On the Feasibility and Impact of Standardising Sparse-secret LWE Parameter Sets for Homomorphic Encryption

Benjamin R. Curtis
benjamin.curtis.2015@rhul.ac.uk
Royal Holloway University of London
United Kingdom

Rachel Player
rachel.player@rhul.ac.uk
Royal Holloway University of London
United Kingdom

ABSTRACT

In November 2018, the HomomorphicEncryption.org consortium published the Homomorphic Encryption Security Standard. The Standard recommends several sets of Learning with Errors (LWE) parameters that can be selected by application developers to achieve a target security level $\lambda \in \{128, 192, 256\}$. These parameter sets all involve a power-of-two dimension $n \leq 2^{15}$, an error distribution of standard deviation $\sigma \approx 3.19$, and a secret whose coefficients are either chosen uniformly in $\mathbb{Z}_q$, chosen according to the error distribution, or chosen uniformly in $\{-1, 0, 1\}$. These parameter sets do not necessarily reflect implementation choices in the most commonly used homomorphic encryption libraries. For example, several libraries support dimensions that are not a power of two. Moreover, all known implementations for bootstrapping for the CKKS, BFV and BGV schemes use a sparse secret and a large ring dimension such as $n \in \{2^{16}, 2^{17}\}$, and advanced applications such as logistic regression have used equally large dimensions. This motivates the community to consider widening the recommended parameter sets, and the purpose of this paper is to investigate such possible extensions. We explore the security of possible sparse-secret LWE parameter sets, taking into account hybrid attacks, which are often the most competitive in the sparse-secret regime. We present a conservative analysis of the hybrid decoding and hybrid dual attacks for parameter sets of varying sparsity, with the goal of balancing security requirements with bootstrapping efficiency. We also show how the methodology in the Standard can be easily adapted to support parameter sets with power-of-two dimension $n \geq 2^{16}$. We conclude with a number of discussion points to motivate future improvements to the Standard.

KEYWORDS

Cryptanalysis; Learning with Errors; Homomorphic Encryption; Parameter Selection; Bootstrapping.

1 INTRODUCTION

Homomorphic Encryption [15, 29, 30] is a powerful tool enabling computation on encrypted data. Many libraries, implementing a variety of schemes, are available [28, 33, 39, 42, 43] and there are numerous interesting applications [10], including privacy-preserving machine learning [36]. The iDash competition [1] runs each year, challenging participants to produce a privacy-preserving solution using homomorphic encryption.

Motivated in part by the increasing commercial interest in homomorphic encryption, the HomomorphicEncryption.org consortium have begun an effort to standardise both an API [17] and advice on secure parameter selection. An important output of the consortium is the Homomorphic Encryption Security Standard [19], which recommends parameter sets achieving certain target security levels.

The Homomorphic Encryption Security Standard focuses on Learning with Errors (LWE) [44] based homomorphic encryption schemes. Informally, LWE asks an adversary to recover $\mathbf{s}$ (resp. $\mathbf{e}$) from a noisy system of linear equations:

$$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q$$

where the entries of the public matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ are each drawn uniformly at random from $\mathbb{Z}_q$, the components of the secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ are drawn from some secret distribution $\chi_{\mathbf{s}}$, and the components of the error vector $\mathbf{e} \in \mathbb{Z}_q^m$ are drawn from some error distribution $\chi_{\mathbf{e}}$. Typically, $\chi_{\mathbf{e}}$ is a Discrete Gaussian distribution with standard deviation σ.

In the original definition, the secret distribution $\chi_{\mathbf{s}}$ is taken to be the uniform distribution on $\mathbb{Z}_q^n$. However, we can always transform an LWE instance into one where the secret follows the error distribution [9]. For error distributions such that $\sigma = O(\sqrt{n})$, there are reductions from worst-case hard lattice problems to LWE [44]. In this setting, hardness results for a binary secret $\mathbf{s} \in \{0, 1\}^n$ can be obtained, at the cost of increasing the dimension [16]. However, implementations of LWE-based homomorphic encryption schemes typically choose a much narrower error for which these reductions would not apply. An early example [31] uses $\sigma = 3.19$, which remains a popular choice today.

A commonly used tool to estimate security of LWE instances is the LWE Estimator [6]. The Homomorphic Encryption Security Standard specifies parameters (n, q, σ) achieving a security level $\lambda \in \{128, 192, 256\}$ according to the LWE Estimator. The Standard limits its consideration to power-of-two ring dimensions $n \in \{1024, 2048, ..., 32768\}$ and a fixed Discrete Gaussian error distribution $\chi_{\mathbf{e}}$ with standard deviation $\sigma = 3.19$. For each dimension, a specific modulus q is not given; rather, a bit-length $\log q$ is standardised, since only the size of q, and not its specific form, affects security. For a given modulus q, the constraint on the error distribution can equivalently be expressed as fixing the parameter $\alpha = \frac{8}{q}$, where α is defined such that $\sigma = \frac{\alpha q}{\sqrt{2\pi}}$.

The Standard allows for the secret distribution $\chi_{\mathbf{s}}$ to be *uniform ternary*, i.e. $\mathbf{s}$ is chosen uniformly at random from the set $\{-1, 0, 1\}^n$; *uniform*, i.e. $\mathbf{s}$ is chosen uniformly at random from the set $\mathbb{Z}_q^n$; or *error*, i.e. each coefficient of $\mathbf{s}$ is sampled from the error distribution $\chi_{\mathbf{e}}$. All major implementations of homomorphic encryption use a binary or ternary secret distribution, with coefficients chosen from $\{-1, 0, 1\}$. Moreover, many implementations use a sparse secret, for which all but a certain Hamming weight h of the coefficients are

zero. For example, HEAAN [39] uses by default a sparse ternary secret of Hamming weight $h = 64$.

An important issue motivating the use of sparse secrets is the complexity of bootstrapping. For the CKKS scheme [26], bootstrapping can be implemented by evaluating a Chebyshev interpolant in degree $d = O(K + \log q)$, for q the ciphertext modulus, and K a constant depending on the secret distribution. This evaluation requires $O(\sqrt{d})$ ciphertext multiplications [20]. The heuristic argument of [24] shows that for a sparse ternary secret with Hamming weight h we have $K = O(\sqrt{h})$, while for a uniform ternary secret we have $K = O(\sqrt{n})$. In the case of the BGV scheme [14], bootstrapping requires the evaluation of a circuit of depth $\log(\|\mathbf{s}\|_1) + \log t$, where $\|\mathbf{s}\|_1$ is the 1-norm of the secret and t is the plaintext modulus. This evaluation requires $O(\log^{3/2}\|\mathbf{s}\|_1 + \log^{1/2}\|\mathbf{s}\|_1 \cdot \log t + \log^2 t)$ ciphertext multiplications [21]. For the BFV scheme [29], bootstrapping requires the evaluation of a circuit of depth $\log(\|\mathbf{s}\|_1) + \log\log t$ and requires $O((\log\|\mathbf{s}\|_1 + \log t)^{1/2}\log\|\mathbf{s}\|_1)$ ciphertext multiplications [21]. For sparse ternary secret with Hamming weight h, we have $\|\mathbf{s}\|_1 = h$, whereas for a uniform ternary secret we expect $\|\mathbf{s}\|_1 = O(n)$. Current implementations for bootstrapping in CKKS, BGV or BFV use sparse secrets for efficiency reasons [21, 24].

Sparse secret distributions are not currently supported in the Homomorphic Encryption Security Standard. This is likely due to the loss of security as compared to (e.g.) a uniform ternary secret for a fixed set of LWE parameters (n, q, σ). This loss is intuitive, as we are shrinking the size of the keyspace. Moreover, several additional attacks are known which can exploit the sparsity of an LWE secret [3, 25, 35]. At a high level, all of these techniques combine a combinatorial search in some dimension τ, followed by solving a lattice problem in dimension $(d-\tau)$. For sparse secrets, this is typically more efficient than solving the original lattice problem in dimension d.

