{F}_{2^d}$ be a primitive root of unity (so ζ is a root of $[\Phi_m \bmod 2]$ over $\mathbb{F}_{2^d}$), the assembly formula implies in particular that

$$a(\zeta^j) \ = \ \sum_{k=0}^{u-1} \zeta^{jk} \cdot \tilde{a}_k(\zeta^{ju}) \ = \ \sum_{k=0}^{u-1} \zeta^{jk} \cdot \tilde{a}_k(\tau^j) \quad \text{for every } j \in S_i$$

(where $\tau = \zeta^u$). Observing that S_i satisfies the conditions of Lemma 6, let $h \in R_{w,2}$ be the polynomial satisfying $h(\tau^j) = \zeta^j$ for all $j \in S_i$. Further, let us denote $h_k = (h^k \bmod (\Phi_w, 2)) \in R_{w,2}$. Since for all $j \in S_i$, τ^j is a primitive w'th root of unity (and hence a root of $[\Phi_w \bmod 2]$ over $\mathbb{F}_{2^d}$), then we get

$$h_k(\tau^j) \ = \ h(\tau^j)^k \ = \ \zeta^{jk} \ \text{ for every } j \in S_i.$$

We now set $\mathbf{c}^* = \sum_{k=0}^{u-1} h_k \cdot \mathbf{c}_k \bmod (\Phi_w, q)$, and note that this is a linear combination of the valid ciphertexts $\mathbf{c}_k$ with low-norm coefficients. (The h_k's have low norm because they are 0-1 polynomials.) Using the additive homomorphism of the cryptosystem (over the small ring R_w), this means that $\mathbf{c}^*$ is still a valid small-ring ciphertext, encrypting the polynomial $a^* = \sum_{k=0}^{u-1} h_k \cdot \tilde{a}_k \bmod (\Phi_w, 2) \in R_{w,2}$. Moreover, by our definition of the h_k's we have that for all $j \in T_m \cap S_i$,

$$a^*(\tau^j) \;=\; \sum_{k=0}^{u-1} h_k(\tau^j) \cdot \tilde{a}_k(\tau^j) \;=\; \sum_{k=0}^{u-1} \zeta^{jk} \cdot \tilde{a}_k(\zeta^{ju}) \;=\; a(\zeta^j).$$

Using our encoding conventions from the beginning of this section, this means that the content of the plaintext slots of a^* is exactly the content of the plaintext slots in a corresponding to $T_m \cap S_i$.

Ring-switching for "sparsely populated" ciphertexts. We mentioned that when the original big-ring ciphertext was sparsely populated, we would like to reduce it to only a small number of small-ring ciphertexts, only as many as needed to hold all the plaintext slots that contain real data. If the full slots are not already packed together in one (or a few) of the parts S_i, then we can apply the slot permutation techniques of Gentry et al. [12] to pack them as needed inside the big-ring ciphertext, before breaking it into the small-ring.

4.2 The General Case

The above treatment relies on the the order of 2 in $(\mathbb{Z}/w\mathbb{Z})^*$ and $(\mathbb{Z}/m\mathbb{Z})^*$ being the same d. However, the only part that relies on this fact was Lemma 6, where we needed it in order to prove that the polynomial h is defined over the base field. In the general case this no longer holds, so although we can define the polynomials h_k (and therefore a^*) just as above, all of these polynomials will now have coefficients from the extension field $\mathbb{F}_{2^d}$ rather than 0-1 coefficients.[6] This is unavoidable in general, since we know that we cannot always encode $\mathbb{F}_{2^d}$ elements as polynomials in the small ring $R_{q,2}$.

In principle there is no problem with using plaintext arithmetic over $\mathbb{F}_{2^d}[X]/\Phi_w$ (rather than $R_{w,2} = \mathbb{F}_2[X]/\Phi(w)$).Fixing a representation $\mathbb{F}_{2^d} = \mathbb{F}_2[Y]/G(Y)$, we can represent the plaintext polynomial $A(X) \in \mathbb{F}_{2^d}[X]/\Phi_w(X)$ as a bivariate polynomial $A(X,Y) \in \mathbb{F}_2[X,Y]/(\Phi_w(X), G(Y))$, writing each coefficient from $\mathbb{F}_{2^d}$ as a degree-$(d-1)$ polynomial in Y. This means that A can be written as $A(X,Y) = \sum_{i=0}^{d-1} a_i(X)Y^i$ with the a_i's 0-1 polynomials in $R_{w,2}$. An encryption of a A then consists of d small-ring ciphertexts encrypting the a_i's, and arithmetic operations can be implemented naturally using our basic operations on encryptions of the a_i's. However, this is likely to be quite inefficient, probably even less efficient than keeping everything in the big ring.

