Which Ring Based Somewhat Homomorphic Encryption Scheme is Best?

Ana Costache and Nigel P. Smart

Dept. Computer Science,
University of Bristol,
Bristol, UK.
anamaria.costache@bristol.ac.uk,nigel@cs.bris.ac.uk

Abstract. The purpose of this paper is to compare side-by-side the NTRU and BGV schemes in their non-scale invariant (messages in the lower bits), and their scale invariant (message in the upper bits) forms. The scale invariant versions are often called the FV and YASHE schemes. As an additional optimization, we also investigate the affect of modulus reduction on the scale-invariant schemes. We compare the schemes using the "average case" noise analysis presented by Gentry et al. In addition we unify notation and techniques so as to show commonalities between the schemes. We find that the BGV scheme appears to be more efficient for large plaintext moduli, whilst YASHE seems more efficient for small plaintext moduli (although the benefit is not as great as one would have expected).

1 Introduction

Some of the more spectacular advances in implementation improvements for Somewhat Homomorphic Encryption (SHE) schemes have come in the context of the ring based schemes such as BGV [2]. The main improvements here have come through the use of SIMD techniques (first introduced in the context of Gentry's original scheme [6] by Smart and Vercauteren [15], but then extended to the Ring-LWE based schemes by Gentry et al [2]). SIMD techniques in the ring setting allow for a small overall asymptotic overhead in using SHE schemes [7] by exploiting the Galois group to move data between slots. The Galois group can also be used to perform cheap exponentiation via the Frobenius endomorphism [8]. Other improvements in the ring based setting have come from the use of modulus switching to a larger modulus perform key switching [8], the use of scale invariant versions [5, 1], and the use of NTRU to enable key homomorphic schemes [12].

However each paper which analyses the schemes uses a different methodology for deriving parameters, and examining the noise growth. In addition not all papers utilize all optimizations and improvements available. For example papers on the NTRU scheme [4, 12], and its scale invariant version YASHE [1], rarely, if at all, make mention of the use of SIMD techniques. Papers working on scale invariant systems [5, 1] usually focus on plaintext moduli of two, and discount larger moduli. But many applications, e.g. usage in the SPDZ [3] MPC system, require the use of large moduli.

We have therefore conducted a systematic study of the main ring-based SHE schemes with a view to producing a fair comparison over a range of possible application spaces, from low characteristic plaintext spaces through to large characteristic ones, from low depth circuits through to high depth ones. The schemes we have studied are BGV, whose details can be found in [2, 7, 8], and its scale-invariant version [5] (called FV in what follows), the basic NTRU scheme [4, 12], and its scale-invariant version YASHE [1]. A previous study [10] only compared FV and YASHE, restricted to small plaintext spaces (in particular characteristic two), and did not consider the various variants in relation to key switching and modulus switching which we consider.

On the face of it one expects that YASHE should be the most efficient, since it is scale invariant (which often leads to smaller parameters) and a ciphertext consists of only a single ring element, as opposed to two for the BGV style schemes. Yet this initial impression hides a number of details, wherein one can find a number of devils. It turns out that which is the most efficient scheme depends on the context (message characteristic and depth of admissible circuits).

To compare all four schemes fairly we apply the same API to all schemes, and the same optimizations. In particular we also investigate whether applying modulus switching to the scale invariant schemes (where its use is often discounted as being not needed). The use of modlus switching can be beneficial as it means ciphertexts become smaller as the function evaluation proceeds, resulting in increased performance. We also examine two forms of key switching (one based on the traditional decomposition technique and one based on raising the modulus to a larger value). For the decomposition technique we also examine the most efficient modulus to take in the modular decomposition, which turns out not to the two often seen in many treatments.

To compare the schemes we use the average distributional analysis first introduced in [8], which measures the noise in terms of the expected size in the canonical embedding norm. The use of the canonical embedding norm also deviates from some other treatments. For general rings the canonical embedding norm provides a more accurate measure of noise growth, over norms in the polynomial embedding, when analysed over a number of homomorphic operations. The noise growth of all of our schemes is analysed in the same way, and this is the first time (to our knowledge) that all schemes have been analysed on an equal footing.

The first question when performing such a comparison is how to compare security of differing schemes. On one hand one could take the standpoint of an exact security analysis and derive parameter sizes from the security theorems. However, even this is tricky when comparing schemes as the theorems may reduce security of different schemes to different hard problems. So instead we side-step this issue and select parameters according to an analysis of the best known attack on each scheme; which is luckily the same in all four cases. Thus we select parameters according to the Lindner-Peikert analysis [11]. To also afford a fair comparison we use similar distributions for the various parameters for each scheme; e.g. small Hamming weight for the secret key distributions etc.

The next question is how to measure what is "better". In the context of a given specific scheme we consider one set of parameters to be better than another, for a given plaintext modulus, level bound and security parameter, if the number of bits to represent a ring element is minimized. After all this corresponds directly to the computational overhead when implementing the scheme. When comparing schemes one has to be a little more careful, as ciphertexts in the BGV family consist of two ring elements and in the NTRU family they consist of one element, but still ciphertext size is a good crude measure of overall performance.

As one can appreciate much of the analysis is an intricate following through of various inequalities. The full derivations can be found in the Appendice of this paper. We find that the BGV scheme appears to be more efficient for large plaintext moduli, whilst YASHE seems more efficient for small plaintext moduli (although the benefit is not as great as one would have expected).

2 Preliminaries

In this section we outline the basic mathematical background which forms the basis of our four ring-based SHE schemes. Much of what follows can be found in [8], we recap on it here for convenience of the reader. We utilize rings defined by cyclotomic polynomials, $\mathbb{A} = \mathbb{Z}[X]/\Phi_m(X)$. We let $\mathbb{A}_q$ denote the set of elements of this ring reduced modulo various (possibly composite) moduli q. The ring $\mathbb{A}$ is the ring of integers of a the mth cyclotomic number field $K = \mathbb{Q}(\zeta_m)$. We let $[a]_q$ for an element $a \in \mathbb{A}$ denote the reduction of a modulo q, with the set of representatives of coefficients lying in $(-q/2, \ldots, q/2]$, hence $[a]_q \in \mathbb{A}_q$. Assignment of variables will be denoted by $a \leftarrow b$, with equality being denoted by $=$ or $\equiv$.

2.1 Plaintext Slots

We will always use p for the plaintext modulus, and thus plaintexts will be elements of $\mathbb{A}_p$, and the polynomial $\Phi_m(X)$ factors modulo p into ℓ irreducible factors, $\Phi_m(X) = F_1(X) \cdot F_2(X) \cdots F_\ell(X) \pmod{p}$, all of degree $d = \phi(m)/\ell$. Just as in [2, 7, 15, 8] each factor corresponds to a "plaintext slot". That is, we view a polynomial $a \in \mathbb{A}_p$ as representing an ℓ-vector $(a \bmod F_i)_{i=1}^{\ell}$. We assume that p does not divide m so as to enable the slots to exist. In practice p is likely to split completely in $\mathbb{A}$, i.e. $p \equiv 1 \pmod{m}$.

2.2 Canonical Embedding Norm

Following [13], we use as the "size" of a polynomial $a \in \mathbb{A}$ the l_∞ norm of its canonical embedding. Recall that the canonical embedding of $a \in \mathbb{A}$ into $\mathbb{C}^{\phi(m)}$ is the $\phi(m)$-vector of complex numbers $\sigma(a) = (a(\zeta_m^i))_i$ where ζ_m is a complex primitive m-th root of unity and the indexes i range over all of $(\mathbb{Z}/m\mathbb{Z})^*$. We call the norm of $\sigma(a)$ the *canonical embedding norm* of a, and denote it by

$$\|a\|_\infty^{\mathsf{can}} = \|\sigma(a)\|_\infty.$$

We will make use of the following properties of $\|\cdot\|_\infty^{\mathsf{can}}$:

- For all $a, b \in \mathbb{A}$ we have $\|a \cdot b\|_\infty^{\mathsf{can}} \le \|a\|_\infty^{\mathsf{can}} \cdot \|b\|_\infty^{\mathsf{can}}$.
- For all $a \in \mathbb{A}$ we have $\|a\|_\infty^{\mathsf{can}} \le \|a\|_1$.

– There is a ring constant c_m (depending only on m) such that $\|a\|_\infty \le c_m \cdot \|a\|_\infty^{\mathsf{can}}$ for all $a \in \mathbb{A}$.

The ring constant c_m is defined by $c_m = \|\mathsf{CRT}_m^{-1}\|_\infty$ where CRT_m is the CRT matrix for m, i.e. the Vandermonde matrix over the complex primitive m-th roots of unity. Asymptotically the value c_m can grow super-polynomially with m, but for the "small" values of m one would use in practice values of c_m can be evaluated directly. See [3] for a discussion of c_m.

2.3 Sampling From $\mathbb{A}_q$

At various points we will need to sample from $\mathbb{A}_q$ with different distributions, as described below. We denote choosing the element $a \in \mathbb{A}$ according to distribution $\mathcal{D}$ by $a \leftarrow \mathcal{D}$. The distributions below are described as over $\phi(m)$-vectors, but we always consider them as distributions over the ring $\mathbb{A}$, by identifying a polynomial $a \in \mathbb{A}$ with its coefficient vector.

The uniform distribution $\mathcal{U}_q$: This is just the uniform distribution over $(\mathbb{Z}/q\mathbb{Z})^{\phi(m)}$, which we identify with $(\mathbb{Z} \cap (-q/2, q/2])^{\phi(m)}$).

The "discrete Gaussian" $\mathcal{DG}_q(\sigma^2)$: Let $\mathcal{N}(0, \sigma^2)$ denote the normal (Gaussian) distribution on real numbers with zero-mean and variance σ^2, we use drawing from $\mathcal{N}(0, \sigma^2)$ and rounding to the nearest integer as an approximation to the discrete Gaussian distribution. The distribution $\mathcal{DG}_{q_t}(\sigma^2)$ draws a real ϕ-vector according to $\mathcal{N}(0, \sigma^2)^{\phi(m)}$, rounds it to the nearest integer vector, and outputs that integer vector reduced modulo q (into the interval $(-q/2, q/2]$).

Sampling small polynomials, $\mathcal{ZO}(p)$ and $\mathcal{HWT}(h)$: These distributions produce vectors in $\{0, \pm 1\}^{\phi(m)}$.

– For a real parameter $\rho \in [0, 1]$, $\mathcal{ZO}(p)$ draws each entry in the vector from $\{0, \pm 1\}$, with probability $\rho/2$ for each of -1 and $+1$, and probability of being zero $1 - \rho$.
– For an integer parameter $h \le \phi(m)$, the distribution $\mathcal{HWT}(h)$ chooses a vector uniformly at random from $\{0, \pm 1\}^{\phi(m)}$, subject to the conditions that it has exactly h nonzero entries.

2.4 Canonical embedding norm of random polynomials

In the coming sections we will need to bound the canonical embedding norm of polynomials that are produced by the distributions above, as well as products of such polynomials. Following [8] we use a heuristic approach, which we now recap on.

Let $a \in \mathbb{A}$ be a polynomial that was chosen by one of the distributions above, hence all the (nonzero) coefficients in a are independently identically distributed. For a complex primitive m-th root of unity ζ_m, the evaluation $a(\zeta_m)$ is the inner product between the coefficient vector of a and the fixed vector $\mathbf{z}_m = (1, \zeta_m, \zeta_m^2, \ldots)$, which has Euclidean norm exactly $\sqrt{\phi(m)}$. Hence the random variable $a(\zeta_m)$ has variance $V = \sigma^2 \phi(m)$, where σ^2 is the variance of each coefficient of a. Specifically, when $a \leftarrow$

$\mathcal{U}_q$ then each coefficient has variance $q^2/12$, so we get variance $V_U = q^2 \cdot \phi(m)/12$. When $a \leftarrow \mathcal{DG}_q(\sigma^2)$ we get variance $V_G \approx \sigma^2 \cdot \phi(m)$, and when $a \leftarrow \mathcal{ZO}(\rho)$ we get variance $V_Z = \rho \cdot \phi(m)$. When choosing $a \leftarrow \mathcal{HWT}(h)$ we get a variance of $V_H = h$ (but not $\phi(m)$, since a has only h nonzero coefficients).

Moreover, the random variable $a(\zeta_m)$ is a sum of many independent identically distributed random variables, hence by the law of large numbers it is distributed similarly to a complex Gaussian random variable of the specified variance.[1] We therefore use $6\sqrt{V}$ (i.e. six standard deviations) as a high-probability bound on the size of $a(\zeta_m)$. Since the evaluation of a at all the roots of unity obeys the same bound, we use six standard deviations as our bound on the canonical embedding norm of a. (We chose six standard deviations since $\text{erfc}(6) \approx 2^{-55}$, which is good enough for us even when using the union bound and multiplying it by $\phi(m) \approx 2^{16}$.)

In this paper we model all canonical embedding norms as if from a random distribution. In [8] the messages were always given a norm of $\|m\|_\infty^{\mathsf{can}} \leq p \cdot \phi(m)/2$, i.e. a worst case bound. We shall assume that messages, and similar quantities, behave as if selected uniformly at random and hence estimate $\|m\|_\infty^{\mathsf{can}} \leq 6 \cdot p \cdot \sqrt{\phi(m)/12} = p \cdot \sqrt{3 \cdot \phi(m)}$. This makes our bounds better, and does not materially affect the decryption ability due to the larger effect of other terms. However, this simplification makes the formulae somewhat easier to parse.

In many cases we need to bound the canonical embedding norm of a product of two or more such "random polynomials". In this case our task is to bound the magnitude of the product of two random variables, both are distributed close to Gaussians, with variances σ_a^2, σ_b^2, respectively. For this case we use $16 \cdot \sigma_a \cdot \sigma_b$ as our bound, since $\text{erfc}(4) \approx 2^{-25}$, so the probability that both variables exceed their standard deviation by more than a factor of four is roughly 2^{-50}. For a product of three variables we use $40 \cdot \sigma_a \cdot \sigma_b \cdot \sigma_c$, since $\text{erfc}(3.4) \approx 2^{-28}$, and $3.4^3 \approx 40$.

3 Ring Based SHE Schemes

We refer to our four schemes as BGV, FV, NTRU and YASHE. The various schemes have been used/defined in various papers: for example one can find BGV in [2, 7, 8], FV in [5], NTRU in [4, 12] and YASHE in [1]. In all four schemes we shall use a chain of moduli for our homomorphic evaluation[2] by choosing L "small primes" $p_0, p_1, \ldots, p_{L-1}$ and the t^{th} modulus in our chain is defined as $q_t = \prod_{j=0}^{t} p_j$. A chain of L primes allows us to perform $L-1$ multiplications. The primes p_i's are chosen so that for all i, $\mathbb{Z}/p_i\mathbb{Z}$ contains a primitive m-th root of unity, i.e. $p_i \equiv 1 \pmod{m}$. Hence we can use the double-CRT representation, see [8], for all $\mathbb{A}_{q_t}$.

For the BGV and NTRU schemes we additionally assume that $p_i \equiv 1 \pmod{p}$, this is to enable Scaling to work without having to additionally scale by $p_i \pmod{p}$, which would result in slightly more noise growth. A disadvantage of this is that the moduli p_i

[1] The mean of $a(\zeta_m)$ is zero, since the coefficients of a are chosen from a zero-mean distribution.

[2] This is not strictly needed for the Scale invariant version if modulus switching is not performed.

will need to be slightly larger than would otherwise be the case. The two scale invariant schemes (FV and YASHE) will make use of a scaling factor Δ_q defined by

$$\Delta_q = \left\lfloor \frac{q}{p} \right\rfloor = \frac{q}{p} - \epsilon_q,$$

where $0 \leq \epsilon_q < p$.

3.1 Key Generation

$\underline{\mathsf{KeyGen}^{\mathsf{BGV}}()}$: Sample $\mathfrak{sk} \leftarrow \mathcal{HWT}(h)$, $a \leftarrow \mathcal{U}_{q_{L-1}}$, and $e \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$. Then set the secret key as $\mathfrak{sk}$ and the public key as $\mathfrak{pk} \leftarrow (a, b)$ where $b \leftarrow [a \cdot s + p \cdot e]_{q_{L-1}}$.

$\underline{\mathsf{KeyGen}^{\mathsf{FV}}()}$: Sample $\mathfrak{sk} \leftarrow \mathcal{HWT}(h)$, $a \leftarrow \mathcal{U}_{q_{L-1}}$, and $e \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$. Then set the secret key as $\mathfrak{sk}$ and the public key as $\mathfrak{pk} \leftarrow (a, b)$ where $b \leftarrow [a \cdot s + e]_{q_{L-1}}$.

$\underline{\mathsf{KeyGen}^{\mathsf{NTRU}}()}$: Sample $f, g \leftarrow \mathcal{HWT}(h)$. Then set the secret key as $\mathfrak{sk} \leftarrow p \cdot f + 1$ and the public key as $\mathfrak{pk} \leftarrow [p \cdot g/\mathfrak{sk}]_{q_{L-1}}$. Note, if $p \cdot f + 1$ is not invertible in $\mathbb{A}_{q_{L-1}}$ we repeat the sampling again until it is.

$\underline{\mathsf{KeyGen}^{\mathsf{YASHE}}()}$: Sample $f, g \leftarrow \mathcal{HWT}(h)$. Then set the secret key as $\mathfrak{sk} \leftarrow p \cdot f + 1$ and the public key as $\mathfrak{pk} \leftarrow [p \cdot g/\mathfrak{sk}]_{q_{L-1}}$. Again, if $p \cdot f + 1$ is not invertible in $\mathbb{A}_{q_{L-1}}$ we repeat the sampling until it is.

3.2 Encryption and Decryption

The encryption algorithms for all four schemes are given in Fig. 1. The output of each algorithm is a tuple $\mathfrak{c}$ consisting of the ciphertext data, the current level, plus a bound on the current "noise" B^*_{clean}. This bound is on the canonical embedding norm of a particular critical quantity which comes up in the decryption process; a different critical quantity depending on which scheme we are using. If the critical quantity has canonical embedding norm less than a specific value then decryption will work, otherwise decryption will likely fail. Thus having each ciphertext carry around an upper bound on the norm of this quantity allows us to analyse noise growth dynamically.

To understand the critical quantity we have to first look at the decryption procedure in each case. Then we can apply our heuristic noise analysis to obtain an upper bound on the canonical embedding norm of the critical quantity for a fresh ciphertext, and so obtain B^*_{clean}; a process which is done in the Appendix.

$\underline{\mathsf{Dec}^{\mathsf{BGV}}_{\mathfrak{pk}}(\mathfrak{c})}$: Decryption of a ciphertext (c_0, c_1, t, ν) at level t is performed by setting

$$m' \leftarrow [c_0 - \mathfrak{sk} \cdot c_1]_{q_t},$$

and outputting $m' \bmod p$. If we define the critical quantity to be $c_0 - \mathfrak{sk} \cdot c_1 \pmod{q_t}$, then this procedure will work when ν is an upper bound on the canonical embedding norm of this quantity and $c_m \cdot \nu < q_t/2$. If ν satisfies this inequality then the value of

$\underline{\mathsf{Enc}^{\mathsf{BGV}}_{\mathfrak{pk}}(m)}$:
- $v \leftarrow \mathcal{ZO}(0.5)$.
- $e_0, e_1 \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
- $c_0 \leftarrow [b \cdot v + p \cdot e_0 + m]_{q_{L-1}}$,
- $c_1 \leftarrow [a \cdot v + p \cdot e_1]_{q_{L-1}}$,
- Output $\mathfrak{c} \leftarrow (c_0, c_1, L-1, B^{\mathsf{BGV}}_{\mathsf{clean}})$.

$\underline{\mathsf{Enc}^{\mathsf{FV}}_{\mathfrak{pk}}(m)}$:
- $v \leftarrow \mathcal{ZO}(0.5)$.
- $e_0, e_1 \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
- $c_0 \leftarrow [b \cdot v + e_0 + \Delta_{q_{L-1}} \cdot m]_{q_{L-1}}$,
- $c_1 \leftarrow [a \cdot v + e_1]_{q_{L-1}}$,
- Output $\mathfrak{c} \leftarrow (c_0, c_1, L-1, B^{\mathsf{FV}}_{\mathsf{clean}})$.

$\underline{\mathsf{Enc}^{\mathsf{NTRU}}_{\mathfrak{pk}}(m)}$:
- $e_0, e_1 \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
- $c \leftarrow [e_1 \cdot \mathfrak{pk} + p \cdot e_0 + m]_{q_{L-1}}$,
- Output $\mathfrak{c} \leftarrow (c, L-1, B^{\mathsf{NTRU}}_{\mathsf{clean}})$.