Another way in which the recommended parameter sets in the Homomorphic Encryption Security Standard do not always reflect implementation choices is in the maximal supported dimension $n = 2^{15}$. For example, many implementations of bootstrapping, such as [20, 24, 34], choose ring dimension $n = 2^{16}$. In addition, advanced applications of homomorphic encryption, such as logistic regression training [37, 38], have been reported using dimension $n = 2^{16}$ or $n = 2^{17}$.

1.1 Contribution and structure of paper

The above discussion motivates the homomorphic encryption community to consider widening the recommended parameter sets to include parameter sets with a sparse secret, or parameter sets for larger dimension $n > 2^{15}$, and in this paper we consider such possible extensions. We stress that our goal is to spark this discussion, and we do not endorse any specific parameter set we present. In Section 2 we introduce necessary background. In Section 3 we assess the impact on security and performance of using a sparse ternary secret of Hamming weight h instead of a uniform ternary secret, for various choices of h. In Section 4 we show how the methodology of the Standard could be used to select parameters with a larger power-of-two dimension $n \geq 2^{16}$. In Section 5 we raise a number of points intended to prompt further discussion and highlight directions for future work.

2 BACKGROUND

Notation. We denote (column) vectors by lower case bold letters, e.g $\mathbf{a}$. Matrices are denoted by upper case bold letters, e.g $\mathbf{A}$. We denote the inner product of two vectors $\mathbf{a}$ and $\mathbf{b}$ of length d as $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{d} a_i b_i$. All logarithms are to the base two, unless stated otherwise.

2.1 Lattice background

Lattices. A *lattice* $\Lambda = \Lambda(\mathbf{B})$ is a discrete additive subgroup of $\mathbb{R}^d$ generated by a basis $\mathbf{B}$ of linearly-independent integer vectors. The rank of the lattice $\Lambda(\mathbf{B})$ is defined to be the rank of $\mathbf{B}$. If the rank equals d we say that Λ is full-rank. The volume $\mathrm{vol}(\Lambda)$ of a full-rank lattice Λ is the absolute value of the determinant of any basis of the lattice. The i^{th} *successive minimum* of a lattice, $\lambda_i(\Lambda)$, is the radius of the smallest ball centred at the origin containing at least i linearly independent lattice vectors. The *Gaussian heuristic* states that the length of a shortest lattice vector $\lambda_1(\Lambda) \approx \sqrt{\frac{d}{2\pi e}}\mathrm{vol}(\Lambda)^{1/d}$.

Learning with Errors. The homomorphic encryption schemes considered in this work all base their hardness on the Learning with Errors problem (LWE).

Definition 2.1 (LWE [44]). Let n, q be positive integers, χ be a probability distribution on $\mathbb{Z}$ and $\mathbf{s}$ be a secret vector in $\mathbb{Z}_q^n$. We denote the LWE Distribution $L_{\mathbf{s},\chi,q}$ as the distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ given by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}$ according to χ and considering it as an element of $\mathbb{Z}_q$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.
Search-LWE is the problem of recovering the vector $\mathbf{s}$ from a collection $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ of samples drawn according to $L_{\mathbf{s},\chi,q}$.
Decision-LWE is the problem of distinguishing whether samples $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ are drawn from the LWE distribution $L_{\mathbf{s},\chi,q}$ or uniformly from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Lattice Reduction. To solve the Learning with Errors problem, we may employ *lattice reduction algorithms* such as BKZ2.0 [23] (BKZ). Lattice reduction algorithms take as input a (public) "long" basis, and output a shorter, more orthogonal, basis which can be used to find the LWE secret, or distinguish LWE samples from random. The quality of the lattice reduction is characterised by the *root-Hermite* factor δ, defined such that $\|\mathbf{b}_1\| = \delta^d \cdot \det(\Lambda)^{1/d}$, where $\mathbf{b}_1$ is the shortest vector in the output basis.

Following the Homomorphic Encryption Security Standard, we only consider *sieving*-based cost models for BKZ. More precisely, BKZ takes as input a (column) lattice basis $\mathbf{B}$ and its main subroutine solves the Shortest Vector Problem (SVP) via a sieving algorithm on projected sublattices of dimension β, where β is referred to as the *blocksize*. In this work, we view BKZ as a black box which runs in (pre-quantum) time:

$$T_{\mathrm{BKZ}}(\beta, d) = 2^{0.292\beta + 16.4 + \log(8d)}$$

and, if instantiated with quantum algorithms to solve SVP, runs in time:

$$T_{\mathrm{BKZ}}(\beta, d) = 2^{0.265\beta + 16.4 + \log(8d)}.$$

When given a basis $\mathbf{B}$ of dimension d, we assume that BKZ outputs a basis following the *Geometric Series Assumption*.

Definition 2.2 (Geometric Series Assumption (GSA) [45]). Let $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}$ be a basis of some lattice Λ, of quality δ, that is output by BKZ. Then the lengths $\|\mathbf{b}_i^*\|$ for $(1 \leq i \leq d)$ of the Gram-Schmidt vectors of this basis are approximated by $\|\mathbf{b}_i^*\| = \alpha^{i-1}\|\mathbf{b}_1\|$ for some $0 < \alpha < 1$.

Using $\|\mathbf{b}_1\| = \delta^d \cdot \det(\Lambda)^{1/d}$ and $\prod \|\mathbf{b}_i^*\| = \det(\Lambda)$ we can determine the value α as $\alpha \approx \delta^{-2}$. We could alternatively use the BKZ Simulator [22] to estimate output quality, but for homomorphic encryption-style parameter sets we have $\beta \ll (m+n)$ and the output of the BKZ Simulator is very close to the GSA.

2.2 Small and sparse secret LWE

Definition 2.3 specifies notation for some LWE secret distributions of interest.

Definition 2.3 (Small Secret Distributions). Let n, q be positive integers. $\mathcal{B}^-$ is the probability distribution on $\mathbb{Z}_q^n$ where each component is independently sampled uniformly at random from $\{-1, 0, 1\}$. $\mathcal{B}^+$ is the probability distribution on $\mathbb{Z}_q^n$ where each component is independently sampled uniformly at random from $\{0, 1\}$. $\mathcal{B}_h^-$ is the probability distribution on $\mathbb{Z}_q^n$ where components are sampled uniformly at random from $\{-1, 0, 1\}$ with the additional guarantee that exactly h components are non-zero.

The uniform ternary distribution is denoted by $\mathcal{B}^-$. The uniform binary distribution is denoted by $\mathcal{B}^+$, and is used in the TFHE library [28]. The HEAAN library [39] uses $\mathcal{B}_{64}^-$, where the choice $h = 64$ seems to originate from [31].

Keyspace size. The size of the keyspace $\mathcal{S}$ when the LWE secret is drawn uniformly at random from $\mathbb{Z}_q^n$ is $\|\mathcal{S}\| = q^n$. When the secret is drawn from $\mathcal{B}^-$, the size of the keyspace is $\|\mathcal{S}\| = 3^n$. When the secret is drawn from $\mathcal{B}_h^-$, we have $\|\mathcal{S}_h\| = \binom{n}{h} \cdot 2^h$. In Figure 1, we highlight the size of the keyspace for each h when $n = 1024$. In this case, if $h = 64$ then the size of the keyspace is $\approx 2^{405}$, whereas if the secret is is drawn from $\mathcal{B}^-$ then the size of the keyspace is $\approx 2^{1623}$.

Density. Keeping a fixed Hamming weight (e.g. $h = 64$) for a variety of ring dimensions means that the *density* h/n of the secret decreases as n grows. One approach to scaling sparse secrets is to fix the *density parameter* $\kappa = \frac{h}{n}$. For example, we could consider $\kappa = \frac{1}{16}$ such that $(n, h) \in \{(1024, 64), (2048, 128), \ldots\}$. This follows the approach of several submissions to the ongoing NIST post-quantum standardisation effort: for example, Lizard [27] uses $h = \frac{n}{8}$, i.e. $\kappa = \frac{1}{8}$. However, for larger ring dimensions used in homomorphic encryption libraries, this approach can lead to a large Hamming weight h. For example, for $n = 32768$, choosing $\kappa = \frac{1}{16}$ would require $h = 2048$.

