Practical Cryptanalysis of a Public-Key Encryption Scheme Based on New Multivariate Quadratic Assumptions

Martin R. Albrecht[1], Jean-Charles Faugère[3,2,4], Robert Fitzpatrick[5],
Ludovic Perret[2,3,4], Yosuke Todo[6], and Keita Xagawa[6]

[1] Technical University of Denmark, Denmark
[2] Sorbonne Universités, UPMC Univ Paris 06, POLSYS,
UMR 7606, LIP6, F-75005, Paris, France
[3] INRIA, Paris-Rocquencourt Center, POLSYS Project
[4] CNRS, UMR 7606, LIP6, F-75005, Paris, France
[5] Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, United Kingdom
[6] NTT Secure Platform Laboratories
maroa@dtu.dk, jean-charles.faugere@inria.fr,
robert.fitzpatrick.2010@live.rhul.ac.uk, ludovic.perret@lip6.fr,
{todo.yosuke,xagawa.keita}@lab.ntt.co.jp

Abstract. In this paper, we investigate the security of a public-key encryption scheme introduced by Huang, Liu and Yang (HLY) at PKC'12. This new scheme can be provably reduced to the hardness of solving a set of quadratic equations whose coefficients of highest degree are chosen according to a discrete Gaussian distributions. The other terms being chosen uniformly at random. Such a problem is a variant of the classical problem of solving a system of non-linear equations (PoSSo), which is known to be hard for random systems. The main hypothesis of Huang, Liu and Yang is that their variant is not easier than solving PoSSo for random instances. In this paper, we disprove this hypothesis. To this end, we exploit the fact that the new problem proposed by Huang, Liu and Yang reduces to an easy instance of the Learning With Errors (LWE) problem. The main contribution of this paper is to show that security and efficiency are essentially incompatible for the HLY proposal. That is, one cannot find parameters which yield a secure and a practical scheme. For instance, we estimate that a public-key of at least 1.03 GB is required to achieve 80-bit security against the simplest of our attacks. As a proof of concept, we present 3 practical attacks against all the parameters proposed by Huang, Liu and Yang. With the most efficient attack, we have been able to recover the private-key in roughly 5 minutes for the first challenge (i.e. Case 1) proposed by HLY and less than 30 minutes for the second challenge (i.e. Case 2).

1 Introduction

At PKC 2012 Huang, Liu and Yang (HLY) proposed a new public-key encryption scheme [17]. It follows a line of research, called Multivariate Quadratic ($\mathcal{MQ}$)

H. Krawczyk (Ed.): PKC 2014, LNCS 8383, pp. 446–464, 2014.

cryptography, to construct public-key encryption schemes from the known hard problem of solving systems of polynomial equations. This line of research dates back to the mid eighties with the design of C* [24], later followed by many other proposals. While this family of designs is commonly considered to be an interesting alternative to constructions based on number-theoretic problems (in the post-quantum setting), it suffers from a lack of clear security reductions to well-understood problems, leading to a series of attacks.

In contrast, [17] is part of a recent trend in $\mathcal{MQ}$ cryptography of designing cryptosystems whose security can be provably reduced to the the hardness of solving a system of non-linear equations (other examples include [3,8]). The key innovation of Huang-Liu-Yang [17] is a $\mathcal{MQ}$ scheme in which the public key is noise-free and non-linear but ciphertexts are noisy and linear. Hence, the scheme proposed by Huang, Liu, and Yang can be viewed as a hybrid between the Learning with Errors (LWE) problem [27] and $\mathcal{MQ}$ cryptosystems. The semantic security of the scheme [17] can be provably reduced to the difficulty of solving a system of non-linear equations which is somewhat structured as the coefficients of the non-linear parts of the polynomials are chosen according to a discrete Gaussian. The main assumption of [17] is that this new problem is not easier than the problem of solving a random system of quadratics equations.

1.1 Organisation of the Paper and Overview of the Results

After this introduction, the paper is organized as follows. We first provide a brief introduction to lattices and algorithms for solving LWE in Section 2. In particular, we briefly recall in Section 2.2 Micciancio and Regev's [26,21] distinguishing approach and Kannan's embedding technique [18] for solving LWE. We then describe the HLY proposal in Section 3. The new hard problem introduced by Huang, Liu and Yang is as follows:

Definition 1 ($\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$)**.** *Let n be positive integer, $m = cn$ for some $c \geq 1$, q be a polynomially bounded prime, a constant $\beta, 0 < \beta < q/2$ and $\mathbf{s}$ be a secret vector in $H_\beta := [-\beta, \ldots, \beta]^n \subseteq \mathbb{Z}_q^n$. We denote by $\mathbb{Z}_q^{\Phi_\zeta}[x_1, \ldots, x_n]$ the distribution on quadratic polynomials of $\mathbb{Z}_q[x_1, \ldots, x_n]$ obtained by sampling the monomials of degree 2 according to a discrete Gaussian distribution Φ_ζ of standard deviation $\zeta \in \mathcal{O}(1)$ and centred on zero and by sampling the others coefficients (linear, and constant parts) uniformly at random.* $\mathrm{MQ}_{\mathbf{s},\Phi}^{(n)}$ *is the probability distribution on the $\mathbb{Z}_q[x_1, \ldots, x_n]^m \times \mathbb{Z}_q^m$ obtained by sampling $\mathbf{p} = (p_1, \ldots, p_m)$ from $\mathbb{Z}_q^{\Phi_\zeta}[\mathbf{x}]^m$, and returning $(\mathbf{p}, \mathbf{c}) = (\mathbf{p}, \mathbf{p}(\mathbf{s})) \in \mathbb{Z}_q[x_1, \ldots, x_n]^m \times \mathbb{Z}_q^m$.* $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ *is the problem of finding $\mathbf{s} \in H_\beta^n$ given a pair $(\mathbf{p}, \mathbf{p}(\mathbf{s})) \leftarrow_\$ \mathrm{MQ}_{\mathbf{s},\Phi}^{(n)}$.*

The main assumption from [17] is that $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ is not easier than the problem of solving a random system of quadratic equations (Assumption 1). Remark that the latter problem is notoriously known as a hard problem from a theoretical [14] and practical point of view [5,6,7]. In this paper, we show that $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ is in fact related to a much easier problem. The starting point of our analysis is to simply remark

(Fact 1) that $\mathrm{MQ}(n, m\Phi_\zeta, H_\beta)$ resembles to a LWE problem with a discrete Gaussian with variance $\gamma^2 = O\left(n^2\beta^2\zeta^2\right)$ (centred at zero).

We use this fact, together with the Micciancio-Regev distinguisher and the lattice-reduction complexity model of Lindner and Peikert to derive a new necessary conditions on the security of the HLY scheme (Section 4). In particular, such scheme has at most τ-bit security – with regard to constructing a distinguisher of advantage d – if $(n, \beta, c, k, \tau, d)$ verifies

$$\exp\left(-\frac{\pi^2}{12\beta^2} \cdot (c\,k)^{-2} \cdot n^{-4} \cdot 2^{\frac{3.6cn}{\tau+78.9}}\right) = d.$$

For example, with $\beta = c = 2$, $k = 12$, $d = 0.5$, setting $n = 1140$ satisfies this condition for $\tau = 80$. With $n = 1140$, however, the public-key is of size ≈ 1.03 GB.

It appears then that all parameters suggested in [17] (reproduced Table 1) are too small to verify our new security condition. Indeed, we have been able to mount several practical attacks: distinguishing attack with Micciancio-Regev, and a key-recovery attack with the embedding technique, and an improved key-recovery attack exploiting the presence of a small secret (Section 5). We successfully run the two first attacks in roughly one day for the first challenge (i.e. Case 1)) and in roughly three days for the second challenge (i.e. Case 2) proposed by the authors [17]. The last practical attack is attack even more efficient. For the first challenge, we recovered the secret-key in less than 5 minutes and less than 30 minutes for the second challenge. The experimental results are detailed in Section 6.