$\underline{\mathsf{Enc}^{\mathsf{YASHE}}_{\mathfrak{pk}}(m)}$:
- $e_0, e_1 \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
- $c \leftarrow [e_1 \cdot \mathfrak{pk} + e_0 + \Delta_{q_{L-1}} \cdot m]_{q_{L-1}}$,
- Output $\mathfrak{c} \leftarrow (c, L-1, B^{\mathsf{YASHE}}_{\mathsf{clean}})$.

Fig. 1: Encryption Algorithms for BGV, FV, NTRU and YASHE

$c_0 - \mathfrak{sk} \cdot c_1 \pmod{q_t}$ will be produced exactly with no wrap-around, and will hence be equal to $m + p \cdot v$, if $c_0 = \mathfrak{sk} \cdot c_1 + p \cdot v + m \pmod{q_t}$. Thus we must pick the smallest prime $q_0 = p_0$ large enough to ensure that this always holds.

$\underline{\mathsf{Dec}^{\mathsf{FV}}_{\mathfrak{pk}}(\mathfrak{c})}$: Decryption of a ciphertext (c_0, c_1, t, ν) at level t is performed by setting

$$m' \leftarrow \left\lceil \frac{p}{q_t} \cdot [c_0 - \mathfrak{sk} \cdot c_1]_{q_t} \right\rfloor,$$

and outputting $m' \bmod p$. Consider the value of $[c_0 - \mathfrak{sk} \cdot c_1]_{q_t}$ computed during decryption, suppose this is equal to (over the integers before reduction mod q_t) to $m \cdot \Delta_{q_t} + w + r \cdot q_t$, then another way of looking at decryption is that we perform rounding on the value

$$\begin{aligned} \frac{p \cdot \Delta_{q_t} \cdot m}{q_t} + \frac{p \cdot w}{q_t} + \frac{p \cdot r \cdot q_t}{q_t} &= \frac{p \cdot (\frac{q_t}{p} - \epsilon_{q_t}) \cdot m}{q_t} + \frac{p \cdot w}{q_t} + p \cdot r \\ &= m + p \cdot \frac{w - \epsilon_{q_t} \cdot m}{q_t} + p \cdot r \end{aligned}$$

and then take the result modulo p. Thus the critical quantity in this case is the value of $w - \epsilon_{q_t} \cdot m$. So that the rounding is correct we require that ν is an upper bound on $\left\| w - \epsilon_{q_t} \cdot m \right\|^{\mathsf{can}}_{\infty}$. The decryption procedure will then work when $c_m \cdot \nu < \Delta_{q_t}/2$, since in this case we have

$$\left\| p \cdot \frac{w - \epsilon_{q_t} \cdot m}{q_t} \right\|_{\infty} \leq \frac{c_m \cdot p}{q_t} \cdot \left\| w - \epsilon_{q_t} \cdot m \right\|^{\mathsf{can}}_{\infty} \leq \frac{\Delta_{q_t} \cdot p}{2 \cdot q_t} < \frac{1}{2}.$$

Thus again we must pick the smallest prime $q_0 = p_0$ large enough, to ensure that $c_m \cdot \nu < \Delta_{q_t}/2$.

$\underline{\mathsf{Dec}^{\mathsf{NTRU}}_{\mathfrak{pk}}(\mathfrak{c})}$: Decryption of a ciphertext (c, t, ν) at level t is performed by setting

$$m' \leftarrow [c \cdot \mathfrak{sk}]_{q_t},$$

and outputting $m' \bmod p$. Much as with BGV the critical quantity is $[c \cdot \mathfrak{sk}]_{q_t}$. If ν is an upper bound on the canonical embedding norm of $c \cdot \mathfrak{sk}$, and we have $c = a \cdot \mathfrak{pk} + p \cdot e + m$ modulo q_t, for some values of a and e, then over the integers we have

$$[c \cdot \mathfrak{sk}]_{q_t} = m + p \cdot (a \cdot g + e + f \cdot m) + p^2 \cdot e \cdot f,$$

which will decrypt to m. Thus for decryption to work we require that $c_m \cdot \nu < q_t/2$.

$\underline{\mathsf{Dec}^{\mathsf{YASHE}}_{\mathfrak{pk}}(\mathfrak{c})}$: Decryption of a ciphertext (c, t, ν) at level t is performed by setting

$$m' \leftarrow \left\lceil \frac{p}{q_t} \cdot [c \cdot \mathfrak{sk}]_{q_t} \right\rfloor,$$

and outputting $m' \bmod p$. Following the same reasoning as for the FV scheme, suppose $c \cdot \mathfrak{sk}$ is equal to (again over the integers before reduction mod q_t) $m \cdot \Delta_{q_t} + w + r \cdot q_t$. Then for decryption to work we require ν to be an upper bound on $\|w - \epsilon_{q_t} \cdot m\|_\infty^{\mathsf{can}}$ and $c_m \cdot \nu < q_t/2$.

3.3 Scale

These operations scale a ciphertext, reducing the corresponding level and more importantly scaling the noise. The syntax is $\mathsf{Scale}^*(\mathfrak{c}, t_{out})$ where $\mathfrak{c}$ is at level t_{in} and the output ciphertext is at level t_{out} with $t_{out} \le t_{in}$. The noise is scaled by a factor of approximately $q_{t_{in}}/q_{t_{out}}$, however an additive term of B^*_{scale} is added. For each of our variants see the Appendix for a justification of the proposed method and an estimate on B^*_{scale}.

For use in one of the $\mathsf{SwitchKey}^*$ variants we also use a Scale which takes a ciphertext with respect to modulus Q and produces a ciphertext with respect to modulus q, where $q|Q$. The syntax for this is $\mathsf{Scale}^*(\mathfrak{c}, Q)$; the idea here is that Q is a "temporary" modulus unrelated to the actual level t of the ciphertext, and we aim to reduce Q down to q_t. The former scale function can be defined in terms of the latter via

$\underline{\mathsf{Scale}^*(\mathfrak{c}, t_{out})}$:

- Write $\mathfrak{c} = (c, t, \nu)$.
- $\mathfrak{c}' \leftarrow \mathsf{Scale}^*((c, t_{out}, \nu), q_t)$.
- Output $\mathfrak{c}'$.

The Scale^* function was originally presented in [2] as a form of noise control for the non-scale invariant schemes. However, the use of such a function within the scale invariant schemes can also provide more efficient schemes, as alluded to in [5]. This is due to the modulus one is working with decreases as homomorphic operations are applied. It is also needed for our second key switching variant. We thus present a Scale^* function for all our four schemes in Fig. 2.

$\underline{\mathsf{Scale}^{\mathsf{BGV}}(\mathfrak{c}, Q)}$:
- Write $\mathfrak{c} = ((c_0, c_1), t, \nu)$.
- Fix δ_i such that $\delta_i \equiv -c_i \pmod{P}$ and $\delta_i \equiv 0 \pmod{p}$.
- Write $c_i' \leftarrow (c_i + \delta_i)/P$.
- $\nu' \leftarrow \nu/P + B_{\mathsf{scale}}^{\mathsf{BGV}}$.
- Output $((c_0', c_1'), t, \nu')$.

$\underline{\mathsf{Scale}^{\mathsf{NTRU}}(\mathfrak{c}, Q)}$:
- Write $\mathfrak{c} = (c, t, \nu)$.
- Fix δ such that $\delta \equiv -c \pmod{P}$ and $\delta \equiv 0 \pmod{p}$.
- Write $c' \leftarrow (c + \delta)/P$.
- $\nu' \leftarrow \nu/P + B_{\mathsf{scale}}^{\mathsf{NTRU}}$.
- Output (c', t, ν').

$\underline{\mathsf{Scale}^{\mathsf{FV}}(\mathfrak{c}, Q)}$:
- Write $\mathfrak{c} = ((c_0, c_1), t, \nu)$.
- Fix δ_i such that $\delta_i \equiv -c_i \pmod{P}$.
- Write $c_i' \leftarrow (c_i + \delta_i)/P$.
- $\nu' \leftarrow \nu/P + B_{\mathsf{scale}}^{\mathsf{FV}}$.
- Output $((c_0', c_1'), t, \nu')$.

$\underline{\mathsf{Scale}^{\mathsf{YASHE}}(\mathfrak{c}, Q)}$:
- Write $\mathfrak{c} = (c, t, \nu)$.
- Fix δ such that $\delta \equiv -c \pmod{P}$.
- Write $c' \leftarrow (c + \delta)/P$.
- $\nu' \leftarrow \nu/P + B_{\mathsf{Scale}}^{\mathsf{YASHE}}$.
- Output (c', t, ν').

Fig. 2: Scale Algorithms for BGV, FV, NTRU and YASHE. In all methods $Q = q_t \cdot P$, and for the BGV and NTRU schemes we assume that $P \equiv 1 \pmod{p}$.

3.4 Reduce Level

For all schemes we can define a $\mathsf{ReduceLevel}^*$ operation which reduces a ciphertext level from level t' to level t where $t' \geq t$. For the non-scale invariant schemes when we reduce a level we only perform a scaling (which could be an expensive operation) if the noise is above some global bound B. This is because for small noise we can easily reduce the level by just dropping terms off the modulus, since the modulus is a product of primes. For the scale invariant schemes we actually need to perform a Scale operation since we need to modify the Δ_{q_t} term. See the Appendix for details. In our parameter estimation evaluation we examine the case, for FV and YASHE, of applying modulus switching to reduce levels and not applying it. In the case of not applying it all ciphertexts remain at level $L - 1$, and $\mathsf{ReduceLevel}^*$ becomes a NOP.

3.5 Switch Key

The switch key operation is needed to relinearize after a multiplication, or after the application of a Galois automorphism (see [7] for more details on the later). For all schemes we present two switch key operations:

- One based on decomposition modulo a general modulus T. See [9] for this method explained in the case of the BGV scheme. Unlike prior work we do not take $T = 2$, as we treat T as a parameter to be optimized to achieve the most efficient scheme. Although to ease parameter search we restrict to T being a power of two.
- Our second method is based on the raising the modulus idea from [8], where it was applied to the BGV scheme. Here we adopt a more complex switching operation, and a potentially larger parameter set, but we gain by reducing the size of the switching "matrices".

For each variant we require algorithms $\mathsf{SwitchKeyGen}$ and $\mathsf{SwitchKey}$; the first generates the public switching "matrix", whilst the second performs the actual switch key. In the BGV and FV schemes we perform a general key switch of the underlying decryption equation of the form $d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}' \cdot d_2 \longrightarrow c_0 - \mathfrak{sk} \cdot c_1$. For the NTRU and YASHE schemes the underlying key switch is of the form $c \cdot \mathfrak{sk}' \longrightarrow c' \cdot \mathfrak{sk}$. In Fig. 3 we present the key switching methods for the BGV algorithm. See the Appendix for the methods for the other schemes, plus derivations of upper bounds on the constants $B_{\mathsf{Ks},*} * ()$.

$\underline{\mathsf{SwitchKeyGen}_1^{\mathsf{BGV}}(\mathfrak{sk}', \mathfrak{sk}, T)}$:

- For $i = 0$ to $\lceil \log_T(q_{L-1}) \rceil - 1$ do
  - ⋆ $a_i \leftarrow \mathcal{U}_{q_{L-1}}$.
  - ⋆ $e_i \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
  - ⋆ $b_i \leftarrow [a_i \cdot \mathfrak{sk} + p \cdot e_i + T^i \cdot \mathfrak{sk}']_{q_{L-1}}$.
- $\mathfrak{ksd} \leftarrow (T, \{a_i, b_i\}_{i=0}^{\lceil \log_T q_{L-1} \rceil - 1})$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_1^{\mathsf{BGV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))}$:

- Write d_2 in base T as $d_2 = \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot T^i$.
- $c_0 \leftarrow d_0 + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot b_i \pmod{q_t}$.
- $c_1 \leftarrow d_1 + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot a_i \pmod{q_t}$.
- $\nu' \leftarrow \nu + B_{\mathsf{Ks},1}^{\mathsf{BGV}}(t)$.
- Output $((c_0, c_1), t, \nu')$.

$\underline{\mathsf{SwitchKeyGen}_2^{\mathsf{BGV}}(\mathfrak{sk}', \mathfrak{sk})}$:

- $a \leftarrow \mathcal{U}_{q_{L-1}}$.
- $e \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
- $b \leftarrow [a \cdot \mathfrak{sk} + p \cdot e + P \cdot \mathfrak{sk}']_{q_{L-1} \cdot P}$.
- $\mathfrak{ksd} \leftarrow\leftarrow (a, b)$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_2^{\mathsf{BGV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))}$:

- $c_0 \leftarrow [P \cdot d_0 + b \cdot d_2]_{q_t \cdot P}$.
- $c_1 \leftarrow [P \cdot d_1 + a \cdot d_2]_{q_t \cdot P}$.
- $\nu' \leftarrow P \cdot \nu + B_{\mathsf{Ks},2}^{\mathsf{BGV}}(t)$.
- Output $\mathsf{Scale}^{\mathsf{BGV}}(((c_0, c_1), t, \nu'), q_t \cdot P)$.

Fig. 3: The two variants of Key Switching for BGV.

In the context of BGV the first method requires us to store $\log_T(q_{L-1})$ "encryptions" of $\mathfrak{sk}'$, each of which is an element in $R_{q_{L-1}}^2$. The second method requires us to store a single "encryption" of $P \cdot \mathfrak{sk}'$, but this time as an element in $R_{P \cdot q_{L-1}}^2$. The former will require more space than the latter as soon as

$$\log_2 P < \log_T(q_{L-1}).$$

In terms of noise the output noise of the first method is modified by an additive constant of

$$B_{\mathsf{Ks},1}^{\mathsf{BGV}}(t) = \frac{8}{\sqrt{3}} \cdot p \cdot \lceil \log_T q_t \rceil \cdot \sigma \cdot \phi(m) \cdot T.$$

whilst the output noise of the second method is modified by the additive constant

$$\frac{B_{\mathsf{Ks},2}^{\mathsf{BGV}}(t)}{P} + B_{\mathsf{scale}}^* = \frac{8 \cdot p \cdot q_t \cdot \sigma \cdot \phi(m)}{\sqrt{3} \cdot P} + B_{\mathsf{scale}}^*.$$

As the level decreases this becomes closer and closer to B_{scale}^*, as the P in the denominator will wipe out the numerator term. Thus the noise will grow of the order of

$O(\sqrt{\phi(m)})$ using the second method and as $O(\phi(m))$ using the first method. A similar outcomes arises when comparing the two methods with respect to the other three schemes.

3.6 Addition and Multiplication

We can now turn to presenting the homomorphic addition, for reasons of space we only give the multiplication method in Fig 4. The addition method is immediate and is given in the the Appendix. In all methods the input ciphertexts $\mathfrak{c}_i$ have level t_i, and recall our parameters are such that we can evaluate circuits with multiplicative depth $L-1$. The algebraic expressions for the functions F^{FV} and F^{YASHE} are given in the the Appendix.

$\underline{\mathsf{Mult}^{\mathsf{BGV}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{BGV}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c_{i,0}, c_{i,1}, t, \nu_i)$.
- $d_0 \leftarrow c_{0,0} \cdot c_{1,0}$.
- $d_1 \leftarrow c_{0,0} \cdot c_{1,1} + c_{0,1} \cdot c_{1,0}$.
- $d_2 \leftarrow c_{0,1} \cdot c_{1,1}$.
- $\mathfrak{d} \leftarrow (d_0, d_1, d_2)$.
- $\nu \leftarrow F^{\mathsf{BGV}}(\nu_0, \nu_1) = \nu_0 \cdot \nu_1$.
- $\mathfrak{c} \leftarrow \mathsf{SwitchKey}_*^{\mathsf{BGV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))$.
- $\mathfrak{c} \leftarrow \mathsf{ReduceLevel}^{\mathsf{BGV}}(\mathfrak{c}, t-1)$.

$\underline{\mathsf{Mult}^{\mathsf{NTRU}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{NTRU}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c, t, \nu_i)$.
- $d \leftarrow c_0 \cdot c_1$.
- $\nu \leftarrow F^{\mathsf{NTRU}}(\nu_0, \nu_1) = \nu_0 \cdot \nu_1$.
- $\mathfrak{c} \leftarrow \mathsf{SwitchKey}_*^{\mathsf{NTRU}}(\mathfrak{ksd}, (d, t, \nu))$.
- $\mathfrak{c} \leftarrow \mathsf{ReduceLevel}^{\mathsf{NTRU}}(\mathfrak{c}, t-1)$.

$\underline{\mathsf{Mult}^{\mathsf{FV}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{FV}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c_{i,0}, c_{i,1}, t, \nu_i)$.
- $d_0'' \leftarrow \frac{p}{q_t} \cdot (c_{0,0} \cdot c_{1,0})$.
- $d_1'' \leftarrow \frac{p}{q_t} \cdot (c_{0,0} \cdot c_{1,1} + c_{0,1} \cdot c_{1,0})$.
- $d_2'' \leftarrow \frac{p}{q_t} \cdot (c_{0,1} \cdot c_{1,1})$.
- $d_0' \leftarrow \left\lceil d_0'' \right\rfloor, d_1' \leftarrow \left\lceil d_1'' \right\rfloor, d_2' \leftarrow \left\lceil d_2'' \right\rfloor$.
- $d_0 \leftarrow [d_0']_{q_t}, d_1 \leftarrow [d_1']_{q_t}, d_2 \leftarrow [d_2']_{q_t}$.
- $\mathfrak{d} \leftarrow (d_0, d_1, d_2)$.
- $\nu \leftarrow F^{\mathsf{FV}}(\nu_0, \nu_1)$.
- $\mathfrak{c} \leftarrow \mathsf{SwitchKey}_*^{\mathsf{FV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))$.
- $\mathfrak{c} \leftarrow \mathsf{ReduceLevel}^{\mathsf{FV}}(\mathfrak{c}, t-1)$.

$\underline{\mathsf{Mult}^{\mathsf{YASHE}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{YASHE}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c_i, t, \nu_i)$.
- $d'' \leftarrow \frac{p}{q_t} \cdot (c_0 \cdot c_1)$.
- $d' \leftarrow \left\lceil d'' \right\rfloor$.
- $d \leftarrow [d']_{q_t}$.
- $\nu \leftarrow F^{\mathsf{YASHE}}(\nu_0, \nu_1)$.
- $\mathfrak{c} \leftarrow \mathsf{SwitchKey}_*^{\mathsf{YASHE}}(\mathfrak{ksd}), (d, t, \nu))$.
- $\mathfrak{c} \leftarrow \mathsf{ReduceLevel}^{\mathsf{YASHE}}(\mathfrak{c}, t-1)$.

Fig. 4: The Multiplication Methods for BGV, FV, NTRU and YASHE.

3.7 Security and Parameters

In this section we outline how we select parameters in the case where $\mathsf{ReduceLevel}^*$ is not a NOP operation. An analysis, for the FV and YASHE schemes, where $\mathsf{ReduceLevel}^*$

is a NOP we defer the analysis to the Appendix. We let B denote an upper bound on ν at the output of any $\mathsf{ReduceLevel}^*$ operation. Following [8] we set $B = 2 \cdot B^*_{\mathsf{scale}}$. We assume that operations are performed as follows. We encrypt, perform up to ζ additions, then do a multiplication, then do ζ additions, then do a multiplication and so on, where we assume decryption occurs after a multiplication.

Security: We assume, as a heuristic assumption, that if we set the parameters of the ring and modulus as per the BGV scheme then the other schemes will also be secure. Following the analysis in [8], which itself follows on from the analysis by Lindner and Peikert [11], we therefore have one of two possible lower bounds for $\phi(m)$, for security parameter k

$$\phi(m) \geq \begin{cases} \frac{\log(q_{L-1}/\sigma)\cdot(k+110)}{7.2} & \text{If the first variant of } \mathsf{SwitchKey} \text{ is used,} \\ \\ \frac{\log(P \cdot q_{L-1}/\sigma)\cdot(k+110)}{7.2} & \text{If the second variant of } \mathsf{SwitchKey} \text{ is used.} \end{cases} \quad (1)$$

Note the logs here are natural logarithms.