We remark that in many settings, even though our plaintext slots can hold elements in $\mathbb{F}_{2^d}$, we really only use them to hold elements from a much smaller sub-field (e.g. bits or $\mathbb{F}_{2^8}$ elements). One could therefore hope that the technique from above could be generalized to map the $\mathbb{F}_{2^d}$ plaintext slots over the big ring into $\mathbb{F}_{2^{d'}}$ slots over the small ring, such that if the content of the slots happened to already belong to the subfield $\mathbb{F}_{2^{d'}}$ then it will be copied intact. Finding such a generalization for every $d'|d$ is an interesting open problem.

For the case where we use the plaintext slots to hold just bits, it turns out that we can use a slight adaptation of the procedure for $d' = d$. In this case, the transformation from above yields an encryption of a polynomial $A(X)$ over $\mathbb{F}_{2^d}$, that contains in its slots whatever we had in the original big-ring polynomial. In particular it means that $A(\tau^k) \in \{0,1\}$ for every k, hence in this case *A must be a 0-1 polynomial*. So after we compute an encryption of A (as a set of d encryptions as above), we can just discard all the ciphertexts except the one corresponding to a_0.

[6] Sometimes it is possible to show that the coefficients are drawn from a smaller extension field.

References

[1] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.

[2] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[3] Sanjeev Arora and Rong Ge. New algorithms for learning in the presence of errors. Manuscript, 2011.

[4] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. Manuscript available at `http://eprint.iacr.org/2012/078`.

[5] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. In *Innovations in Theoretical Computer Science (ITCS'12)*, 2012. Available at `http://eprint.iacr.org/2011/277`.

[6] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, 2011.

[7] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS'11*. IEEE Computer Society, 2011.

[8] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty Computation from Somewhat Homomorphic Encryption. Available at `http://eprint.iacr.org/2011/535`.

[9] Leo Ducas and Alain Durmus. Ring-LWE in Polynomial Rings. To appear in PKC 2012, manuscript availabale from `http://eprint.iacr.org/2012/235`

[10] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 2008.

[11] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.

[12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 446-464, 2012. Full version at `http://eprint.iacr.org/2011/566`, 2012.

[13] Craig Gentry, Shai Halevi, and Nigel P. Smart. Better bootstrapping for fully homomorphic encryption. To appear *PKC 2012*. `http://eprint.iacr.org/2011/680`, 2011.

[14] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. Manuscript, 2012.

[15] Adriana Lòpez-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption In *STOC* 2012.

[16] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, 2010.

[17] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.

[18] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic SIMD operations. Manuscript at http://eprint.iacr.org/2011/133, 2011.

A Proofs of Lemmas

Proof of Lemma 1.

Proof. **a.** Since $h(X) \equiv f(X) \cdot g(X) \pmod{\Phi_w(X)}$ then for every primitive w-th root of unity τ (say, over the complex field) we have $h(\tau) = f(\tau) \cdot g(\tau)$. Let us denote $\tilde{f}(X) = f(X^u) \bmod \Phi_m(X)$, $\tilde{g}(X) = g(X^u) \bmod \Phi_m(X)$, and $\tilde{h}(X) = h(X^u) \bmod \Phi_m(X)$, then for every primitive m-th root of unity ζ we have

$$\tilde{f}(\zeta) \cdot \tilde{g}(\zeta) \;=\; f(\zeta^u) \cdot g(\zeta^u) \;\stackrel{(\star)}{=}\; h(\zeta^u) \;=\; \tilde{h}(\zeta)$$

where the equality $(\star)$ follows since ζ^u is a primitive w-th of unity whenever ζ is a primitive m-th of unity. Since $\tilde{f} \cdot \tilde{g}$ has the same evaluations as $\tilde{h}$ on all the primitive m-th roots of unity then it follows that $\tilde{f} \cdot \tilde{g} \equiv \tilde{h} \pmod{\Phi_m}$, as needed.

b. The proof is identical to Part a, except that we consider all w-th and m-th roots of unity, not just the primitive roots. □

Proof of Lemma 2.