2 Preliminaries

Notation. In the following we always start counting at zero, denote vectors and matrices in bold, vectors in lower case, and matrices in upper case. Given a vector $\mathbf{a}$, we denote by $\mathbf{a}_{(i)}$ the i-th entry in $\mathbf{a}$, and by $\mathbf{A}_{(i,j)}$ the entry at index (i, j). When given a list of vectors, we index its elements by subscript, e.g. $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2$, to denote the first three vectors of the list. Let q be a prime. We represent elements in $\mathbb{Z}_q$ as integers in $[-\frac{q}{2}, \ldots, \frac{q}{2}]$. We work in the Euclidean norm throughout. We denote by $\overline{\mathbb{Z}_q}$ the algebraic closure of $\mathbb{Z}_q$.

2.1 Background on Lattices

A lattice Λ in $\mathbb{R}^m$ is a discrete additive subgroup. For a general introduction, the reader is referred to [25]. We view a lattice as being generated by a (non-unique) basis $\mathbf{B} = \{\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}\} \subset \mathbb{Z}^m$ of linearly-independent integer vectors. We assume that the vectors $\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}$ form the rows of the $n \times m$ matrix $\mathbf{B}$. That is: $\Lambda = \mathcal{L}(\mathbf{B}) = \mathbb{Z}^n \cdot \mathbf{B} = \left\{\sum_{i=0}^{n-1} x_i \cdot \mathbf{b}_i \mid x_0, \ldots, x_{n-1} \in \mathbb{Z}\right\}$. In this work, we are concerned only with q-ary lattices which are those such that $q\mathbb{Z}^m \subseteq \Lambda \subseteq \mathbb{Z}^m$. We also restrict our attention to *full-rank* lattices i.e. those in which $\dim(\mathrm{span}(\Lambda)) = m$. The determinant or volume $\mathrm{vol}(\Lambda)$ of a (full-rank) lattice Λ is the determinant of any given basis of Λ, hence $\mathrm{vol}(\Lambda) = \det(\mathbf{B})$.

The *dual* of a lattice Λ, denoted by Λ^*, is the lattice consisting of the set of all vectors $\mathbf{z} \in \mathbb{R}^m$ such that $\langle \mathbf{y}, \mathbf{z} \rangle \in \mathbb{Z}$ for all vectors $\mathbf{y} \in \Lambda$. Given a lattice Λ, we denote by $\lambda_i(\Lambda)$ the i-th minimum of Λ defined as $\lambda_i(\Lambda) := \inf \left\{ r \mid \dim(\operatorname{span}(\Lambda \cap \bar{\mathcal{B}}_m(\mathbf{0}, r))) \geq i \right\}$, where $\bar{\mathcal{B}}_m(\mathbf{0}, r)$ denotes the closed, zero-centered m-dimensional (Euclidean) ball of radius r. We define the minimum distance from a given point $\mathbf{t} \in \mathbb{R}^m$ to the lattice by $\operatorname{dist}(\Lambda, \mathbf{t}) = \min \{ \|\mathbf{t} - \mathbf{x}\|_2 \mid \mathbf{x} \in \Lambda \}$.

Minkowski's second theorem gives us a bound on the geometric mean of the successive minima. Given an m-dimensional lattice Λ and any $1 \leq k \leq m$ we have $\left(\prod_{i=1}^{k} \lambda_i(\Lambda) \right)^{1/k} \leq \sqrt{\gamma_m} \cdot \operatorname{vol}(\Lambda)^{1/m}$, where γ_m denotes Hermite's constant of dimension m. However, determining the exact value of γ_m is a long-standing open problem in the geometry of numbers, with the exact values being known for only $1 \leq m \leq 8$ and $m = 24$. Heuristically speaking, given a *random* lattice Λ of dimension m and a Euclidean ball $\bar{\mathcal{B}}_m(\mathbf{x}, r)$. We expect that the number of lattice points which lie in $\Lambda \cap \bar{\mathcal{B}}_m(\mathbf{x}, r)$ to be approximately equal to $\frac{\operatorname{vol}(\bar{\mathcal{B}}_m(\mathbf{x}, r))}{\operatorname{vol}(\Lambda)}$.

The lattices we consider here are not random, rather they are 'Ajtai' lattices, possessing reductions from worst-case Approx-SVP to average-case Hermite-SVP. For more details on the nature of random lattices, the reader is referred to [16]. However, it is generally assumed in the literature, as in this work, that the Gaussian heuristic holds reasonably well for Ajtai lattices. If this approximate equality was to hold for any such ball, then by considering the unit ball in $\bar{\mathcal{B}}_m(\mathbf{0}, 1) \subset \mathbb{R}^m$, we would have $|\Lambda \cap \bar{\mathcal{B}}_m(\mathbf{0}, 1)| \approx \frac{\pi^{m/2}}{\Gamma(1+m/2) \cdot \operatorname{vol}(\Lambda)}$, where Γ denotes the standard gamma function $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$, $z \in \mathbb{C}$.

Hence we would expect that

$$\lambda_1(\Lambda) \approx \left(\frac{\operatorname{vol}(\Lambda)}{\operatorname{vol}(\bar{\mathcal{B}}_m(\mathbf{0}, 1))} \right)^{1/m} = \frac{\operatorname{vol}(\Lambda)^{1/m} \cdot \Gamma(1+m/2)^{1/m}}{\sqrt{\pi}} \approx \operatorname{vol}(\Lambda)^{1/m} \cdot \sqrt{\frac{m}{2\pi e}}$$

For random lattices, it is known that, with overwhelming probability, the above holds (for all successive minima) [1]. This provides the motivation for the Hermite-SVP problem, which we define below. More generally, we list below the four main lattice problems of relevance to this work.

First, the *approximate Shortest Vector problem* (γ-SVP) is as follows: Given a lattice $\Lambda = \mathcal{L}(\mathbf{B})$, find a vector $\mathbf{v} \in \Lambda$ such that $0 < \|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\Lambda)$. In the same vain, the *approximate Hermite Shortest Vector problem* (γ-HSVP) is: Given a lattice $\Lambda = \mathcal{L}(\mathbf{B})$, find a vector $\mathbf{v} \in \Lambda$ such that $0 < \|\mathbf{v}\| \leq \gamma \cdot \det(\Lambda)^{\frac{1}{m}}$. Any algorithm which solves γ-SVP also solves Hermite-SVP with factor $\gamma \sqrt{\gamma_n}$. Note also that (γ-SVP) ($\gamma \geq 1$) is NP-Hard under randomized reduction for any $\gamma < 2^{(\log n)^{1/2-\epsilon}}$, where $\epsilon > 0$ is an arbitrarily small constant [19].

We also consider the *bounded distance decoding problem* (BDD_η): Given a lattice Λ and a vector $\mathbf{t}$ such that $\operatorname{dist}(\mathbf{t}, \Lambda) < \eta \cdot \lambda_1(\Lambda)$, find the lattice vector $\mathbf{y}$ which is closest to $\mathbf{t}$. We note that, when considering BDD_η from a complexity theory approach, arbitrary values for η can be considered while in practical settings, the problem is often defined with the restriction that $\eta \leq \frac{1}{2}$. The case of solving $\text{BDD}_{\eta > \frac{1}{2}}$ corresponds to list-decoding in coding parlance. BDD_η is known to be NP-hard for any constant factor $\eta > \frac{1}{\sqrt{2}}$[22]. Finally:

Finally, the GapSVP (promise) problem (GapSVP$_\gamma$) is: Given a lattice Λ, a radius $r > 0$ and approximation factor $\gamma > 1$, is $\lambda_1(\Lambda) \leq r$? If so return YES, else if $\lambda_1(\Lambda) > \gamma r$ return NO, and otherwise return YES or NO. Note that GapSVP$_\gamma$ is NP-Hard for any constant γ[19].

Lattice Reduction. The predominant approaches for solving the Learning with Errors (LWE) problem [27] rely on reducing a lattice basis (determined by a subset of the LWE samples) to obtain either a single short vector in the (scaled) dual lattice [26] or a 'good' (relatively orthogonal) basis of the primal lattice [21], as measured by the norms of the Gram-Schmidt vectors of such a basis. In the first case, since we do not know $\lambda_1(\Lambda)$ *a priori*, it is customary to measure the 'strength' of a basis reduction algorithm by the γ-HSVP factor it can attain. In the latter case, similar notions are used, with the added heuristic that the norms of the Gram-Schmidt vectors of a reduced-basis decrease geometrically.