Bottom Modulus: To ensure decryption correctness at level zero we require that

$$4 \cdot c_m \cdot B^*_{\mathsf{scale}} = 2 \cdot c_m \cdot B < \begin{cases} p_0 & \text{For BGV and NTRU} \\ \\ \left\lfloor \frac{p_0}{p} \right\rfloor & \text{For FV and YASHE.} \end{cases} \quad (2)$$

Top Modulus: At the top level we take as input a ciphertext with noise B^*_{clean}, perform ζ additions to produce a ciphertext with noise $B_1 = \zeta \cdot B^*_{\mathsf{clean}}$. We then perform a multiplication to produce something with noise

$$B_2 = \begin{cases} F^*(B_1, B_1) + B^*_{\mathsf{Ks},1}(L-1) & \text{If the first variant of } \mathsf{SwitchKey} \text{ is used,} \\ \\ F^*(B_1, B_1) + \frac{B^*_{\mathsf{Ks},2}(L-1)}{P} + B^*_{\mathsf{scale}} & \text{If the second variant of } \mathsf{SwitchKey} \text{ is used.} \end{cases}$$

We then scale down a level to obtain something at the next level down. Thus we obtain something with noise bounded by $B_3 = \frac{B_2}{p_{L-1}} + B^*_{\mathsf{scale}}$. We require, for our invariant, $B_3 \leq B = 2 \cdot B^*_{\mathsf{scale}}$. Thus we require,

$$p_{L-1} \geq \frac{B_2}{B^*_{\mathsf{scale}}}. \quad (3)$$

Middle Moduli: A similar argument applies for the middle moduli, but now we start off with a ciphertext with bound $B = 2 \cdot B^*_{\mathsf{scale}}$ as opposed to B^*_{clean}. Thus we form

$$B'(t) = \begin{cases} F^*(\zeta \cdot B, \zeta \cdot B) + B^*_{\mathsf{Ks},1}(t) & \text{First variant of } \mathsf{SwitchKey}, \\ \\ F^*(\zeta \cdot B, \zeta \cdot B) + \frac{B^*_{\mathsf{Ks},2}(t)}{P} + B^*_{\mathsf{scale}} & \text{Second variant of } \mathsf{SwitchKey}. \end{cases}$$

after which a Scale operation is performed. Hence, the modulus p_t for $t \neq 0, L-1$ needs to be selected so that

$$p_t \geq \frac{B'(t)}{B^*_{\mathsf{scale}}}. \tag{4}$$

Note, in practice we can do a bit better in the second variant of $\mathsf{SwitchKey}$ by merging the final two final scalings into one.

Putting It All Together: We are looking for parameters which satisfy equations (1), (2), (3) and (4), and which also minimize the size of data being processed, which is

$$\phi(m) \cdot \left(\sum_{t=0}^{L-1} p_t \right).$$

To do this we iterate through all possible values of $\log_2 q_{L-1}$ and $\log_2 T$ (resp. $\log_2 P$). We then determine $\phi(m)$, as the smallest value which satisfies equation (1). Here, we might need to take a larger value than the right hand side of equation (1) due to application requirements on p or the amount of packing required.

We then determine the size of p_{L-1} from equation (3), via

$$p_{L-1} \approx \left\lceil \frac{B_2}{B^*_{\mathsf{scale}}} \right\rceil.$$

We can now iterate downwards for $t = L-2, \ldots, 1$ by determining the size of $\log_2 q_t$, via

$$\log_2 q_t = \log_2 q_{t+1} - \log_2 p_{t+1}.$$

If we obtain $\log_2 q_t < 0$ then we abort, and pass to the next pair of $(\log_2 q_{L-1}, T)$ (resp. $(\log_2 q_{L-1}, \log_2 P)$) values. The value of p_t being determined by equation (4), via

$$p_t \approx \left\lceil \frac{B'(t)}{B^*_{\mathsf{scale}}} \right\rceil.$$

Finally we check whether a prime p_0 the size of $\log_2 q_0$, will satisify equation (2), if so we accept this set of values as a valid set of parameters, otherwise we pass to the next pair of $(\log_2 q_{L-1}, T)$ (resp. $(\log_2 q_{L-1}, \log_2 P)$) values.

4 Results

In the Appendix one can find a full set of parameters for each scheme, and variant of key switching, for various values of the plaintext modulus p and the number of levels L. In this section we summarize the overall conclusion. As a measure of efficiency we examine the size of a ciphertext in kBytes; this is a very crude measure but it will capture both the size of any data needed to be transmitted as well as the computational cost of dealing with a single ciphertext element within a calculation. In the Appendix we also examine the size of the associated key switching matrices, which is significantly smaller for the case of our second key switching method. In a given application this

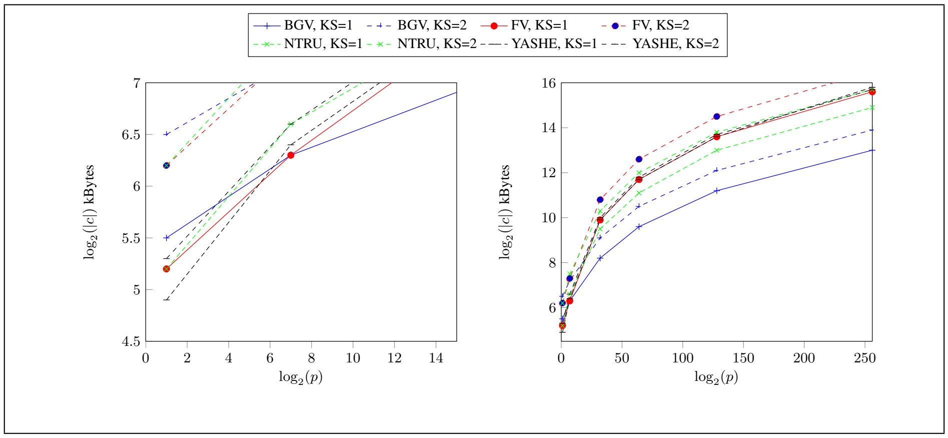

Fig. 5: Size of required ciphertext for various sizes of plaintext modulus when $L = 5$. The graph on the left zooms into the portion of the right graph for small values of $\log_2 p$.

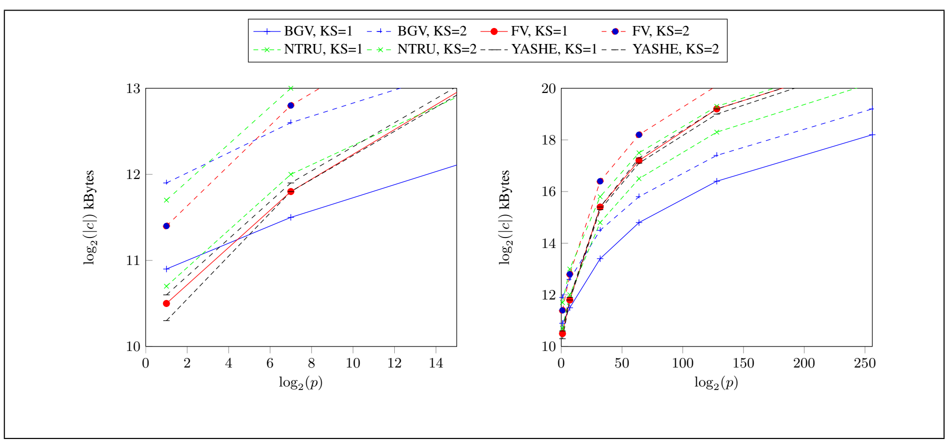

Fig. 6: Size of required ciphertext for various sizes of plaintext modulus when $L = 30$. The graph on the left zooms into the portion of the right graph for small values of $\log_2 p$.

additional cost of holding key switching data may impact on the overall choices, but for this section we ignore this fact.

For all schemes we used a Hamming weight of $h = 64$ to generate the secret key data, we used a security level of $k = 80$ bits of security, a standard deviation of $\sigma = 3.2$ for the discrete Gaussians, a tolerance factor of $\zeta = 8$ and a ring constant of $c_m = 1.3$. These are all consistent with the prior estimates for parameters given in [8]. The use of a small ring constant can be justified by either selecting $\phi(m)$ to be a power of two, or selecting m to be prime, as explained in [3]. As a general conclusion we find that for FV and YASHE the use of modulus switching to lower levels results in slightly bigger parameters to start for large values of L; approximately a factor of two for $L = 20$ or 30. But as a homomorphic calculation progresses this benefit will drop away, leaving, for most calculations, the variant in which modulus switching is applied the most efficient. Thus in what follows we assume that modulus switching is applied in all schemes.

Firstly examine the graphs in Figures 5 and 6. We see that for a fixed number of levels and very small plaintext moduli the most efficient scheme seems to be YASHE. However, quite rapidly, as the plaintext modulus increases the BGV scheme quickly outperforms all other schemes. In particular for the important case of the SPDZ MPC system [3] which requires an SHE scheme supporting circuits of multiplicative depth one, i.e. $L = 2$, for a large plaintext modulus p, the BGV scheme is seen to be the most efficient.

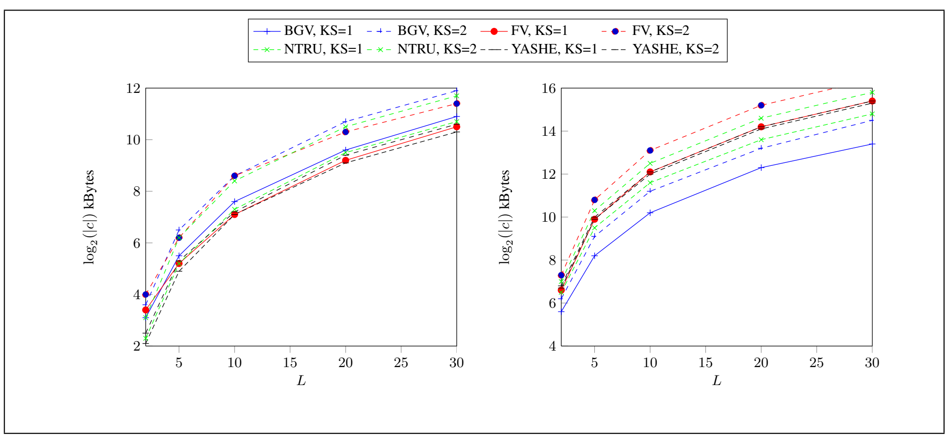

Fig. 7: Size of required ciphertext for various values of L when $p = 2$ and $p \approx 2^{32}$.

Examining Fig. 7 we see that if we fix the prime and just increase the number of levels then the choice of which is the better scheme is be very consistent. Thus one is led to conclude that the main choice of which scheme to adopt depends on the plaintext modulus, where one selects YASHE for very small plaintext moduli and BGV for larger plaintext moduli.

Acknowledgements

This work has been supported in part by an ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO and by the European Union's H2020 Programme under grant agreement number ICT-644209.

References

1. Joppe W. Bos, Kristin E. Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.
2. 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`.
3. Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Safavi-Naini and Canetti [14], pages 643–662.
4. Yarkin Doröz, Yin Hu, and Berk Sunar. Homomorphic AES evaluation using the modified LTV scheme. *Des. Codes and Cryptography*, XXXX:XXXX–XXXX, 2015.
5. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.
6. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.
7. Craig Gentry, Shai Halevi, and Nigel Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2012. Full version at `http://eprint.iacr.org/2011/566`.
8. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Safavi-Naini and Canetti [14], pages 850–867.
9. Kristin Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *CCSW*, pages 113–124. ACM, 2011.
10. Tancrède Lepoint and Michael Naehrig. A comparison of the homomorphic encryption schemes FV and YASHE. In David Pointcheval and Damien Vergnaud, editors, *Progress in Cryptology - AFRICACRYPT 2014 - 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28-30, 2014. Proceedings*, volume 8469 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2014.
11. 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.
12. Adriana Lòpez-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*. ACM, 2012.
13. 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.
14. Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012.
15. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic SIMD operations. *Des. Codes Cryptography*, 71(1):57–81, 2014.

A Estimating B^*_{clean}

$\underline{B^{\mathsf{BGV}}_{\mathsf{clean}}}$: The initial value of ν for a fresh ciphertext is $B^{\mathsf{BGV}}_{\mathsf{clean}}$, where our invariant is that ν is an upper bound on the canonical embedding norm of the value $c_0 - \mathfrak{sk} \cdot c_1 \pmod{q_t}$. We have, using our above estimates for bounding the norm of random variables, for a fresh ciphertext,

$$\begin{aligned}
\|c_0 - \mathfrak{sk} \cdot c_1\|_\infty^{\mathsf{can}} &= \|((a \cdot s + p \cdot e) \cdot v + p \cdot e_0 + m - (a \cdot v + p \cdot e_1) \cdot \mathfrak{sk}\|_\infty^{\mathsf{can}} \\
&= \|m + p \cdot (e \cdot v + e_0 - e_1 \cdot \mathfrak{sk})\|_\infty^{\mathsf{can}} \\
&\leq \|m\|_\infty^{\mathsf{can}} + p \cdot \left(\|e \cdot v\|_\infty^{\mathsf{can}} + \|e_0\|_\infty^{\mathsf{can}} + \|e_1 \cdot \mathfrak{sk}\|_\infty^{\mathsf{can}}\right) \\
&\leq p \cdot \Big(\sqrt{3 \cdot \phi(m)} + \frac{16 \cdot \sigma \cdot \phi(m)}{\sqrt{2}} \\
&\qquad + 6 \cdot \sigma \cdot \sqrt{\phi(m)} + 16 \cdot \sigma \cdot \sqrt{h \cdot \phi(m)}\Big) \\
&= B^{\mathsf{BGV}}_{\mathsf{clean}}.
\end{aligned}$$

$\underline{B^{\mathsf{FV}}_{\mathsf{clean}}}$: For a fresh ciphertext we need to upperbound the canonnical embedding of $w - \epsilon_{q_t} \cdot m$, namely $v \cdot e + e_0 + e_1 \cdot \mathfrak{sk} - \epsilon_{q_{L-1}} \cdot m$. We have

$$\begin{aligned}
\|w - \epsilon_{q_{L-1}} \cdot m\|_\infty^{\mathsf{can}} &\leq p \cdot \|m\|_\infty^{\mathsf{can}} + \|v \cdot e\|_\infty^{\mathsf{can}} + \|e_0\|_\infty^{\mathsf{can}} + \|e_1 \cdot \mathfrak{sk}\|_\infty^{\mathsf{can}} \\
&\leq p^2 \cdot \sqrt{3 \cdot \phi(m)} \\
&\qquad + \Big(\frac{16 \cdot \sigma \cdot \phi(m)}{\sqrt{2}} + 6 \cdot \sigma \cdot \sqrt{\phi(m)} + 16 \cdot \sigma \cdot \sqrt{h \cdot \phi(m)}\Big) \\
&\leq p^2 \cdot \sqrt{3 \cdot \phi(m)} \\
&\qquad + 2 \cdot \sigma \cdot \Big(\frac{8 \cdot \phi(m)}{\sqrt{2}} + 3 \cdot \sqrt{\phi(m)} + 8 \cdot \sqrt{h \cdot \phi(m)}\Big) \\
&= B^{\mathsf{FV}}_{\mathsf{clean}}.
\end{aligned}$$

Note, compared to the $B^{\mathsf{BGV}}_{\mathsf{clean}}$ we do not have a dependence on p in the latter terms, but we have a squared dependence on p in the first term. Hence, for small p we are likely to have $B^{\mathsf{FV}}_{\mathsf{clean}} < B^{\mathsf{BGV}}_{\mathsf{clean}}$.

$\underline{B^{\mathsf{NTRU}}_{\mathsf{clean}}}$: For a fresh ciphertext we have, assuming $\mathfrak{pk}$ is distributed as a uniformly random element in A_{q_t},

$$\begin{aligned}
\|c \cdot \mathfrak{sk}\|_\infty^{\mathsf{can}} &= \|e_1 \cdot \mathfrak{pk} \cdot \mathfrak{sk} + (p \cdot e_0 + m) \cdot (1 + p \cdot f)\|_\infty^{\mathsf{can}} \\
&\leq p \cdot \|e_1 \cdot g\|_\infty^{\mathsf{can}} + p \cdot \|e_0\|_\infty^{\mathsf{can}} + \|m\|_\infty^{\mathsf{can}} \\
&\qquad + p^2 \cdot \|e_0 \cdot f\|_\infty^{\mathsf{can}} + p \cdot \|m \cdot f\|_\infty^{\mathsf{can}} \\
&\leq 16 \cdot p \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} + 6 \cdot p \cdot \sigma \cdot \sqrt{\phi(m)} + p \cdot \sqrt{3 \cdot \phi(m)} \\
&\qquad + 16 \cdot p^2 \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} + 16 \cdot p^2 \cdot \sqrt{h \cdot \phi(m)/12}
\end{aligned}$$

$$
\begin{aligned}
&= \left(16 \cdot p \cdot (1+p) \cdot \sigma + \frac{8}{\sqrt{3}} \cdot p^2\right) \cdot \sqrt{h \cdot \phi(m)} \\
&\qquad\qquad + p \cdot (6 \cdot \sigma + \sqrt{3}) \cdot \sqrt{\phi(m)} \\
&= B_{\mathsf{clean}}^{\mathsf{NTRU}}.
\end{aligned}
$$

$B_{\mathsf{clean}}^{\mathsf{YASHE}}$: For a fresh ciphertext we have that

$$
\begin{aligned}
w - \epsilon_{q_{L-1}} \cdot m &= (e_1 \cdot \mathfrak{pk} + e_0) \cdot \mathfrak{sk} - \epsilon_{q_{L-1}} \cdot m \\
&= e_1 \cdot p \cdot g + e_0 \cdot \mathfrak{sk} - \epsilon_{q_{L-1}} \cdot m.
\end{aligned}
$$

Hence, we have

$$
\begin{aligned}
\left\|w - \epsilon_{q_{L-1}} \cdot m\right\|_\infty^{\mathsf{can}} &\le p \cdot \left\|m\right\|_\infty^{\mathsf{can}} + p \cdot \left\|e_1 \cdot g\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|e_0 \cdot (1 + p \cdot f)\right\|_\infty^{\mathsf{can}} \\
&\le p^2 \cdot \sqrt{3 \cdot \phi(m)} + p \cdot 16 \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} \\
&\qquad + \left\|e_0\right\|_\infty^{\mathsf{can}} + p \cdot \left\|e_0 \cdot f\right\|_\infty^{\mathsf{can}} \\
&\le p^2 \cdot \sqrt{3 \cdot \phi(m)} + 16 \cdot p \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} \\
&\qquad + 6 \cdot \sigma \cdot \sqrt{\phi(m)} + 16 \cdot p \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} \\
&= (6 \cdot \sigma + p^2 \cdot \sqrt{3}) \cdot \sqrt{\phi(m)} + 32 \cdot p \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} \\
&= B_{\mathsf{clean}}^{\mathsf{YASHE}}.
\end{aligned}
$$

B Estimating B_{scale}^{*}

$\mathsf{Scale}^{\mathsf{BGV}}(\mathfrak{c}, Q)$: For correctness of the method presented we appeal to the proof of Lemma 13 in the full version of [7]. Basically the idea is that we have that $c_0 - \mathfrak{sk} \cdot c_1 = m + p \cdot v + Q \cdot u$. Now adding on $\delta_0 - \mathfrak{sk} \cdot \delta_1$ to both sides makes no difference modulo p, since $\delta_i \equiv 0 \pmod{p}$. In addition it makes the left hand side divisible exactly by P over the integers. When dividing by P we do not affect the output modulo p, since $P \equiv 1 \pmod{p}$.