Proof. Our goal is to show that under the hardness of ring-LWE in R_w, it is infeasible to distinguish the case where the β_i's where chosen as prescribed in the scheme from the case where they are uniformly random according to $\mathcal{U}_m$. That is, we show that an adversary $\mathcal{A}$ that given the old secret key $\mathfrak{s}$ and the matrix of (β_i, α_i)'s can distinguish between these two distributions, can be used to solve the ring-LWE problem in the small ring $R_{w,q}$.

The reduction. A ring-LWE distinguisher $\mathcal{B}$ gets $l \cdot u$ pairs $(\gamma_{i,k}, \delta_{i,k})$, for $i = 0, 1, \ldots, l-1$ and $k = 0, 1, \ldots, u-1$, where the $\gamma_{i,k}$'s are uniform in $R_{w,q}$ and the $\delta_{i,k}$'s are either set as $\mathfrak{s}' \cdot \gamma_{i,k} + \eta_{i,k}$, for $\eta \leftarrow \mathcal{E}_w$, or chosen at random $\delta_{i,k} \leftarrow \mathcal{U}_w$. $\mathcal{B}$ begins by choosing an "old secret key" in the big ring $\mathfrak{s} \in R_{m,q}$ (according to whatever distribution the scheme specifies). Then $\mathcal{B}$ assembles the α_i's and β_i's by setting

$$\alpha_i(X) = 2\sum_{k=0}^{u-1} X^k \cdot \gamma_{i,k}(X^u) \bmod (\Phi_m, q) \;\text{ and }\; \beta_i(X) = 2^i \cdot \mathfrak{s} - 2 \cdot \sum_{k=0}^{u-1} X^k \cdot \delta_{i,k}(X^u) \bmod (\Phi_m, q).$$

Finally, $\mathcal{B}$ runs the adversary $\mathcal{A}$ on $\mathfrak{s}$ and the matrix with columns $(\beta_i, \alpha_i)^t$ and outputs whatever $\mathcal{A}$ does.

Analysis. We observe that when we have polynomials $f_0, f_1, \ldots, f_{u-1} \in R_{w,q}$ and we set $g(X) = \sum_{k=0}^{u-1} X^k f_k(X^u) \bmod (\Phi_m, q)$, then the coefficients of g are related to those of the f_k's via a $(\phi(w) \cdot u) \times \phi(m)$ matrix of full rank (i.e., rank $\phi(m)$) over $\mathbb{Z}/q\mathbb{Z}$. When the f_k's are drawn from $\mathcal{U}_w$ then all their coefficients are uniform in $\mathbb{Z}/q\mathbb{Z}$, and therefore so are all the coefficients of g.

Applying this observation to the reduction above, since the $\gamma_{i,k}$'s are uniform in the small ring $R_{w,q}$ then the α_i's are set as twice a uniform element in the big ring $R_{m,q}$, which is also uniform since q is odd.

Similarly, if the $\delta_{i,k}$'s are uniform in $R_{w,q}$ then also the β_i's are uniform in the big ring $R_{m,q}$. On the other hand, if the $\delta_{i,k}$'s are chosen as $\delta_{i,k} = \mathfrak{s}' \cdot \gamma_{i,k} + \eta_{i,k} \bmod (\Phi_w, q)$, with $\eta_{i,k} \leftarrow \mathcal{E}_w$, then we have

$$\begin{aligned}
\beta_i(X) &\equiv 2^i \cdot \mathfrak{s}(X) - 2\sum_{k=0}^{u-1} X^k \cdot \delta_{i,k}(X^u) \\
&= 2^i \cdot \mathfrak{s}(X) - 2 \cdot \sum_{k=0}^{u-1} X^k \cdot \overbrace{\left[(\mathfrak{s}' \cdot \gamma_{i,k} + \eta_{i,k}) \bmod (\Phi_w, q)\right](X^u)}^{\delta_{i,k} \text{ evaluated at } X^u} \\
&\overset{(\star)}{\equiv} 2^i \cdot \mathfrak{s}(X) - 2 \cdot \sum_{k=0}^{u-1} X^k \cdot \overbrace{\left[\mathfrak{s}' \cdot \gamma_{i,k} + \eta_{i,k}\right]}^{\text{no modular reduction}}(X^u) \\
&\equiv 2^i \cdot \mathfrak{s}(X) - \underbrace{\mathfrak{s}'(X^u)}_{\mathfrak{s}''(X)} \cdot \underbrace{2 \cdot \sum_{k=0}^{u-1} X^k \cdot \gamma_{i,k}(X^u)}_{\alpha_i(X)} - \underbrace{2\sum_{k=0}^{u-1} X^k \cdot \eta_{i,k}(X^u)}_{\epsilon_i(X)} \pmod{\Phi_m, q},
\end{aligned}$$