We briefly recall some notions of lattice basis reduction (from a Hermite-SVP perspective). While finding the shortest vector in low-dimensional lattices is relatively easy, only approximation algorithms can be realistically run in higher dimensions. With respect to the Hermite-SVP problem, we aim to find a vector $\mathbf{v}$ in the lattice such that $\gamma = \|\mathbf{v}\|/\mathrm{vol}\,(\Lambda)^{\frac{1}{m}}$ is small. The famed LLL algorithm [20] discloses lattice vectors with Hermite factor $\leq (4/3)^{(m-1)/4}$ while the more powerful Block Korkine-Zolotarev (BKZ) algorithm, parameterised by a block-size β, discloses lattice vectors with Hermite factor $\leq \sqrt{\gamma_\beta}^{1+(m-1)/(\beta-1)}$ [13].

In practice, however, both LLL and BKZ perform much better than their worst-case provable bounds and both are commonly characterised by a 'root Hermite-factor' δ_0 such that $\delta_0^m \approx \|\mathbf{v}\|/\mathrm{vol}\,(\Lambda)^{\frac{1}{m}}$. Given a fixed algorithm, the value of δ_0 appears to rapidly converge to a fixed value as the lattice dimension increases. In [13], the authors report the results of extensive experiments, partly aimed at determining root Hermite factors for LLL and BKZ with selected block-sizes. The results of [13] indicate that, in practise, LLL achieves a $\delta_0 \approx 1.0219$ while BKZ-20 and BKZ-28 achieve $\delta_0 \approx 1.0128$ and $\delta_0 \approx 1.0109$, respectively, conjecturing that the current limits of 'practical' lattice reduction appear to be a root Hermite factor of ≈ 1.01, with $\delta_0 = 1.005$ being far beyond reach (in high dimension). However, estimation of the running time of BKZ in high dimension with a large block-size is difficult, with the asymptotic running time being doubly-exponential in the block-size. To attempt a conservative prediction of the running time of BKZ with large block-size, the authors of [21] assume that δ_0 is the dominant influence on the running-time of BKZ in high dimension and proposed a simple extrapolation of running times as a function of δ_0 leading to the model

$$\log_2 T_{\mathrm{sec}} = 1.8/\log_2 \delta_0 - 110. \tag{1}$$

We can translate this figure into bit operations by assuming $2.3 \cdot 10^9$ bit operations per second on a 2.3 GHz CPU.

However, the accuracy and hence utility of such models is debatable, with such models giving infeasibly low complexity estimates for the application of LLL.

2.2 Learning with Errors (LWE)

We briefly review the results on LWE required in our cryptanalysis. The central idea of our attack is to observe that the security of HLY scheme actually relies on weak instances of LWE. After providing the definition of LWE, we recall a modulus-switching result from [9] which we exploit to improve our basic attack. Finally, we briefly review some known techniques for solving LWE. In this work, we consider the short dual-lattice vector distinguishing attack [26] to distinguish LWE instances arising in our attack of HLY scheme. The LWE problem is as follows:

Definition 2 (LWE [27]). *Let n, q be a positive integers, χ be a probability distribution on $\mathbb{Z}_q$ and $\mathbf{s}$ be a secret vector in $\mathbb{Z}_q^n$. We denote by $L_{\mathbf{s},\chi}^{(n)}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}_q$ according to χ, and returning $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. We shall call* Decision-LWE *the problem of deciding whether pairs $(\mathbf{a}, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ are sampled according to $L_{\mathbf{s},\chi}^{(n)}$ or the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

The noise follows some distribution χ which is classically chosen to be a discrete Gaussian distribution over $\mathbb{Z}$ with mean 0, reduced modulo q. This distribution (over $\mathbb{Z}$) is obtained by rounding the (continuous) Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and standard deviation $\sigma = s/\sqrt{2\pi} = \alpha q/\sqrt{2\pi}$, i.e. we consider $\lceil \mathcal{N}(\mu, \sigma^2) \rfloor$. The modulus q is typically taken to be polynomial in n. It was shown [27,9] that if $\alpha q > 2\sqrt{n}$, then (worst-case) $\mathrm{GapSVP}_{\tilde{O}(n/\alpha)}$ reduces to (average-case) LWE.

Remark 1 (Modulus reduction). Modulus switching was introduced to improve the performance of homomorphic encryption schemes [10] and was recently used to reduce the hardness of LWE with polynomially sized moduli to GAPSVP [9]. It also possible to use such technique for attacking schemes. As soon as the secret $\mathbf{s}$ follows a distribution with small standard deviation σ_s, then we can perform modulus reduction. That is, given $p \ll q$ we can consider a new LWE sample $(\lfloor p/q \cdot \mathbf{a}_i \rfloor, \lfloor p/q \cdot c_i \rfloor)$ in place of the initial LWE $(\mathbf{a}_i, c_i)$ at the cost of a slight increase in the noise level. We do not consider this approach further in this work as it only provides a moderate improvement over the results presented in Section 6.

Solving LWE with Lattice Reduction. For solving LWE, several approaches exist in the literature. Asymptotically, combinatorial approaches are superior [2] while in practise lattice-based approaches are often more efficient. The most straight-forward approach [26] is to apply lattice basis reduction to the (scaled) dual lattice determined by the LWE samples. This allows to obtain a short vector in this lattice and leads to a distinguisher of valid LWE samples and uniformly random samples. Note that thanks to the classical decision to search equivalence for LWE [27] any distinguisher can be actually used to recover the secret key. This multiplies the cost of the distinguisher by a polynomial factor q (more precisely, by the size of the secret space).

Given a set of m LWE samples $(\mathbf{a}_i, c_i)$, we denote by $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ the matrix whose columns are the $\mathbf{a}_i^T$'s. We then consider the following q-ary lattice

$$\Lambda_q(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m \mid \exists \mathbf{s} \in \mathbb{Z}^n \text{ such that } \mathbf{sA} \equiv \mathbf{z} \bmod q\}$$

and a corresponding (scaled) dual lattice

$$\Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{y}^T \equiv \mathbf{0} \bmod q\}.$$

In [26], the authors briefly examine an approach for solving LWE by distinguishing between valid matrix-LWE samples of the form $(\mathbf{A}, \mathbf{c}) = (\mathbf{A}, \mathbf{sA} + \mathbf{e})$ and samples drawn from the uniform distribution over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$. Given a matrix of samples $\mathbf{A}$, one way of constructing such a distinguisher is to find a short vector $\mathbf{u}$ in the (scaled) dual lattice $\Lambda_q^{\perp}(\mathbf{A})$, the vector $\mathbf{u}$ is such that $\mathbf{A}\mathbf{u}^T = \mathbf{0} \mod q$. If $\mathbf{c}$ belongs to the uniform distribution over $\mathbb{Z}_q^m$, then $\langle \mathbf{u}, \mathbf{c} \rangle$ belongs to the uniform distribution on $\mathbb{Z}_q$. On the other hand, if $\mathbf{c} = \mathbf{sA} + \mathbf{e}$, then $\langle \mathbf{u}, \mathbf{c} \rangle = \langle \mathbf{u}, \mathbf{sA} + \mathbf{e} \rangle = \langle \mathbf{u}, \mathbf{e} \rangle$. Each sample of the form $\langle \mathbf{u}, \mathbf{e}_i \rangle$ are governed by another discrete, wrapped Gaussian distribution. Following the work of Micciancio and Regev [26], the authors of [21] investigates the algorithmic hardness of Decision-LWE by estimating the cost of the BKZ algorithm in finding a short enough vector, using the model mentioned above (Section 2.1).

In particular, given $m, n, q, \sigma = \alpha q$, we set $s = \sigma\sqrt{2\pi}$. Then, given a vector $\mathbf{v}$ in the dual lattice, a good approximation for the distinguishing advantage obtained through this approach is

$$\epsilon \approx \exp\left(-\pi \cdot (\|\mathbf{v}\| \cdot s/q)^2\right). \tag{2}$$

Thus, given a target distinguishing advantage ϵ, we can compute the required norm of a vector in the (scaled) dual lattice to be: $v = (q/s) \cdot \sqrt{-\log(\epsilon)/\pi}$. We also let

$$\lambda_1(\Lambda_q(\mathbf{A})) = \min\left\{q, q^{n/m} \cdot \sqrt{m/(2\pi \cdot e)}\right\}$$

be the length of the shortest vector according to the Gaussian heuristic. Once again, we note that while the q-ary lattices derived from LWE instances are not random in a strict sense and thus we cannot *a priori* expect the Gaussian heuristic to be verified, in practice the heuristic holds extremely well. Hence, as do other works, we assume this also in our case.