Another approach is to fix the ratio between the Hamming weight h of the secret and the security parameter λ, that is fix the value ζ, where $\zeta = \frac{h}{\lambda}$. Using this approach, for each target security level λ_{target}, we would fix the value of the Hamming weight h for *every* ring dimension $n = 2^k$. For example, if $\zeta = 1$, then for a fixed security level λ we consider secrets of Hamming weight $h = \lambda$. Such an approach means that the (theoretical) complexity of bootstrapping would remain the same for each dimension n. In this work we consider the *second* approach, i.e. fixing the value of $\zeta = \frac{h}{\lambda}$. We consider $\zeta \in \{\frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}\}$.

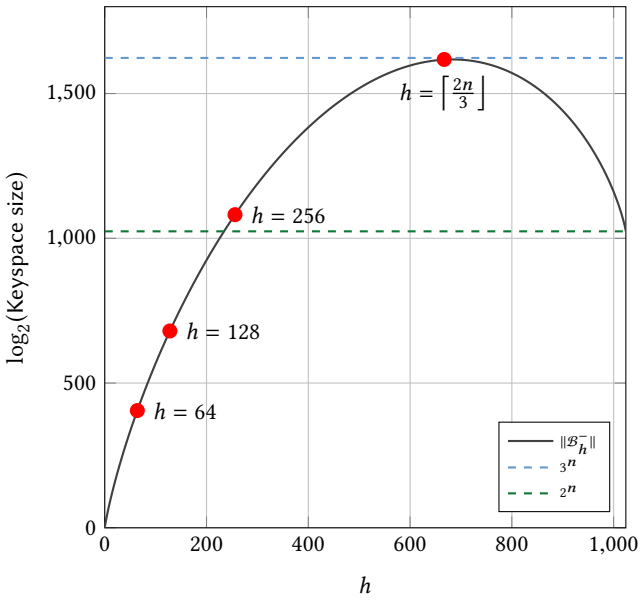

Figure 1: Example keyspace sizes with $n = 1024$ for fixed Hamming weight ternary secrets (black line). The keyspace sizes of uniform ternary secrets (blue dashed line), and uniform binary secrets (green dashed line), are provided for reference.

2.3 Algorithms for solving LWE

The security of homomorphic encryption parameter sets is typically determined by considering the best known attacks. That is, given an LWE parameter set (n, q, σ) and a corresponding secret distribution, we set λ to be the logarithm of the running time of the fastest attack. Several tools are available to estimate the running time of algorithms for solving LWE, the most popular being the *LWE Estimator* of Albrecht *et al.* [6]. The current version of the Homomorphic Encryption Security Standard [2] uses the LWE Estimator to determine parameters, based on the running time of three attacks: `usvp`, `dec` and `dual`. Hybrid attacks [25, 35] are typically among the most competitive in the case of sparse secrets, although they are not currently supported by the LWE Estimator.

Estimates for the primal decoding attack `dec` [40, 41] reported by the LWE Estimator do not assume state-of-the-art techniques, hence may be inaccurate and are often not competitive. More precisely, the Estimator currently assumes the decoding attack is implemented with the Nearest Planes algorithm [40] as opposed to the more efficient pruned enumeration [41]. For this reason, we do not report `dec` estimates in this work. We note that an accurate evaluation of security against `dec` should be performed before standardising any homomorphic encryption parameter sets.

In the remainder of this section we describe the four attacks that we consider as part of this work, namely `usvp`, `hybrid-dec`, `dual` and

hybrid-dual. We do not expect other algorithms for LWE (e.g. [11]) to be competitive in our setting.

Primal uSVP. The primal uSVP attack (usvp) [7, 13] solves search LWE by finding a unique shortest vector in the lattice $\Lambda(\mathbf{B})$ where

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ -\mathbf{A} & q\mathbf{I}_m & \mathbf{b} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}.$$

Lattice reduction is performed on the basis $\mathbf{B}$ to find the unique shortest vector $(\mathbf{s}, \mathbf{e}, 1)$ in $\Lambda(\mathbf{B})$. The blocksize β is chosen using the success condition from [7], which was experimentally verified in [5], in order to guarantee the success of the attack. This algorithm has a combinatorial variant, in which τ components of $\mathbf{s}$ are guessed as zero, and lattice reduction is performed in dimension $(d - \tau)$. The complexity of the usvp attack is therefore determined as

$$\min_{\beta, \tau, d} \frac{T_{\mathsf{BKZ}}(\beta, d)}{p_\tau}.$$

In this paper, we estimate complexity of this combinatorial variant of usvp using the LWE estimator[1].

Hybrid decoding. The hybrid decoding attack (hybrid-dec) was proposed by Howgrave-Graham [35] as an efficient attack on NTRU, and can also be applied in the LWE setting [18, 32]. Given an LWE sample $(\mathbf{A}, \mathbf{As} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, the main idea of hybrid-dec is as follows. We split $\mathbf{s}$ into a guessing part and a decoding part as $\mathbf{s} = (\mathbf{s}_g, \mathbf{s}_l) \in \mathbb{Z}_q^r \times \mathbb{Z}_q^{n-r}$, and split $\mathbf{A} = (\mathbf{A}_1 || \mathbf{A}_2)$ with $\mathbf{A}_1 \in \mathbb{Z}_q^{m \times r}$ and $\mathbf{A}_2 \in \mathbb{Z}_q^{m \times (n-r)}$. It can be verified that the lattice with basis

$$\mathbf{M} = \begin{pmatrix} \mathbf{I}_r & \mathbf{0} \\ -\mathbf{A}_1 & \mathbf{M}' \end{pmatrix}, \text{ where } \mathbf{M}' = \begin{pmatrix} \mathbf{I}_{n-r} & \mathbf{0} \\ -\mathbf{A}_2 & q\mathbf{I}_m \end{pmatrix},$$

contains a lattice point which is separated by $(\mathbf{s}_l, \mathbf{e})$ from the point $(0, \mathbf{b})$. Denote $\mathbf{w} = (\mathbf{w}_g, \mathbf{w}_l)$ where $\mathbf{w}_g = \mathbf{s}_g$ and $\mathbf{w}_l = (\mathbf{s}_l, \mathbf{e})$. We can see that for some $\mathbf{x} \in \mathbb{Z}^m$,

$$\mathbf{w} = \begin{pmatrix} \mathbf{I}_r & \mathbf{0} \\ -\mathbf{A}_1 & \mathbf{M}' \end{pmatrix} \cdot \begin{pmatrix} \mathbf{w}_g \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{w}_g \\ -\mathbf{A}_1\mathbf{s}_g + \mathbf{M}'\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{w}_g \\ \mathbf{w}_l \end{pmatrix}$$

Hence $-\mathbf{A}_1\mathbf{s}_g$ is close to the lattice $\Lambda(\mathbf{M}')$, the offset being the short vector $\mathbf{w}_l$. If we can guess $\mathbf{w}_g$ correctly, we can hope to recover $\mathbf{w}_l$ as the output of Babai's Nearest Plane algorithm [12]. The guessing part is sped up using a meet-in-the-middle process. Such a meet-in-the-middle procedure also carries a probability of failure, which we denote p_{mitm}.

A recent analysis of the hybrid decoding attack can be found in Wunderer's thesis [47]. Wunderer's analysis of hybrid-dec can be applied to sparse-secret LWE parameter sets typically used in homomorphic encryption[2]. However, in this paper, estimates of hybrid-dec are obtained using a more conservative analysis than Wunderer's, generated using custom code[3]. We use a conservative analysis as we focus only on parameter selection for security: this approach mitigates against future improvements to attacks. Of course, a more accurate estimation of the attack complexity might be preferred in practice to ensure the smallest possible dimension n can be chosen to maximise efficiency while retaining security.

[1] All estimates produced using the LWE Estimator in this document were obtained with commit 3019847.

