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 YASHE and FV schemes. As an additional optimization, we also investigate the ffect 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 [3]. The main improvements here have come through the use of SIMD techniques (first introduced in the context of Gentry's original scheme [7] by Smart and Vercauteren [17], but then extended to the Ring-LWE based schemes by Gentry et al [3]). SIMD techniques in the ring setting allow for a small overall asymptotic overhead in using SHE schemes [8] 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 [9]. Other improvements in the ring based setting have come from the use of modulus switching to a larger modulus, so as to perform key switching [9], the use of scale invariant versions [6, 1], and the use of NTRU to enable key homomorphic schemes [14].

The scale invariant schemes, originally introduced in [2], are particularly interesting, they place the message space in the "upper bits" of the decryption equation, as opposed to the lower bits. This enables a more effective noise control mechanism to be employed which does not on the face of it require modulus switching to keep the noise within bounds. However, the downside is that they require a more complex rounding operation to be performed in the multiplication procedure.

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 [5, 14], 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 [6, 1] usually focus on plaintext moduli of two, and discount larger moduli. But many applications, e.g. usage in the SPDZ [4] 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 [3, 8, 9], and its scale-invariant version [6] (called FV in what follows), the basic NTRU scheme [5, 14], and its scale-invariant version YASHE [1]. A previous study [12] 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. Our results are broadly in line with [12] (where we have a direct comparison) for YASHE, but our estimates for FV appear slightly better.

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 applying modulus switching to the scale invariant schemes (where its use is often discounted as not being needed). The use of modulus 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 [9], 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 [13]. 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. In addition, the operations needed for the scale invariant schemes are not directly compatible with the efficient double-CRT representation of ring elements introduced in [9], thus even if ciphertext sizes for the scale invariant schemes are smaller than for the non-scale invariant schemes, the actual computation times might be much larger.

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, 9], 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 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 $\equiv$.

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 [3, 8, 17, 9] 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 a number of applications p is likely to split completely in $\mathbb{A}$, i.e. $p \equiv 1 \pmod{m}$. This is especially true in applications not requiring bootstrapping, and hence only requiring evaluation of low depth arithmetic circuits.

Canonical Embedding Norm: Following the work in [15], 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 $a \in \mathbb{A}$.

where $\|a\|_\infty$ and $\|a\|_1$ refer to the relevant norms on the coefficient vectors of a in the power basis. 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 [4] for a discussion of c_m.

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 "rounded 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 condition that it has exactly h nonzero entries.

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 the work in [9] 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 $(q-1)^2/12 \approx q^2/12$, 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 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 [9] the messages were always given a norm of $\|m\|_\infty^{\mathsf{can}} \le p \cdot \phi(m)/2$, i.e. a worst case bound. We shall assume that messages, and similar quantities, behave as if selected uniformly at random and hence estimate $\|m\|_\infty^{\mathsf{can}} \le 6 \cdot p \cdot \sqrt{\phi(m)/12} = p \cdot \sqrt{3 \cdot \phi(m)}$. 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 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 erfc$(3.4) \approx 2^{-19}$, 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 [3, 8, 9], FV in [6], NTRU in [5, 14] 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"

---

[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.

$p_0, p_1, \ldots, p_{L-1}$ 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 [9], 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 the Scaling operation 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 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 $0 \le \epsilon_q < 1$.

3.1 Key Generation

We utilize the following methods for key generation, they sample the secret key in all cases, from a sparse distribution, this follows the choices made in [9]. This leads to more efficient homomorphic operations (since noise growth depends on the size of the secret key in many situations). However, such choices might lead to security weaknesses, which would need to be considered in any commercial deployment.

$\underline{\mathsf{KeyGen}^{\mathsf{BGV}}(): \text{Sample}}$ $\mathfrak{sk} \leftarrow \mathcal{HWT}(h)$, $a \leftarrow \mathcal{U}_{q_{L-1}}$, 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 $b \leftarrow [a \cdot \mathfrak{sk} + p \cdot e]_{q_{L-1}}$.

$\underline{\mathsf{KeyGen}^{\mathsf{FV}}(): \text{Sample}}$ $\mathfrak{sk} \leftarrow \mathcal{HWT}(h)$, $a \leftarrow \mathcal{U}_{q_{L-1}}$, 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 $b \leftarrow [a \cdot \mathfrak{sk} + e]_{q_{L-1}}$.

$\underline{\mathsf{KeyGen}^{\mathsf{NTRU}}(): \text{Sample}}$ $f, g \leftarrow \mathcal{HWT}(h)$. 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.

$\underline{\mathsf{KeyGen}^{\mathsf{YASHE}}(): \text{Sample}}$ $f, g \leftarrow \mathcal{HWT}(h)$. 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. As for key generation we select slightly simpler distributions than the theory would imply so as to ensure noise growth is not as bad as it would otherwise be. 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.

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

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 $c_0 - \mathfrak{sk} \cdot c_1 \pmod{q_t}$ 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) $m \cdot \Delta_{q_t} + w + r \cdot q_t$. 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

$\left\|w - \epsilon_{q_t} \cdot m\right\|_\infty^{\mathsf{can}}$. The decryption procedure will then work when $c_m \cdot \nu < \Delta_{q_t}/2$, since in this case we have

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

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 $\left\|w - \epsilon_{q_t} \cdot m\right\|_\infty^{\mathsf{can}}$ 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 $t_{out} \leq t_{in}$. 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 $\mathfrak{c}'$.

$\underline{\mathsf{Scale}^{\mathsf{BGV}}(\mathfrak{c}, Q)}$:
- 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 $c'_i \leftarrow (c_i + \delta_i)/P$.
- $\nu' \leftarrow \nu/P + B^{\mathsf{BGV}}_{\mathsf{scale}}$.
- 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 $c' \leftarrow (c + \delta)/P$.
- $\nu' \leftarrow \nu/P + B^{\mathsf{NTRU}}_{\mathsf{scale}}$.
- 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 $c'_i \leftarrow (c_i + \delta_i)/P$.
- $\nu' \leftarrow \nu/P + B^{\mathsf{FV}}_{\mathsf{scale}}$.
- 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 $c' \leftarrow (c + \delta)/P$.
- $\nu' \leftarrow \nu/P + B^{\mathsf{YASHE}}_{\mathsf{Scale}}$.
- 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 $P \equiv 1 \pmod{p}$.

The Scale^* function was originally presented in [3] 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 [6]. This is due to the modulus one is working with which 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.

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 [8] for more details on the latter). For all schemes we present two switch key operations:

– One based on decomposition modulo a general modulus T. See [11] 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 [9], 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 SwitchKeyGen and 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 $\left\lceil \log_T(q_{L-1}) \right\rceil - 1$ do
  ⋆ $a_i \leftarrow \mathcal{U}_{q_{L-1}}$.
  ⋆ $e_i \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$.
  ⋆ $b_i \leftarrow [a_i \cdot \mathfrak{sk} + p \cdot e_i + T^i \cdot \mathfrak{sk}']_{q_{L-1}}$.
– $\mathfrak{ksd} \leftarrow (T, \{a_i, b_i\}_{i=0}^{\lceil \log_T q_{L-1} \rceil - 1})$.
– 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

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

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

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

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 and multiplication operations. For reasons of space we give the addition and multiplication methods in 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 $L - 1$.