To estimate the root Hermite factor δ_0 we need to achieve, we rely on the heuristic – but experimentally sound – model in which we expect the norm of the shortest vectors found to be approximately $q^{n/m}\delta_0^m$. Then, the optimal sub-lattice dimension for the attack is $m_{\text{opt}} = \sqrt{n \log(q)/\log(\delta_0)}$. Assuming that we have enough LWE samples to construct a lattice of the optimal dimension, we then require the application of a basis-reduction algorithm with root-factor given by $\delta_0 = 2^{\frac{\log^2 v}{4n \log q}}$.

An alternative method for solving LWE (and for BDD in general) using lattice reduction is to employ Kannan's embedding method. Here, we take a lattice $\Lambda = \mathcal{L}(\mathbf{B}) \subset \mathbb{R}^m$ and a point $\mathbf{t} \in \mathbb{R}^m$ which is close to a lattice point $\mathbf{y}$ with $\|\mathbf{y} - \mathbf{t}\| < \lambda_1(\Lambda)/2$. We then construct

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{t} & \|\mathbf{y} - \mathbf{t}\| \end{pmatrix}.$$

It can be shown [23] that if $\sqrt{2} \cdot ||\mathbf{y} - \mathbf{t}|| < \lambda_1(\Lambda)$ then $[\mathbf{t} \mid ||\mathbf{y} - \mathbf{t}||]$ is a shortest (non-zero) vector in $\mathcal{L}(\mathbf{B}')$. This leads to an instance of unique-SVP - an instance of SVP in which we are given the additional guarantee that there is a certain 'gap' between $\lambda_1(\mathcal{L}(\mathbf{B}'))$ and $\lambda_2(\mathcal{L}(\mathbf{B}'))$. Note that, in practise, one would choose the embedding factor to be smaller than $||\mathbf{y} - \mathbf{t}||$ to (probabilistically) maximise this gap. In this work, we employ both large and small embedding factors, 'small' meaning an embedding factor of 1. In the latter case (see section 5.1) we make the simplifying assumption that the second minimum of the embedding lattice is approximately equal to the first minimum of the original lattice, to gain an estimation of the gap. However, compared to alternative approaches for solving LWE, the efficacy of the embedding approach is poorly understood at present with no good models (to the best of our knowledge) to predict when the approach will succeed. It is known, however, that the presence of a λ_2/λ_1 gap makes finding the shortest vector somewhat easier, with an exponential gap clearly allowing disclosure of a shortest non-zero vector by application of LLL. With smaller gaps, the success of the approach is known to be probabilistic [13].

3 A New Multivariate Quadratic Assumption and LWE with Small Secrets

in this section we describe the public-key encryption scheme proposed by Huang, Liu and Yang (HLY) [17] at PKC'12 as well as the new hard problem underlying their scheme. We will revisit the fact that the hardness of this new problem is related to the difficulty of solving a LWE-style problem for a very small secret. In [17] the authors introduced a variant of the classical Polynomial System Solving Problem (PoSSo).

Definition 3. *Let $f_0, \ldots, f_{m-1} \in \mathbb{Z}_q[x_0, \ldots, x_{n-1}]$ be non-linear polynomials.* PoSSo *is the problem of finding – if any – $\mathbf{s} \in \overline{\mathbb{Z}_q}^n$ such that $f_0(\mathbf{s}) = 0, \ldots, f_{m-1}(\mathbf{s}) = 0$.*

It is well known [14] that this problem is NP-hard. Note that PoSSo remains NP-hard [14] even if we suppose that the input polynomials are quadratics. In this case, PoSSo is also called MQ. Huang, Liu and Yang proposed a variant of MQ where the monomials of highest degree (i.e. 2) in the system have their coefficients chosen according to a discrete Gaussian distribution of standard deviation $\zeta \in O(1)$ and centered on zero. Following [17], we denote this distribution by Φ_ζ.[1] The remaining coefficients (linear, and constant parts) are chosen uniformly at random. We denote this distribution on $\mathbb{Z}_q[x_1, \ldots, x_n]$ by $\mathbb{Z}_q^{\Phi_\zeta}[\mathbf{x}]$. The problem introduced by Huang, Liu and Yang will be the main concern of this work:

Definition 4 (MQ$(n, m, \Phi_\zeta, H_\beta)$). *Let n be positive integer, $m \in O(n)$, q be a polynomially bounded prime, a constant $\beta, 0 < \beta < q/2$ and $\mathbf{e}$ be a secret vector in $H_\beta := [-\beta, \ldots, \beta]^n \subseteq \mathbb{Z}_q^n$. We denote by $\mathrm{MQ}_{\mathbf{s},\Phi}^{(n)}$ the probability distribution on $\mathbb{Z}_q[x_1, \ldots, x_n]^m \times \mathbb{Z}_q^m$ obtained by sampling $\mathbf{p} = (p_1, \ldots, p_m)$ from $\mathbb{Z}_q^{\Phi_\zeta}[\mathbf{x}]^m$, and returning $(\mathbf{p}, \mathbf{c}) = (\mathbf{p}, \mathbf{p}(\mathbf{s})) \in \mathbb{Z}_q[x_1, \ldots, x_n]^m \times \mathbb{Z}_q^m$.*

[1] The parameter ζ is called α in [17] but this notation clashes with the standard notation for LWE.

$\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ *is the problem of finding* $\mathbf{s} \in H_\beta^n$ *given a pair* $(\mathbf{p}, \mathbf{p}(\mathbf{s})) \leftarrow_\$ \mathrm{MQ}^{(n)}_{\mathbf{s},\Phi}$.

The decision problem *associated to* $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ *is the task of distinguishing* $\mathrm{MQ}^{(n)}_{\mathbf{s},\Phi}$ *from the uniform distribution on* $\mathbb{Z}_q[x_1, \ldots, x_n]^m \times \mathbb{Z}_q^m$.

As mentioned in [17], $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ is rather close to LWE:

Fact 1. *Each* $(\mathbf{p}, \mathbf{p}(\mathbf{s})) \leftarrow_\$ \mathrm{MQ}^{(n)}_{\mathbf{s},\Phi}$ *can be mapped to a* LWE *instance. To do so, we just consider the matrix* $A_\mathbf{p} \in \mathbb{Z}_q^{n \times m}$ *corresponding to the linear part of* $\mathbf{p}$. *We then remark that each component of* $\mathbf{p}(\mathbf{s}) - \mathbf{s} \cdot A_\mathbf{p} - \mathbf{p}(\mathbf{0})$ *is the sum of* $\frac{n(n+1)}{2}$ *discrete Gaussians each having variance* $\left(\frac{(2\beta+1)^2-1}{12}\right) \cdot \zeta^2$. *From now, we assume that this sum is a discrete Gaussian of variance* $\gamma^2 = \frac{n(n+1)}{2} \cdot \left(\frac{(2\beta+1)^2-1}{12}\right) \cdot \zeta^2$ *(centered at zero).*

It is proven in [17] that $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ has decision to search equivalence. Such equivalence makes the problem appealing to design an encryption scheme. The public-key of the scheme proposed in [17] is a pair of the form $(\mathbf{p}, \mathbf{p}(\mathbf{s})) = (\mathbf{p}, \mathbf{c}) \in \mathbb{Z}_q^{\Phi_\zeta}[\mathbf{x}]^m \times \mathbb{Z}_q^m$. To encrypt a bit b, we choose $\mathbf{r} \in H_{n^\lambda} := [-n^\lambda, \ldots, n^\lambda]^m \subset \mathbb{Z}_q^m$ with λ being a new parameter. We then compute : $c = (A_\mathbf{p} \cdot \mathbf{r}^T, \langle\, \mathbf{r}, \mathbf{c} - p(\mathbf{0})\, \rangle + b \cdot \lfloor q/2 \rceil)$. Thus, each encryption of zero produces a LWE sample whose error has variance: $m \cdot n^{2\lambda} \cdot \gamma^2$. As a consequence, we expect the noise to have size $\sqrt{\frac{2}{\pi}} \cdot \sqrt{m} \cdot n^\lambda \cdot \gamma$. Note that [17] also proposed a Key Encapsulation Mechanism (KEM) scheme, based on the same new hard problem, but which we do not discuss here.