[2] Code and a preprint detailing the application are available at https://github.com/rachelplayer/LatRedHybrid

[3] The code is available at https://github.com/bencrts/hybridattack

Our analysis for hybrid-dec is given under the following assumptions:

- The output basis-shape of lattice reduction is given by the Geometric Series Assumption [45].
- The (heuristic) success probability of Babai's Nearest Plane algorithm follows the analysis in [47] and is given by:

$$p_{\mathsf{babai}} \approx \prod_{1 \le i \le d} \left(1 - \frac{2}{B(\frac{d-1}{2}, \frac{1}{2})} \int_{\min(r_i, 1)}^{1} (1 - t^2)^{(d-3)/2}\, dt\right)$$

  where d is the dimension of the lattice under consideration; $r_i = \|\mathbf{b}_i^*\|/2\|\mathbf{v}\|$, where $\|\mathbf{v}\|$ is the (expected) norm of the target vector; and $B(\cdot, \cdot)$ denotes the Beta function.
- The cost of running Babai's Nearest Plane algorithm in a lattice of dimension d is given by $T_{\mathsf{babai}} = \frac{d^2}{2^{1.06}}$.
- The meet-in-the-middle search phase provides a square-root speed-up compared to an exhaustive search.
- The associated meet-in-the-middle probability is set to be $p_{\mathsf{mitm}} = 1$, thus providing an explicit *underestimate* of security.
- The meet-in-the-middle search phase has access to unlimited memory.

Under these assumptions, the cost of hybrid-dec is given by:

$$\min_{\beta, \tau, d, t} \frac{T_{\mathsf{BKZ}}(\beta, d - \tau) + \frac{d^2}{2^{1.06}} \| \sum_{i=0}^{t} \mathcal{S}_i \|}{p \cdot \sum_{i=0}^{t} p_i}$$

where β is the BKZ blocksize, $d - \tau$ is the dimension of the lattice reduction, τ is the guessing dimension (i.e. the number of guessed components of the secret), and t is the maximal Hamming weight considered in the search space which is a union of sets $\mathcal{S}_i$, each containing all length τ ternary vectors of Hamming weight i.

Dual. The dual attack (dual) [3] on small secret LWE instances involves searching for short vectors $(\mathbf{w}, \mathbf{v})$ in the *dual lattice*

$$\Lambda^\top(A) = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^m \times \mathbb{Z}^n \mid \mathbf{y} \cdot \mathbf{A} \equiv \mathbf{x} \mod q\}.$$

Such a short vector $(\mathbf{w}, \mathbf{v})$ can then be used to distinguish LWE from random. If $\mathbf{b}$ is an LWE sample, then

$$\mathbf{w} \cdot \mathbf{b} = \mathbf{w} \cdot (\mathbf{A} \cdot \mathbf{s} + \mathbf{e}) = \langle \mathbf{v}, \mathbf{s} \rangle + \langle \mathbf{w}, \mathbf{e} \rangle \mod q$$

which is short, since both $\mathbf{s}$ and $\mathbf{e}$ are short; whereas if $\mathbf{b}$ is uniformly random then so is $\mathbf{w} \cdot \mathbf{b}$. In the case of a sparse secret, combinatorial techniques can be leveraged to improve the attack [3]. In this paper, we estimate complexity of dual using the LWE estimator, which supports the small and sparse secret dual variants described in [3].

The hybrid dual attack. The dual attack was recently improved by Cheon *et al.* [25], who add a meet-in-the-middle step to the combinatorial search, giving rise to a hybrid dual attack (hybrid-dual). It is shown in [25] that when fixing a maximal memory of 2^{80}, the hybrid dual attack outperforms the dual attack for certain homomorphic encryption parameter sets with a sparse ternary secret. Cheon *et al.* [25] provide a script[4] that can be used to estimate the security of a given parameter set against the hybrid dual attack.

In this paper, we instead use a more conservative analysis for the hybrid dual attack. We estimate complexity of hybrid-dual by assuming a square-root speed-up over Albrecht's variant [3] to account for the meet-in-the-middle procedure. In addition, to align

[4] The script is available at https://github.com/swanhong/HybridLWEAttack.

with our hybrid-dec estimates, we assume that any probabilities associated to the meet-in-the-middle phase are set to one, thus providing an explicit *underestimate* of security.

In Figure 2 we show a comparison of the four attacks discussed above, under our assumptions, for parameter sets of varying sparsity. We note that as the secret becomes more dense, hybrid attacks are less effective compared to the dual and usvp attacks.

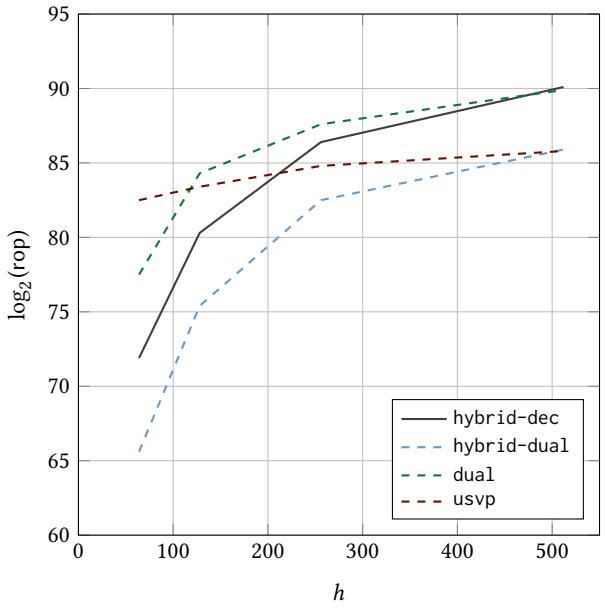

Figure 2: A comparison of the usvp, dual, hybrid-dual and hybrid-dec attacks, for the parameter set $n = 1024, q = 2^{40}$ and $\sigma \approx 3.2$ with a sparse ternary secret with a variety of Hamming weights $h \in \{64, 128, 256, 512\}$.

2.4 Currently recommended parameters

In Table 1 we reproduce the LWE parameter sets $(n, \log q, \alpha)$ that are recommended in the current version of the Homomorphic Encryption Security Standard [2] to achieve target security level λ for $\lambda \in \{128, 192, 256\}$ for power-of-two ring dimensions between 1024 and 32768 and for a secret having coefficients chosen uniformly in $\{-1, 0, 1\}$. Table 1 reports the estimated cost of running the usvp [8], dec [40] and dual [3] attacks on these parameter sets under a sieving cost model for BKZ according to the LWE Estimator [6].

n	$\log q$	α	usvp	dec	dual	λ_{target}
1024	27	8/q	**131.6**	160.2	138.7	128
2048	54	8/q	**129.7**	144.4	134.2	
4096	109	8/q	**128.1**	134.9	129.9	
8192	218	8/q	**128.5**	131.5	129.2	
16384	438	8/q	**128.1**	129.9	129.0	
32768	881	8/q	**128.5**	129.1	**128.5**	
1024	19	8/q	**193.0**	259.5	207.7	192
2048	37	8/q	**197.5**	233.0	207.8	
4096	75	8/q	**194.7**	212.2	198.5	
8192	152	8/q	**192.2**	200.4	194.6	
16384	305	8/q	**192.1**	196.2	193.2	
32768	611	8/q	**192.7**	194.2	193.7	
1024	14	8/q	**265.6**	406.4	293.8	256
2048	29	8/q	**259.1**	321.7	273.5	
4096	58	8/q	**260.4**	292.6	270.1	
8192	118	8/q	**256.7**	270.4	260.6	
16384	237	8/q	**256.9**	264.2	259.8	
32768	476	8/q	**256.4**	260.2	258.2	

Table 1: Currently standardised LWE parameters at the 128-, 192- and 256-bit security level for a uniform ternary secret specified in [2, Table 1] and estimates of their security against usvp, dec, and dual attacks under the BKZ cost model $T(\beta, d) = 2^{0.292\beta + 16.4 + \log(8d)}$, where β is the blocksize and d is the lattice dimension. The best performing attack for each parameter set is highlighted in bold.