If we let (τ_0, τ_1) denote the rounding error $\tau_i = c_i' - c_i/P = \delta_i/P$, then the coefficients of τ_i will behave as if they are drawn from a uniform distribution modulo p. We then have that

$$
\begin{aligned}
\left\|c_0' - \mathfrak{sk} \cdot c_1'\right\|_\infty^{\mathsf{can}} &= \left\|\frac{1}{P} \cdot \Big(c_0 - \mathfrak{sk} \cdot c_1 + \delta_0 - \mathfrak{sk} \cdot \delta_1\Big)\right\|_\infty^{\mathsf{can}} \\
&\le \frac{\nu}{P} + \left\|\tau_0 - \mathfrak{sk} \cdot \tau_1\right\|_\infty^{\mathsf{can}} \\
&\le \frac{\nu}{P} + \left\|\tau_0\right\|_\infty^{\mathsf{can}} + \left\|\mathfrak{sk} \cdot \tau_1\right\|_\infty^{\mathsf{can}}.
\end{aligned}
$$

Thus we set

$$
B_{\mathsf{scale}}^{\mathsf{BGV}} = 6 \cdot p \cdot \sqrt{\phi(m)/12} + 16 \cdot p \cdot \sqrt{\phi(m) \cdot h/12}
$$

$$= p \cdot \Big(\sqrt{3 \cdot \phi(m)} + 8 \cdot \sqrt{\phi(m) \cdot h/3} \Big) .$$

$\underline{\mathsf{Scale}^{\mathsf{FV}}(\mathfrak{c}, Q)}$: We assume that $Q = q_t \cdot P$, note we make no assumption on P. To show correctness we suppose $\mathfrak{c}$ decrypts correctly modulo Q, i.e. if $\mathfrak{c} = ((c_0, c_1), t, \nu)$ then

$$c_0 - \mathfrak{sk} \cdot c_1 = m \cdot \Delta_Q + w + r \cdot Q$$

where

$$\Delta_Q = \Big\lfloor \frac{Q}{p} \Big\rfloor = \frac{Q}{p} - \epsilon_Q = \frac{q_t \cdot P}{p} - \epsilon_Q = P \cdot (\Delta_{q_t} + \epsilon_{q_t}) - \epsilon_Q$$

and

$$\|w - \epsilon_Q \cdot m\|_\infty^{\mathsf{can}} \le \nu.$$

The output ciphertext satisfies

$$\begin{aligned} c_0' - \mathfrak{sk} \cdot c_1' &= \frac{1}{P} \cdot \Big(c_0 + \delta_0 - \mathfrak{sk} \cdot c_1 - \mathfrak{sk} \cdot \delta_1 \Big) \\ &= \frac{1}{P} \cdot \Big(m \cdot \Delta_Q + w + r \cdot q_t \cdot P + \delta_0 - \mathfrak{sk} \cdot \delta_1 \Big) \\ &= \Delta_{q_t} \cdot m + r \cdot q_t + \epsilon_{q_t} \cdot m + \frac{1}{P} \Big(- \epsilon_Q \cdot m + w + \delta_0 - \mathfrak{sk} \cdot \delta_1 \Big) \\ &= \Delta_{q_t} \cdot m + r \cdot q_t + w' \end{aligned}$$

As the left hand side is exactly divisible by P, and hence so must the right hand side be. To bound the noise of the output ciphertext we need to bound

$$\begin{aligned} \|w' - \epsilon_{q_t} \cdot m\|_\infty^{\mathsf{can}} &= \Big\| \epsilon_{q_t} \cdot m + \frac{1}{P} \Big(- \epsilon_Q \cdot m + w + \delta_0 - \mathfrak{sk} \cdot \delta_1 \Big) - \epsilon_{q_t} \cdot m \Big\|_\infty^{\mathsf{can}} \\ &= \frac{1}{P} \cdot \|w - \epsilon_Q \cdot m + \delta_0 - \mathfrak{sk} \cdot \delta_1 \Big) \|_\infty^{\mathsf{can}} \\ &\le \frac{1}{P} \cdot \Big(\nu + \|\delta_0\|_\infty^{\mathsf{can}} + \|\mathfrak{sk} \cdot \delta_1\|_\infty^{\mathsf{can}} \Big) \\ &\le \frac{1}{P} \cdot \Big(\nu + P \cdot \sqrt{3 \cdot \phi(m)} + 16 \cdot P \cdot \sqrt{h \cdot \phi(m)/12} \Big). \end{aligned}$$

Thus

$$B_{\mathsf{scale}}^{\mathsf{FV}} = \sqrt{3 \cdot \phi(m)} + 8 \cdot \sqrt{h \cdot \phi(m)/3}.$$

$\underline{\mathsf{Scale}^{\mathsf{NTRU}}(\mathfrak{c}, Q)}$: For showing correctness we note that we have $c \cdot \mathfrak{sk} = m + p \cdot v + Q \cdot u$. Adding $\delta \cdot \mathfrak{sk}$ to both sides make no difference to the value modulo p, as $\delta \equiv 0 \pmod p$, in addition it makes the left hand side divisible by P. When dividing by P we do not affect $m \pmod p$ since $P \equiv 1 \pmod P$.

All that remains is to establish the value of $B_{\mathsf{scale}}^{\mathsf{NTRU}}$. We let τ denote the rounding error $\tau = c' - c/P = \delta/P$. The coefficients of τ will act like they are drawn from a uniform distribution modulo p, since the coefficients of δ are in the range $[-p \cdot P/2, \ldots, p \cdot P/2]$. We then have that

$$\|c' \cdot \mathfrak{sk}\|_\infty^{\mathsf{can}} = \Big\| \frac{1}{P} \cdot (c \cdot \mathfrak{sk} + \delta \cdot \mathfrak{sk}) \Big\|_\infty^{\mathsf{can}}$$

$$
\begin{aligned}
&\leq \frac{\nu}{P} + \left\|\tau \cdot \mathfrak{sk}\right\|_\infty^{\mathsf{can}} \\
&= \frac{\nu}{P} + \left\|\tau \cdot (1 + p \cdot f)\right\|_\infty^{\mathsf{can}} \\
&\leq \frac{\nu}{P} + \left\|\tau\right\|_\infty^{\mathsf{can}} + \left\|p \cdot \tau \cdot f\right\|_\infty^{\mathsf{can}} \\
&\leq \frac{\nu}{P} + 6 \cdot p \cdot \sqrt{\phi(m)/12} + 16 \cdot p^2 \cdot \sqrt{h \cdot \phi(m)/12} \\
&= \frac{\nu}{P} + p \cdot \sqrt{3 \cdot \phi(m)} + \frac{8}{\sqrt{3}} \cdot p^2 \cdot \sqrt{h \cdot \phi(m)} \\
&= \frac{\nu}{P} + B_{\mathsf{scale}}^{\mathsf{NTRU}}.
\end{aligned}
$$

$\underline{\mathsf{Scale}^{\mathsf{YASHE}}(\mathfrak{c}, Q)}$: To show correctness we assume that $\mathfrak{c} = (c, t, \nu)$ decrypts correctly modulo Q, i.e. we have $c \cdot \mathfrak{sk} = m \cdot \Delta_Q + w + r \cdot Q$, where Δ_Q is as above and

$$
\left\|w - \epsilon_Q \cdot m\right\|_\infty^{\mathsf{can}} \leq \nu.
$$

We then have that

$$
\begin{aligned}
c' \cdot \mathfrak{sk} &= \frac{1}{P} \cdot (c \cdot \mathfrak{sk} + \delta \cdot \mathfrak{sk}) \\
&= \frac{1}{P} \cdot (m \cdot \Delta_Q + w + r \cdot Q + \delta \cdot \mathfrak{sk}) \\
&= m \cdot \Delta_{q_t} + r \cdot q_t + m \cdot \epsilon_{q_t} + \frac{1}{P} \cdot (w + \delta \cdot \mathfrak{sk} - m \cdot \epsilon_Q) \\
&= m \cdot \Delta_{q_t} + r \cdot q_t + w'.
\end{aligned}
$$

To bound the noise of the output ciphertext we need to bound

$$
\begin{aligned}
\left\|w' - m \cdot \epsilon_{q_t}\right\|_\infty^{\mathsf{can}} &\leq \left\|m \cdot \epsilon_{q_t} + \frac{1}{P}(w + \delta \cdot \mathfrak{sk} - m \cdot \epsilon_Q) - m \cdot \epsilon_{q_t}\right\|_\infty^{\mathsf{can}} \\
&\leq \frac{1}{P} \cdot \left\|w + \delta \cdot \mathfrak{sk} - m \cdot \epsilon_Q\right\|_\infty^{\mathsf{can}} \\
&\leq \frac{1}{P} \cdot \left(\nu + \left\|\delta \cdot \mathfrak{sk}\right\|_\infty^{\mathsf{can}}\right) \\
&\leq \frac{1}{P} \cdot \left(\nu + \left\|\delta \cdot (1 + pf)\right\|_\infty^{\mathsf{can}}\right) \\
&\leq \frac{1}{P} \cdot \left(\nu + \left\|\delta\right\|_\infty^{\mathsf{can}} + p \cdot \left\|\delta \cdot f\right\|_\infty^{\mathsf{can}}\right) \\
&\leq \frac{1}{P} \cdot \left(\nu + P \cdot \sqrt{3 \cdot \phi(m)} + 16 \cdot P \cdot p\sqrt{h \cdot \phi(m)/12}\right) \\
&= \frac{\nu}{P} + \left(\sqrt{3 \cdot \phi(m)} + \frac{8}{\sqrt{3}} \cdot p \cdot \sqrt{h \cdot \phi(m)}\right) \\
&= \frac{\nu}{P} + B_{\mathsf{Scale}}^{\mathsf{YASHE}},
\end{aligned}
$$

on letting $B_{\mathsf{Scale}}^{\mathsf{YASHE}} = \sqrt{3 \cdot \phi(m)} + \frac{8}{\sqrt{3}} \cdot p \cdot \sqrt{h \cdot \phi(m)}$.

C Reduce Level

The ReduceLevel* operations for our four schemes are presented in Fig. 8.

$\underline{\mathsf{ReduceLevel}^{\mathsf{BGV}}(((c_0', c_1'), t', \nu), t)}$:
- If $t' \le t$ then abort.
- If $\nu > B$ then
  - $\star$ $\mathfrak{c} \leftarrow \mathsf{Scale}^{\mathsf{BGV}}(((c_0', c_1'), t', \nu), t)$
- Else
  - $\star$ $c_0 \leftarrow c_0' \pmod{q_t}$.
  - $\star$ $c_1 \leftarrow c_1' \pmod{q_t}$.
  - $\star$ $\mathfrak{c} \leftarrow ((c_0, c_1), t, \nu)$.
- Return $\mathfrak{c}$.

$\underline{\mathsf{ReduceLevel}^{\mathsf{FV}}(((c_0', c_1'), t', \nu), t)}$:
- If $t' \le t$ then abort.
- $\mathfrak{c} \leftarrow \mathsf{Scale}^{\mathsf{FV}}(((c_0', c_1'), t', \nu), t)$
- Return $\mathfrak{c}$.

$\underline{\mathsf{ReduceLevel}^{\mathsf{NTRU}}((c', t', \nu), t)}$:
- If $t' \le t$ then abort.
- If $\nu > B$ then
  - $\star$ $\mathfrak{c} \leftarrow \mathsf{Scale}^{\mathsf{NTRU}}((c', t', \nu), t)$
- Else
  - $\star$ $c \leftarrow c' \pmod{q_t}$.
  - $\star$ $\mathfrak{c} \leftarrow (c, t, \nu)$.
- Return $\mathfrak{c}$.

$\underline{\mathsf{ReduceLevel}^{\mathsf{YASHE}}((c', t', \nu), t)}$:
- If $t' \le t$ then abort.
- $\mathfrak{c} \leftarrow \mathsf{Scale}^{\mathsf{YASHE}}((c', t', \nu), t)$
- Return $\mathfrak{c}$.

Fig. 8: The ReduceLevel* Operations for BGV, FV, NTRU and YASHE.

D Switch Key

D.1 BGV

In each of the variants we switch from a key $\mathfrak{sk}'$ to a key $\mathfrak{sk}$. The input ciphertext will involve both keys; thus we aim a switch of the form

$$d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}' \cdot d_2 \longrightarrow c_0 - \mathfrak{sk} \cdot c_1.$$

For ease of reference we recap on the algorithms in Fig. 9.

SwitchKey First Variant: This is the bit-decomposition method generalised for an arbitrary decomposition modulus T. We first establish that the output ciphertext encrypts the same message as the input ciphertext.

$$\begin{aligned} c_0 - \mathfrak{sk} \cdot c_1 &= d_0 + \left(\sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot b_i \right) - d_1 \cdot \mathfrak{sk} - \mathfrak{sk} \cdot \left(\sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot a_i \right) \\ &= d_0 - d_1 \cdot \mathfrak{sk} + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} (d_{2,i} \cdot b_i - d_{2,i} \cdot a_i \cdot \mathfrak{sk}) \end{aligned}$$

$\underline{\mathsf{SwitchKeyGen}_1^{\mathsf{BGV}}(\mathfrak{sk}', \mathfrak{sk}, T)}$:
- For $i = 0$ to $\lceil \log_T(q_{L-1}) \rceil - 1$ do
  - ⋆ $a_i \leftarrow \mathcal{U}_{q_{L-1}}$.
  - ⋆ $e_i \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
  - ⋆ $b_i \leftarrow [a_i \cdot \mathfrak{sk} + p \cdot e_i + T^i \cdot \mathfrak{sk}']_{q_{L-1}}$.
- $\mathfrak{ksd} \leftarrow (T, \{a_i, b_i\}_{i=0}^{\lceil \log_T q_{L-1} \rceil - 1})$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_1^{\mathsf{BGV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))}$:
- Write d_2 in base T as $d_2 = \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot T^i$.
- $c_0 \leftarrow d_0 + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot b_i \pmod{q_t}$.
- $c_1 \leftarrow d_1 + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot a_i \pmod{q_t}$.
- $\nu' \leftarrow \nu + B_{\mathsf{Ks},1}^{\mathsf{BGV}}(t)$.
- Output $((c_0, c_1), t, \nu')$.

$\underline{\mathsf{SwitchKeyGen}_2^{\mathsf{BGV}}(\mathfrak{sk}', \mathfrak{sk})}$:
- $a \leftarrow \mathcal{U}_{q_{L-1}}$.
- $e \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
- $b \leftarrow [a \cdot \mathfrak{sk} + p \cdot e + P \cdot \mathfrak{sk}']_{q_{L-1} \cdot P}$.
- $\mathfrak{ksd} \leftarrow (a, b)$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_2^{\mathsf{BGV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))}$:
- $c_0 \leftarrow [P \cdot d_0 + b \cdot d_2]_{q_t \cdot P}$.
- $c_1 \leftarrow [P \cdot d_1 + a \cdot d_2]_{q_t \cdot P}$.
- $\nu' \leftarrow P \cdot \nu + B_{\mathsf{Ks},2}^{\mathsf{BGV}}(t)$.
- Output $\mathsf{Scale}^{\mathsf{BGV}}(((c_0, c_1), t, \nu'), q_t \cdot P)$.

Fig. 9: The two variants of Key Switching for BGV.

$$= d_0 - d_1 \cdot \mathfrak{sk} + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} \left(p \cdot e_i + T^i \cdot \mathfrak{sk}'\right) \cdot d_{2,i}$$

$$= d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}' + p \cdot \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot e_i.$$

So assuming no wrap around the two ciphertexts encrypt the same value. We also have, for the noise term, that

$$\left\|c_0 - \cdot \mathfrak{sk} c_1\right\|_\infty^{\mathsf{can}} \leq \left\|d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}'\right\|_\infty^{\mathsf{can}} + p \cdot \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} \left\|d_{2,i} \cdot e_i\right\|_\infty^{\mathsf{can}}$$
$$\leq \nu + \frac{16}{\sqrt{12}} \cdot p \cdot \left\lceil \log_T q_t \right\rceil \cdot \sigma \cdot \phi(m) \cdot T.$$

So we set

$$B_{\mathsf{Ks},1}^{\mathsf{BGV}}(t) = \frac{8}{\sqrt{3}} \cdot p \cdot \left\lceil \log_T q_t \right\rceil \cdot \sigma \cdot \phi(m) \cdot T.$$

Note, that the size of this term depend on the size of the current modulus q_t as well as T.

SwitchKey Second Variant: Our second variant uses the raising the modulus idea. A large prime P is selected which is congruent to one modulo p. Note that unlike [8] the keyswitch constant $B_{\mathsf{Ks},2}^{\mathsf{BGV}}(t)$ is the addition *before* the scaling takes place, thus it will look larger than in [8].

Again, we establish that the output ciphertext encrypts the same message as the input ciphertext. We look at the ciphertext before the scaling operation.

$$\begin{aligned} c_0 - \mathfrak{sk} \cdot c_1 &= P \cdot (d_0 - \mathfrak{sk} \cdot d_1) + b \cdot d_2 - a \cdot d_2 \cdot \mathfrak{sk} \\ &= P \cdot (d_0 - \mathfrak{sk} \cdot d_1) + d_2 \cdot (p \cdot e + P \cdot \mathfrak{sk}') \\ &= P \cdot (d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}' \cdot d_2) + p \cdot e \cdot d_2. \end{aligned}$$

So we will encrypt the same thing as long as the noise term $p \cdot e \cdot d_2$ does not create wrap around modulo $P \cdot q_t$. The large P is to cater for the large value of d_2. We have

$$\begin{aligned} \left\|c_0 - \mathfrak{sk} \cdot c_1\right\|_\infty^{\mathsf{can}} &\leq P \cdot \left\|d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}' \cdot d_2\right\|_\infty^{\mathsf{can}} + p \cdot \left\|e \cdot d_2\right\|_\infty^{\mathsf{can}} \\ &\leq P \cdot \nu + \frac{16}{\sqrt{12}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \end{aligned}$$

So we set

$$B_{\mathsf{Ks},2}^{\mathsf{BGV}}(t) = \frac{8}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m).$$

D.2 FV

In each of the variants we switch from a key $\mathfrak{sk}'$ to a key $\mathfrak{sk}$. The input ciphertext will involve both keys; thus we aim a switch of the form

$$d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}' \cdot d_2 \longrightarrow c_0 - \mathfrak{sk} \cdot c_1.$$

The two variants are described in Fig. 10.

$\underline{\mathsf{SwitchKeyGen}_1^{\mathsf{FV}}(\mathfrak{sk}', \mathfrak{sk}, T)}$:

- For $i = 0$ to $\left\lceil \log_T(q_{L-1}) \right\rceil - 1$ do
  - ⋆ $a_i \leftarrow \mathcal{U}_{q_{L-1}}$.
  - ⋆ $e_i \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
  - ⋆ $b_i \leftarrow [a_i \cdot \mathfrak{sk} + e_i + T^i \cdot \mathfrak{sk}']_{q_{L_1}}$.
- $\mathfrak{ksd} \leftarrow (T, \{a_i, b_i\}_{i=0}^{\lceil \log_T q_{L-1} \rceil - 1})$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_1^{\mathsf{FV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))}$:

- Write d_2 in base T as $d_2 = \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot T^i$.
- $c_0 \leftarrow d_0 + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot b_i \pmod{q_t}$.
- $c_1 \leftarrow d_1 + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot a_i \pmod{q_t}$.
- $\nu' \leftarrow \nu + B_{\mathsf{Ks},1}^{\mathsf{FV}}(t)$.
- Output $((c_0, c_1), t, \nu')$.

$\underline{\mathsf{SwitchKeyGen}_2^{\mathsf{FV}}(\mathfrak{sk}', \mathfrak{sk})}$:

- $a \leftarrow \mathcal{U}_{q_{L-1}}$.
- $e \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
- $b \leftarrow [a \cdot \mathfrak{sk} + e + P \cdot \mathfrak{sk}']_{q_{L-1} \cdot P}$.
- $\mathfrak{ksd} \leftarrow (a, b)$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_2^{\mathsf{FV}}(((\mathfrak{sk}, \mathfrak{sk}') \to \mathfrak{sk}), (\mathfrak{d}, t, \nu))}$:

- $c_0 \leftarrow P \cdot d_0 + b \cdot d_2 \pmod{q_t \cdot P}$.
- $c_1 \leftarrow P \cdot d_1 + a \cdot d_2 \pmod{q_t \cdot P}$.
- $\nu' \leftarrow P \cdot \nu + B_{\mathsf{Ks},2}^{\mathsf{FV}}(t)$.
- Output $\mathsf{Scale}^{\mathsf{BGV}}(((c_0, c_1), t, \nu'), q_t \cdot P)$.

Fig. 10: The two variants of Key Switching for FV.

SwitchKey First Variant: This is the bit-decomposition method generalised for an arbitrary decomposition modulus t. Note, that the (a_i, b_i) do not even "look like" encryptions of $T^i \cdot \mathfrak{sk}'$ in the FV scheme. As before, we first establish that the output ciphertext encrypts the same message as the input ciphertext. We write $d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}' = m \cdot \Delta_{q_t} + w + r \cdot q_t$

$$\begin{aligned}
c_0 - \mathfrak{sk} \cdot c_1 &= d_0 + \left(\sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot b_i \right) - d_1 \cdot \mathfrak{sk} - \mathfrak{sk} \cdot \left(\sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot a_i \right) \\
&= d_0 - d_1 \cdot \mathfrak{sk} + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} \left(d_{2,i} \cdot b_i - d_{2,i} \cdot a_i \cdot \mathfrak{sk} \right) \\
&= d_0 - d_1 \cdot \mathfrak{sk} + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} \left(e_i + T^i \cdot \mathfrak{sk}' \right) \cdot d_{2,i} \\
&= d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}' + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot e_i \\
&= m \cdot \Delta_{q_t} + w + r \cdot q_t + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot e_i \\
&= m \cdot \Delta_{q_t} + w' + r \cdot q_t.
\end{aligned}$$

So assuming no wrap around the two ciphertexts encrypt the same value. We also have, for the noise term, that

$$\begin{aligned}
\left\| w' - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} &\leq \left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} \left\| d_{2,i} \cdot e_i \right\|_\infty^{\mathsf{can}} \\
&\leq \nu + \frac{16}{\sqrt{12}} \cdot \left\lceil \log_T q_t \right\rceil \cdot \sigma \cdot \phi(m) \cdot T.
\end{aligned}$$

So we set

$$B_{\mathsf{Ks},1}^{\mathsf{FV}}(t) = \frac{8}{\sqrt{3}} \cdot \left\lceil \log_T q_t \right\rceil \cdot \sigma \cdot \phi(m) \cdot T.$$

Note, that the size of this term depend on the size of the current modulus q_t as well as T.