Regarding the security, [17] showed that breaking the semantic security of the encryption scheme is equivalent to solving $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$. More precisely:

Theorem 2 ([17]). *Let* $\mathcal{A}$ *be an adversary breaking the semantic security of the scheme working in time* T *with advantage* ϵ. *Then, there exists a probabilistic algorithm* $\mathcal{B}$ *solving* $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ *in time at most* $T \cdot \frac{128}{\epsilon^2} \cdot (2\beta + 1) \cdot (n^2 \log q)^2$ *with success probability at least* $\epsilon/(4\,q)$.

A similar result holds for the KEM scheme, i.e. breaking the semantic security of the KEM scheme allows to solve $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$.

Such reduction is then used to establish concrete parameters for the proposed encryption scheme. The basic hypothesis for setting the parameter is to assume that solving $\mathbf{p} - \mathbf{p}(\mathbf{s}) = \mathbf{0}$, for $(\mathbf{p}, \mathbf{p}(\mathbf{s})) \leftarrow_\$ \mathrm{MQ}^{(n)}_{\mathbf{s},\Phi}$, is essentially not easier than solving a random system of equations [17].

Assumption 1 (HLY Hardness Hypothesis). *Solving* $\mathrm{MQ}(n, m, \Phi_\zeta, H_\beta)$ *is as hard as solving a random system of* m *quadratic equations in* n *variables modulo* q *with a pre-assigned solution in* H_β^n.

Remark 2. The fact that the secret is in H_β^n implies that one can always add n equations of degree $2\beta + 1$ of the form $\prod_{j \in H_\beta}(x_i - j)$. Clearly, the evaluation of such equations on any $\mathbf{s} \in H_\beta^n$ will be zero.

Arguably, this connection between the semantic security and hardness of PoSSo is the main difference between the HLY scheme and the classical encryption scheme based on LWE. Indeed, the HLY scheme is very similar to a textbook LWE encryption scheme equipped with a Gaussian of standard deviation $\sqrt{m} \cdot n^{\lambda} \cdot \gamma$ with a very small secret. A noteworthy difference lies in the fact that we also consider small (i.e. of norm bounded by n^{λ}) linear combinations of public samples. In the classical LWE encryption scheme due to Regev [27], we consider only linear combinations with coefficients in $\{-1, 0, 1\}$ of the public samples.

Assumption 1 allows to estimate the cost of the best attack against $\mathrm{MQ}(n, m, \Phi_{\zeta}, H_{\beta})$. A well-established approach to solve PoSSo is to compute a Gröbner basis [11]. The cost of solving a (zero-dimensional, i.e. finite number of solutions) system of m non-linear equations in n variables with the F_5 algorithm [5,12] is $O\left(\binom{n+D_{reg}}{D_{reg}}^{\omega}\right)$, where D_{reg} is the maximum degree reached during the Gröbner basis computation, and ω is the matrix multiplication exponent (or the linear-algebra constant) as defined in [31, Chapter 12]. We recall that $\omega \in [2, 2.3727]$).

In general, it is a hard problem to predict *a priori* the degree of regularity of a given system of equations. However, Assumption 1 implies that the system of non-linear equations involved is no easier to solve than semi-regular equations [5,6,7]. Precisely, D_{reg} is bounded from below by the index of the first non-positive coefficient of: $\sum_{k\geq 0} c_k z^k = (1-z^2)^m (1-z^{(2\beta+1)})^n / (1-z)^n$. This is the degree of regularity of a system of m equations of degree 2 plus n equations of degree $2\beta + 1$ in n variables.[2] From now on, we will denote by $T_{\mathrm{ref}}(m, n, q)$ the cost of solving such system with F_5 algorithm, and by ϵ_{ref} the success probability. Usually, a Gröbner basis computation always succeeds, but one can relax this condition by randomly fixing variables. Precisely, a success probability ϵ_{ref} allows to fix $r_{\mathrm{ref}} = \left\lceil \log_{2\beta+1}(1/\epsilon_{\mathrm{ref}}) \right\rceil$ variables for systems sampled according to $\mathrm{MQ}^{(n)}_{\mathbf{s},\Phi}$.

It is worth mentioning and commending that [17] propose concrete parameters for their scheme (reproduced in Table 1). The parameters are chosen as follows. Assume there exist an adversary $\mathcal{A}$ breaking the semantic security of the HLY encryption in time $T_{\mathrm{dist}} = 2^{\ell}$ with advantage $\epsilon_{\mathrm{dist}} = 2^{-s}$. According to Theorem 2, we can construct an algorithm $\mathcal{B}$ solving $\mathrm{MQ}(n, m, \Phi_{\zeta}, H_{\beta})$ in time $T_{\mathrm{search}}(T_{\mathrm{dist}}, \epsilon_{\mathrm{dist}}, n, q)$ with success probability $\epsilon_{\mathrm{search}}(\epsilon_{\mathrm{dist}}, q)$. From Assumption 1, the best algorithm for solving $\mathrm{MQ}(n, m, \Phi_{\zeta}, H_{\beta})$ works in time $T_{\mathrm{ref}}(m, n - r_{\mathrm{ref}}, q)$ with a success probability ϵ_{ref}. The parameters m, n, q are chosen such that

$$T_{\mathrm{search}}(T_{\mathrm{dist}}, \epsilon_{\mathrm{dist}}, n, q) < T_{\mathrm{ref}}(m, n, q) \text{ and } \epsilon_{\mathrm{search}}(\epsilon_{\mathrm{dist}}, q) < \epsilon_{\mathrm{ref}}.$$

Under the HLY hypothesis (Assumption 1), this means that no adversary can break the semantic security of the scheme in time less than 2^{ℓ} with success probability better than 2^{-s}.

4 Analysis of the Parameters

In this part, we show that security and efficiency are essentially incompatible for HLY. To do so, we derive a set of conditions on the parameters that would thwart the simplest

[2] Note that this quantity can be explicitly computed for any value of n, m and β.

known attack against LWE-style systems such as those discussed above. That is, we want to find parameters such that both computing a Gröbner basis and lattice attacks (in particular the non-optimal Micciancio-Regev approach) are exponentially hard in the security parameter τ. Below, we recall the constraints on the parameters from [17]:

1. $k \cdot \zeta \cdot n^{2+\lambda} \cdot m \cdot \beta^2 \leq q/4$ (to allow for correct decryption)
2. $m \cdot \log(2n^\lambda + 1) \geq (n+1)\log q + 2k$ (to make sure the subset sum problem is hard)
3. n, m, q, ζ, β (to satisfy the condition in the MQ assumption such that $MQ(n, m, q, \Psi_\zeta, H_\beta)$ is hard to solve).

For the number of equations, we may restrict $m = c \cdot n$ where c is a constant (we remark that the challenges proposed in [17] have $c = 2$). In this case, we can assume that MQ is hard (that is, the cost of computing a Gröbner basis is exponential in the number of variables [5,6,7]. From Condition 2, we then get $m \cdot \log(2n^\lambda + 1) \geq (n+1)\log q + 2k \geq n \log q$ and

$$c \cdot \log(2n^\lambda + 1) \geq \log q$$

by replacing m with cn. This means that $2n^\lambda$ should be roughly (or at least) $q^{1/c}$. Hence, the first condition yields:

$$\begin{aligned} k \cdot \zeta \cdot n^{2+\lambda} \cdot m \cdot \beta^2 &\leq q/4 \\ k \cdot \zeta \cdot n^{2+\lambda} \cdot c \cdot n \cdot \beta^2 &\leq 2^{(c-2)} n^{c\lambda} \\ \zeta \cdot n^2 \cdot \beta^2 &\leq (ck)^{-1} 2^{(c-2)} n^{(c-1)\lambda - 1} \end{aligned}$$

as a bound on the noise in each of the m samples. As explained in Section 2.2, (heuristically) lattice reduction will produce vectors of length

$$v = q^{n/m} \cdot \delta_0^m = q^{1/c} \cdot \delta_0^{cn} \leq 2n^\lambda \cdot \delta_0^{cn}.$$