3.7 Security and Parameters

In this section we outline how we select parameters in the case where $\mathsf{ReduceLevel}^*$ is not a NOP (a no-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 [9] 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. We follow the analysis in [9], which itself follows on from the analysis by Lindner and Peikert [13][3]. 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 $\log$s here are natural logarithms.

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

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

[3] One could take into account a more elaborate analysis here, for example looking at BKW style attacks e.g. [10]. But for simplicity we follow the same analysis as in [9].

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

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

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,

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

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

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

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 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

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

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.

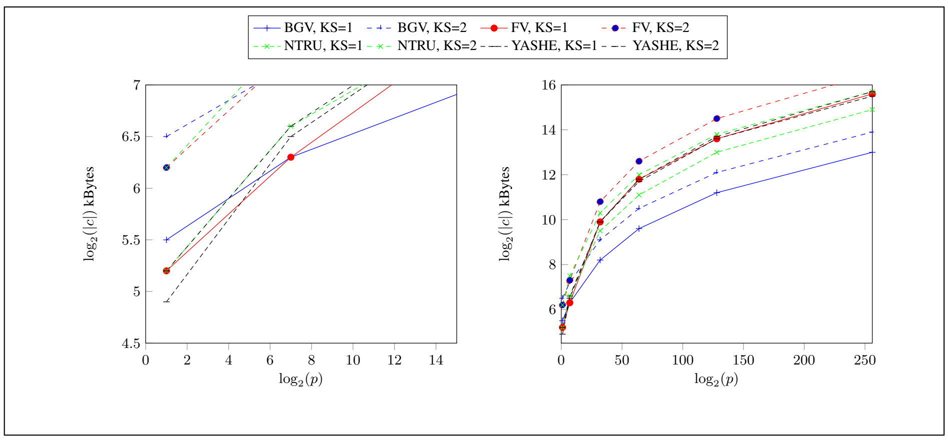

Fig. 4: Size of required ciphertext for various sizes of plaintext modulus when $L = 5$. The graph on the left zooms into the portion of the right graph for small values of $\log_2 p$.

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

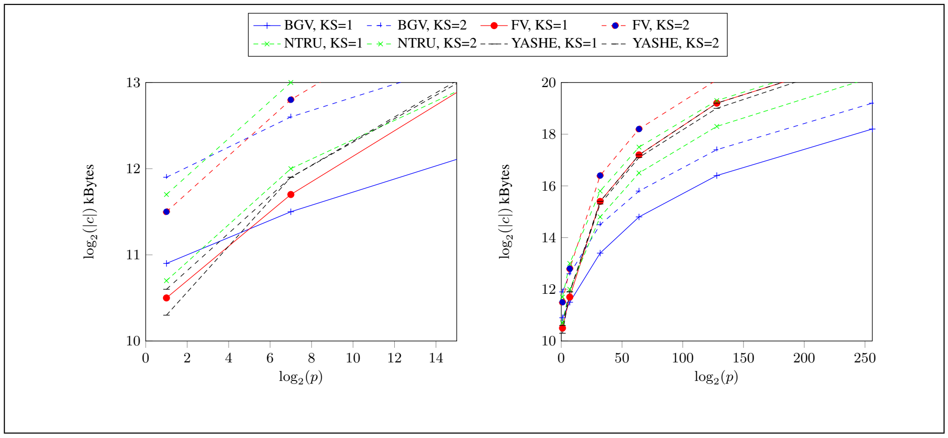

Fig. 5: Size of required ciphertext for various sizes of plaintext modulus when $L = 30$. The graph on the left zooms into the portion of the right graph for small values of $\log_2 p$.

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 rounded 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 [9]. 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 [4]. 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 4 and 5. 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 [4] 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.

Examining Fig. 6 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 agree-

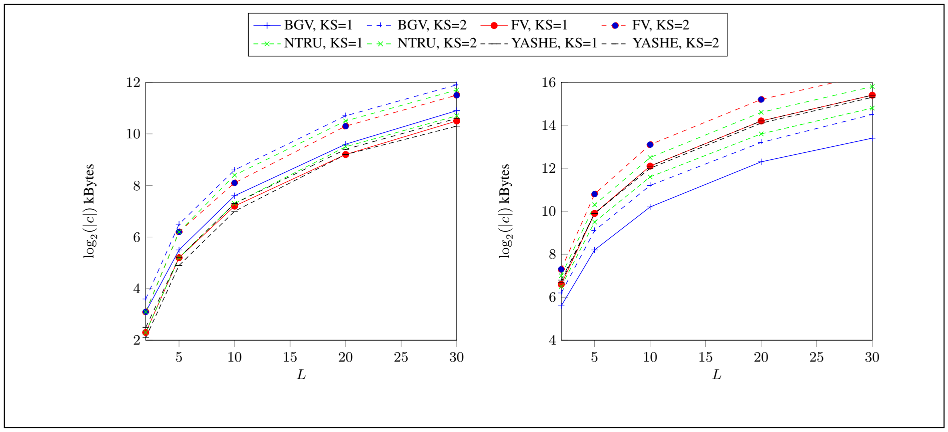

Fig. 6: Size of required ciphertext for various values of L when $p = 2$ and $p \approx 2^{32}$.

ment number ICT-644209. The authors would like to thank Steven Galbraith for comments on an earlier version of this manuscript.

References

1. J. W. Bos, K. E. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In M. 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. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In Safavi-Naini and Canetti [16], pages 868–886.
3. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. In *Innovations in Theoretical Computer Science (ITCS'12)*, 2012. Available at `http://eprint.iacr.org/2011/277`.
4. I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In Safavi-Naini and Canetti [16], pages 643–662.
5. Y. Doröz, Y. Hu, and B. Sunar. Homomorphic AES evaluation using the modified LTV scheme. *Des. Codes and Cryptography*, XXXX:XXXX–XXXX, 2015.
6. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.
7. C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.
8. C. Gentry, S. Halevi, and N. 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`.
9. C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the AES circuit. In Safavi-Naini and Canetti [16], pages 850–867.
10. P. Kirchner and P. Fouque. An improved BKW algorithm for LWE with applications to cryptography and lattices. In R. Gennaro and M. Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 43–62. Springer, 2015.