SwitchKey Second Variant: Our second variant uses the raising the modulus idea from [8], hence a large prime P is selected. Note as we are using a scale invariant version we do not require $P \equiv 1 \pmod{p}$, and again note that (a, b) does not "look like" an encryption of $P \cdot \mathfrak{sk}'$ To establish that the output ciphertext encrypts the same message as the input ciphertext, we write $d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}' = m \cdot \Delta_{q_t} + w + r \cdot q_t$ We look at the ciphertext before the scaling operation.

$$c_0 - \mathfrak{sk} \cdot c_1 = P \cdot (d_0 - \mathfrak{sk} \cdot d_1) + b \cdot d_2 - a \cdot d_2 \cdot \mathfrak{sk}$$

$$
\begin{aligned}
&= P \cdot (d_0 - \mathfrak{sk} \cdot d_1) + d_2 \cdot (e + P \cdot \mathfrak{sk}') \\
&= P \cdot (d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}' \cdot d_2) + e \cdot d_2 \\
&= P \cdot (m \cdot \Delta_{q_t} + w + r \cdot q_t) + e \cdot d_2 \\
&= m \cdot P \cdot \Delta_{q_t} + P \cdot w + P \cdot r \cdot q_t + e \cdot d_2 \\
&= m \cdot (\Delta_{P \cdot q_t} + \epsilon_Q - P \cdot \epsilon_{q_t}) + P \cdot w + P \cdot r \cdot q_t + e \cdot d_2 \\
&= m \cdot \Delta_{P \cdot q_t} + w' + r' \cdot q_t
\end{aligned}
$$

We have

$$
w' = (\epsilon_Q - P \cdot \epsilon_{q_t}) \cdot m + P \cdot w + e \cdot d_2,
$$

and we know by our invariant that $\left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} \leq \nu$. This leads us to consider the inequalities

$$
\begin{aligned}
\left\| w' - \epsilon_{P \cdot q_t} \cdot m \right\|_\infty^{\mathsf{can}} &= \left\| P \cdot w - P \cdot \epsilon_{q_t} \cdot m + e \cdot d_2 \right\|_\infty^{\mathsf{can}} \\
&\leq P \cdot \left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} + \left\| e \cdot d_2 \right\|_\infty^{\mathsf{can}} \\
&\leq P \cdot \nu + \frac{16}{\sqrt{12}} \cdot q_t \cdot \sigma \cdot \phi(m).
\end{aligned}
$$

So we set

$$
B_{\mathsf{Ks},2}^{\mathsf{FV}}(t) = \frac{8}{\sqrt{3}} \cdot q_t \cdot \sigma \cdot \phi(m).
$$

D.3 NTRU

Let $\mathfrak{c}'$ be a ciphertext with respect to the secret key $\mathfrak{sk}'$. In both variants, we want to obtain a ciphertext $\mathfrak{c}$ with respect to another secret key $\mathfrak{sk}$ such that both decrypt to the same message. The two variants are described in Fig. 11.

SwitchKey First Variant: Recall we have $\mathfrak{pk} = [p \cdot g / \mathfrak{sk}]_{q_t}$ (see $\mathsf{KeyGen}^{\mathsf{NTRU}}$) and c, $\mathfrak{sk}'$ are such that $c \cdot \mathfrak{sk}' = m + p \cdot e \pmod{q_t}$. we then see that

$$
\begin{aligned}
\mathfrak{sk} \cdot c' &= \sum_i \left(e_{1,i} \cdot \mathfrak{pk} + p \cdot e_{0,i} + T^i \cdot \mathfrak{sk}' \right) \cdot c_i \cdot \mathfrak{sk} \\
&= c \cdot \mathfrak{sk}' \cdot \mathfrak{sk} + p \cdot \left(\sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} \right) \\
&= (m + p \cdot e) \cdot (1 + p \cdot f) + p \cdot \left(\sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} \right) \\
&= m + p \cdot \left(e + f \cdot (m + p \cdot e) + \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} \right).
\end{aligned}
$$

Thus assuming $\left\| \mathfrak{sk} \cdot c' \right\|_\infty^{\mathsf{can}}$ is suitably small we will obtain m upon decryption. All that remains is to bound ν', by deriving an estimate for $B_{\mathsf{Ks},1}^{\mathsf{NTRU}}(t)$,

$$
\left\| \mathfrak{sk} \cdot c' \right\|_\infty^{\mathsf{can}} = \left\| c \cdot \mathfrak{sk}' \cdot \mathfrak{sk} + p \cdot \left(\sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} \right) \right\|_\infty^{\mathsf{can}}
$$

$\underline{\mathsf{SwitchKeyGen}_1^{\mathsf{NTRU}}(\mathfrak{sk}', \mathfrak{sk})}$:

- For $i = 0$ to $\left\lceil \log_T(q_{L-1}) \right\rceil - 1$ do
  - ⋆ $e_{0,i}, e_{1,i} \leftarrow \mathcal{DG}_{q_t}(\sigma^2)$.
  - ⋆ $b_i \leftarrow [e_{1,i} \cdot \mathfrak{pk} + p \cdot e_{0,1} + T^i \cdot \mathfrak{sk}']_{q_t}$.
- $\mathfrak{ksd} \leftarrow (T, \{b_i\}_{i=0}^{\lceil \log_T q_{L-1} \rceil - 1})$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_1^{\mathsf{NTRU}}(\mathfrak{ksd}, (c, t, \nu))}$:

- Write c in base T as $c = \sum_{i=0}^{\lceil \log_T(q_t) \rceil - 1} c_i \cdot T^i$.
- $c' \leftarrow \sum_i b_i \cdot c_i$.
- $\nu' \leftarrow \nu + B_{\mathsf{Ks},1}^{\mathsf{NTRU}}(t)$.
- Output (c', t, ν').

$\underline{\mathsf{SwitchKeyGen}_2^{\mathsf{NTRU}}(\mathfrak{sk}', \mathfrak{sk})}$:

- $s', e' \leftarrow \mathcal{DG}_q(\sigma)$.
- $a \leftarrow [\mathfrak{pk} \cdot s' + p \cdot e' + P \cdot \mathfrak{sk}']_{q_t}$.
- $\mathfrak{ksd} \leftarrow a$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_2^{\mathsf{NTRU}}(\mathfrak{ksd}, (c, t, \nu))}$:

- $c' \leftarrow a \cdot c$.
- $\nu' \leftarrow P \cdot \nu + B_{\mathsf{Ks},2}^{\mathsf{NTRU}}(t)$.
- Output $\mathsf{Scale}(c', t, \nu')$.

Fig. 11: The two variants of Key Switching for NTRU.

$$
\begin{aligned}
&\leq \left\| c \cdot \mathfrak{sk}' + p \cdot c \cdot \mathfrak{sk}' \cdot f \right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\| p \cdot \left(\sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i + p \cdot \sum_i e_{0,i} \cdot c_i \cdot f \right) \right\|_\infty^{\mathsf{can}} \\
&\leq \nu + p \cdot \Big(6 \cdot \nu \cdot \sqrt{h} + 40 \cdot \left\lceil \log_T(q_t) \right\rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&\qquad + 16 \cdot \left\lceil \log_T(q_t) \right\rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{1/12} \\
&\qquad + 40 \cdot p \cdot \left\lceil \log_T(q_t) \right\rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \Big) \\
&\leq \nu + p \cdot \Big(6 \cdot \nu \cdot \sqrt{h} + \frac{20}{\sqrt{3}} \cdot (1+p) \cdot \left\lceil \log_T(q_t) \right\rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h} \\
&\qquad + \frac{8}{\sqrt{3}} \cdot \left\lceil \log_T(q_t) \right\rceil \cdot T \cdot \sigma \cdot \phi(m) \Big).
\end{aligned}
$$

So we let

$$
\begin{aligned}
B_{\mathsf{Ks},1}^{\mathsf{NTRU}}(t) = p \cdot \Big(&6 \cdot \nu \cdot \sqrt{h} + \frac{20}{\sqrt{3}} \cdot (1+p) \cdot \left\lceil \log_T(q_t) \right\rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h} \\
&+ \frac{8}{\sqrt{3}} \cdot \left\lceil \log_T(q_t) \right\rceil \cdot T \cdot \sigma \cdot \phi(m) \Big).
\end{aligned}
$$

Note that $B_{\mathsf{Ks},1}^{\mathsf{NTRU}}(t)$ depends on ν, which is not the case for the BGV and FV schemes.

SwitchKey Second Variant: Since c decrypts under $\mathfrak{sk}'$, let $c \cdot \mathfrak{sk}' = m + p \cdot e$. We look at the ciphertext before the scaling operation, and see

$$
c' \cdot \mathfrak{sk} = (\mathfrak{pk} \cdot s' + p \cdot e' + P \cdot \mathfrak{sk}') \cdot c \cdot \mathfrak{sk}
$$

$$= \mathfrak{pk} \cdot s' \cdot \mathfrak{sk} \cdot c + (p \cdot e' + P \cdot \mathfrak{sk}') \cdot c \cdot (1 + p \cdot f)$$
$$= P \cdot \mathfrak{sk}' \cdot c + p \cdot g \cdot s' \cdot c + p \cdot e' \cdot c + p^2 \cdot e' \cdot c \cdot f + p \cdot P \cdot \mathfrak{sk}' \cdot c \cdot f$$

Thus we will obtain, assuming no wrap around, the "message" $P \cdot m = m$ modulo p. To guarantee no wrap around we need to bound $\left\|c' \cdot \mathfrak{sk}\right\|_\infty^{\mathsf{can}}$

$$\begin{aligned}
\left\|c' \cdot \mathfrak{sk}\right\|_\infty^{\mathsf{can}} &= \left\|P \cdot \mathfrak{sk}' \cdot c + p \cdot g \cdot s' \cdot c + p \cdot e' \cdot c + p^2 \cdot e' \cdot c \cdot f + p \cdot P \cdot \mathfrak{sk}' \cdot c \cdot f\right\|_\infty^{\mathsf{can}} \\
&\leq P \cdot \left\|c \cdot \mathfrak{sk}'\right\|_\infty^{\mathsf{can}} + p \cdot \left\|g \cdot s' \cdot c\right\|_\infty^{\mathsf{can}} + p \cdot \left\|e' \cdot c\right\|_\infty^{\mathsf{can}} + p^2 \cdot \left\|e' \cdot c \cdot f\right\|_\infty^{\mathsf{can}} \\
&\qquad + p \cdot P \cdot \left\|\mathfrak{sk}' \cdot c \cdot f\right\|_\infty^{\mathsf{can}} \\
&\leq P \cdot \nu + 40 \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&\qquad + \frac{8}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \\
&\qquad + 40 \cdot p^2 \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&\qquad + 6 \cdot p \cdot P \cdot \nu \cdot \sqrt{h} \\
&= P \cdot \nu + 40 \cdot p \cdot (1+p) \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&\qquad + \frac{8}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \\
&\qquad + 6 \cdot p \cdot P \cdot \nu \cdot \sqrt{h}.
\end{aligned}$$

Thus we set

$$B_{\mathsf{Ks},2}^{\mathsf{NTRU}}(t) = 40 \cdot p \cdot (1+p) \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} + \frac{8}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) + 6 \cdot p \cdot P \cdot \nu \cdot \sqrt{h}.$$

Note again that $B_{\mathsf{Ks},2}^{\mathsf{NTRU}}(t)$ depends on ν.

D.4 YASHE

Again let $\mathfrak{c}'$ be a ciphertext with respect to the secret key $\mathfrak{sk}'$. In both variants, we want to obtain a ciphertext $\mathfrak{c}$ with respect to another secret key $\mathfrak{sk}$ such that both decrypt to the same message. The two variants are described in Fig. 12.

SwitchKey First variant: Since we start with a ciphertext c which decrypts under $\mathfrak{sk}'$, let $c \cdot \mathfrak{sk}' = \Delta_{q_t} \cdot m + w + r \cdot q_t$. Then notice that

$$\begin{aligned}
\mathfrak{sk} \cdot c' &= \sum_i (e_{1,i} \cdot \mathfrak{pk} + e_{0,i} + T^i \cdot \mathfrak{sk}') \cdot c_i \cdot \mathfrak{sk} \\
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + c \cdot \mathfrak{sk}' \cdot \mathfrak{sk} \\
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + c \cdot \mathfrak{sk}' \cdot (1 + p \cdot f)
\end{aligned}$$

$\underline{\mathsf{SwitchKeyGen}_1^{\mathsf{YASHE}}(\mathfrak{sk}', \mathfrak{sk})}$:

- For $i = 0$ to $\left\lceil \log_T(q_{L-1}) \right\rceil - 1$ do
  - $\star$ $e_{0,i}, e_{1,i} \leftarrow \mathcal{DG}_{q_t}(\sigma^2)$.
  - $\star$ $b_i \leftarrow [e_{1,i} \cdot \mathfrak{pk} + e_{0,1} + T^i \cdot \mathfrak{sk}']_{q_t}$.
- $\mathfrak{ksd} \leftarrow (T, \{b_i\}_{i=0}^{\lceil \log_T(q_{L-1}) \rceil - 1})$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_1^{\mathsf{YASHE}}(\mathfrak{ksd}, c, t, \nu)}$:

- $\nu' \leftarrow \nu + B_{\mathsf{Ks},1}^{\mathsf{YASHE}}(t)$.
- Write c in base T as $\sum_{i=0}^{\lceil \log_T(q_t) \rceil - 1} c_i \cdot T^i$.
- Set $c' = \sum_i b_i \cdot c_i$.
- Output $\mathfrak{c} = (c', t, \nu')$.

$\underline{\mathsf{SwitchKeyGen}_2^{\mathsf{YASHE}}(\mathfrak{sk}', \mathfrak{sk})}$:

- $e_0, e_1 \leftarrow \mathcal{DG}_q(\sigma)$.
- $a \leftarrow [\mathfrak{pk} \cdot e_1 + e_0 + P \cdot \mathfrak{sk}']_Q$.
- $\mathfrak{ksd} \leftarrow a$.
- Output $\mathfrak{ksd}$.

$\underline{\mathsf{SwitchKey}_2^{\mathsf{YASHE}}(\mathfrak{ksd}, (c, t, \nu))}$:

- $\nu' \leftarrow \nu + B_{\mathsf{Scale}}^{\mathsf{YASHE}}$.
- $d \leftarrow a \cdot c$.
- $c' \leftarrow \mathsf{Scale}(d, P, q_t)$.
- Output $\mathfrak{c} = (c', \nu', t)$.

Fig. 12: The two variants of Key Switching for YASHE.

$$
\begin{aligned}
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + (\Delta_{q_t} \cdot m + w + r \cdot q_t) \\
&\qquad + p \cdot f \cdot (\Delta_{q_t} \cdot m + w + r \cdot q_t) \\
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + (\Delta_{q_t} \cdot m + w + r \cdot q_t) \\
&\qquad - p \cdot f \cdot m \cdot \epsilon_{q_t} - p \cdot f \cdot m \cdot \epsilon_{q_t} + p \cdot f \cdot (w + r \cdot q_t) \\
&= \Delta_{q_t} \cdot m + w' + r' \cdot q_t,
\end{aligned}
$$

where we have $w' = p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} - p \cdot f \cdot m \cdot \epsilon_{q_t} + w \cdot (1 + p \cdot f)$ and $r' = r \cdot (1 + p \cdot f) + p \cdot f \cdot m$. We therefore want to bound

$$
\begin{aligned}
\left\| w' - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} &\leq \left\| p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} - p \cdot f \cdot m \cdot \epsilon_{q_t} \right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\| w \cdot (1 + p \cdot f) - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} \\
&\leq \left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} \\
&\qquad + p \cdot \left\| \sum_i e_{1,i} \cdot g \cdot c_i \right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\| \sum_i e_{0,i} \cdot c_i \cdot (1 + pf) \right\|_\infty^{\mathsf{can}} \\
&\qquad - p \cdot \epsilon_{q_t} \cdot \left\| f \cdot m \right\|_\infty^{\mathsf{can}} \\
&\qquad + p \cdot \left\| f \cdot w \right\|_\infty^{\mathsf{can}} \\
&\leq \left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} \\
&\qquad + p \cdot \left\| \sum_i e_{1,i} \cdot g \cdot c_i \right\|_\infty^{\mathsf{can}}
\end{aligned}
$$

$$
\begin{aligned}
&+ \Big\| \sum_i e_{0,i} \cdot c_i \Big\|_\infty^{\mathsf{can}} \\
&+ p \cdot \Big\| \sum_i f \cdot e_{0,i} \cdot c_i \Big\|_\infty^{\mathsf{can}} \\
&+ p^2 \cdot \big\| f \cdot m \big\|_\infty^{\mathsf{can}} \\
&+ p \cdot \big\| f \cdot (w - \epsilon_{q_t} + \epsilon_{q_t}) \big\|_\infty^{\mathsf{can}} \\
\leq\ & \nu + 40 \cdot p \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&+ 16 \cdot \lceil \log_T(q_t) \rceil \cdot \sigma \cdot T \cdot \phi(m) \cdot \sqrt{1/12} \\
&+ 40 \cdot p \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&+ 16 \cdot p^3 \cdot \sqrt{h \cdot \phi(m)/12} \\
&+ p \cdot \big\| f \cdot (w - \epsilon_{q_t}) \big\|_\infty^{\mathsf{can}} \\
&+ p \cdot \big\| f \cdot \epsilon_{q_t} \big\|_\infty^{\mathsf{can}} \\
\leq\ & \nu + \frac{40}{\sqrt{3}} \cdot p \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h} \\
&+ \frac{8}{\sqrt{3}} \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) \\
&+ \frac{8}{\sqrt{3}} \cdot p^3 \cdot \sqrt{h \cdot \phi(m)} \\
&+ 6 \cdot p \cdot \nu \cdot \sqrt{h} \\
&+ 6 \cdot p^2 \cdot \sqrt{h} \\
\leq\ & \nu + \frac{8}{\sqrt{3}} \cdot \left(1 + 5 \cdot p \cdot \sqrt{h}\right) \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) \\
&+ \frac{8}{\sqrt{3}} \cdot p^3 \cdot \sqrt{h \cdot \phi(m)} \\
&+ 6 \cdot p \cdot (\nu + p) \cdot \sqrt{h}.
\end{aligned}
$$

Let $B_{\mathsf{Ks},1}^{\mathsf{YASHE}}(t) = \frac{8}{\sqrt{3}} \cdot \left(1 + 5 \cdot p \cdot \sqrt{h}\right) \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) + \frac{8}{\sqrt{3}} \cdot p^3 \cdot \sqrt{h \cdot \phi(m)} + 6 \cdot p \cdot (\nu + p) \cdot \sqrt{h}$. Note that as in the previous section, this depends on ν.