By combining this with the above, we get a distinguishing advantage (as defined in (2)) of

$$\begin{aligned} \exp\left(-\tfrac{\pi s^2 v^2}{q^2}\right) &= \exp\left(-\frac{\pi s^2 4 n^{2\lambda} \delta_0^{2cn}}{q^2}\right) = \exp\left(-\frac{2\pi^2 \sigma^2 4 n^{2\lambda} \delta_0^{2cn}}{q^2}\right) \\ &= \exp\left(-\frac{2\pi^2 \sigma^2 4 n^{2\lambda} \delta_0^{2cn}}{4^c n^{2c\lambda}}\right) = \exp\left(-(4^{(1-c+\frac{1}{2})} \pi^2 \sigma^2 n^{2\lambda(1-c)} \delta_0^{2cn})\right). \end{aligned}$$

Now, we can write:

$$\begin{aligned} \sigma^2 &= \zeta^2 \cdot \tfrac{n(n+1)}{2} \cdot \left(\tfrac{(2\beta+1)^2-1}{12}\right) = \left(\zeta^{\frac{1}{2}} \cdot n \cdot \beta\right) \cdot \left(\zeta^{\frac{3}{2}} \cdot \tfrac{n+1}{2} \cdot \tfrac{\beta+1}{3}\right) \\ &= \left(\zeta^{\frac{1}{2}} \cdot n \cdot \beta\right) \cdot \left(\tfrac{1}{6} \cdot \zeta^{\frac{3}{2}} \cdot (n+1) \cdot (\beta+1)\right) \approx \tfrac{1}{6} \cdot \zeta \cdot \left(\zeta \cdot n^2 \cdot \beta^2\right). \end{aligned}$$

This gives:

$$\zeta \lessapprox \frac{(ck)^{-1} 2^{(c-2)} n^{(c-1)\lambda-1}}{n^2 \cdot \beta^2}.$$

Thus:

$$\sigma^2 \lessapprox \frac{1}{6} \cdot n^2 \cdot \beta^2 \cdot \left(\frac{(ck)^{-2} 2^{2(c-2)} n^{2(c-1)\lambda-2}}{n^4 \cdot \beta^4}\right) = \frac{(ck)^{-2} 2^{2(c-2)} n^{2(c-1)\lambda-4}}{6\beta^2}.$$

Hence we can lower-bound the distinguishing advantage by:

$$\exp\left(-(4^{\frac{3}{2}-c}\pi^2\sigma^2 n^{2\lambda(1-c)}\delta_0^{2cn})\right) = \exp\left(-\tfrac{\pi^2}{12\beta^2}\cdot(ck)^{-2}n^{-4}\cdot\delta_0^{2cn}\right)$$

We now introduce a parameter τ, representing the bit-complexity of solving such instances using the model of Lindner and Peikert. We then replace δ_0 by $2^{(1.8/(\tau+78.9))}$ (employing (1) to deliver an estimate of the number of bit operations required to obtain such a root Hermite factor) and require that the advantage is constant in terms of τ. In other words

$$\exp\left(-\tfrac{\pi^2}{12\beta^2}\cdot(ck)^{-2}\cdot n^{-4}\cdot 2^{3.6cn/(\tau+78.9)}\right) = d. \tag{3}$$

For example, for $\tau = 80$, with $\beta = 2$, $c = 2$, $k = 12$ and $d = 0.5$, setting $n = 1140$ satisfies this condition. For $\tau = 128$, the same parameters require $n = 1530$. We note, however, that setting $n = 1140$ already results in a public key of considerable size (optimistically setting $\zeta = 10$):

$$\tfrac{m\cdot\binom{n+2}{2}\cdot\log_2(2\pi\zeta)}{8\cdot 1024^3} \approx 1.03\text{ GB}, \tag{4}$$

while setting $n = 1530$ results in a public-key of size 2.49 GB.

Furthermore, we stress that these parameters do not take potential other attack vectors into account and should be viewed as a somewhat loose *upper-bound* on the complexity of solving such instances. In particular, this discussion does not reflect the possibility of exploiting the small secret for example through modulus reduction (Remark 1) and the approach discussed next.

5 Improved Embedding Attack

We present an improved version of the embedding attack described in Section 2.2. To do so, we exploit the fact that the secret key **s** is *extremely short*. Recall that the coefficients of the secret lie in a small subset $H = [-\beta, \beta] \subset \mathbb{Z}_q$. Typically, Huang, Liu and Yang suggested to take $\beta = 2$ (Table 1).

Let $(\mathbf{p}, \mathbf{p}(\mathbf{s})) = (\mathbf{p}, \mathbf{c}) \leftarrow_{\$} \mathrm{MQ}_{\mathbf{s},\Phi}^{(n)}$ be a public-key of HLY scheme. Let $A_{\mathbf{p}} \in \mathbb{Z}_q^{n\times m}$ be the matrix corresponding to the linear part of **p**. According to Fact 1, we can write:

$$\mathbf{c} \equiv \mathbf{s}\cdot A_{\mathbf{p}} + \mathbf{e} + \mathbf{p}(\mathbf{0}) \bmod q,$$

where $\mathbf{e} \equiv \mathbf{p}(\mathbf{s}) - \mathbf{s}\cdot A_{\mathbf{p}} - \mathbf{p}(\mathbf{0}) \bmod q$. Notice that each coefficient of **e** is the sum of $n(n+1)/2$ discrete Gaussians. From now on, we let $\mathbf{y} \equiv \mathbf{c} - \mathbf{p}(\mathbf{0}) \equiv \mathbf{s}\cdot A_{\mathbf{p}} + \mathbf{e} \bmod q$ to ignore the constant part.

The basic idea is to consider the lattice defined by the following basis **B**:

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{0} & \mathbf{0} \\ A_{\mathbf{p}} & \mathbf{I}_n & \mathbf{0} \\ -\mathbf{y} & \mathbf{0} & 1 \end{pmatrix}.$$

Since $\mathbf{y} \equiv \mathbf{s}A_{\mathbf{p}} + \mathbf{e} \pmod{q}$, there exists $\mathbf{k} \in \mathbb{Z}^m$ satisfying $\mathbf{y} = \mathbf{s} \cdot A_{\mathbf{p}} + \mathbf{e} + q\mathbf{k} \in \mathbb{Z}^m$. Notice that the lattice $\mathcal{L}(\mathbf{B})$ contains a short vector $\mathbf{w} = [-\mathbf{e} \mid \mathbf{s} \mid 1] \in \mathbb{Z}^{m+n+1}$, since $[\mathbf{k} \mid \mathbf{s} \mid 1] \cdot \mathbf{B} = [q\mathbf{k} + \mathbf{s}A_{\mathbf{p}} - \mathbf{y} \mid \mathbf{s} \mid 1] = \mathbf{w}$. Applying the reduction algorithm to the lattice $\mathcal{L}(\mathbf{B})$ is less efficient than the basic embedding attack. The dimension $m + n + 1$ is larger than $m + 1$ and the short vector $\mathbf{w} = [-\mathbf{e} \mid \mathbf{s} \mid 1] \in \mathbb{Z}^{m+n+1}$ contains $\mathbf{e}$ entirely.

However, we can consider *a truncated lattice* defined by an $(m' + n + 1)$-dimension right-bottom submatrix $\mathbf{B}'$ of $\mathbf{B}$. By this truncation, we have the following relations:

$$[\mathbf{k}' \mid \mathbf{s} \mid 1] \cdot \begin{pmatrix} q\mathbf{I}_{m'} & \mathbf{0} & \mathbf{0} \\ A'_{\mathbf{p}} & \mathbf{I}_n & \mathbf{0} \\ -\mathbf{y}' & \mathbf{0} & 1 \end{pmatrix} = [-\mathbf{e}' \mid \mathbf{s} \mid 1] \in \mathbb{Z}^{m'+n+1}.$$

We note that $\mathbf{w}' = [-\mathbf{e}' \mid \mathbf{s} \mid 1]$ should be shorter than the previous $\mathbf{w}$. Hence, we could expect a ‘less powerful’ basis reduction algorithm to be required for recovery of $\pm\mathbf{w}'$ as compared to one required for recovery of the previous $\mathbf{w}$. We finally note that, if $m' < m - n$, then the dimension is smaller than that of the lattice in the direct approach.