11. K. Lauter, M. Naehrig, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *CCSW*, pages 113–124. ACM, 2011.
12. T. Lepoint and M. Naehrig. A comparison of the homomorphic encryption schemes FV and YASHE. In D. Pointcheval and D. 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.
13. R. Lindner and C. 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.
14. A. Lòpez-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*. ACM, 2012.
15. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, 2010.
16. R. Safavi-Naini and R. 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.
17. N. P. Smart and F. 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,

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

$\underline{B^{\mathsf{FV}}_{\mathsf{clean}}}$: 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

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

$$
\begin{aligned}
&\leq p \cdot \sqrt{3 \cdot \phi(m)} \\
&\qquad + 2 \cdot \sigma \cdot \left(\frac{8 \cdot \phi(m)}{\sqrt{2}} + 3 \cdot \sqrt{\phi(m)} + 8 \cdot \sqrt{h \cdot \phi(m)} \right) \\
&= B^{\mathsf{FV}}_{\mathsf{clean}}.
\end{aligned}
$$

Note, compared to the $B^{\mathsf{BGV}}_{\mathsf{clean}}$ we do not have a dependence on p in the latter terms, but we still have a dependence on p in the first term.

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

$$
\begin{aligned}
\left\| c \cdot \mathfrak{sk} \right\|^{\mathsf{can}}_{\infty} &= \left\| e_1 \cdot \mathfrak{pk} \cdot \mathfrak{sk} + (p \cdot e_0 + m) \cdot (1 + p \cdot f) \right\|^{\mathsf{can}}_{\infty} \\
&\leq p \cdot \left\| e_1 \cdot g \right\|^{\mathsf{can}}_{\infty} + p \cdot \left\| e_0 \right\|^{\mathsf{can}}_{\infty} + \left\| m \right\|^{\mathsf{can}}_{\infty} \\
&\qquad + p^2 \cdot \left\| e_0 \cdot f \right\|^{\mathsf{can}}_{\infty} + p \cdot \left\| m \cdot f \right\|^{\mathsf{can}}_{\infty} \\
&\leq 16 \cdot p \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} + 6 \cdot p \cdot \sigma \cdot \sqrt{\phi(m)} + p \cdot \sqrt{3 \cdot \phi(m)} \\
&\qquad + 16 \cdot p^2 \cdot \sigma \cdot \sqrt{h \cdot \phi(m)} + 16 \cdot p^2 \cdot \sqrt{h \cdot \phi(m)/12} \\
&= \left(16 \cdot p \cdot (1 + p) \cdot \sigma + \frac{8}{\sqrt{3}} \cdot p^2 \right) \cdot \sqrt{h \cdot \phi(m)} \\
&\qquad + p \cdot (6 \cdot \sigma + \sqrt{3}) \cdot \sqrt{\phi(m)} \\
&= B^{\mathsf{NTRU}}_{\mathsf{clean}}.
\end{aligned}
$$

$\underline{B^{\mathsf{YASHE}}_{\mathsf{clean}}}$: 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

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

B Estimating B^*_{scale}

$\underline{\mathsf{Scale}^{\mathsf{BGV}}(\mathfrak{c}, Q)}$: For correctness of the method presented we appeal to the proof of Lemma 13 in the full version of [8]. 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

$$\begin{aligned} \left\| c'_0 - \mathfrak{sk} \cdot c'_1 \right\|_\infty^{\mathsf{can}} &= \left\| \frac{1}{P} \cdot \Big(c_0 - \mathfrak{sk} \cdot c_1 + \delta_0 - \mathfrak{sk} \cdot \delta_1 \Big) \right\|_\infty^{\mathsf{can}} \\ &\leq \frac{\nu}{P} + \left\| \tau_0 - \mathfrak{sk} \cdot \tau_1 \right\|_\infty^{\mathsf{can}} \\ &\leq \frac{\nu}{P} + \left\| \tau_0 \right\|_\infty^{\mathsf{can}} + \left\| \mathfrak{sk} \cdot \tau_1 \right\|_\infty^{\mathsf{can}}. \end{aligned}$$

Thus we set

$$\begin{aligned} B^{\mathsf{BGV}}_{\mathsf{scale}} &= 6 \cdot p \cdot \sqrt{\phi(m)/12} + 16 \cdot p \cdot \sqrt{\phi(m) \cdot h/12} \\ &= p \cdot \Big(\sqrt{3 \cdot \phi(m)} + 8 \cdot \sqrt{\phi(m) \cdot h/3} \Big). \end{aligned}$$

$\underline{\mathsf{Scale}^{\mathsf{FV}}(\mathfrak{c}, Q)}$: 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

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

and

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

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

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

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 $\delta \equiv 0 \pmod p$ and 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

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

$\underline{\mathsf{Scale}^{\mathsf{YASHE}}(\mathfrak{c}, Q)}$: 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

$$c' \cdot \mathfrak{sk} = \frac{1}{P} \cdot (c \cdot \mathfrak{sk} + \delta \cdot \mathfrak{sk})$$

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

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 $\mathsf{ReduceLevel}^*$ operations for our four schemes are presented in Fig. 7.

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. 8.

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.

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

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

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

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

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

Fig. 7: The $\mathsf{ReduceLevel}^*$ Operations for BGV, FV, NTRU and YASHE.

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

$\mathsf{SwitchKey}_1^{\mathsf{BGV}}(\mathfrak{ksd}, (\mathfrak{d}, t, \nu))$:
- 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 $((c_0, c_1), t, \nu')$.

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

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

Fig. 8: The two variants of Key Switching for BGV.

$$= d_0 - d_1 \cdot \mathfrak{sk} + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} (d_{2,i} \cdot b_i - d_{2,i} \cdot a_i \cdot \mathfrak{sk})$$
$$= d_0 - d_1 \cdot \mathfrak{sk} + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} \left(p \cdot e_i + T^i \cdot \mathfrak{sk}'\right) \cdot d_{2,i}$$
$$= d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}' + p \cdot \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot e_i.$$

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 depends 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 [9] 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 [9].

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

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

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. 9.

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

- 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. 9: 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 $d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}' = m \cdot \Delta_{q_t} + w + r \cdot q_t$

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

$$
\begin{aligned}
&= d_0 - d_1 \cdot \mathfrak{sk} + d_2 \cdot \mathfrak{sk}' + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot e_i \\
&= m \cdot \Delta_{q_t} + w + r \cdot q_t + \sum_{i=0}^{\lceil \log_T q_t \rceil - 1} d_{2,i} \cdot e_i \\
&= m \cdot \Delta_{q_t} + w' + r \cdot q_t.
\end{aligned}
$$

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 depends 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 [9], 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.

$$
\begin{aligned}
c_0 - \mathfrak{sk} \cdot c_1 &= P \cdot (d_0 - \mathfrak{sk} \cdot d_1) + b \cdot d_2 - a \cdot d_2 \cdot \mathfrak{sk} \\
&= P \cdot (d_0 - \mathfrak{sk} \cdot d_1) + d_2 \cdot (e + P \cdot \mathfrak{sk}') \\
&= P \cdot (d_0 - \mathfrak{sk} \cdot d_1 + \mathfrak{sk}' \cdot d_2) + e \cdot d_2 \\
&= P \cdot (m \cdot \Delta_{q_t} + w + r \cdot q_t) + e \cdot d_2 \\
&= m \cdot P \cdot \Delta_{q_t} + P \cdot w + P \cdot r \cdot q_t + e \cdot d_2 \\
&= m \cdot (\Delta_{P \cdot q_t} + \epsilon_Q - P \cdot \epsilon_{q_t}) + P \cdot w + P \cdot r \cdot q_t + e \cdot d_2 \\
&= m \cdot \Delta_{P \cdot q_t} + w' + r' \cdot q_t
\end{aligned}
$$

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

$$
\left\|w' - \epsilon_{P \cdot q_t} \cdot m\right\|_\infty^{\mathsf{can}} = \left\|P \cdot w - P \cdot \epsilon_{q_t} \cdot m + e \cdot d_2\right\|_\infty^{\mathsf{can}}
$$

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

So we set

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

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. 10.