SwitchKey Second Variant: Here again we use the idea of raising the modulus to some large P, then use the Scale function at the end of the operation. We let $Q = q_t \cdot P$ and recall that $\Delta_Q = \left\lfloor \frac{Q}{p} \right\rfloor = \frac{Q}{p} - \epsilon_Q = \frac{q_t \cdot P}{p} - \epsilon_Q = P \cdot (\Delta_{q_t} + \epsilon_{q_t}) - \epsilon_Q$. We first check that the output decrypts correctly. Since c decrypts under $\mathfrak{sk}'$, we have that $c \cdot \mathfrak{sk}' = \Delta_{q_t} \cdot m + w + r \cdot q_t$.

$$
\begin{aligned}
d \cdot \mathfrak{sk} &= a \cdot c \cdot \mathfrak{sk} \\
&= \mathfrak{sk} \cdot (\mathfrak{pk} \cdot e_1 + e_0 + P \cdot \mathfrak{sk}') \cdot c \\
&= P \cdot \mathfrak{sk} \cdot \mathfrak{sk}' \cdot c + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c
\end{aligned}
$$

$$
\begin{aligned}
&= P \cdot \mathfrak{sk} \cdot (\Delta_{q_t} \cdot m + w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= P \cdot \mathfrak{sk} \cdot \Delta_{q_t} \cdot m + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= P \cdot \mathfrak{sk} \cdot m \cdot \Delta_{q_t} + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= (1 + p \cdot f) \cdot m \cdot P \cdot \Delta_{q_t} + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= \Delta_Q \cdot m + m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) + p \cdot f \cdot m \cdot P \cdot \Delta_{q_t} \\
&\qquad + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= \Delta_Q \cdot m + m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) + p \cdot f \cdot m \cdot P \cdot \Delta_{q_t} \\
&\qquad + P \cdot (1 + p \cdot f) \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= \Delta_Q \cdot m + m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) + p \cdot f \cdot m \cdot P \cdot (\frac{q_t}{p} - \epsilon_{q_t}) \\
&\qquad + P \cdot (w + r \cdot q_t) + p \cdot f \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= \Delta_Q \cdot m + m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) + f \cdot m \cdot Q - p \cdot f \cdot m \cdot P \cdot \epsilon_{q_t} \\
&\qquad + P \cdot (w + r \cdot q_t) + p \cdot f \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= \Delta_Q \cdot m + w' + r' \cdot Q,
\end{aligned}
$$

where $w' = m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) - p \cdot P \cdot \epsilon_{q_t} \cdot f \cdot m + P \cdot w + p \cdot f \cdot w + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c$ and $r' = r + r \cdot p \cdot f + f \cdot m$. Thus we indeed have a ciphertext modulo Q which correctly decrypts to the initial message m, so long as the noise is not too big. We know that $\left\|w - m \cdot e_{q_t}\right\|_\infty^{\mathsf{can}} \leq \nu$ and so we consider

$$
\begin{aligned}
\left\|w' - \epsilon_Q m\right\|_\infty^{\mathsf{can}} &\leq \left\| - p \cdot P \cdot \epsilon_{q_t} \cdot f \cdot m + p \cdot f \cdot w + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|P \cdot w + m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) - \epsilon_Q \cdot m\right\|_\infty^{\mathsf{can}} \\
&\leq \left\|p \cdot P \cdot \epsilon_{q_t} \cdot f \cdot m\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|p \cdot f \cdot (w - \epsilon_{q_t} \cdot m + \epsilon_{q_t} \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|(1 + p \cdot f) \cdot e_0 \cdot c\right\|_\infty^{\mathsf{can}} \\
&\qquad + p \cdot \left\|e_2 \cdot g \cdot c\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|P \cdot w + m \cdot \epsilon_Q - P \cdot m \cdot \epsilon_{q_t} - \epsilon_Q \cdot m\right\|_\infty^{\mathsf{can}} \\
&\leq \left\|p \cdot P \cdot \epsilon_{q_t} \cdot f \cdot m\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|p \cdot f \cdot (w - \epsilon_{q_t} \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|p \cdot f \cdot \epsilon_{q_t} \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|(1 + p \cdot f) \cdot e_0 \cdot c\right\|_\infty^{\mathsf{can}} \\
&\qquad + p \cdot \left\|e_2 \cdot g \cdot c\right\|_\infty^{\mathsf{can}} \\
&\qquad + P \cdot \left\|w - m \cdot \epsilon_{q_t}\right\|_\infty^{\mathsf{can}} \\
&\leq P \cdot p^2 \left\|f \cdot m\right\|_\infty^{\mathsf{can}} \\
&\qquad + p \cdot \nu \cdot \left\|f\right\|_\infty^{\mathsf{can}} \\
&\qquad + p^2 \cdot \left\|f \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\qquad + \left\|e_0 \cdot c\right\|_\infty^{\mathsf{can}}
\end{aligned}
$$

$$
\begin{aligned}
&+ p \cdot \left\| f \cdot e_0 \cdot c \right\|_\infty^{\mathsf{can}} \\
&+ p \cdot \left\| e_2 \cdot g \cdot c \right\|_\infty^{\mathsf{can}} \\
&+ P \cdot \nu \\
\leq\ & 16 \cdot P \cdot p^3 \cdot \sqrt{h \cdot \phi(m)/12} \\
&+ 6 \cdot p \cdot \nu \cdot \sqrt{h} \\
&+ 16 \cdot p^3 \cdot \sqrt{h \cdot \phi(m)/12} \\
&+ 16 \cdot q_t \cdot \sigma \cdot \phi(m)/\sqrt{12} \\
&+ 40 \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&+ 40 \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&+ P \cdot \nu \\
\leq\ & \frac{8}{\sqrt{3}} \cdot P \cdot p^3 \cdot \sqrt{h \cdot \phi(m)} \\
&+ 6 \cdot p \cdot \nu \cdot \sqrt{h} \\
&+ \frac{8}{\sqrt{3}} \cdot p^3 \cdot \sqrt{h \cdot \phi(m)} \\
&+ \frac{8}{\sqrt{3}} \cdot q_t \cdot \sigma \cdot \phi(m) \\
&+ \frac{40}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h} \\
&+ P \cdot \nu
\end{aligned}
$$

Therefore, we set $B_{\mathsf{Ks},2}^{\mathsf{YASHE}}(t) = \frac{8}{\sqrt{3}} \cdot P \cdot p^3 \cdot \sqrt{h \cdot \phi(m)} + 6 \cdot p \cdot \nu \cdot \sqrt{h} + \frac{8}{\sqrt{3}} \cdot p^3 \cdot \sqrt{h \cdot \phi(m)} + \frac{8}{\sqrt{3}} \cdot q_t \cdot \sigma \cdot \phi(m) + \frac{40}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h}$. Again note this depends on the previous noise bound ν.

E Addition and Multiplication

The homomorphic addition method for all schemes is given in Fig. 13.

E.1 BGV

These methods are standard. The fact that the output ciphertext satisfies $\left\| c_0 - \mathfrak{sk} \cdot c_1 \right\|_\infty^{\mathsf{can}} \leq \nu$ in both cases is obvious.

E.2 FV

To see that the output ν is correct for the addition operation, we write $c_{i,0} - \mathfrak{sk} \cdot c_{i,1} = \Delta_{q_t} \cdot m_i + w_i + r_i \cdot q_t$ and $c_0 - \mathfrak{sk} \cdot c_1 = \Delta_{q_t} \cdot m + w + r \cdot q_t$, where $m_i \in \mathbb{A}_p$, and

$\underline{\mathsf{Add}^{\mathsf{BGV}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{BGV}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c_{i,0}, c_{i,1}, t, \nu_i)$.
- $c_0 \leftarrow c_{0,0} + c_{1,0} \pmod{q_t}$.
- $c_1 \leftarrow c_{0,1} + c_{1,1} \pmod{q_t}$.
- $\nu \leftarrow \nu_0 + \nu_1$
- Output $((c_0, c_1), t, \nu)$.

$\underline{\mathsf{Add}^{\mathsf{FV}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{FV}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c_{i,0}, c_{i,1}, t, \nu_i)$.
- $c_0 \leftarrow c_{0,0} + c_{1,0} \pmod{q_t}$.
- $c_1 \leftarrow c_{0,1} + c_{1,1} \pmod{q_t}$.
- $\nu \leftarrow \nu_0 + \nu_1$
- Output $\mathfrak{c} = ((c_0, c_1), t, \nu)$.

$\underline{\mathsf{Add}^{\mathsf{NTRU}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{NTRU}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c_i, t, \nu_i)$.
- $c \leftarrow c_0 + c_1 \pmod{q_t}$.
- $\nu \leftarrow \nu_0 + \nu_1$
- Output (c, t, ν).

$\underline{\mathsf{Add}^{\mathsf{YASHE}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{YASHE}}(\mathfrak{c}_i, t)$ for $i = 1, 2$.
- Write $\mathfrak{c}_i = (c_i, t, \nu_i)$.
- $c \leftarrow c_0 + c_1 \pmod{q_t}$.
- $\nu \leftarrow \nu_0 + \nu_1$
- Output $\mathfrak{c} = (c, t, \nu)$.

Fig. 13: The Addition Methods for BGV, FV, NTRU and YASHE.

write $m = [m_0 + m_1]_p = m_0 + m_1 + p \cdot r_a$. Then, decrypting $\mathfrak{c}$ results in the taking the value (modulo q_t)

$$\begin{aligned}\Delta_{q_t} \cdot (m_0 + m_1) + w_0 + w_1 &= \Delta_{q_t} \cdot (m - p \cdot r_a) + w_0 + w_1 \pmod{q_t} \\ &= \Delta_{q_t} \cdot m + w_0 + w_1 - p \cdot r_a \cdot \Delta_{q_t} \\ &= \Delta_{q_t} \cdot m + w_0 + w_1 - p \cdot r_a \cdot \left(\frac{q_t}{p} - \epsilon_{q_t}\right) \\ &= \Delta_{q_t} \cdot m + w_0 + w_1 + p \cdot r_a \cdot \epsilon_{q_t} \pmod{q_t} \\ &= \Delta_{q_t} \cdot m + w\end{aligned}$$

multiplying the result by p/q_t and rounding. Thus $w = w_0 + w_1 + p \cdot r_a \cdot \epsilon_{q_t}$ and so the ν value on $\mathfrak{c}$ is an upper bound on

$$\begin{aligned}\left\|w - \epsilon_{q_t} \cdot m\right\|_\infty^{\mathsf{can}} &= \left\|w_0 + w_1 + p \cdot r_a \cdot \epsilon_{q_t} - \epsilon_{q_t} \cdot (m_0 + m_1 + p \cdot r_a)\right\|_\infty^{\mathsf{can}} \\ &\le \left\|w_0 - \epsilon_{q_t} \cdot m_0\right\|_\infty^{\mathsf{can}} + \left\|w_1 - \epsilon_{q_t} \cdot m_1\right\|_\infty^{\mathsf{can}} \\ &\le \nu_0 + \nu_1.\end{aligned}$$

For the multiplication operation the triple $\mathfrak{d} = (d_0, d_1, d_2)$ decrypts via the equation

$$\left\lceil \frac{p}{q_t} \cdot [d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}^2 \cdot d_2]_{q_t} \right\rfloor$$

which is why we need the SwitchKey operation. To establish correctness, and the bound on ν, we write $[c_{i,0} - \mathfrak{sk} \cdot c_{i,1}]_{q_t} = \Delta_{q_t} \cdot m_i + w_i + r_i \cdot q_t$. Recall that $\left\|w_i - \epsilon_{q_t} \cdot m_i\right\|_\infty^{\mathsf{can}} \le \nu_i$, which means that

$$\left\|w_i\right\|_\infty^{\mathsf{can}} \leq \left\|w_i - \epsilon_{q_t} \cdot m_i\right\|_\infty^{\mathsf{can}} + \left\|\epsilon_{q_t} \cdot m_i\right\|_\infty^{\mathsf{can}} \\ \leq \nu_i + p^2 \cdot \sqrt{3 \cdot \phi(m)} = B_{w_i}.$$

Note that this means that

$$\begin{aligned}\left\|r_i\right\|_\infty^{\mathsf{can}} &= \left\|\frac{1}{q_t}\left(c_{i,0} - \mathfrak{sk} \cdot c_{i,1} - \Delta_{q_t} \cdot m_i - w_i\right)\right\|_\infty^{\mathsf{can}} \\ &\leq \left\|c_{i,0}\right\|_\infty^{\mathsf{can}}/q_t + \left\|\mathfrak{sk} \cdot c_{i,1}\right\|_\infty^{\mathsf{can}}/q_t + \left\|m_i\right\|_\infty^{\mathsf{can}}/p + \left\|w_i - \epsilon_{q_t} \cdot m_i\right\|_\infty^{\mathsf{can}}/q_t \\ &\leq \sqrt{3 \cdot \phi(m)} + \frac{16}{\sqrt{12}} \cdot \sqrt{\phi(m) \cdot h} + \sqrt{3 \cdot \phi(m)} + \frac{\nu_i}{q_t} \\ &= 2 \cdot \sqrt{3 \cdot \phi(m)} + \frac{8}{\sqrt{3}} \cdot \sqrt{\phi(m) \cdot h} + \frac{\nu_i}{q_t} \\ &= B_{r_i}.\end{aligned}$$

We also write $d_i' = d_i'' + \delta_i$. Note that

$$\begin{aligned}\left\|\delta_0 - \delta_1 \cdot \mathfrak{sk} + \delta_2 \cdot \mathfrak{sk}^2\right\|_\infty^{\mathsf{can}} &\leq \sqrt{3 \cdot \phi(m)} + 12 \cdot \sqrt{\phi(m) \cdot h/12} + 40 \cdot h \cdot \sqrt{\phi(m)/12} \\ &= \sqrt{3 \cdot \phi(m)} + 2 \cdot \sqrt{3 \cdot \phi(m) \cdot h} + 20 \cdot h \cdot \sqrt{\phi(m)/3} \\ &= B_\delta.\end{aligned}$$

We set $r_a = (\delta_0 - \mathfrak{sk} \cdot \delta_1 + \mathfrak{sk}^2 \cdot \delta_2)$ and $[m]_p = [m_0 \cdot m_1]_p = m_0 \cdot m_1 - p \cdot r_m$. We can take $\left\|r_m\right\|_\infty^{\mathsf{can}} \leq 16 \cdot p \cdot \phi(m)/12 = 4 \cdot p \cdot \phi(m)/3$.

We now need to examine the value of $d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}^2 \cdot d_2$, we note that as we only take the result modulo q_t we might as well examine $d_0' - \mathfrak{sk} \cdot d_1' + \mathfrak{sk}^2 \cdot d_2'$ We then have that,

$$\begin{aligned}d_0' - \mathfrak{sk} \cdot d_1' + \mathfrak{sk}^2 \cdot d_2' &= \frac{p}{q_t} \cdot \Big(c_{0,0} \cdot c_{1,0} - \mathfrak{sk} \cdot (c_{0,0} \cdot c_{1,1} + c_{0,1} \cdot c_{1,0}) + \mathfrak{sk}^2 \cdot c_{0,1} \cdot c_{1,1}\Big) \\ &\quad + \Big(\delta_0 - \mathfrak{sk} \cdot \delta_1 + \mathfrak{sk}^2 \cdot \delta_2\Big), \\ &= \frac{p}{q_t} \cdot \Big(c_{0,0} - \mathfrak{sk} \cdot c_{0,1}\Big) \cdot \Big(c_{1,0} - \mathfrak{sk} \cdot c_{1,1}\Big) + r_a, \\ &= \frac{p}{q_t} \cdot \Big(\Delta_{q_t} \cdot m_0 + w_0 + r_0 \cdot q_t\Big) \cdot \Big(\Delta_{q_t} \cdot m_1 + w_1 + r_1 \cdot q_t\Big) \\ &\quad + r_a, \\ &= \frac{p}{q_t} \cdot \Big(\Delta_{q_t}^2 \cdot m_0 \cdot m_1 \\ &\qquad + \Delta_{q_t} \cdot (m_0 \cdot (w_1 + r_1 \cdot q_t) + m_1 \cdot (w_0 + r_0 \cdot q_t)) \\ &\qquad + (w_0 + r_0 \cdot q_t) \cdot (w_1 + r_1 \cdot q_t)\Big) + r_a, \\ &= \frac{p}{q_t} \cdot \Big(\Delta_{q_t} \cdot \frac{q_t}{p} \cdot m_0 \cdot m_1 - \Delta_{q_t} \cdot \epsilon_{q_t} \cdot m_0 \cdot m_1\end{aligned}$$

$$
\begin{aligned}
&\quad + \left(\frac{q_t}{p} - \epsilon_{q_t}\right) \cdot \Big(m_0 \cdot (w_1 + r_1 \cdot q_t) \\
&\qquad\qquad\qquad\qquad + m_1 \cdot (w_0 + r_0 \cdot q_t)\Big) \\
&\quad + (w_0 + r_0 \cdot q_t) \cdot (w_1 + r_1 \cdot q_t)\Big) + r_a, \\
&= \mathit{\Delta}_{q_t} \cdot [m]_p + \mathit{\Delta}_{q_t} \cdot p \cdot r_m - \frac{p}{q_t} \cdot \mathit{\Delta}_{q_t} \cdot \epsilon_{q_t} \cdot [m]_p \\
&\quad - \frac{p}{q_t} \cdot \mathit{\Delta}_{q_t} \cdot \epsilon_{q_t} \cdot p \cdot r_m \\
&\quad + \Big(m_0 \cdot (w_1 + r_1 \cdot q_t) + m_1 \cdot (w_0 + r_0 \cdot q_t)\Big) \\
&\quad - \frac{\epsilon_{q_t} \cdot p}{q_t} \cdot \Big(m_0 \cdot (w_1 + r_1 \cdot q_t) + m_1 \cdot (w_0 + r_0 \cdot q_t)\Big) \\
&\quad + \frac{p}{q_t} \cdot w_0 \cdot w_1 + p \cdot (r_0 \cdot w_1 + r_1 \cdot w_0) \\
&\quad + p \cdot q_t \cdot r_0 \cdot r_1 + r_a \\
&= \mathit{\Delta}_{q_t} \cdot [m]_p + \mathit{\Delta}_{q_t} \cdot p \cdot \left(r_m - \frac{\epsilon_{q_t}}{q_t} \cdot [m]_p - \frac{\epsilon_{q_t}}{q_t} \cdot p \cdot r_m\right) \\
&\quad + q_t \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 + p \cdot r_0 \cdot r_1\Big) \\
&\quad + m_0 \cdot w_1 + m_1 \cdot w_0 + p \cdot (r_0 \cdot w_1 + r_1 \cdot w_0) \\
&\quad + \frac{p}{q_t} \cdot \Big(w_0 \cdot w_1 - \epsilon_{q_t} \cdot (m_0 \cdot w_1 + m_1 \cdot w_0)\Big) \\
&\quad - \epsilon_{q_t} \cdot \Big(p \cdot m_0 \cdot r_1 + p \cdot m_1 \cdot r_0\Big) + r_a \\
&= \mathit{\Delta}_{q_t} \cdot [m]_p + (q_t - p \cdot \epsilon_{q_t}) \cdot \left(r_m - \frac{\epsilon_{q_t}}{q_t} \cdot [m]_p - \frac{\epsilon_{q_t}}{q_t} \cdot p \cdot r_m\right) \\
&\quad + q_t \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 + p \cdot r_0 \cdot r_1\Big) \\
&\quad + m_0 \cdot w_1 + m_1 \cdot w_0 + p \cdot (r_0 \cdot w_1 + r_1 \cdot w_0) \\
&\quad + \frac{p}{q_t} \cdot \Big(w_0 \cdot w_1 - \epsilon_{q_t} \cdot (m_0 \cdot w_1 + m_1 \cdot w_0)\Big) \\
&\quad - \epsilon_{q_t} \cdot \Big(p \cdot m_0 \cdot r_1 + p \cdot m_1 \cdot r_0\Big) + r_a \\
&= \mathit{\Delta}_{q_t} \cdot [m]_p + q_t \cdot r_m - \epsilon_{q_t} \cdot [m]_p - 2 \cdot \epsilon_{q_t} \cdot p \cdot r_m \\
&\quad + \frac{p}{q_t} \cdot \epsilon_{q_t}^2 \cdot [m]_p + \frac{p}{q_t} \cdot p \cdot \epsilon_{q_t}^2 \cdot r_m \\
&\quad + q_t \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 + p \cdot r_0 \cdot r_1\Big) \\
&\quad + m_0 \cdot w_1 + m_1 \cdot w_0 + p \cdot (r_0 \cdot w_1 + r_1 \cdot w_0) \\
&\quad + \frac{p}{q_t} \cdot \Big(w_0 \cdot w_1 - \epsilon_{q_t} \cdot (m_0 \cdot w_1 + m_1 \cdot w_0)\Big) \\
&\quad - \epsilon_{q_t} \cdot p \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0\Big) + r_a.
\end{aligned}
$$

We know the expression on the right hand side is integral, and we can take the expression on the left modulo q_t. Thus we are interested in bounding the canonical norm of the term

$$\begin{aligned}w - \epsilon_{q_t} \cdot [m]_p = -2 \cdot \epsilon_{q_t} \cdot [m]_p - 2 \cdot \epsilon_{q_t} \cdot p \cdot r_m + \frac{p}{q_t} \cdot \epsilon_{q_t}^2 \cdot [m]_p + \frac{p}{q_t} \cdot p \cdot \epsilon_{q_t}^2 \cdot r_m \\ + m_0 \cdot w_1 + m_1 \cdot w_0 + p \cdot (r_0 \cdot w_1 + r_1 \cdot w_0) \\ + \frac{p}{q_t} \cdot \Big(w_0 \cdot w_1 - \epsilon_{q_t} \cdot (m_0 \cdot w_1 + m_1 \cdot w_0) \Big) \\ - \epsilon_{q_t} \cdot p \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 \Big) + r_a\end{aligned}$$

We obtain a bound of, recalling $\epsilon_{q_t} \le p$,

$$\begin{aligned}\Big\| w - \epsilon_{q_t} \cdot [m]_p \Big\|_\infty^{\mathsf{can}} &\le 2 \cdot p \cdot \|m\|_\infty^{\mathsf{can}} + 2 \cdot p^2 \cdot \|r_m\|_\infty^{\mathsf{can}} \\ &\quad + \frac{p^3}{q_t} \cdot \|m\|_\infty^{\mathsf{can}} + \frac{p^4}{q_t} \cdot \|r_m\|_\infty^{\mathsf{can}} \\ &\quad + S + p \cdot T + \frac{p}{q_t} (B_{w_0} \cdot B_{w_1} + p \cdot S) + p^2 \cdot U + \|r_a\|_\infty^{\mathsf{can}}, \\ &\le 2 \cdot p^2 \cdot \sqrt{3 \cdot \phi(m)} + \frac{8}{3} \cdot p^3 \cdot \phi(m) \\ &\quad + \frac{p^4}{q_t} \cdot \sqrt{3 \cdot \phi(m)} + \frac{p^5}{3 \cdot q_t} \cdot \phi(m) \\ &\quad + S + p \cdot T + \frac{p}{q_t} \cdot (B_{w_0} \cdot B_{w_1} + p \cdot S) + p^2 \cdot U + B_\delta \\ &= F^{\mathsf{FV}}(\nu_0, \nu_1).\end{aligned}$$

where

$$\begin{aligned}\|m_0 \cdot w_1 + m_1 \cdot w_0\|_\infty^{\mathsf{can}} &\le (B_{w_1} + B_{w_2}) \cdot p \cdot \sqrt{3 \cdot \phi(m)} = S, \\ \|r_0 \cdot w_1 + r_1 \cdot w_0\|_\infty^{\mathsf{can}} &\le B_{w_1} \cdot B_{r_0} + \cdot B_{w_0} \cdot B_{r_1} = T, \\ \|r_0 \cdot m_1 + r_1 \cdot m_0\|_\infty^{\mathsf{can}} &\le (B_{r_0} + B_{r_1}) \cdot p \cdot \sqrt{3 \cdot \phi(m)} = U.\end{aligned}$$

Notice that this new value of ν grows as $\nu_0 \cdot \nu_1 / q_t$, in terms of the input noise values.