3 STANDARDISING SPARSE SECRETS

Ternary secrets with coefficients chosen from $\{-1, 0, 1\}$, are ubiquitous in current implementations of homomorphic encryption. The TFHE library [28], whose bootstrapping process requires a binary secret, is the only major exception. Some implementations use a uniform ternary distribution, while others use a sparse ternary secret, for which all but a certain Hamming weight h of the coefficients are zero. For example, the HEAAN [39] library uses a sparse ternary secret of Hamming weight $h = 64$. All known methods for bootstrapping for the CKKS, BFV and BGV schemes use sparse secrets [20, 21, 24]. Concretely, a sparse secret of Hamming weight $h = 128$ is used in [21], while $h = 64$ is used in [24].

It may be desirable to extend the Homomorphic Encryption Security Standard to include parameter sets with such secret distributions. In this section, we consider the feasibility and impact of including sparse ternary secret distributions parameter sets in the Standard.

3.1 Using sparse secrets with existing recommended parameter sets

As a starting point, we give a rough idea of the impact of using a sparse ternary secret of Hamming weight $h = 128$ for the sets of parameters $(n, \log q, \alpha)$ as recommended in [2, Table 1] for uniform ternary secret (see also Table 1). In Table 2 we report the concrete security of the parameter sets $(n, \log q, \alpha, h)$ with a sparse ternary secret of Hamming weight $h = 128$. For consistency, we used the

same sieving cost model $T(\beta, d) = 2^{0.292\beta+16.4+\log(8d)}$ (known as `BKZ.sieve` in the LWE Estimator) as was used to generate [2, Table 1].

It can be seen from Table 2 that introducing a sparse secret of Hamming weight $h = 128$ results in a noticeable security loss. Considering only `dual` and `usvp` attacks, the security drops by up to 10 bits at the target 128-bit security level, up to 25 bits at the target 192-bit security level, and up to 50 bits at the target 256-bit security level. Considering also the hybrid attacks `hybrid-dec` and `hybrid-dual`, we see that these are the most effective attacks according to our conservative analysis. The security drops by approximately 25 bits at the target 128-bit security level, by approximately 50 bits at the target 192-bit security level, and by approximately 85 bits at the target 256-bit security level.

One of the main arguments not to standardise sparse secrets is the wider range of attacks that can apply. Moreover, cryptanalysis in the space is very fast-moving: indeed, the `hybrid-dual` attack due to Cheon *et al.* [25] was only announced in June 2019. This serves to remind us that further away we move from LWE as originally defined, the greater the potential for more efficient attacks.

3.2 Sparsity vs. performance trade-off

In Table 3 we illustrate the effect of using a sparse ternary secret with various Hamming weights h on the bit size $\log q$ of the required modulus q to achieve security with $n = 2048$ and $\sigma = 3.19$ fixed. For comparison, we also note the bit size $\log Q$ which is currently recommended to achieve target security level λ for the same n, σ with a uniform ternary secret.

n	$\log q$	α	h	usvp	dual	hybrid-dec	hybrid-dual	λ_{target}
1024	27	8/q	128	124.9	127.8	111.5	**106.2**	128
2048	54	8/q	128	125.0	122.0	108.9	**103.9**	
4096	109	8/q	128	124.9	117.9	108.3	**103.0**	
8192	218	8/q	128	126.4	117.2	110.0	**103.9**	
1024	19	8/q	128	178.2	178.8	146.2	**141.8**	192
2048	37	8/q	128	186.5	173.8	143.6	**141.7**	
4096	75	8/q	128	186.6	165.2	143.7	**139.2**	
8192	152	8/q	128	186.4	167.5	143.6	**138.6**	
1024	14	8/q	128	235.5	238.5	181.5	**176.6**	256
2048	29	8/q	128	231.9	217.3	170.7	**168.4**	
4096	58	8/q	128	234.3	210.2	170.0	**167.5**	
8192	118	8/q	128	232.8	207.9	170.6	**172.1**	

Table 2: Impact of using a sparse ternary secret of Hamming weight $h = 128$, using the currently standardised LWE parameter sets at the target 128-, 192- and 256-bit security level for a uniform ternary secret specified in [2, Table 1]. An estimate of the security of each parameter set against `usvp`, `dual`, `hybrid-dec` and `hybrid-dual` attacks under the BKZ cost model $T(\beta, d) = 2^{0.292\beta+16.4+\log(8d)}$ is presented, where β is the blocksize and d is the dimension. A conservative analysis for both the `hybrid-dec` and `hybrid-dual` attacks is used which assumes a square-root speed-up in the search space, and ignores any meet-in-the-middle probabilities. The best performing attack for each parameter set is highlighted in bold.

A smaller modulus q may impact on practical performance of the schemes. For example, in a levelled setting, we need to ensure that q is large enough to support the full computation to ensure correct decryption. The lower q required by introducing a sparse secret may necessitate moving to a higher dimension n to support the computation, which in turn will be slower.

h	n	λ	$\log q$	$\log Q$
128	2048	128	41	54
		192	24	37
		256	15	29
256	2048	128	48	54
		192	31	37
		256	22	29
512	2048	128	52	54
		192	35	37
		256	26	29
$\lceil \frac{2n}{3} \rfloor$	2048	128	54	54
		192	37	37
		256	28	29

Table 3: Bit size $\log q$ of moduli required to provide target security level λ, for $\lambda \in \{128, 192, 256\}$, for various secret densities. For comparison, the bit size $\log Q$ recommended for a uniform ternary secret and the same n, σ, λ is also given.

We note that the LWE Estimator treats uniform ternary secrets as fixed weight ternary secrets with Hamming weight $h = \lceil \frac{2n}{3} \rfloor$. For security level $\lambda = 256$, the currently standardised modulus for uniform ternary secrets is $\log(Q) = 29$. In Table 3, we see that under our conservative analysis of the `hybrid-dec` and `hybrid-dual` attack, a smaller modulus of size $\log(q) = 28$ is required in order to attain security level $\lambda = 256$ for a fixed weight ternary secret of Hamming weight $h = \lceil \frac{2n}{3} \rfloor$. Under a less conservative analysis, the currently standardised parameters for uniform ternary secrets would all attain their required security levels. However, this example highlights the need to consider hybrid attacks even in the uniform ternary case, as future improvements could affect the currently standardised parameters.

3.3 Sparsity as a proportion of target security: an exploration of choices for ζ

In Table 4 we present an exploration of possible choices for the value $\zeta = \frac{h}{\lambda}$, illustrating the reduction in bitsize $\log q$ of the modulus q required to retain the desired level of security when using a sparse ternary secret compared to a uniform ternary secret. Table 4 uses the cost model $T(\beta, d) = 2^{0.292\beta+16.4+\log(8d)}$ (that is, `BKZ.sieve` in the LWE Estimator) and considers the following attacks: `usvp`, `dual`, `hybrid-dec`, and `hybrid-dual`. We provide as a point of comparison $\log Q$, the bit size of the modulus Q currently recommended in [2] for the given parameters ($n, \sigma = 3.19$) with uniform ternary secret at target security level λ.

Table 4 indicates that a choice such as $\zeta = 1$, i.e. a Hamming weight $h = \lambda$ for target security level λ, gives a reasonable trade-off between performance and security. In this case, we can retain secure parameters with at most a 27% drop in the bitsize of modulus $\log q$ compared to that recommended at target security level λ for the same $(n, \sigma = 3.19)$ and uniform ternary secret. This choice corresponds to a sparse ternary secret with Hamming weight $h \in \{128, 192, 256\}$ depending on the desired security level, which allows for practicable bootstrapping. We consider parameter sets with $\zeta = 1$ as an example in Section 4.

n	λ	$\log q^{(\zeta=\frac{1}{2})}$	$\log q^{(\zeta=\frac{3}{4})}$	$\log q^{(\zeta=1)}$	$\log q^{(\zeta=\frac{3}{2})}$	$\log Q$
1024	128	14	19	21	23	27
	192	9	13	14	16	19
	256	7	10	11	12	14
2048	128	27	37	41	46	54
	192	19	26	29	32	37
	256	15	19	22	24	29
4096	128	55	74	83	92	109
	192	37	52	57	64	75
	256	30	39	44	49	58
8192	128	111	148	171	186	218
	192	84	100	114	130	152
	256	60	79	89	98	118
16384	128	223	300	342	377	438
	192	157	201	232	265	305
	256	115	161	176	202	237
32768	128	496	619	699	767	881
	192	350	411	479	523	611
	256	263	313	361	408	476

Table 4: The reduction in bitsize $\log q$ of the modulus q required to retain the desired level of security against `dual`, `usvp`, `hybrid-dual` and `hybrid-dec` when using a sparse ternary secret compared to a uniform ternary secret (which has recommended bitsize $\log Q$). The cost model $T(\beta, d) = 2^{0.292\beta+16.4+\log(8d)}$ is used. A conservative analysis for both the `hybrid-dual` and `hybrid-dec` attacks is used which assumes a square-root speed-up in the search space, and ignores any meet-in-the-middle probabilities.