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

- 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 $\mathfrak{ksd}$.

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

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

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

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

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

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

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

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

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

$$= m + p \cdot \left(e + f \cdot (m + p \cdot e) + \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} \right).$$

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 $B_{\mathsf{Ks},1}^{\mathsf{NTRU}}(t)$,

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

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

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

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 $\left\| c' \cdot \mathfrak{sk} \right\|_\infty^{\mathsf{can}}$

$$\begin{aligned}
\left\| c' \cdot \mathfrak{sk} \right\|_\infty^{\mathsf{can}} &= \left\| P \cdot \mathfrak{sk}' \cdot c + p \cdot g \cdot s' \cdot c + p \cdot e' \cdot c + p^2 \cdot e' \cdot c \cdot f + p \cdot P \cdot \mathfrak{sk}' \cdot c \cdot f \right\|_\infty^{\mathsf{can}} \\
&\leq P \cdot \left\| c \cdot \mathfrak{sk}' \right\|_\infty^{\mathsf{can}} + p \cdot \left\| g \cdot s' \cdot c \right\|_\infty^{\mathsf{can}} + p \cdot \left\| e' \cdot c \right\|_\infty^{\mathsf{can}} + p^2 \cdot \left\| e' \cdot c \cdot f \right\|_\infty^{\mathsf{can}}
\end{aligned}$$

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

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. 11.

$\underline{\mathsf{SwitchKeyGen}_1^{\mathsf{YASHE}}(\mathfrak{sk}', \mathfrak{sk})}$:
- 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. 11: The two variants of Key Switching for YASHE.

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

$$\begin{aligned}
\mathfrak{sk} \cdot c' &= \sum_i (e_{1,i} \cdot \mathfrak{pk} + e_{0,i} + T^i \cdot \mathfrak{sk}') \cdot c_i \cdot \mathfrak{sk} \\
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + c \cdot \mathfrak{sk}' \cdot \mathfrak{sk} \\
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + c \cdot \mathfrak{sk}' \cdot (1 + p \cdot f) \\
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + (\Delta_{q_t} \cdot m + w + r \cdot q_t) \\
&\quad + p \cdot f \cdot (\Delta_{q_t} \cdot m + w + r \cdot q_t) \\
&= p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} + (\Delta_{q_t} \cdot m + w + r \cdot q_t) \\
&\quad - p \cdot f \cdot m \cdot \epsilon_{q_t} - p \cdot f \cdot m \cdot \epsilon_{q_t} + p \cdot f \cdot (w + r \cdot q_t) \\
&= \Delta_{q_t} \cdot m + w' + r' \cdot q_t,
\end{aligned}$$

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

$$\begin{aligned}
\left\| w' - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} &\le \Big\| p \cdot \sum_i e_{1,i} \cdot g \cdot c_i + \sum_i e_{0,i} \cdot c_i \cdot \mathfrak{sk} - p \cdot f \cdot m \cdot \epsilon_{q_t} \Big\|_\infty^{\mathsf{can}} \\
&\quad + \left\| w \cdot (1 + p \cdot f) - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} \\
&\le \left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \Big\| \sum_i e_{1,i} \cdot g \cdot c_i \Big\|_\infty^{\mathsf{can}} \\
&\quad + \Big\| \sum_i e_{0,i} \cdot c_i \cdot (1 + pf) \Big\|_\infty^{\mathsf{can}} \\
&\quad - p \cdot \epsilon_{q_t} \cdot \left\| f \cdot m \right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \left\| f \cdot w \right\|_\infty^{\mathsf{can}} \\
&\le \left\| w - \epsilon_{q_t} \cdot m \right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \Big\| \sum_i e_{1,i} \cdot g \cdot c_i \Big\|_\infty^{\mathsf{can}} \\
&\quad + \Big\| \sum_i e_{0,i} \cdot c_i \Big\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \Big\| \sum_i f \cdot e_{0,i} \cdot c_i \Big\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \left\| f \cdot m \right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \left\| f \cdot (w - \epsilon_{q_t} + \epsilon_{q_t}) \right\|_\infty^{\mathsf{can}} \\
&\le \nu + 40 \cdot p \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12}
\end{aligned}$$

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

Let $B_{\mathsf{Ks},1}^{\mathsf{YASHE}}(t) = \frac{8}{\sqrt{3}} \cdot \left(1 + 5 \cdot p \cdot \sqrt{h}\right) \cdot \lceil \log_T(q_t) \rceil \cdot T \cdot \sigma \cdot \phi(m) + \frac{8}{\sqrt{3}} \cdot p^2 \cdot \sqrt{h \cdot \phi(m)} + 6 \cdot p \cdot (\nu + 1) \cdot \sqrt{h}$. 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 $c \cdot \mathfrak{sk}' = \Delta_{q_t} \cdot m + w + r \cdot q_t$.

$$
\begin{aligned}
d \cdot \mathfrak{sk} &= a \cdot c \cdot \mathfrak{sk} \\
&= \mathfrak{sk} \cdot (\mathfrak{pk} \cdot e_1 + e_0 + P \cdot \mathfrak{sk}') \cdot c \\
&= P \cdot \mathfrak{sk} \cdot \mathfrak{sk}' \cdot c + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= P \cdot \mathfrak{sk} \cdot (\Delta_{q_t} \cdot m + w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= P \cdot \mathfrak{sk} \cdot \Delta_{q_t} \cdot m + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= P \cdot \mathfrak{sk} \cdot m \cdot \Delta_{q_t} + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= (1 + p \cdot f) \cdot m \cdot P \cdot \Delta_{q_t} + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= \Delta_Q \cdot m + m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) + p \cdot f \cdot m \cdot P \cdot \Delta_{q_t} \\
&\qquad + P \cdot \mathfrak{sk} \cdot (w + r \cdot q_t) + (\mathfrak{sk} \cdot e_0 + p \cdot e_2 \cdot g) \cdot c \\
&= \Delta_Q \cdot m + m \cdot (\epsilon_Q - P \cdot \epsilon_{q_t}) + p \cdot f \cdot m \cdot P \cdot \Delta_{q_t}
\end{aligned}
$$