E.3 NTRU

Recall for NTRU that our invariant on ν is that $\|c \cdot \mathfrak{sk}\|_\infty^{\mathsf{can}} \le \nu$. It is immediate that the output noise level satisfies the requied inequality for the addition and multiplication operations, and that both operations will be correct if the noise remains within the decryption bound.

E.4 YASHE

To see that ν is correct for addition, write $c_i \cdot \mathfrak{sk} = \Delta_{q_t} \cdot m_i + w_i + r_i \cdot q_t$ and $c \cdot \mathfrak{sk} = \Delta_{q_t} \cdot m + w + r \cdot q_t$, where $m_i \in \mathbb{A}_p$, and write $m = [m_0 + m_1]_p = m_0 + m_1 + p \cdot r_a$.

Then, decrypting $\mathfrak{c}$ results in the taking the value (modulo q_t) of

$$\begin{aligned}
\Delta_{q_t} \cdot (m_0 + m_1) + w_0 + w_1 &= \Delta_{q_t} \cdot (m - p \cdot r_a) + w_0 + w_1 \pmod{q_t} \\
&= \Delta_{q_t} \cdot m + w_0 + w_1 - p \cdot r_a \cdot \Delta_{q_t} \\
&= \Delta_{q_t} \cdot m + v_0 + v_1 - p \cdot r_a \cdot \left(\frac{q_t}{p} - \epsilon_{q_t} \right) \pmod{q_t} \\
&= \Delta_{q_t} \cdot m + w_0 + w_1 + p \cdot r_a \cdot \epsilon_{q_t} \\
&= \Delta_{q_t} \cdot m + w,
\end{aligned}$$

multiplying the result by p/q_t, and rounding. Thus $w = w_0 + w_1 + p \cdot r_a \cdot \epsilon_{q_t}$ and so the ν value on $\mathfrak{c}$ is a correct upper bound since

$$\begin{aligned}
\left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} &= \left\| w_0 + w_1 + p \cdot r_a \cdot \epsilon_{q_t} - \epsilon_{q_t} \cdot (m_0 + m_1 + p \cdot r_a) \right\|_\infty^{\mathsf{can}} \\
&\leq \left\| w_0 - \epsilon_{q_t} \cdot m_0 \right\|_\infty^{\mathsf{can}} + \left\| w_1 - \epsilon_{q_t} \cdot m_1 \right\|_\infty^{\mathsf{can}} \\
&\leq \nu_0 + \nu_1 = \nu.
\end{aligned}$$

We now turn to multiplication for YASHE. We write $\mathfrak{sk} \cdot c_i = \Delta_{q_t} \cdot m_i + w_i + r_i \cdot q_t$. Recall that $\left\| w_i - \epsilon_{q_t} \cdot m_i \right\|_\infty^{\mathsf{can}} \leq \nu_i$, which means that

$$\left\| w_i \right\|_\infty^{\mathsf{can}} \leq \nu_i + p^2 \cdot \sqrt{3 \cdot \phi(m)} = B_{w_i}.$$

Note that this means that

$$\begin{aligned}
\left\| r_i \right\|_\infty^{\mathsf{can}} &= \left\| \frac{1}{q_t} \left(\mathfrak{sk} \cdot c_i - \Delta_{q_t} \cdot m_i - w_i \right) \right\|_\infty^{\mathsf{can}} \\
&\leq \left\| \mathfrak{sk} \cdot c_i \right\|_\infty^{\mathsf{can}} / q_t + \left\| m_i \right\|_\infty^{\mathsf{can}} / p + \left\| w_i - \epsilon_{q_t} \cdot m_i \right\|_\infty^{\mathsf{can}} / q_t \\
&\leq \left\| c_i \right\|_\infty^{\mathsf{can}} / q_t + p \cdot \left\| c_i \cdot f \right\|_\infty^{\mathsf{can}} / q_t + \left\| m_i \right\|_\infty^{\mathsf{can}} / p + \left\| w_i - \epsilon_{q_t} \cdot m_i \right\|_\infty^{\mathsf{can}} / q_t \\
&\leq \sqrt{3 \cdot \phi(m)} + \frac{16}{\sqrt{12}} \cdot p \cdot \sqrt{\phi(m) \cdot h} + \sqrt{3 \cdot \phi(m)} + \frac{\nu_i}{q_t} \\
&\leq 2 \cdot \sqrt{3 \cdot \phi(m)} + \frac{8}{\sqrt{3}} \cdot p \cdot \sqrt{\phi(m) \cdot h} + \frac{\nu_i}{q_t} \\
&= B_{r_i}.
\end{aligned}$$

We write $d' = d'' + \delta$, and note that

$$\begin{aligned}
\left\| \delta \cdot \mathfrak{sk}^2 \right\|_\infty^{\mathsf{can}} &\leq \left\| \delta \right\|_\infty^{\mathsf{can}} + 2 \cdot p \cdot \left\| \delta \cdot f \right\|_\infty^{\mathsf{can}} + p^2 \cdot \left\| \delta \cdot f^2 \right\|_\infty^{\mathsf{can}} \\
&\leq \sqrt{3 \cdot \phi(m)} + \frac{32}{\sqrt{12}} \cdot p \cdot \sqrt{\phi(m) \cdot h} + \frac{40}{\sqrt{12}} \cdot p^2 \cdot h \cdot \sqrt{\phi(m)} \\
&= \sqrt{3 \cdot \phi(m)} + \frac{16}{\sqrt{3}} \cdot p \cdot \sqrt{\phi(m) \cdot h} + \frac{20}{\sqrt{3}} \cdot p^2 \cdot h \cdot \sqrt{\phi(m)} \\
&= B_\delta.
\end{aligned}$$

We set $r_a = \delta \cdot \mathfrak{sk}^2$ and $[m]_p = [m_0 \cdot m_1]_p = [m_0]_p \cdot [m_1]_p - p \cdot r_m$, where we can assume that $\left\| r_m \right\|_\infty^{\mathsf{can}} \leq 16 \cdot p \cdot \phi(m)/12$. We now examine the value of $\mathfrak{sk}^2 \cdot d$, as we take the result modulo q_t we might as well restrict to examining $\mathfrak{sk}^2 \cdot d'$.

$$\begin{aligned}
\mathfrak{sk}^2 \cdot d' &= \frac{p}{q_t} \cdot \left(\mathfrak{sk}^2 \cdot c_0 \cdot c_1\right) + \mathfrak{sk}^2 \cdot \delta \\
&= \frac{p}{q_t} \cdot \left(\mathfrak{sk} \cdot c_0\right) \cdot \left(\mathfrak{sk} \cdot c_1\right) + r_a \\
&= \frac{p}{q_t} \cdot \left(\Delta_{q_t} \cdot m_0 + w_0 + r_0 \cdot q_t\right) \cdot \left(\Delta_{q_t} \cdot m_0 + w_0 + r_0 \cdot q_t\right) + r_a \\
&= \cdots
\end{aligned}$$

The analysis now continues exactly as for the case of the FV scheme, bar the different definitions for B_{r_i} and B_δ. Hence we obtain

$$\begin{aligned}
\left\|w - \epsilon_{q_t} \cdot [m]_p\right\|_\infty^{\mathsf{can}} &\leq 2 \cdot p \cdot \left\|m\right\|_\infty^{\mathsf{can}} + 2 \cdot p^2 \cdot \left\|r_m\right\|_\infty^{\mathsf{can}} \\
&\qquad + \frac{p^3}{q_t} \cdot \left\|m\right\|_\infty^{\mathsf{can}} + \frac{p^4}{q_t} \cdot \left\|r_m\right\|_\infty^{\mathsf{can}} \\
&\qquad + S + p \cdot T + \frac{p}{q} \left(B_{w_0} \cdot B_{w_1} + p \cdot S\right) + p^2 \cdot U + \left\|r_a\right\|_\infty^{\mathsf{can}}, \\
&\leq 2 \cdot p^2 \cdot \sqrt{3 \cdot \phi(m)} + \frac{8}{3} \cdot p^3 \cdot \phi(m) \\
&\qquad + \frac{p^4}{q_t} \cdot \sqrt{3 \cdot \phi(m)} + \frac{p^5}{3 \cdot q_t} \cdot \phi(m) \\
&\qquad + S + p \cdot T + \frac{p}{q} \cdot \left(B_{w_0} \cdot B_{w_1} + p \cdot S\right) + p^2 \cdot U + B_\delta \\
&= F^{\mathsf{YASHE}}(\nu_0, \nu_1),
\end{aligned}$$

where

$$\begin{aligned}
\left\|m_0 \cdot w_1 + m_1 \cdot w_0\right\|_\infty^{\mathsf{can}} &\leq (B_{w_1} + B_{w_2}) \cdot p \cdot \sqrt{3 \cdot \phi(m)} = S, \\
\left\|r_0 \cdot w_1 + r_1 \cdot w_0\right\|_\infty^{\mathsf{can}} &\leq B_{w_1} \cdot B_{r_0} + \cdot B_{w_0} \cdot B_{r_1} = T, \\
\left\|r_0 \cdot m_1 + r_1 \cdot m_0\right\|_\infty^{\mathsf{can}} &\leq (B_{r_0} + B_{r_1}) \cdot p \cdot \sqrt{3 \cdot \phi(m)} = U.
\end{aligned}$$

Again, notice that this new value of ν grows as $\nu_0 \cdot \nu_1 / q_t$, in terms of the input noise values.

E.5 To Scale or Not to Scale

In this section we examine parameter setting for the scale invariant schemes FV and YASHE in the situation where we do not perform a scale operation, and hence do not have a chain of moduli $q_0, \ldots, q_{L-1}$. The ciphertexts are always defined with respect to a single modulus q_{L_1}, which may of course be a product of primes as before for implementation reasons.

At the start of an encryption we have as input a ciphertext with noise $B_0 = B^*_{\mathsf{clean}}$, we perform ζ additions to produce a ciphertext with noise $\zeta \cdot B_0$. We then perform a multiplication to produce something with noise

$$B_1 = \begin{cases} F^*(\zeta \cdot B_0, \zeta \cdot B_0) + B^*_{\mathsf{Ks},1}(L-1) & \text{First variant of } \mathsf{SwitchKey}, \\ F^*(\zeta \cdot B_0, \zeta \cdot B_0) + \frac{B^*_{\mathsf{Ks},2}(L-1)}{P} + B^*_{\mathsf{scale}} & \text{Second variant of } \mathsf{SwitchKey}. \end{cases}$$

Then for the next $L-2$ levels we repeat the procedure; we add ζ times and then perform a multiplication, so that at a bound on the noise after performing a multiplication at multiplicative depth i is

$$B_i = \begin{cases} F^*(\zeta \cdot B_i, \zeta \cdot B_i) + B^*_{\mathsf{Ks},1}(L-1) & \text{First variant of } \mathsf{SwitchKey}, \\ F^*(\zeta \cdot B_i, \zeta \cdot B_i) + \frac{B^*_{\mathsf{Ks},2}(L-1)}{P} + B^*_{\mathsf{scale}} & \text{Second variant of } \mathsf{SwitchKey}. \end{cases}$$

At this point we need to be able to still decrypt the ciphertext, hence we require

$$2 \cdot c_m \cdot B_{L-1} \leq \left\lfloor \frac{q_{L-1}}{p} \right\rfloor.$$

Combined with the equations for security in the main body, this gives us a search space for determining parameters.

E.6 Example Parameters

We outline our example parameters in the following tables; all figures are to be taken as approximate values in any implementation. For the FV and YASHE schemes the line denoted FV-NOP and YASHE-NOP is for the case where ReduceLevel is a NOP command, and hence we keep all ciphertexts at the top level, and make no use of a chain of levels with modulus switching between them.

$L = 2, p = 2, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	793	14	-	31	45	26	8	8	23
BGV	2	1159	15	-	30	45	20	12	12	31
FV	1	610	14	-	21	35	14	5	5	18
FV-NOP	1	592	-	-	-	34	13	4	4	17
FV	2	1067	14	-	21	35	25	9	9	24
FV-NOP	2	976	-	-	-	35	20	8	8	21
NTRU	1	884	15	-	35	50	25	5	5	16
NTRU	2	1342	15	-	35	50	25	8	8	32
YASHE	1	793	15	-	30	45	19	4	4	14
YASHE-NOP	1	738	-	-	-	42	15	3	3	14
YASHE	2	1159	15	-	25	40	25	5	5	24
YASHE-NOP	2	1177	-	-	-	36	30	5	5	24

$L = 2, p = 101, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	976	20	-	34	55	29	13	13	37
BGV	2	1525	20	-	35	55	30	20	20	52
FV	1	976	20	-	35	55	23	13	13	44
FV-NOP	1	884	-	-	-	50	17	10	10	42
FV	2	1616	23	-	32	55	35	21	21	57
FV-NOP	2	1543	-	-	-	51	35	19	19	51
NTRU	1	1433	27	-	53	80	43	13	13	40
NTRU	2	2165	29	-	51	80	40	21	21	84
YASHE	1	1342	27	-	48	75	37	12	12	37
YASHE-NOP	1	1268	-	-	-	71	31	10	10	36
YASHE	2	1799	27	-	33	60	40	13	13	57
YASHE-NOP	2	1799	-	-	-	60	40	13	13	57

$L = 2, p \approx 2^{32}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	1982	46	-	64	110	58	53	53	154
BGV	2	2988	48	-	62	110	55	80	80	200
FV	1	2805	46	-	109	155	70	106	106	341
FV-NOP	1	2768	-	-	-	153	68	103	103	336
FV	2	4360	46	-	109	155	85	164	164	420
FV-NOP	2	4341	-	-	-	154	85	163	163	416
NTRU	1	3720	78	-	127	205	116	93	93	257
NTRU	2	5549	78	-	127	205	100	138	138	552
YASHE	1	4085	78	-	147	225	135	112	112	299
YASHE-NOP	1	4049	-	-	-	223	133	110	110	295
YASHE	2	5183	82	-	108	190	95	120	120	480
YASHE-NOP	2	5018	-	-	-	186	90	113	113	452

$L = 2, p \approx 2^{64}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	3171	78	-	97	175	91	135	135	396
BGV	2	4726	79	-	96	175	85	201	201	501
FV	1	5183	78	-	207	285	136	360	360	1116
FV-NOP	1	5128	-	-	-	282	132	353	353	1107
FV	2	7927	80	-	205	285	150	551	551	1393
FV-NOP	2	7963	-	-	-	282	155	548	548	1397
NTRU	1	6646	142	-	223	365	211	296	296	808
NTRU	2	10122	147	-	223	370	185	457	457	1828
YASHE	1	7652	143	-	277	420	265	392	392	1014
YASHE-NOP	1	7579	-	-	-	416	261	384	384	998
YASHE	2	9573	146	-	204	350	175	409	409	1636
YASHE-NOP	2	9408	-	-	-	346	170	397	397	1582

$$L = 2, p \approx 2^{128}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	5549	142	-	163	305	157	413	413	1215
BGV	2	8292	144	-	161	305	150	617	617	1538
FV	1	9847	143	-	397	540	261	1298	1298	3984
FV-NOP	1	9829	-	-	-	539	260	1293	1293	3974
FV	2	14969	143	-	397	540	280	1973	1973	4970
FV-NOP	2	14950	-	-	-	539	280	1967	1967	4956
NTRU	1	12591	271	-	419	690	408	1060	1060	2854
NTRU	2	18901	274	-	416	690	345	1592	1592	6368
YASHE	1	14694	271	-	534	805	522	1443	1443	3670
YASHE-NOP	1	14621	-	-	-	801	517	1429	1429	3644
YASHE	2	18353	273	-	397	670	335	1501	1501	6004
YASHE-NOP	2	18206	-	-	-	667	330	1482	1482	5913

$$L = 2, p \approx 2^{256}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	10213	271	-	289	560	282	1396	1396	4169
BGV	2	15335	271	-	289	560	280	2096	2096	5241
FV	1	19267	271	-	784	1055	520	4962	4962	15030
FV-NOP	1	19212	-	-	-	1052	516	4934	4934	14994
FV	2	29053	272		783	1055	535	7483	7483	18761
FV-NOP	2	29090	-	-	-	1052	540	7471	7471	18777
NTRU	1	24297	527	-	803	1330	791	3944	3944	10577
NTRU	2	36461	530	-	800	1330	665	5919	5919	23678
YASHE	1	28687	527	-	1043	1570	1025	5497	5497	13919
YASHE-NOP	1	28687	-	-	-	1570	1025	5497	5497	13919
YASHE	2	35912	529	-	781	1310	655	5742	5742	22971
YASHE-NOP	2	35784	-	-	-	1308	650	5713	5713	22819

$\boldsymbol{L = 5, p = 2, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	1890	15	20	30	105	10	48	48	557
BGV	2	3537	16	21	31	110	85	94	94	263
FV	1	1616	14	18	22	90	9	35	35	390
FV-NOP	1	1488	-	-	-	83	13	30	30	222
FV	2	3079	15	18	26	95	75	71	71	199
FV-NOP	2	2896	-	-	-	85	75	60	60	173
NTRU	1	2439	17	28	34	135	14	40	40	427
NTRU	2	4543	16	28	35	135	115	74	74	352
YASHE	1	2165	16	25, 26	28	120	11	31	31	377
YASHE-NOP	1	2055	-	-	-	114	14	28	28	261
YASHE	2	3262	16	19	22	95	85	37	37	181
YASHE-NOP	2	3061	-	-	-	89	80	33	33	159

$\boldsymbol{L = 5, p = 101, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	2439	21	26	36	135	17	80	80	718
BGV	2	4451	22	26	40	140	105	152	152	418
FV	1	2531	21	29	32	140	13	86	86	1018
FV-NOP	1	2092	-	-	-	116	16	59	59	488
FV	2	4817	21	29	32	140	125	164	164	476
FV-NOP	2	3957	-	-	-	118	100	113	113	324
NTRU	1	3902	27	46	50	215	33	102	102	769
NTRU	2	7286	31	46	51	220	180	195	195	907
YASHE	1	3720	28	43, 44	46	205	30	93	93	729
YASHE-NOP	1	3537	-	-	-	195	32	84	84	597
YASHE	2	5274	28	31	34	155	135	99	99	473
YASHE-NOP	2	4945	-	-	-	147	125	88	88	417

$L = 5, p \approx 2^{32}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	4817	47	52	62	265	43	311	311	2232
BGV	2	8750	48	52	66	270	210	576	576	1602
FV	1	8658	48	106, 107	108	475	65	1004	1004	8341
FV-NOP	1	5402	-	-	-	297	68	391	391	2102
FV	2	16066	48	106	109	475	405	1863	1863	5314
FV-NOP	2	9646	-	-	-	299	230	704	704	1949
NTRU	1	10487	79	123	127	575	110	736	736	4583
NTRU	2	18901	81	122	128	575	460	1326	1326	6102
YASHE	1	12042	79	144, 145	147	660	131	970	970	5858
YASHE-NOP	1	10524	-	-	-	577	132	741	741	3981
YASHE	2	16798	80	106	112	510	410	1045	1045	4818
YASHE-NOP	2	13908	-	-	-	427	335	724	724	3312