4 STANDARDISING LARGER DIMENSIONS

With current progress in applied homomorphic encryption it is becoming necessary to work in dimensions larger than $n = 2^{15}$, the largest dimension currently standardised. Several recent papers [20, 24, 34, 37] have reported implementations in dimension $n = 2^{16}$, and an implementation in dimension $n = 2^{17}$ was reported in [38]. A natural extension of the current standard would therefore be to standardise parameter sets for dimension $n = 2^k$ for some $k \geq 16$, since power-of-two n remain the most widely used in practice. Moreover, power-of-two n enable convenient coefficient-wise error sampling and would require no change to the currently standardised error distribution, while other choices for n would make the choice of error distribution more complex.[5]

λ	n	$\log q$	`usvp`	`dual`
128	65536	1782	**128.3**	128.4
192	65536	1242	192.5	**192.0**
256	65536	963	**256.7**	257.7

Table 5: Required bit size $\log q$ of moduli required to attain target security level λ, with $\lambda \in \{128, 192, 256\}$, for dimension $n = 65536$, for a uniform ternary secret distribution. An estimate of the security of each parameter set against the `usvp` and `dual` attacks under the sieving-based cost model $T(\beta, d) = 2^{0.292\beta+16.4+\log(8d)}$ is presented. The best performing attack for each parameter set is highlighted in bold.

For $n = 2^{16}$, it is straightforward to apply the methodology in the current Homomorphic Encryption Security Standard [2] to use the LWE Estimator to find an appropriate $\log q$ to meet security requirements for fixed $\sigma = 3.19$ and a currently standardised secret distribution. We present the results of such an analysis for a uniform ternary secret distribution in Table 5, which gives an estimate of the security of the proposed parameter sets $(n = 2^{16}, \sigma = 3.19, \log q)$ against the `usvp` and `dual` attacks under a sieving lattice reduction cost model. Since we use the current methodology of the Homomorphic Encryption Security Standard to generate Table 5, we do not consider hybrid attacks.[6] We also omit `dec` estimates in Table 5 (as in the rest of this paper) as these are known to be inaccurate.

For $n \geq 2^{17}$ a larger power of two, the same methodology works in theory, although it can become cumbersome: the LWE estimator can take hours to run for each such large input parameter set, and we must run it for every candidate modulus q. To find a suitable q achieving target security λ for higher values of n, we can extrapolate using the data we already have using the apparent linear relationship between n and $\log q$. That is, $n/\log q$ is essentially constant for a fixed target security level. This means we can easily extrapolate entries for larger values of n, without having to explicitly run the LWE estimator many times. Such an extrapolation could help to identify a good place to start the search for $\log q$, and then the security of an identified parameter set can be confirmed using the LWE estimator.

4.1 Extrapolating Sparse Secrets

We illustrate in Figure 3 such an extrapolation for a sparse ternary secret of fixed Hamming weight $h = \lambda$ (i.e. $\zeta = 1$). When considering pre-quantum estimates, we can represent $\log(q)$ as a linear function of n by extrapolation from our data:

$$\lg q^{\text{sieve}}_{\lambda=128}(n) = 0.021370n - 3.601989$$
$$\lg q^{\text{sieve}}_{\lambda=192}(n) = 0.014630n - 3.139303$$
$$\lg q^{\text{sieve}}_{\lambda=256}(n) = 0.011007n - 1.184080$$

[5] We note that the current Homomorphic Encryption Security Standard [2] already states that extension to general cyclotomic rings is eventually envisaged, and this will lead to standardising n and σ of a different form.

[6] We note that our script for estimating the cost of the hybrid decoding attack requires more memory than we have available when attempting to run in dimension $n = 2^{16}$.

These linear models were found using the `find_fit` function in SageMath [46]. For readability, we round the coefficients to six decimal places in each case.

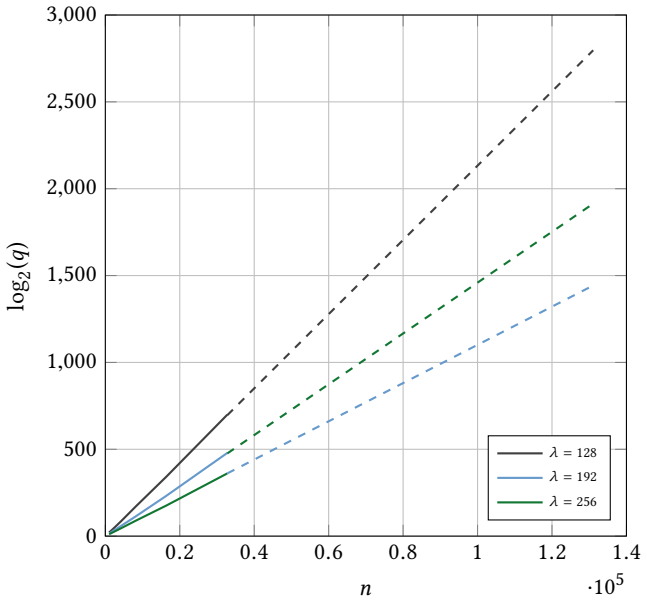

Figure 3: Extrapolation to $n = 65536$ and $n = 131072$ using the data from Table 4 for the value $\zeta = 1$. We consider the lattice reduction cost model $T_{\mathbf{BKZ}}(\beta, d) = 2^{0.292\beta+16.4+\log(8d)}$ and extrapolate using the Sage function `find_fit`. The solid lines represents data points and the dashed lines represent extrapolation.

As an example, in the case of $\zeta = 1$ and $\lambda = 128$, for $n = 65536$ we have

$$\lg q^{\text{sieve}}_{\lambda=128}(65536) = 1396$$

and for a security level of $\lambda = 256$, we have:

$$\lg q^{\text{sieve}}_{\lambda=256}(65536) = 720.$$

5 DISCUSSION

We conclude with a number of points for discussion which arise from our work, and highlight some natural directions for future work.

Should the Standard be based on implementation choices? Historically, the parameter sets in the standard have been based on implementation choices. For example, the choice of standard deviation $\sigma = 3.19$, appearing in all currently standardised parameter sets, originates in an early implementation [31]. This narrow error distribution already deviates from the regime of provably secure LWE instances, and using a sparse secret would introduce a further nonstandard aspect. An important issue for the community to consider is whether any standardised parameter sets should be driven by implementation progress, or if instead implementors should be encouraged to select parameters for which we have strong confidence in their security.

Should we standardise sparse-secret parameter sets? If we do decide to choose parameters based on implementation choices, then from the point of view of current bootstrapping approaches it would make sense to standardise a parameter set including a sparse secret. If the community does wish to standardise sparse-secret parameter sets, a natural question would be to decide if there is an appropriate sparseness that balances performance (e.g. efficiency of bootstrapping in practice) with security. For example, what would be an appropriate choice of ζ and how would this be justified?

On the other hand, it is clear that cryptanalysis of sparse-secret LWE is fast-paced, illustrated by the very recent `hybrid-dual` attack [25]. It is plausible that further attacks on sparse-secret LWE variants will be discovered in the short-term. Moreover, increasing the number of standardised LWE variants would spread out cryptanalytic efforts, making it hard to get higher confidence in the concrete security of any version of LWE.