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

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 \cdot 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}} \\
&\quad + \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}} \\
&\quad + \left\|p \cdot f \cdot (w - \epsilon_{q_t} \cdot m + \epsilon_{q_t} \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\quad + \left\|(1 + p \cdot f) \cdot e_0 \cdot c\right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \left\|e_2 \cdot g \cdot c\right\|_\infty^{\mathsf{can}} \\
&\quad + \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}} \\
&\quad + \left\|p \cdot f \cdot (w - \epsilon_{q_t} \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\quad + \left\|p \cdot f \cdot \epsilon_{q_t} \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\quad + \left\|(1 + p \cdot f) \cdot e_0 \cdot c\right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \left\|e_2 \cdot g \cdot c\right\|_\infty^{\mathsf{can}} \\
&\quad + P \cdot \left\|w - m \cdot \epsilon_{q_t}\right\|_\infty^{\mathsf{can}} \\
&\leq P \cdot p \cdot \left\|f \cdot m\right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \nu \cdot \left\|f\right\|_\infty^{\mathsf{can}} \\
&\quad + p^2 \cdot \left\|f \cdot m)\right\|_\infty^{\mathsf{can}} \\
&\quad + \left\|e_0 \cdot c\right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \left\|f \cdot e_0 \cdot c\right\|_\infty^{\mathsf{can}} \\
&\quad + p \cdot \left\|e_2 \cdot g \cdot c\right\|_\infty^{\mathsf{can}} \\
&\quad + P \cdot \nu \\
&\leq 16 \cdot P \cdot p^2 \cdot \sqrt{h \cdot \phi(m)/12} \\
&\quad + 6 \cdot p \cdot \nu \cdot \sqrt{h} \\
&\quad + 16 \cdot p^2 \cdot \sqrt{h \cdot \phi(m)/12}
\end{aligned}
$$

$$
\begin{aligned}
&+ 16 \cdot q_t \cdot \sigma \cdot \phi(m)/\sqrt{12} \\
&+ 40 \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&+ 40 \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h/12} \\
&+ P \cdot \nu \\
\leq \frac{8}{\sqrt{3}} &\cdot P \cdot p^2 \cdot \sqrt{h \cdot \phi(m)} \\
&+ 6 \cdot p \cdot \nu \cdot \sqrt{h} \\
&+ \frac{8}{\sqrt{3}} \cdot p^2 \cdot \sqrt{h \cdot \phi(m)} \\
&+ \frac{8}{\sqrt{3}} \cdot q_t \cdot \sigma \cdot \phi(m) \\
&+ \frac{40}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h} \\
&+ P \cdot \nu
\end{aligned}
$$

Therefore, we set $B^{\mathsf{YASHE}}_{\mathsf{Ks},2}(t) = \frac{8}{\sqrt{3}} \cdot P \cdot p^2 \cdot \sqrt{h \cdot \phi(m)} + 6 \cdot p \cdot \nu \cdot \sqrt{h} + \frac{8}{\sqrt{3}} \cdot p^2 \cdot \sqrt{h \cdot \phi(m)} + \frac{8}{\sqrt{3}} \cdot q_t \cdot \sigma \cdot \phi(m) + \frac{40}{\sqrt{3}} \cdot p \cdot q_t \cdot \sigma \cdot \phi(m) \cdot \sqrt{h}$. 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. 12, whilst those for multiplication are 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 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)

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

$\underline{\mathsf{Add}^{\mathsf{BGV}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{BGV}}(\mathfrak{c}_i, t)$ 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. 12: The Addition Methods for BGV, FV, NTRU and YASHE.

$$= \Delta_{q_t} \cdot m + w$$

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

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

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 $\mathsf{SwitchKey}$ 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 $\left\|w_i - \epsilon_{q_t} \cdot m_i\right\|_\infty^{\mathsf{can}} \leq \nu_i$, which means that

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

Note that this means that

$$\left\|r_i\right\|_\infty^{\mathsf{can}} = \left\| \frac{1}{q_t} \left(c_{i,0} - \mathfrak{sk} \cdot c_{i,1} - \Delta_{q_t} \cdot m_i - w_i\right) \right\|_\infty^{\mathsf{can}}$$

$\underline{\mathsf{Mult}^{\mathsf{BGV}}(\mathfrak{c}_0, \mathfrak{c}_1)}$:
- $t = \min(t_0, t_1)$.
- $\mathfrak{c}_i \leftarrow \mathsf{ReduceLevel}^{\mathsf{BGV}}(\mathfrak{c}_i, t)$ 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. 13: The Multiplication Methods for BGV, FV, NTRU and YASHE.

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

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,

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

$$
\begin{aligned}
&+ \Big(m_0 \cdot (w_1 + r_1 \cdot q_t) + m_1 \cdot (w_0 + r_0 \cdot q_t)\Big) \\
&- \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) \\
&+ \frac{p}{q_t} \cdot w_0 \cdot w_1 + p \cdot (r_0 \cdot w_1 + r_1 \cdot w_0) \\
&+ 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) \\
&+ q_t \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 + p \cdot r_0 \cdot r_1\Big) \\
&+ 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 \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) \\
&+ q_t \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 + p \cdot r_0 \cdot r_1\Big) \\
&+ 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 \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 \\
&+ \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 \\
&+ q_t \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 + p \cdot r_0 \cdot r_1\Big) \\
&+ 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 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

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

$$- \epsilon_{q_t} \cdot p \cdot \Big(m_0 \cdot r_1 + m_1 \cdot r_0 \Big) + r_a$$

We obtain a bound of, recalling $\epsilon_{q_t} \leq 1$,

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

where

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

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 $\|c \cdot \mathfrak{sk}\|_\infty^{\mathsf{can}} \leq \nu$. 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

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

$$= \Delta_{q_t} \cdot m + w,$$

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

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

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 $\left\|w_i - \epsilon_{q_t} \cdot m_i\right\|_\infty^{\mathsf{can}} \le \nu_i$, which means that

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

Note that this means that

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

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

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

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 $\left\|r_m\right\|_\infty^{\mathsf{can}} \le 16 \cdot p \cdot \phi(m)/12$. 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 $\mathfrak{sk}^2 \cdot d'$.

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

$$
\begin{aligned}
&= \frac{p}{q_t} \cdot \Big(\Delta_{q_t} \cdot m_0 + w_0 + r_0 \cdot q_t\Big) \cdot \Big(\Delta_{q_t} \cdot m_0 + w_0 + r_0 \cdot q_t\Big) + r_a \\
&= \cdots
\end{aligned}
$$

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

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

where

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

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.