5.1 Estimation of the Expected Gap

For an N dimensional lattice Λ, we define $\sigma(\Lambda) \approx \sqrt{N/2\pi e} \cdot \mathrm{vol}(\Lambda)^{1/N}$ to be the expected first minimum $\lambda_1(\Lambda)$ according to the Gaussian heuristic. We have

$$\sigma(\mathcal{L}(\mathbf{B}')) \approx \sqrt{(n + m' + 1)/2\pi e} \cdot q^{m'/(n+m'+1)}.$$

Next, we estimate $\|\mathbf{w}'\| = \sqrt{\|\mathbf{s}\|^2 + \|\mathbf{e}'\|^2 + 1}$. Since $\mathbf{s}$ is chosen from $\{-\beta, \ldots, \beta\}$ uniformly at random, the expected value of $\|\mathbf{s}\|^2$ is $n \cdot \frac{1}{2\beta+1} \sum_{i=-\beta}^{\beta} i^2 = n\beta(\beta + 1)/3$. As mentioned above, each coefficient of $\mathbf{e}'$ follows a discrete Gaussian of standard deviation γ. Hence, $\mathrm{E}[\|\mathbf{e}'\|]$ can be estimated as $\sqrt{m'} \cdot \gamma$. Summarizing the above, we obtain (using $\beta = 2$):

$$\mathrm{E}[\|\mathbf{w}'\|] \approx \sqrt{m'\gamma^2 + 2n + 1} \approx \sqrt{m' \cdot \zeta^2 \cdot n \cdot (n + 1) + 2n}.$$

Hence, the expected gap is expected to be

$$\frac{\sigma(\mathcal{L}(\mathbf{B}'))}{\|\mathbf{w}'\|} \approx \sqrt{\frac{n+m'+1}{2\pi e n(m'\zeta^2(n+1)+2)}} \cdot q^{m'/(n+m'+1)}. \tag{5}$$

We finally note that, when comparing the efficacy of embedding attacks, the expected gaps should be compared with those of lattices of *the same dimension*. If the dimensions differ, we derive less information regarding the success of the lattice-reduction algorithm in finding the shortest vector.

6 Practical Attacks against HLY Challenges

From the discussion in Section 4, we expect that all parameters suggested in [17] should be weak against a lattice-reduction attack. To mount such attacks practically we make

Table 1. Suggested parameters in [17]

Case	n	m	ζ	β	q	Hardness (T, μ)
1	200	400	10	2	$180313175469726327885l9 \approx 2^{73.93}$	$(2^{156}, 2^{-100})$
2	256	512	10	2	$52324402795762678724873 \approx 2^{75.47}$	$(2^{205}, 2^{-104})$

use of the fact that we can view the hard problem from [17] as an LWE instance and then solve these instances using lattice reduction. In particular, we consider all the parameter sets proposed in [17] (Table 1).

The column "Hardness" (T, μ) is a strict lower bound [17] on the complexity of solving MQ$(n, m, \Phi_\zeta, H_\beta)$ under Assumption 1. The parameters of Case (1) are chosen such that no adversary running in time less than 2^{82} can break the semantic security of the HLY bit-encryption scheme with advantage better than 2^{-11}. For the KEM, Case (1) provides a security of $(2^{85}, 2^{-10})$ (which denotes (time, advantage). Case (2) was expected to provide a security level of $(2^{130}, 2^{-11})$ for the bit encryption scheme (and a security level of $(2^{130}, 2^{-10})$ for the KEM scheme).

Case (1)

Distinguishing. We have $m = 400$ equations in $n = 200$ unknowns. Coefficients for quadratic terms are chosen from a discrete Gaussian with standard deviation $\zeta = 10$ and the secret is in $[-\beta, \ldots, \beta]$ for $\beta = 2$. If we ignore all quadratic terms and only consider the linear part, we have an LWE-style instance with $m = 400, n = 200, q \approx 2^{73.93}$ and standard deviation $\gamma = \sqrt{\frac{200 \cdot 201}{2} \cdot 10^2 \cdot \left(\frac{5^2-1}{12}\right)^2} \approx 2^{11.47}$. In this instance, the optimal sub-lattice dimension for applying LLL is $\sqrt{n \log(q) / \log(1.0219)} \approx 688$. However, applying LLL in dimension 400 is expected to return a vector of norm $v = q^{n/m} \cdot \delta_0^m \approx 2^{49.47}$ which is more than sufficient to distinguish between such LWE samples and random with advantage $\epsilon = \exp\left(-\frac{\pi s^2 v^2}{q^2}\right) \approx 0.9999$. We ran the LLL algorithm as implemented in fpLLL [29] on lattice instances as in Case (1), i.e., with $m = 400, n = 200, q = 180313175469726327885l9$. More precisely, we ran LLL (using Sage's default parameters [30]) on the 400×400 dual lattice. The shortest vector recovered by LLL had norm $2^{49.76}$ while we predicted a norm of $2^{49.47}$. The entire computation took 26 hours on a single core.

Modulus Reduction. A slightly more efficient variant is to perform modulus reduction before performing LLL in order to keep coefficients small. We may apply modulus reduction technique (Remark 1) with the above parameters and pick $p \approx 2^{65.00}$ and $\gamma \approx 2^{3.59}$. Applying LLL in dimension 400 is expected to return a vector of norm $v = 2^{45.00}$ which translates into a distinguishing advantage of $\epsilon \approx 1$.

Embedding. We may also consider the embedding attack as described in Section 2.2. We apply LLL to the 401×401 extended primal lattice and using a (conservative) embedding factor $\lceil \sqrt{m} \cdot \sigma \rceil$. The λ_2/λ_1 gap in this case is approximately

$$\frac{\mathrm{vol}(\mathcal{L}(\mathbf{B}))^{1/m} \cdot \Gamma(1+m/2)^{1/m}}{\sqrt{2\pi m}\sigma} \approx \frac{q^{\frac{m-n}{m}} \sqrt{\frac{m}{2\pi e}}}{\sqrt{2m}\sigma} \approx 2^{22.94}.$$

The attack recovered the 'noise' from the public key, allowing the private key (or an equivalent) to be recovered by simple linear algebra. We note that this attack obviates the need for a separate search-to-distinguishing phase, as required in the dual-lattice method, the attack taking again ~26 hours using a single core.

Improved Embedding. We set $m' = 66 \approx 200/3$. Our attack can recover the secret key $\mathbf{s}$ from every vector $\mathbf{y} = \mathbf{c} - \mathbf{p}(\mathbf{0})$. The running times vary from 268.69 to 295.34 seconds and the average (on 10 instances) of them is 278.16 seconds. We notice that the expected gap (5) is $\approx 2^{6.267}$. This attack was mounted on a Core i7 PC using the NTL library [28] with GMP [15]. In each case, we ran the BKZ algorithm (G_BKZ_FP with $\delta = 0.99$, block size = 30, and prune = 10) on 267-dimensional lattices constructed from the public-keys. We computed m' by incrementing m' from 1 until we success to recover in a test case.

Case (2)

Distinguishing. We have $m = 512$ equations in $n = 256$ unknowns modulo $q \approx 2^{75.47}$. Coefficients for quadratic terms are chosen from a discrete Gaussian with standard deviation $\zeta = 10$ and the secret is in $[-2, \ldots, 2]$ for $\beta = 2$. This gives a standard deviation $\gamma = \sqrt{\frac{256 \cdot 257}{2} \cdot 10^2 \cdot \left(\frac{5^2-1}{12}\right)^2} \approx 2^{11.82}$. Applying LLL in dimension 512 is expected to return a vector of norm $v = q^{n/m} \cdot \delta_0^m \approx 2^{53.74}$ which is more than sufficient to distinguish between such LWE samples and random with advantage $\epsilon = \exp\left(-\frac{\pi s^2 v^2}{q^2}\right) \approx 1$.

Modulus Reduction. Using modulus reduction, we pick $p \approx 2^{66.36}$ and $\gamma \approx 2^{3.76}$. Applying LLL in dimension 512 is expected to return a vector of norm $v = 2^{16.00}$ which translates into a distinguishing advantage of $\epsilon \approx 1$.