In this paper, we estimated the cost of running hybrid attacks using a conservative analysis. This is reasonable in the context of setting secure parameters and can mitigate against the threat of ongoing cryptanalytic improvements. However, such a conservative analysis may lead to choosing a larger dimension n than is needed, and hence lower efficiency. If sparse secrets were standardised, a more realistic analysis such as that presented by Wunderer [47] may be preferable.[7]

Improving the LWE Estimator. The methodology in the current Standard relies on the LWE Estimator [6], but we note that the Estimator has a number of limitations. In this paper we excluded `dec` estimates output from the Estimator as these are known to be inaccurate: however, any standardised parameter sets should be shown to be secure against a state-of-the-art primal decoding attack.

More generally, it is cumbersome to recommend parameter sets which are not easily verified by the Estimator, such as those with binary secrets[8], as in [28], and those vulnerable to hybrid attacks. We note that the security of the uniform ternary secret parameter sets recommended in [2] remain unaffected when also considering `hybrid-dec`.

The code to estimate `hybrid-dual` used in [25] seems to be based on the Estimator and therefore may not be too difficult to integrate, although it currently only supports sparse secret distributions. Improving the Estimator to add support for binary secrets and hybrid attacks would be beneficial to the community and equally so for the Estimator itself, as it provides an opportunity for additional code review. In summary, an important direction for the community is improving the LWE Estimator.

Should we alter the methodology used in the Standard? At present, sets of LWE parameters themselves are standardised. Their security is assured using estimates of relevant attacks output by the LWE Estimator. Perhaps it would be better to standardise the process itself. That is, the community could standardise the process of using the Estimator (or another tool) to verify estimated security of parameters. Then, implementors could select any set of parameters appropriate to their needs, demonstrate they were

[7] A preprint detailing the application of Wunderer's analysis to the homomorphic encryption setting is available at https://github.com/rachelplayer/LatRedHybrid

[8] Combinatorial techniques are not currently supported for a binary secret distribution, leading to an overestimation of security.

σ	n	$\log q$	usvp	dual	hybrid-dec	hybrid-dual
3.2	4096	109	**128.1**	129.9	131.3	128.7
32	4096	111	129.6	132.6	133.1	**129.3**
320	4096	115	**128.7**	130.5	132.6	129.6
3200	4096	119	**128.1**	130.2	131.3	129.0

Table 6: Bit size $\log q$ of moduli required to attain a target security level of $\lambda = 128$ for a fixed dimension $n = 4096$, a fixed uniform ternary secret distribution, and varying σ. An estimate of the security of each parameter set against the usvp, dual, hybrid-dec, and hybrid-dual attacks under the sieving-based cost model $T(\beta, d) = 2^{0.292\beta+16.4+\log(8d)}$ is also given. The best performing attack for each parameter set is highlighted in bold.

secure according to the Estimator, and the parameters would be deemed secure according to the Standard. Of course, this would involve agreeing an appropriate BKZ cost model (for which there is generally a lack of consensus [4]), and a list of relevant attacks (which may in turn involve incorporating those attacks into the Estimator).

Should we standardise different values of σ? The choice of the standard deviation σ of the error distribution has an impact on both the security of the underlying LWE problem as well as the noise growth as homomorphic operations are performed[9]. In the current Homomorphic Encryption Standard, the standard deviation is fixed at $\sigma \approx 3.19$, i.e. $\alpha = \frac{8}{q}$. The choice of $\sigma \approx 3.19$ appears to be somewhat arbitrary, originating in an early implementation [31]. Nevertheless, it remains a popular choice in many (though not all [28]) current implementations.

A natural way to extend the standard would be to consider other choices of σ which balance security and noise growth. As a starting point, in Table 6 we present the bitsize of modulus $\log q$ that would be required to attain security level $\lambda = 128$ for a fixed dimension $n = 4096$ and a uniform random ternary secret, when varying $\sigma \in \{3.2, 32, 320, 3200\}$.

ACKNOWLEDGMENTS

The research of Curtis was supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/K035584/1). The research of Player was supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701). The authors would like to thank Martin Albrecht, Kim Laine, Kristin Lauter, Daniele Micciancio, and Yuriy Polyakov for their feedback on earlier versions of this work, and the anonymous reviewers for their useful comments.

REFERENCES

[1] 2019. IDASH Privacy and Security Workshop. http://www.humangenomeprivacy.org/2019. (2019).

[2] Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. 2018. *Homomorphic Encryption Security Standard.* Technical Report. HomomorphicEncryption.org, Toronto, Canada.

[3] Martin R. Albrecht. 2017. On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL. In *EUROCRYPT 2017, Part II (LNCS)*, Jean-Sébastien Coron and Jesper Buus Nielsen (Eds.), Vol. 10211. Springer, Heidelberg, 103–129. https://doi.org/10.1007/978-3-319-56614-6_4

[4] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. 2018. Estimate All the {LWE, NTRU} Schemes!. In *Security and Cryptography for Networks*, Dario Catalano and Roberto De Prisco (Eds.). Springer International Publishing, 351–367.

[5] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. 2017. Revisiting the Expected Cost of Solving uSVP and Applications to LWE. In *ASIACRYPT 2017, Part I (LNCS)*, Tsuyoshi Takagi and Thomas Peyrin (Eds.), Vol. 10624. Springer, Heidelberg, 297–322. https://doi.org/10.1007/978-3-319-70694-8_11

[6] Martin R Albrecht, Rachel Player, and Sam Scott. 2015. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology* 9, 3 (2015), 169–203.

[7] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. 2016. Post-quantum Key Exchange - A New Hope. In *25th USENIX Security Symposium, USENIX Security 16*, Thorsten Holz and Stefan Savage (Eds.). USENIX Association, 327–343. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

[8] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. 2016. Post-quantum Key Exchange - A New Hope. In *USENIX Security 2016*, Thorsten Holz and Stefan Savage (Eds.). USENIX Association, 327–343.

[9] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. 2009. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In *CRYPTO 2009 (LNCS)*, Shai Halevi (Ed.), Vol. 5677. Springer, Heidelberg, 595–618. https://doi.org/10.1007/978-3-642-03356-8_35

[10] David Archer, Lily Chen, Jung Hee Cheon, Ran Gilad-Bachrach, Roger A. Hallman, Zhicong Huang, Xiaoqian Jiang, Ranjit Kumaresan, Bradley A. Malin, Heidi Sofia, Yongsoo Song, and Shuang Wang. 2017. *Applications of Homomorphic Encryption.* Technical Report. HomomorphicEncryption.org, Redmond WA, USA.

[11] Sanjeev Arora and Rong Ge. 2011. New Algorithms for Learning in Presence of Errors. In *ICALP 2011, Part I (LNCS)*, Luca Aceto, Monika Henzinger, and Jiri Sgall (Eds.), Vol. 6755. Springer, Heidelberg, 403–415. https://doi.org/10.1007/978-3-642-22006-7_34

[12] László Babai. 1985. On Lovász' Lattice Reduction and the Nearest Lattice Point Problem (Shortened Version). In *STACS '86 (Lecture Notes in Computer Science)*, Kurt Mehlhorn (Ed.), Vol. 82. Springer, 13–20.

[13] Shi Bai and Steven D. Galbraith. 2014. Lattice Decoding Attacks on Binary LWE. In *ACISP 14 (LNCS)*, Willy Susilo and Yi Mu (Eds.), Vol. 8544. Springer, Heidelberg, 322–337. https://doi.org/10.1007/978-3-319-08344-5_21

[14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2012. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS 2012*, Shafi Goldwasser (Ed.). ACM, 309–325. https://doi.org/10.1145/2090236.2090262

[15] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2014. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)* 6, 3 (2014), 13.

[16] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. 2013. Classical hardness of learning with errors. In *45th ACM STOC*, Dan Boneh, Tim Roughgarden, and Joan Feigenbaum (Eds.). ACM Press, 575–584. https://doi.org/10.1145/2488608.2488680

[17] Michael Brenner, Wei Dai, Shai Halevi, Kyoohyung Han, Amir Jalali, Miran Kim, Kim Laine, Alex Malozemoff, Pascal Paillier, Yuriy Polyakov, Kurt Rohloff, Erkay Savaş, and Berk Sunar. 2017. *A Standard API for RLWE-based Homomorphic Encryption.* Technical Report. HomomorphicEncryption.org, Redmond WA, USA.