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

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	-	-	-	35	15	5	5	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	793	-	-	-	45	20	4	4	14
YASHE	2	1159	15	-	25	40	25	5	5	24
YASHE-NOP	2	1159	-	-	-	40	25	5	5	24

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

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	884	20	-	30	50	17	10	10	42
FV-NOP	1	884	-	-	-	50	18	10	10	40
FV	2	1525	21	-	29	50	35	18	18	50
FV-NOP	2	1525	-	-	-	50	35	18	18	50
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	1342	-	-	-	75	38	12	12	36
YASHE	2	1799	27	-	33	60	40	13	13	57
YASHE-NOP	2	1799	-	-	-	60	40	13	13	57

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

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	2714	46	-	104	150	64	99	99	332
FV-NOP	1	2714	-	-	-	150	65	99	99	328
FV	2	4360	46	-	104	150	90	159	159	415
FV-NOP	2	4268	-	-	-	150	85	156	156	401
NTRU	1	3720	78	-	127	205	116	93	93	257
NTRU	2	5549	78	-	127	205	100	138	138	552
YASHE	1	3994	78	-	142	220	129	107	107	290
YASHE-NOP	1	3994	-	-	-	220	130	107	107	288
YASHE	2	5000	79	-	106	185	90	112	112	448
YASHE-NOP	2	5000	-	-	-	185	90	112	112	448

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

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	5091	78	-	202	280	131	348	348	1091
FV-NOP	1	5091	-	-	-	280	131	348	348	1091
FV	2	7835	79	-	201	280	150	535	535	1358
FV-NOP	2	7835	-	-	-	280	150	535	535	1358
NTRU	1	6646	142	-	223	365	211	296	296	808
NTRU	2	10122	147	-	223	370	185	457	457	1828
YASHE	1	7561	143	-	272	415	260	383	383	994
YASHE-NOP	1	7561	-	-	-	415	261	383	383	992
YASHE	2	9390	144	-	201	345	170	395	395	1576
YASHE-NOP	2	9390	-	-	-	345	170	395	395	1576

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

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	262	1298	1298	3973
FV-NOP	1	9756	-	-	-	535	255	1274	1274	3947
FV	2	14969	145	-	395	540	280	1973	1973	4970
FV-NOP	2	14969	-	-	-	540	280	1973	1973	4970
NTRU	1	12591	271	-	419	690	408	1060	1060	2854
NTRU	2	18901	274	-	416	690	345	1592	1592	6368
YASHE	1	14603	271	-	529	800	516	1426	1426	3637
YASHE-NOP	1	14603	-	-	-	800	517	1426	1426	3632
YASHE	2	18170	272	-	393	665	330	1474	1474	5888
YASHE-NOP	2	18170	-	-	-	665	330	1474	1474	5888

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

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	19176	271	-	779	1050	515	4915	4915	14938
FV-NOP	1	19176	-	-	-	1050	515	4915	4915	14938
FV	2	29053	272		778	1050	540	7447	7447	18725
FV-NOP	2	28962	-	-	-	1050	535	7424	7424	18631
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	1030	5497	5497	13878
YASHE-NOP	1	28687	-	-	-	1570	1030	5497	5497	13878
YASHE	2	35912	532	-	778	1310	655	5742	5742	22971
YASHE-NOP	2	35729	-	-	-	1305	650	5691	5691	22744

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

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	1525	-	-	-	85	16	31	31	199
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	2073	-	-	-	115	16	29	29	238
YASHE	2	3262	16	19	22	95	85	37	37	181
YASHE-NOP	2	3079	-	-	-	90	80	33	33	161

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

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	2439	22	28	29	135	12	80	80	984
FV-NOP	1	2073	-	-	-	115	16	58	58	476
FV	2	4634	21	28	30	135	120	152	152	441
FV-NOP	2	3994	-	-	-	120	100	117	117	331
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	27	44	46	205	31	93	93	708
YASHE-NOP	1	3537	-	-	-	195	33	84	84	581
YASHE	2	5274	28	31	34	155	135	99	99	473
YASHE-NOP	2	5000	-	-	-	150	125	91	91	427

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

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	8567	48	105, 106	106	470	64	983	983	8202
FV-NOP	1	5366	-	-	-	295	66	386	386	2113
FV	2	15883	47	105	108	470	400	1822	1822	5196
FV-NOP	2	9664	-	-	-	295	235	696	696	1946
NTRU	1	10487	79	123	127	575	110	736	736	4583
NTRU	2	18901	81	122	128	575	460	1326	1326	6102
YASHE	1	11951	80	143, 144	144	655	130	955	955	5770
YASHE-NOP	1	10487	-	-	-	575	130	736	736	3991
YASHE	2	16615	80	105	105	500	410	1014	1014	4705
YASHE-NOP	2	13871	-	-	-	425	335	719	719	3293

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

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	16158	79	201, 202	202	885	126	3491	3491	28012
FV-NOP	1	9481	-	-	-	520	129	1203	1203	6055
FV	2	30151	82	202	202	890	760	6551	6551	18697
FV-NOP	2	16706	-	-	-	525	390	2141	2141	5873
NTRU	1	18718	144	219	224	1025	206	2342	2342	13995
NTRU	2	33626	144	219	224	1025	815	4207	4207	19312
YASHE	1	22468	144	271, 272	272	1230	256	3373	3373	19582
YASHE-NOP	1	19267	-	-	-	1055	252	2481	2481	12869
YASHE	2	31431	146	202	203	955	765	3664	3664	16862
YASHE-NOP	2	25029	-	-	-	780	590	2383	2383	10754

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

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	31431	144	394	394	1720	255	13198	13198	102224
FV-NOP	1	17712	-	-	-	970	257	4194	4194	20025
FV	2	58228	144	394	394	1720	1465	24451	24451	69728
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	43595	272	528	529	2385	513	12692	12692	71699
YASHE-NOP	1	36918	-	-	-	2020	515	9103	9103	44809
YASHE	2	60697	273	394	395	1850	1470	13707	13707	62904
YASHE-NOP	2	47344	-	-	-	1485	1105	8582	8582	38519

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

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	61886	273	778	778	3385	509	51143	51143	391263
FV-NOP	1	34175	-	-	-	1870	515	15602	15602	72255
FV	2	114565	273	779	780	3390	2875	94818	94818	270050
FV-NOP	2	59051	-	-	-	1870	1360	26959	26959	73525
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	85756	528	1040, 1041	1041	4690	1024	49096	49096	273960
YASHE-NOP	1	72038	-	-	-	3940	1023	34647	34647	168087
YASHE	2	119321	529	779	779	3645	2880	53091	53091	243171
YASHE-NOP	2	91884	-	-	-	2895	2130	32471	32471	145195

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

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	3079	-	-	-	170	17	127	127	1405
FV	2	6463	17	18	24	185	170	291	291	852
FV-NOP	2	6006	-	-	-	175	155	256	256	740
NTRU	1	5000	16	28	35	275	6	167	167	7860
NTRU	2	9939	17	29	36	285	260	345	345	1668
YASHE	1	4451	16	25	29	245	2	133	133	16439
YASHE-NOP	1	4360	-	-	-	240	18	127	127	1830
YASHE	2	6829	17	19	26	195	180	162	162	787
YASHE-NOP	2	6372	-	-	-	180	170	140	140	684