$L = 5, p \approx 2^{64}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	7835	79	85, 86	95	430	77	822	822	5415
BGV	2	14146	79	85	101	435	340	1502	1502	4178
FV	1	16249	79	202	205	890	125	3530	3530	28669
FV-NOP	1	9536	-	-	-	523	132	1217	1217	6041
FV	2	30242	80	203	206	895	760	6608	6608	18827
FV-NOP	2	16798	-	-	-	525	395	2153	2153	5926
NTRU	1	18718	144	219	224	1025	206	2342	2342	13995
NTRU	2	33626	144	219	224	1025	815	4207	4207	19312
YASHE	1	22560	143	272	275	1235	257	3401	3401	19744
YASHE-NOP	1	19340	-	-	-	1059	259	2500	2500	12722
YASHE	2	31522	144	203	207	960	765	3693	3693	16969
YASHE-NOP	2	25139	-	-	-	781	595	2396	2396	10841

$$L = 5, p \approx 2^{128}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	13688	143	149	160	750	140	2506	2506	15933
BGV	2	24755	145	149	163	755	600	4562	4562	12752
FV	1	31614	144	396	398	1730	258	13352	13352	102887
FV-NOP	1	17767	-	-	-	973	259	4220	4220	20076
FV	2	58411	145	395	400	1730	1465	24670	24670	70232
FV-NOP	2	30882	-	-	-	975	715	7351	7351	20092
NTRU	1	35181	272	412, 413	416	1925	399	8267	8267	48151
NTRU	2	62984	275	411	417	1925	1520	14800	14800	67773
YASHE	1	43686	272	529	531	2390	514	12745	12745	72008
YASHE-NOP	1	36937	-	-	-	2021	515	9112	9112	44872
YASHE	2	60880	272	395	398	1855	1475	13785	13785	63280
YASHE-NOP	2	47399	-	-	-	1488	1105	8609	8609	38615

$$L = 5, p \approx 2^{256}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	25486	271	278	290	1395	269	8679	8679	53692
BGV	2	45973	274	278	292	1400	1115	15713	15713	43941
FV	1	62069	272	780, 781	782	3395	514	51446	51446	391252
FV-NOP	1	34193	-	-	-	1871	514	15618	15618	72473
FV	2	114748	273	780	782	3395	2880	95109	95109	270901
FV-NOP	2	59106	-	-	-	1873	1360	27027	27027	73680
NTRU	1	67922	528	795, 796	801	3715	779	30802	30802	177694
NTRU	2	121608	529	796	803	3720	2930	55222	55222	252657
YASHE	1	85848	529	1041	1043	4695	1024	49201	49201	274786
YASHE-NOP	1	72111	-	-	-	3944	1025	34717	34717	168303
YASHE	2	119595	531	780	784	3655	2885	53359	53359	244314
YASHE-NOP	2	92030	-	-	-	2898	2135	32556	32556	145639

$\boldsymbol{L = 10, p = 2, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	3902	16	21	31	215	11	204	204	4208
BGV	2	7469	16	21	36	220	190	401	401	1148
FV	1	3354	16	18, 19	23	185	7	151	151	4155
FV-NOP	1	3006	-	-	-	166	6	121	121	3492
FV	2	6463	17	18	24	185	170	291	291	852
FV-NOP	2	6043	-	-	-	172	160	253	253	743
NTRU	1	5000	16	28	35	275	6	167	167	7860
NTRU	2	9939	17	29	36	285	260	345	345	1668
YASHE	1	4634	18	26	29	255	10	144	144	3822
YASHE-NOP	1	4287	-	-	-	236	12	123	123	2552
YASHE	2	6829	17	19	26	195	180	162	162	787
YASHE-NOP	2	6372	-	-	-	180	170	140	140	684

$\boldsymbol{L = 10, p = 101, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	5000	22	27	37	275	17	335	335	5766
BGV	2	9573	21	27	38	275	250	642	642	1869
FV	1	5366	23	30	32	295	13	386	386	9156
FV-NOP	1	4159	-	-	-	229	16	232		3560
FV	2	10396	21	30	34	295	275	748	748	2195
FV-NOP	2	8183	-	-	-	234	215	467	467	1364
NTRU	1	8201	28	46, 47	51	450	32	450	450	6785
NTRU	2	15700	30	46	52	450	410	862	862	4158
YASHE	1	7652	29	43	47	420	25	392	392	6893
YASHE-NOP	1	7341	-	-	-	403	30	361	361	5212
YASHE	2	11036	30	31	37	315	290	424	424	2054
YASHE-NOP	2	10487	-	-	-	295	280	377	377	1849

$\boldsymbol{L = 10, p \approx 2^{32}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	9664	50	52	64	530	39	1250	1250	18244
BGV	2	18627	49	53	67	540	480	2455	2455	7094
FV	1	18353	48	106	109	1005	58	4503	4503	82531
FV-NOP	1	9829	-	-	-	539	66	1293	1293	11856
FV	2	35821	49	107	110	1015	945	8876	8876	26017
FV-NOP	2	18609	-	-	-	544	475	2471	2471	7101
NTRU	1	21645	81	122	128	1185	100	3131	3131	40233
NTRU	2	41941	80	123	131	1195	1075	6052	6052	29046
YASHE	1	25212	81	144	147	1380	127	4247	4247	50937
YASHE-NOP	1	21352	-	-	-	1169	130	3046	3046	30445
YASHE	2	36461	80	107	109	1045	950	4651	4651	22409
YASHE-NOP	2	28742	-	-	-	833	740	2922	2922	13960

$\boldsymbol{L = 10, p \approx 2^{64}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	15700	81	85, 86	97	860	75	3296	3296	41094
BGV	2	29785	82	85	98	860	770	6253	6253	18106
FV	1	34906	81	203	205	1910	127	16276	16276	261072
FV-NOP	1	16926	-	-	-	927	130	3830	3830	31146
FV	2	67465	80	203	206	1910	1780	31459	31459	92237
FV-NOP	2	31632	-	-	-	931	800	7189	7189	20557
NTRU	1	38748	144	219	224	2120	203	10027	10027	114748
NTRU	2	73776	146	219	227	2125	1910	19137	19137	91814
YASHE	1	47619	145	273, 274	275	2605	257	15142	15142	168629
YASHE-NOP	1	38985	-	-	-	2133	256	10150	10150	94727
YASHE	2	68745	145	203	206	1975	1785	16573	16573	79679
YASHE-NOP	2	51716	-	-	-	1509	1320	9526	9526	45245

$$L = 10, p \approx 2^{128}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	27407	146	149	162	1500	137	10036	10036	119928
BGV	2	52375	146	150	164	1510	1355	19308	19308	55942
FV	1	67739	147	395	398	3705	252	61272	61272	962127
FV-NOP	1	31047	-	-	-	1699	257	12878	12878	98014
FV	2	131028	146	396	401	3715	3715	118840	118840	348043
FV-NOP	2	57569	-	-	-	1704	1445	23949	23949	68208
NTRU	1	72770	275	411	417	3980	393	35354	35354	393398
NTRU	2	138527	276	412	418	3990	3585	67471	67471	323658
YASHE	1	92067	272	529	531	5035	510	56586	56586	615240
YASHE-NOP	1	74196	-	-	-	4058	514	36753	36753	326923
YASHE	2	133405	273	396	399	3840	3455	62533	62533	300128
YASHE-NOP	2	97518	-	-	-	2858	2475	34021	34021	160990

$$L = 10, p \approx 2^{256}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	51003	275	278	291	2790	267	34740	34740	397762
BGV	2	96823	273	278	293	2790	2505	65951	65951	191116
FV	1	133405	273	780	782	7295	511	237595	237595	$0.362 \cdot 10^7$
FV-NOP	1	59234	-	-	-	3240	515	46855	46855	341632
FV	2	257512	275	780	785	7300	6780	458944	458944	$0.134 \cdot 10^7$
FV-NOP	2	109243	-	-	-	3244	2730	86519	86519	245850
NTRU	1	140813	529	796, 797	801	7700	780	132355	132355	$0.143 \cdot 10^7$
NTRU	2	267207	529	796	803	7700	6910	251158	251158	$0.120 \cdot 10^7$
YASHE	1	181237	529	1042, 1043	1044	9910	1025	219245	219245	$0.233 \cdot 10^7$
YASHE-NOP	1	144527	-	-	-	7903	1025	139428	139428	$0.121 \cdot 10^7$
YASHE	2	262268	531	780	784	7555	6785	241874	241874	$0.116 \cdot 10^7$
YASHE-NOP	2	189030	-	-	-	5551	4785	128089	128089	605094

$\boldsymbol{L = 20, p = 2, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	7835	16	21, 22	32	430	8	822	822	45033
BGV	2	15883	18	22	36	450	420	1744	1744	5118
FV	1	6738	18	18, 19	24	370	2	608	608	113210
FV-NOP	1	6171	-	-	-	339	10	510	510	17824
FV	2	13688	16	19	27	385	365	1286	1286	3792
FV-NOP	2	12499	-	-	-	350	335	1068	1068	3158
NTRU	1	10487	18	29	35	575	11	736	736	39213
NTRU	2	21279	18	30	37	595	570	1545	1545	7597
YASHE	1	9390	17	26	30	515	7	590	590	44020
YASHE-NOP	1	8859	-	-	-	486	12	525	525	21811
YASHE	2	14511	17	20	23	400	395	708	708	3525
YASHE-NOP	2	13249	-	-	-	366	360	591	591	2940

$\boldsymbol{L = 20, p = 101, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	9939	21	27	38	545	14	1322	1322	52803
BGV	2	20182	24	28	42	570	535	2808	2808	8253
FV	1	10853	22	30	33	595	10	1576	1576	95381
FV-NOP	1	8366	-	-	-	459	12	937	937	36796
FV	2	22102	23	31	34	615	595	3318	3318	9847
FV-NOP	2	16871	-	-	-	469	455	1931	1931	5737
NTRU	1	16615	31	46	51	910	28	1845	1845	61829
NTRU	2	33260	31	47	53	930	890	3775	3775	18554
YASHE	1	15883	31	44	47	870	26	1686	1686	58129
YASHE-NOP	1	15060	-	-	-	825	30	1516	1516	43224
YASHE	2	23017	29	32	35	640	620	1798	1798	8878
YASHE-NOP	2	21426	-	-	-	598	575	1564	1564	7699

$\boldsymbol{L = 20, p \approx 2^{32}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	19542	51	53	65	1070	41	5104	5104	138332
BGV	2	38016	48	53	68	1070	1010	9930	9930	29235
FV	1	38107	49	107, 108	109	2085	62	19397	19397	671724
FV-NOP	1	18792	-	-	-	1029	66	4720	4720	78324
FV	2	74946	48	107	111	2085	2015	38159	38159	113196
FV-NOP	2	36699	-	-	-	1038	970	9300	9300	27291
NTRU	1	44326	83	123	128	2425	106	13121	13121	313304
NTRU	2	86580	82	123	129	2425	2310	25629	25629	125716
YASHE	1	51917	83	145	147	2840	127	17998	17998	420486
YASHE-NOP	1	43119	-	-	-	2359	129	12416	12416	239478
YASHE	2	75696	81	107	113	2120	2020	19589	19589	96098
YASHE-NOP	2	58575	-	-	-	1649	1555	11790	11790	57609

$\boldsymbol{L = 20, p \approx 2^{64}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	31248	80	85, 86	98	1710	72	13045	13045	322874
BGV	2	61520	80	86	102	1730	1635	25983	25983	76524
FV	1	72038	80	203	206	3940	121	69294	69294	$0.232 \cdot 10^7$
FV-NOP	1	31797	-	-	-	1740	129	13507	13507	195701
FV	2	142368	81	204	207	3960	3825	137640	137640	408230
FV-NOP	2	61612	-	-	-	1750	1620	26323	26323	77015
NTRU	1	78897	148	219	225	4315	199	41557	41557	942670
NTRU	2	154532	145	220	225	4330	4120	81680	81680	400477
YASHE	1	97554	145	273	276	5335	254	63531	63531	$0.139 \cdot 10^7$
YASHE-NOP	1	78403	-	-	-	4288	258	41039	41039	723114
YASHE	2	143649	145	204	208	4025	3830	70579	70579	346058
YASHE-NOP	2	105145	-	-	-	2970	2780	38120	38120	185723

$\boldsymbol{L = 20, p \approx 2^{128}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	55027	146	150, 151	163	3010	138	40437	40437	922439
BGV	2	107249	145	150	165	3010	2855	78813	78813	232381
FV	1	140356	149	396	398	7675	254	262996	262996	$0.820 \cdot 10^8$
FV-NOP	1	57716	-	-	-	3157	258	44484	44484	588819
FV	2	275895	146	396	401	7675	7410	516966	516966	$0.153 \cdot 10^7$
FV-NOP	2	111108	-	-	-	3166	2910	85880	85880	250698
NTRU	1	148313	277	412	417	8110	394	146828	146828	$0.316 \cdot 10^7$
NTRU	2	289248	275	412	419	8110	7705	286352	286352	$0.140 \cdot 10^7$
YASHE	1	189194	273	530	532	10345	511	238917	238917	$0.507 \cdot 10^7$
YASHE-NOP	1	148807	-	-	-	8137	513	147807	147807	$0.249 \cdot 10^7$
YASHE	2	278365	273	396	399	7800	7420	265044	265044	$0.129 \cdot 10^7$
YASHE-NOP	2	197937	-	-	-	5603	5220	135380	135380	658396

$\boldsymbol{L = 20, p \approx 2^{256}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes			$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes)		
			p_0	p_i	p_{L-1}			Ciphertext	Key	Extended Key
BGV	1	101853	273	278	293	5570	264	138506	138506	$0.306 \cdot 10^7$
BGV	2	199254	273	279	295	5590	5305	271931	271931	801929
FV	1	276170	276	780, 781	783	15100	507	$0.101 \cdot 10^7$	$0.101 \cdot 10^7$	$0.313 \cdot 10^8$
FV-NOP	1	109389	-	-	-	5982	513	159757	159757	$0.202 \cdot 10^7$
FV	2	543498	274	781	783	15115	14600	$0.200 \cdot 10^7$	$0.200 \cdot 10^7$	$0.594 \cdot 10^7$
FV-NOP	2	209790	-	-	-	5991	5480	306848	306848	894373
NTRU	1	286413	530	796	802	15660	776	547513	547513	$0.115 \cdot 10^8$
NTRU	2	559137	530	797	804	15680	14890	$0.107 \cdot 10^7$	$0.107 \cdot 10^7$	$0.524 \cdot 10^7$
YASHE	1	371815	529	1042	1045	29330	1022	922776	922776	$0.192 \cdot 10^8$
YASHE-NOP	1	289449	-	-	-	15826	1020	559182	559182	$0.923 \cdot 10^7$
YASHE	2	553924	533	781	1089	15680	14605	$0.106 \cdot 10^7$	$0.106 \cdot 10^7$	$0.515 \cdot 10^7$
YASHE-NOP	2	383266	-	-	-	10860	10095	508089	508089	$0.246 \cdot 10^7$

$\boldsymbol{L = 30, p = 2, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes			$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes)		
			p_0	p_i	p_{L-1}			Ciphertext	Key	Extended Key
BGV	1	12134	16	22	33	665	9	1969	1969	211970
BGV	2	23877	16	22	35	667	640	3888	3888	11507
FV	1	10432	16	19	24	572	4	1456	1456	302005
FV-NOP	1	9390	-	-	-	515	8	1180	1180	110829
FV	2	20694	16	19	25	573	560	2894	2894	8619
FV-NOP	1	19103	-	-	-	531	515	2476	2476	7354
NTRU	1	15792	17	29	36	865	8	1667	1667	261781
NTRU	2	32327	18	30	36	894	875	3527	3527	17489
YASHE	1	14146	17	26	30	775	4	1338	1338	375415
YASHE-NOP	1	13505	-	-	-	740	12	1219	1219	109752
YASHE	2	21828	17	20	23	600	595	1598	1598	7967
YASHE-NOP	2	20218	-	-	-	557	550	1374	1374	6838

$\boldsymbol{L = 30, p = 101, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	14914	22	27	39	817	12	2974	2974	295168
BGV	2	30370	22	28	41	847	815	6280	6280	18603
FV	1	16341	22	30	33	895	7	3570	3570	662200
FV-NOP	1	12664	-	-	-	694	13	2145	2145	167403
FV	2	33425	22	31	34	924	905	7540	7540	22465
FV-NOP	2	25651	-	-	-	709	695	4440	4440	13232
NTRU	1	25523	29	47	52	1397	30	4352	4352	296759
NTRU	2	50399	29	47	52	1397	1360	8594	8594	42518
YASHE	1	23895	29	44	47	1308	24	3815	3815	303797
YASHE-NOP	1	22834	-	-	-	1250	25	3484	3484	254815
YASHE	2	34723	29	32	35	960	940	4069	4069	20175
YASHE-NOP	2	32693	-	-	-	904	885	3607	3607	17886

$$L = 30, p \approx 2^{32}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	29199	48	53	66	1598	39	11391	11391	684789
BGV	2	58539	48	54	67	1627	1575	23252	23252	69014
FV	1	57661	48	107	110	3154	59	44400	44400	$0.347 \cdot 10^7$
FV-NOP	1	27809	-	-	-	1522	64	10333	10333	364860
FV	2	115169	49	108	110	3183	3115	89497	89497	266581
FV-NOP	2	54972	-	-	-	1537	1470	20627	20627	60984
NTRU	1	66770	80	123	128	3652	103	29766	29766	$0.155 \cdot 10^7$
NTRU	2	132619	81	124	129	3682	3570	59607	59607	294410
YASHE	1	78385	80	145	147	4287	126	41020	41020	$0.205 \cdot 10^7$
YASHE-NOP	1	64959	-	-	-	3553	129	28173	28173	$0.114 \cdot 10^7$
YASHE	2	115827	80	108	110	3214	3120	45442	45442	224556
YASHE-NOP	2	88592	-	-	-	2470	2375	26711	267113	131503

$$L = 30, p \approx 2^{64}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	47290	80	86	99	2587	73	29867	29867	1556920
BGV	2	93036	80	86	100	2588	2500	58783	58783	174351
FV	1	109700	81	204	206	5999	126	160666	160666	$0.112 \cdot 10^7$
FV-NOP	1	46741	-	-	-	2557	128	29178	29178	870117
FV	2	217089	81	204	207	6000	5870	318001	318001	947114
FV-NOP	2	91646	-	-	-	2572	2440	57547	57547	169688
NTRU	1	119413	145	220	225	6530	202	95186	95186	$0.453 \cdot 10^7$
NTRU	2	235106	145	220	225	6530	6325	187407	187407	925270
YASHE	1	147490	145	273	276	8065	250	145203	145203	$0.690 \cdot 10^7$
YASHE-NOP	1	117876	-	-	-	6446	257	92752	92752	$0.344 \cdot 10^7$
YASHE	2	218424	145	204	206	6063	5880	161658	161658	798532
YASHE-NOP	2	158721	-	-	-	4434	4245	85909	85909	422222

$$L = 30, p \approx 2^{128}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3$$

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes p_0	p_i	p_{L-1}	$\log_2 q_{L-1}$	$\log_2 T$ or $\log_2 P$	Sizes (kBytes) Ciphertext	Key	Extended Key
BGV	1	82427	144	150	164	4508	136	90717	90717	$0.442 \cdot 10^7$
BGV	2	162141	145	150	166	4511	4355	178568	178568	529531
FV	1	212735	145	396	399	11632	252	604134	604134	$0.408 \cdot 10^8$
FV-NOP	1	84439	-	-	-	4618	257	952000	95200	$0.256 \cdot 10^7$
FV	2	421769	145	397	399	11660	11400	$0.120 \cdot 10^7$	$0.120 \cdot 10^7$	$0.357 \cdot 10^7$
FV-NOP	2	164720	-	-	-	4632	4375	186275	186275	548490
NTRU	1	223673	274	412	417	12230	393	333925	333925	$0.107 \cdot 10^8$
NTRU	2	440975	273	413	418	12255	11855	659686	659686	$0.325 \cdot 10^7$
YASHE	1	286139	273	530	532	15645	510	546465	546465	$0.173 \cdot 10^8$
YASHE-NOP	1	223491	-	-	-	12220	514	333381	333381	$0.825 \cdot 10^7$
YASHE	2	424202	273	397	399	11788	11405	61041	610411	$0.301 \cdot 10^7$
YASHE-NOP	2	298522	-	-	-	8352	7970	304352	304352	$0.149 \cdot 10^7$