[18] Johannes A. Buchmann, Florian Göpfert, Rachel Player, and Thomas Wunderer. 2016. On the Hardness of LWE with Binary Error: Revisiting the Hybrid Lattice-Reduction and Meet-in-the-Middle Attack. In *AFRICACRYPT 16 (LNCS)*, David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi (Eds.), Vol. 9646. Springer, Heidelberg, 24–43. https://doi.org/10.1007/978-3-319-31517-1_2

[19] Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Jeffrey Hoffstein, Kristin Lauter, Satya Lokam, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. 2017. *Security of Homomorphic Encryption.* Technical Report. HomomorphicEncryption.org, Redmond WA, USA.

[20] Hao Chen, Ilaria Chillotti, and Yongsoo Song. 2019. Improved Bootstrapping for Approximate Homomorphic Encryption. In *EUROCRYPT 2019, Part II (LNCS)*, Yuval Ishai and Vincent Rijmen (Eds.), Vol. 11477. Springer, Heidelberg, 34–54. https://doi.org/10.1007/978-3-030-17656-3_2

[21] Hao Chen and Kyoohyung Han. 2018. Homomorphic Lower Digits Removal and Improved FHE Bootstrapping. In *EUROCRYPT 2018, Part I (LNCS)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.), Vol. 10820. Springer, Heidelberg, 315–337. https://doi.org/10.1007/978-3-319-78381-9_12

[9] This is clear since the noise in a fresh ciphertext depends on the LWE error distribution, see e.g. [14, 26, 29].

[22] Yuanmi Chen and Phong Q. Nguyen. 2011. BKZ 2.0: Better Lattice Security Estimates. In *ASIACRYPT 2011 (LNCS)*, Dong Hoon Lee and Xiaoyun Wang (Eds.), Vol. 7073. Springer, Heidelberg, 1–20. https://doi.org/10.1007/978-3-642-25385-0_1

[23] Yuanmi Chen and Phong Q. Nguyen. 2012. Faster Algorithms for Approximate Common Divisors: Breaking Fully-Homomorphic-Encryption Challenges over the Integers. In *EUROCRYPT 2012 (LNCS)*, David Pointcheval and Thomas Johansson (Eds.), Vol. 7237. Springer, Heidelberg, 502–519. https://doi.org/10.1007/978-3-642-29011-4_30

[24] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2018. Bootstrapping for Approximate Homomorphic Encryption. In *EUROCRYPT 2018, Part I (LNCS)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.), Vol. 10820. Springer, Heidelberg, 360–384. https://doi.org/10.1007/978-3-319-78381-9_14

[25] Jung Hee Cheon, Minki Hhan, Seungwan Hong, and Yongha Son. 2019. A Hybrid of Dual and Meet-in-the-Middle Attack on Sparse and Ternary Secret LWE. *IEEE Access* 7 (2019), 89497–89506. https://doi.org/10.1109/ACCESS.2019.2925425

[26] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. 2017. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *ASIACRYPT 2017, Part I (LNCS)*, Tsuyoshi Takagi and Thomas Peyrin (Eds.), Vol. 10624. Springer, Heidelberg, 409–437. https://doi.org/10.1007/978-3-319-70694-8_15

[27] Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song. 2018. Lizard: Cut Off the Tail! A Practical Post-quantum Public-Key Encryption from LWE and LWR. In *SCN 18 (LNCS)*, Dario Catalano and Roberto De Prisco (Eds.), Vol. 11035. Springer, Heidelberg, 160–177. https://doi.org/10.1007/978-3-319-98113-0_9

[28] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. August 2016. TFHE: Fast Fully Homomorphic Encryption Library. (August 2016). https://tfhe.github.io/tfhe/.

[29] Junfeng Fan and Frederik Vercauteren. 2012. Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive, Report 2012/144. (2012). http://eprint.iacr.org/2012/144.

[30] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *41st ACM STOC*, Michael Mitzenmacher (Ed.). ACM Press, 169–178. https://doi.org/10.1145/1536414.1536440

[31] Craig Gentry, Shai Halevi, and Nigel P. Smart. 2012. Homomorphic Evaluation of the AES Circuit. In *CRYPTO 2012 (LNCS)*, Reihaneh Safavi-Naini and Ran Canetti (Eds.), Vol. 7417. Springer, Heidelberg, 850–867. https://doi.org/10.1007/978-3-642-32009-5_49

[32] Florian Göpfert, Christine van Vredendaal, and Thomas Wunderer. 2017. A Hybrid Lattice Basis Reduction and Quantum Search Attack on LWE. In *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, Tanja Lange and Tsuyoshi Takagi (Eds.). Springer, Heidelberg, 184–202. https://doi.org/10.1007/978-3-319-59879-6_11

[33] Shai Halevi. 2018. HElib. https://github.com/shaih/HElib. (2018).

[34] Kyoohyung Han, Minki Hhan, and Jung Hee Cheon. 2019. Improved Homomorphic Discrete Fourier Transforms and FHE Bootstrapping. *IEEE Access* 7 (2019), 57361–57370. https://doi.org/10.1109/ACCESS.2019.2913850

[35] Nick Howgrave-Graham. 2007. A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU. In *CRYPTO 2007 (LNCS)*, Alfred Menezes (Ed.), Vol. 4622. Springer, Heidelberg, 150–169. https://doi.org/10.1007/978-3-540-74143-5_9

[36] Xiaoqian Jiang, Miran Kim, Kristin E. Lauter, and Yongsoo Song. 2018. Secure Outsourced Matrix Computation and Application to Neural Networks. In *ACM CCS 2018*, David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang (Eds.). ACM Press, 1209–1222. https://doi.org/10.1145/3243734.3243837

[37] Andrey Kim, Yongsoo Song, Miran Kim, Keewoo Lee, and Jung Hee Cheon. 2018. Logistic Regression Model Training based on the Approximate Homomorphic Encryption. Cryptology ePrint Archive, Report 2018/254. (2018). https://eprint.iacr.org/2018/254.

[38] Miran Kim, Yongsoo Song, Shuang Wang, Yuhou Xia, and Xiaoqian Jiang. 2018. Secure Logistic Regression Based on Homomorphic Encryption: Design and Evaluation. *JMIR Med Inform* 6, 2 (17 Apr 2018), e19. https://doi.org/10.2196/medinform.8805

[39] SNU Cryptography Lab. 2018. HEAAN. https://github.com/snucrypto/HEAAN. (2018).

[40] Richard Lindner and Chris Peikert. 2011. Better Key Sizes (and Attacks) for LWE-Based Encryption. In *CT-RSA 2011 (LNCS)*, Aggelos Kiayias (Ed.), Vol. 6558. Springer, Heidelberg, 319–339. https://doi.org/10.1007/978-3-642-19074-2_21

[41] Mingjie Liu and Phong Q. Nguyen. 2013. Solving BDD by Enumeration: An Update. In *CT-RSA 2013 (LNCS)*, Ed Dawson (Ed.), Vol. 7779. Springer, Heidelberg, 293–309. https://doi.org/10.1007/978-3-642-36095-4_19

[42] Microsoft. 2018. SEAL. https://github.com/Microsoft/SEAL. (2018).

[43] New Jersey Institute of Technology. 2019. PALISADE. https://git.njit.edu/palisade/PALISADE. (2019).

[44] Oded Regev. 2005. On lattices, learning with errors, random linear codes, and cryptography. In *37th ACM STOC*, Harold N. Gabow and Ronald Fagin (Eds.). ACM Press, 84–93. https://doi.org/10.1145/1060590.1060603

[45] Claus Peter Schnorr. 2003. Lattice Reduction by Random Sampling and Birthday Methods. In *STACS 2003*, Helmut Alt and Michel Habib (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 145–156.

[46] William Stein et al. 2017. *Sage Mathematics Software Version 8.0.* The Sage Development Team. http://www.sagemath.org.

[47] Thomas Wunderer. 2018. *On the Security of Lattice-Based Cryptography Against Lattice Reduction and Hybrid Attacks.* Ph.D. Dissertation. Technische Universität, Darmstadt. http://tuprints.ulb.tu-darmstadt.de/8082/