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

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	5183	23	29	30	285	12	360	360	8925
FV-NOP	1	4177	-	-	-	230	18	234	234	3231
FV	2	10122	22	29	31	285	270	704	704	2075
FV-NOP	2	8201	-	-	-	235	215	470	470	1371
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, 44	46	420	27	392	392	6494
YASHE-NOP	1	7378	-	-	-	405	34	364	364	4709
YASHE	2	10945	28	31	34	310	290	414	414	2017
YASHE-NOP	2	10396	-	-	-	295	275	374	374	1821

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

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	18170	48	105	106	995	59	4413	4413	78850
FV-NOP	1	9756	-	-	-	535	59	1274	1274	12829
FV	2	35455	48	106	109	1005	935	8699	8699	25491
FV-NOP	2	18536	-	-	-	540	475	2443	2443	7036
NTRU	1	21645	81	122	128	1185	100	3131	3131	40233
NTRU	2	41941	80	123	131	1195	1075	6052	6052	29046
YASHE	1	25029	82	143	144	1370	126	4185	4185	49697
YASHE-NOP	1	21371	-	-	-	1170	132	3052	3052	30106
YASHE	2	36187	81	106	106	1035	945	4571	4571	22064
YASHE-NOP	2	28687	-	-	-	830	740	2906	2906	13902

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

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	34723	82	202	202	1900	126	16106	16106	258988
FV-NOP	1	16889	-	-	-	925	129	3814	3814	31162
FV	2	67099	81	202	203	1900	1770	31125	31125	91245
FV-NOP	2	31614	-	-	-	930	800	7177	7177	20530
NTRU	1	38748	144	219	224	2120	203	10027	10027	114748
NTRU	2	73776	146	219	227	2125	1910	19137	19137	91814
YASHE	1	47436	144	272, 273	273	2595	256	15026	15026	167344
YASHE-NOP	1	38930	-	-	-	2130	254	10122	10122	95005
YASHE	2	68380	146	202	203	1965	1775	16402	16402	78839
YASHE-NOP	2	51734	-	-	-	1510	1320	9535	9535	45279

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

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	67465	144	394	394	3690	251	60777	60777	954284
FV-NOP	1	31065	-	-	-	1700	261	12893	12893	96871
FV	2	130662	144	395	395	3700	3445	118029	118029	345954
FV-NOP	2	57496	-	-	-	1700	1445	23863	23863	68009
NTRU	1	72770	275	411	417	3980	393	35354	35354	393398
NTRU	2	138527	276	412	418	3990	3585	67471	67471	323658
YASHE	1	91884	273	528	528	5025	510	56361	56361	611692
YASHE-NOP	1	74141	-	-	-	4055	511	36699	36699	327924
YASHE	2	133131	275	395	395	3830	3450	62242	62242	298862
YASHE-NOP	2	97463	-	-	-	2855	2475	33966	33966	160792

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

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	133223	274	779	779	7285	510	236945	236945	$0.362 \cdot 10^7$
FV-NOP	1	59234	-	-	-	3240	516	46855	46855	341060
FV	2	257055	273	779	780	7285	6770	457188	457188	$0.133 \cdot 10^7$
FV-NOP	2	109261	-	-	-	3245	2730	86560	86560	245943
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	181055	529	1041, 1042	1042	9900	1024	218804	218804	$0.233 \cdot 10^7$
YASHE-NOP	1	144472	-	-	-	7900	1024	139322	139322	$0.121 \cdot 10^7$
YASHE	2	261902	529	779	779	7540	6780	241057	241057	$0.115 \cdot 10^7$
YASHE-NOP	2	189011	-	-	-	5550	4785	128053	128053	604964

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

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	19	18, 19	24	370	2	608	608	113210
FV-NOP	1	6189	-	-	-	340	13	513	513	13949
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	8933	-	-	-	490	17	534	534	15935
YASHE	2	14511	17	20	23	400	395	708	708	3525
YASHE-NOP	2	13231	-	-	-	370	355	597	597	2939

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

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	10487	22	29	31	575	9	1472	1472	95527
FV-NOP	1	8384	-	-	-	460	16	941	941	28011
FV	2	21371	22	30	33	595	575	3104	3104	9208
FV-NOP	2	16798	-	-	-	470	450	1927	1927	5700
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	30	44	47	870	27	1686	1686	56039
YASHE-NOP	1	15060	-	-	-	825	30	1516	1516	43224
YASHE	2	23017	29	32	35	640	620	1798	1798	8878
YASHE-NOP	2	21462	-	-	-	600	575	1571	1571	7728

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

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	37742	51	106	106	2065	61	19027	19027	663160
FV-NOP	1	18718	-	-	-	1025	61	4684	4684	83391
FV	2	74233	48	106	109	2065	1995	37424	37424	111005
FV-NOP	2	36644	-	-	-	1035	970	9259	9259	27196
NTRU	1	44326	83	123	128	2425	106	13121	13121	313304
NTRU	2	86580	82	123	129	2425	2310	25629	25629	125716
YASHE	1	51552	84	144	144	2820	126	17746	17746	414922
YASHE-NOP	1	43137	-	-	-	2360	131	12427	12427	239305
YASHE	2	74964	81	106	106	2095	2005	19171	19171	94208
YASHE-NOP	2	58594	-	-	-	1650	1555	11801	11801	57649

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

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	71672	82	202	202	3920	122	68592	68592	$0.227 \cdot 10^7$
FV-NOP	1	31797	-	-	-	1740	131	13507	13507	192920
FV	2	141728	83	203	203	3940	3810	136330	136330	404492
FV-NOP	2	61520	-	-	-	1750	1615	26284	26284	76824
NTRU	1	78897	148	219	225	4315	199	41557	41557	942670
NTRU	2	154532	145	220	225	4330	4120	81680	81680	400477
YASHE	1	97189	146	272	273	5315	253	63056	63056	$0.138 \cdot 10^7$
YASHE-NOP	1	78348	-	-	-	4285	255	40981	40981	729663
YASHE	2	143008	147	203	204	4005	3815	69915	69915	342943
YASHE-NOP	2	105145	-	-	-	2980	2780	38120	38120	185723

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

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	139899	145	395	395	7650	253	261285	261285	$0.816 \cdot 10^8$
FV-NOP	1	57679	-	-	-	3155	257	44428	44428	589838
FV	2	275164	145	395	395	7650	7395	513917	513917	$0.152 \cdot 10^7$
FV-NOP	2	110999	-	-	-	3165	2905	85769	85769	250262
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	188828	274	529	529	10325	519	237994	237994	$0.505 \cdot 10^7$
YASHE-NOP	1	148770	-	-	-	8135	512	147734	147734	$0.249 \cdot 10^7$
YASHE	2	277633	274	395	395	7780	7400	263670	263670	$0.129 \cdot 10^7$
YASHE-NOP	2	197883	-	-	-	5600	5220	135271	135271	657999