where the equality $(\star)$ follows from Lemma 3 (part a). In this case the α_i's are still uniformly random, but the ϵ_i's are drawn exactly from our error distribution $\mathcal{E}_m$ in the big ring $R_{m,q}$. This completes the proof. □

Proof of Lemma 4.

Proof. Recall that decryption over $C_{m,q}$ calls for computing $z = c + d \cdot \mathfrak{s}'' \bmod (X^m - 1)$, then reducing z modulo q and then modulo 2. Breaking the polynomials c, d and $\mathfrak{s}''$ into parts, we can write:

$$\begin{aligned}
(d \cdot \mathfrak{s}'')(X) &= \sum_{k=0}^{2u-2} \sum_{\substack{i,j \text{ s.t.} \\ i+j=k}} X^k \cdot d_{(i)}(X^u) \cdot \mathfrak{s}''_{(j)}(X^u) \\
&= \sum_{k=0}^{u-1} X^k \cdot \left(\sum_{\substack{i,j \text{ s.t.} \\ i+j=k}} d_{(i)}(X^u) \cdot \mathfrak{s}''_{(j)}(X^u) + \sum_{\substack{i,j \text{ s.t.} \\ i+j=k+u}} X^u \cdot d_{(i)}(X^u) \cdot \mathfrak{s}''_{(j)}(X^u) \right) \\
&\overset{(\star)}{=} \sum_{k=0}^{u-1} X^k \cdot d_{(k)}(X^u) \cdot \mathfrak{s}''_{(0)}(X^u) \;=\; \sum_{k=0}^{u-1} X^k \cdot d_{(k)}(X^u) \cdot \mathfrak{s}'(X^u)
\end{aligned}$$

where the equality $(\star)$ follows since $\mathfrak{s}''_{(j)} = 0$ for $j > 0$ and $d_{(i)} = 0$ for $i \geq u$. This implies also that

$$\begin{aligned}
(c + d \cdot \mathfrak{s}'')(X) &= \sum_{k=0}^{u-1} X^k \cdot c_{(k)}(X^u) + \sum_{k=0}^{u-1} X^k \cdot d_{(k)}(X^u) \cdot \mathfrak{s}'(X^u) \\
&= \sum_{k=0}^{u-1} X^k \cdot \left[c_{(k)} + d_{(k)} \cdot \mathfrak{s}'\right](X^u)
\end{aligned}$$

Recall from Lemma 3 (part b) that whenever we have $h(X) \equiv f(X) \cdot g(X) \pmod{X^w - 1}$ then also $h(X^u) \equiv f(X^u) \cdot g(X^u) \bmod (X^m - 1)$. Hence we have

$$(c + d \cdot \mathfrak{s}'')(X) \equiv \sum_{k=0}^{u-1} X^k \cdot \left[c_{(k)} + d_{(k)} \cdot \mathfrak{s}' \bmod (X^w - 1)\right](X^u) \pmod{X^m - 1},$$

and since the right-hand side of the last equality is a polynomial of degree less than m, then we get the following equality holding over $\mathbb{Z}[X]$:

$$\big[c + d \cdot \mathfrak{s}'' \bmod (X^m - 1, q)\big](X) \;=\; \sum_{k=0}^{u-1} X^k \cdot \big[c_{(k)} + d_{(k)} \cdot \mathfrak{s}' \bmod (X^w - 1, q)\big](X^u).$$

We note that in the above equality, we have on both sides polynomials that are reduced to a lower degree and have their coefficients reduced modulo q, then evaluated at X or X^u. However, once we perform these modular reduction on both sides, then both polynomials have degrees less than m and coefficients smaller than $q/2$ in absolute value, and since they are congruent modulo $((X^m - 1), q)$ then they must be identical. □

Proof of Lemma 5.