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

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes 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	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	275712	274	779	779	15075	506	$0.101 \cdot 10^7$	$0.101 \cdot 10^7$	$0.312 \cdot 10^8$
FV-NOP	1	10952	-	-	-	5980	513	159649	159649	$0.202 \cdot 10^7$
FV	2	542767	274	780	781	15095	14580	$0.200 \cdot 10^7$	$0.200 \cdot 10^7$	$0.593 \cdot 10^7$
FV-NOP	2	209681	-	-	-	5990	5475	306637	306637	893550
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	371468	531	1041	1041	20310	1021	920961	920961	$0.192 \cdot 10^8$
YASHE-NOP	1	289431	-	-	-	15825	1025	559112	559112	$0.919 \cdot 10^7$
YASHE	2	547522	530	780	780	15350	14585	$0.102 \cdot 10^7$	$0.102 \cdot 10^7$	$0.502 \cdot 10^7$
YASHE-NOP	2	383266	-	-	-	10860	10095	508089	508089	$0.246 \cdot 10^7$

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

Scheme	Key Switch Variant	$\phi(m)$ $\approx$	$\approx \log_2$ primes 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	12134	16	22	33	665	9	1969	1969	211970
BGV	2	23877	16	22	35	667	640	3888	3888	11507
FV	1	10487	19	19	24	575	4	1472	1472	213097
FV-NOP	1	9390	-	-	-	515	8	1180	1180	77183
FV	2	20731	18	19	25	575	560	2910	2910	8654
FV-NOP	1	19176	-	-	-	535	515	2504	2504	7420
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	260629
YASHE-NOP	1	13505	-	-	-	740	12	1219	1219	76449
YASHE	2	21828	17	20	23	600	595	1598	1598	7967
YASHE-NOP	2	20182	-	-	-	560	545	1379	1379	6824

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

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	15792	22	29	31	865	6	3334	3334	484127
FV-NOP	1	12682	-	-	-	695	16	2151	2151	95622
FV	2	32437	23	30	32	895	880	7087	7087	21144
FV-NOP	2	25578	-	-	-	710	690	4443	4443	13176
NTRU	1	25523	29	47	52	1397	30	4352	4352	296759
NTRU	2	50399	29	47	52	1397	1360	8594	8594	42518
YASHE	1	23932	31	44	47	1310	25	3827	3827	204362
YASHE-NOP	1	22834	-	-	-	1250	27	3484	3484	164789
YASHE	2	34723	29	32	35	960	940	4069	4069	20175
YASHE-NOP	2	32711	-	-	-	905	885	3613	3613	17908

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

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	57130	51	106	106	3125	59	43586	43586	$0.235 \cdot 10^7$
FV-NOP	1	27773	-	-	-	1520	64	10306	10306	255083
FV	2	114108	49	107	110	3155	3085	87893	87893	261729
FV-NOP	2	54935	-	-	-	1535	1470	20587	20587	60889
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	77891	83	144	145	4260	125	40504	40504	$0.142 \cdot 10^7$
YASHE-NOP	1	64904	-	-	-	3550	126	28126	28126	82068
YASHE	2	114748	81	107	108	3185	3090	44613	44613	220405
YASHE-NOP	2	88592	-	-	-	2470	2375	26711	267113	131503

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

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	109169	83	203	203	5970	125	159115	159115	$0.775 \cdot 10^7$
FV-NOP	1	46704	-	-	-	2555	128	29132	29132	610654
FV	2	215991	82	203	204	5970	5840	314811	314811	937578
FV-NOP	2	91610	-	-	-	2570	2440	57479	57479	169532
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	146941	147	272	272	8035	249	144124	144124	$0.479 \cdot 10^7$
YASHE-NOP	1	117858	-	-	-	6445	257	92723	92723	$0.241 \cdot 10^7$
YASHE	2	217271	147	203	204	6035	5845	160062	160062	790233
YASHE-NOP	2	158739	-	-	-	4435	4245	85938	85938	422328

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

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	212150	145	395	395	11600	251	600815	600815	$0.283 \cdot 10^8$
FV-NOP	1	84385	-	-	-	4615	254	95077	95077	$0.182 \cdot 10^7$
FV	2	420672	145	396	397	11630	11370	$0.119 \cdot 10^7$	$0.119 \cdot 10^7$	$0.355 \cdot 10^7$
FV-NOP	2	164592	-	-	-	4630	4370	186050	186050	547702
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	285590	274	529	529	15615	509	544371	544371	$0.172 \cdot 10^8$
YASHE-NOP	1	223491	-	-	-	12220	515	333381	333381	$0.824 \cdot 10^7$
YASHE	2	423141	273	396	399	11760	11375	607438	607438	$0.299 \cdot 10^7$
YASHE-NOP	2	298485	-	-	-	8350	7970	304241	304241	$0.149 \cdot 10^7$

$\boldsymbol{L = 30, p \approx 2^{256}, h = 64, k = 80, \zeta = 8, c_m = 1.3, \sigma = 1.3}$

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	152703	273	278	293	8350	262	311296	311296	$0.102 \cdot 10^8$
BGV	2	301320	273	279	295	8380	8095	616470	616470	$0.182 \cdot 10^7$
FV	1	418751	275	780	780	22895	509	$0.234 \cdot 10^7$	$0.234 \cdot 10^7$	$0.107 \cdot 10^9$
FV-NOP	1	159562	-	-	-	8725	511	339887	339887	$0.614 \cdot 10^7$
FV	2	828112	274	780	781	22895	22380	$0.462 \cdot 10^7$	$0.462 \cdot 10^7$	$0.137 \cdot 10^8$
FV-NOP	2	310283	-	-	-	8740	8225	662078	662078	$0.194 \cdot 10^7$
NTRU	1	432506	529	797	802	23647	527	$0.124 \cdot 10^7$	$0.124 \cdot 10^7$	$0.166 \cdot 10^9$
NTRU	2	850757	530	797	802	23648	22865	$0.245 \cdot 10^7$	$0.245 \cdot 10^7$	$0.121 \cdot 10^8$
YASHE	1	561881	531	1041	1041	30720	1018	$0.210 \cdot 10^7$	$0.210 \cdot 10^7$	$0.656 \cdot 10^8$
YASHE-NOP	1	434482	-	-	-	23755	1027	$0.125 \cdot 10^7$	$0.125 \cdot 10^7$	$0.304 \cdot 10^8$
YASHE	2	834057	532	781	785	23185	22415	$0.236 \cdot 10^7$	$0.236 \cdot 10^7$	$0.116 \cdot 10^8$
YASHE-NOP	2	577612	-	-	-	16175	15405	$0.114 \cdot 10^7$	$0.114 \cdot 10^7$	$0.559 \cdot 10^7$