Proof. Recall that for all k we have the equality (over $\mathbb{Z}[X]$)

$$c_{(k)} + d_{(k)} \cdot \mathfrak{s}' \bmod (X^w - 1, q) = 2e'_{(k)} + a'_{(k)}$$

for a low-norm error term $e'_{(k)}$. Denoting $b'_{(k)} = 2e'_{(k)} + a'_{(k)}$, we have that $b'_{(k)}$ is a low-norm polynomial in $C_{w,q}$.

Let us now denote $\tilde{b}_k = (b'_{(k)} \bmod \Phi_w)$ (without reduction modulo q). Since the $b'_{(k)}$'s are low-norm then so are the $\tilde{b}_k$ (because reduction modulo Φ_w increases the norm by at most a factor of the ring constant c_w). This means that $\tilde{b}_k$ has norm much smaller than q, so it is already reduced modulo q. In other words, we also have $\tilde{b}_k = b'_{(k)} \bmod (\Phi_w, q)$.

Observe that $\tilde{a}_k = (a'_{(k)} \bmod \Phi_w) + 2 \cdot \mu_k$ for some low-norm μ_k's. The μ_k's have low norm because $\tilde{a}_k$ has low norm (being a 0-1 polynomial) and also $(a'_{(k)} \bmod \Phi_w)$ has low norm (being at most c_w time more than the norm of the 0-1 polynomial $a'_{(k)}$). Next we argue that for all k we have $\tilde{b}_k = 2\tilde{e}_k + \tilde{a}_k$ for a low-norm error terms $\tilde{e}_k \in R_{w,q}$. This follows because

$$\begin{aligned}\tilde{b}_k = (b'_{(k)} \bmod \Phi_w) \;&=\; (2 \cdot e'_{(k)} + a'_{(k)} \bmod \Phi_w) \;=\; (2 \cdot e'_{(k)} \bmod \Phi_w) + (a'_{(k)} \bmod \Phi_w) \\ &= 2 \cdot (e'_{(k)} \bmod \Phi_w) + \tilde{a}_k - 2 \cdot \mu_k \;=\; 2 \cdot \underbrace{((e'_{(k)} \bmod \Phi_w) - \mu_k)}_{\tilde{e}_k} + \tilde{a}_k,\end{aligned}$$

Finally, we obtain:

$$\begin{aligned}(\tilde{c}_k + \tilde{d}_k \cdot \mathfrak{s}' \bmod (\Phi_w, q)) &= (c_{(k)} + d_{(k)} \cdot \mathfrak{s}' \bmod (\Phi_w, q)) \\ &= (b'_{(k)} \bmod (\Phi_w, q)) \;=\; \tilde{b}_k \;=\; 2 \cdot \tilde{e}_k + \tilde{a}_k\end{aligned}$$

In other words, since $\tilde{e}_k$ has low norm then the pair $(\tilde{c}_k, \tilde{d}_k)$ is a valid ciphertext over $R_{w,q}$ with respect to secret key $\mathfrak{s}'$, encrypting the plaintext polynomial $\tilde{a}_k \in R_{w,2}$. □

Proof of Lemma 6.

Proof. Clearly, since $|S| = \phi(w)$ then there exists a unique polynomial h over $\mathbb{F}_{2^d}$ of degree smaller than $\phi(w)$ such that $h(\tau^j) = \zeta^j$ all $j \in S$. It is left to show only that h is a polynomial over the base field,

i.e. with 0-1 coefficients. To show this, note that by definition of h we have $h(\tau^j) = \zeta^j$ for all $j \in S$, and moreover $2j \in S$ whenever $j \in S$ (and hence $h(\tau^{2j}) = \zeta^{2j}$). Thus, we get for all $j \in S$

$$h(\tau^{2j}) \;=\; \zeta^{2j} \;=\; (\zeta^j)^2 = h(\tau^j)^2.$$

Since $S \bmod w = (\mathbb{Z}/w\mathbb{Z})^*$ then the set $\{\tau^j : j \in S\}$ ranges over all the primitive w'th roots of unity in $\mathbb{F}_{2^d}$, so we have $h(\theta^2) = h(\theta)^2$ for *every primitive w'th root of unity θ*. It is a well-known fact that for an arbitrary polynomial $h(X)$ of degree smaller than $\phi(w)$ over $\mathbb{F}_{2^d}$, if $h(\theta^2) = h(\theta)^2$ holds for every primitive w'th root of unity $\theta \in \mathbb{F}_{2^d}$, then h is in fact a polynomial over the base field, i.e. a polynomial with 0-1 coefficients. This conclude the proof. □