Improved Embedding. We set $m' = 90$ as a slightly larger integer than a third of n. Our attack successfully recovers the secret keys from all ten public-keys. The running times vary from 898.14 to 1119.53 seconds and the average of them is 964.83 seconds (≈ 16 minutes). We note that the expected gap is $\approx 2^{7.176}$.

Beyond the Challenges. To examine how the improved embedding attack scales, we consider larger parameters than those provided by the two challenges of Huang, Liu and Yang. In order to extend these challenges, we fix $\zeta = 10, \beta = 2, m = 2n, k = 12$, and $\lambda = 5$ and calculate q. From the correctness condition in [17] (see also Section 4), we should set $q \geq \mathrm{NextPrime}(4k\zeta\beta^2 mn^{2+\lambda}) = \mathrm{NextPrime}(3840n^8)$. From the provable security side, in order to employ the leftover hash lemma, Huang et al. [17] require q to satisfy $m \cdot \log(2n^\lambda + 1) \geq (n+1)\log q + 2k$. We here take q as small as possible, that is, we take q = NextPrime($3840n^8$), which always satisfies the correctness constraint and the security constraint.

Employing a single core of an i7 machine (3.4GHz), we ran the LLL algorithm on lattices constructed from the public keys with the parameter n increasing from 100 with intervals of 25. We computed m' on each n by incrementing m' from 30 at an interval of 10 until we were able to successfully recover a shortest vector in such a test case. The implementation of LLL in the NTL library consists of a number of variants of LLL capable of handling differing precision levels. Additionally, to enhance numerical stability, Givens orthogonalization can be used in place of Gram-Schmidt orthogonalization, the use of Given orthogonalization being denoted by a G_ prefix. The variants which concern us are: G_LLL_FP - LLL with Givens orthogonalization + double precision; G_LLL_QP - LLL with Givens orthogonalization + quadratic precision and G_LLL_RR - LLL with Givens orthogonalization and arbitrary precision - we used a precision of 150 bits.

Table 2 summarises the results of experiments using G_LLL_FP, G_LLL_QP, and G_LLL_RR (with precision 150), respectively. Due to precision limitations, G_LLL_FP fails at $n = 300$ while G_LLL_QP stops at $n = 450$. We also ran G_LLL_RR with default precision 150. Due to time constraints, we only ran this algorithm for parameters up to and including $n = 325$.

We can approximate the charts of the logarithm of T_{FP}, T_{QP}, and T_{RR}, which are the running times (in seconds) for these algorithms; $\log_2(T_{\mathrm{FP}}) = 6.9675\log(n) - 27.238$, $\log_2(T_{\mathrm{QP}}) = 7.3037\log(n) - 27.208$, and $\log_2(T_{\mathrm{RR}}) = 6.4345\log(n) - 18.502$, By using $T_{\mathrm{cycle}} = T \cdot 3.4 \cdot 10^9$, we obtain bit-operation complexity estimation formulae $\log_2(T_{\mathrm{FP,cycle}}) = 6.9675\log(n) + 4.425$, $\log_2(T_{\mathrm{QP,cycle}}) = 7.3037\log(n) + 4.459$, and $\log_2(T_{\mathrm{RR,cycle}}) = 6.4345\log(n) + 13.161$.

Our experiment shows that G_LLL_QP can find the secret keys up to $n = 425$ in approximately 2 days. Although we could run G_LLL_RR on $n \geq 425$ to avoid the

Table 2. Experimental results using G_LLL_FP, G_LLL_QP, and G_LLL_RR with precision 150

		G_LLL_FP		G_LLL_QP		G_LLL_RR *	
n	q	m'	T_{FP} (sec.)	m'	T_{QP} (sec.)	m'	T_{RR} (sec.)
100	$\approx 2^{65.058}$	30	$31 \approx 2^{4.954}$	30	$115 \approx 2^{6.845}$	30	$2412 \approx 2^{11.237}$
125	$\approx 2^{67.533}$	40	$82 \approx 2^{6.358}$	40	$294 \approx 2^{8.200}$	40	$5960 \approx 2^{12.541}$
150	$\approx 2^{69.737}$	50	$177 \approx 2^{7.468}$	50	$626 \approx 2^{9.290}$	50	$11974 \approx 2^{13.548}$
175	$\approx 2^{71.517}$	70	$466 \approx 2^{8.864}$	70	$1411 \approx 2^{10.463}$	70	$27190 \approx 2^{14.731}$
200	$\approx 2^{73.057}$	80	$810 \approx 2^{9.662}$	80	$2441 \approx 2^{11.253}$	80	$50976 \approx 2^{15.638}$
225	$\approx 2^{74.417}$	100	$1456 \approx 2^{10.508}$	100	$4513 \approx 2^{12.140}$	100	$86427 \approx 2^{16.399}$
250	$\approx 2^{75.633}$	120	$2487 \approx 2^{11.280}$	120	$7587 \approx 2^{12.889}$	120	$135423 \approx 2^{17.047}$
275	$\approx 2^{76.733}$	140	$3784 \approx 2^{11.886}$	130	$11720 \approx 2^{13.517}$	140	$203450 \approx 2^{17.634}$
300	$\approx 2^{77.737}$	—	fail	160	$19285 \approx 2^{14.235}$	160	$292092 \approx 2^{18.156}$
325	$\approx 2^{78.661}$			200	$32016 \approx 2^{14.967}$	190	$439574 \approx 2^{18.746}$
350	$\approx 2^{79.516}$			230	$44158 \approx 2^{15.430}$		
375	$\approx 2^{80.313}$			280	$82369 \approx 2^{16.330}$		
400	$\approx 2^{81.057}$			330	$119767 \approx 2^{16.870}$		
425	$\approx 2^{81.757}$			400	$175007 \approx 2^{17.417}$		
450	$\approx 2^{82.417}$			—	fail		

precision problems with G_LLL_QP beyond this point, we only ran it up to $n = 325$ due to time constraints. If we ran G_LLL_RR on $n = 450$, our model indicates that around $2^{20.808}$ seconds ≈ 21 days would be required. However, we expect that LLL will be insufficient to recover the private key (with probability ∼ 1) in this manner for values of n greater than ∼ 500. For such values of n, lattice reduction algorithms achieving lower root Hermite factors will be required [3]. We expect this to be the case due to observations made in [13] and [4] that we can expect to solve unique-SVP instances with a certain probability p whenever we have

$$\lambda_2(\mathcal{L})/\lambda_1(\mathcal{L}) \geq \tau_p \cdot \delta_0^{\dim(\mathcal{L})}$$

for some $\tau_p \in (0, 1]$. The values of τ_p derived experimentally in [13] ranged from 0.18 to 0.45, though with unspecified p. In [4], values of $\tau_{0.1}$ were derived experimentally for LWE instances, with values between 0.385 and 0.400 being obtained. Though there are 'structural' differences in the lattices employed in this work and [13],[4], we expect the model above to also hold reasonably well.

In any case, our experimental results suggest that the security bounds derived in Section 4 are already very pessimistic; even bigger keys than (4), for example, should be considered to thwart the improved embedding attack.

Acknowledgement. Jean-Charles Faugère and Ludovic Perret have been partially supported supported by the Computer Algebra and Cryptography (CAC) project (ANR-09-JCJCJ-0064-01) and the HPAC grant (ANR ANR-11-BS02-013) of the French National Research Agency.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC, pp. 99–108 (1996)
2. Albrecht, M.R., Cid, C., Faugère, J.-C., Fitzpatrick, R., Perret, L.: On the complexity of the BKW algorithm on LWE. Cryptology ePrint Archive, Report 2012/636 (2012), `http://eprint.iacr.org/`; Des. Codes Cryptogr. (2013)
3. Albrecht, M.R., Farshim, P., Faugère, J.-C., Perret, L.: Polly Cracker, revisited. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 179–196. Springer, Heidelberg (2011), `http://eprint.iacr.org/`
4. Albrecht, M.R., Fitzpatrick, R., Gopfert, F.: On the efficacy of solving lwe by reduction to unique-svp. Cryptology ePrint Archive, Report 2013/602 (2013), `http://eprint.iacr.org/`
5. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. PhD thesis, Université Paris VI (2004)
6. Bardet, M., Faugère, J.-C., Salvy, B.: Complexity of Gröbner basis computation for semi-regular overdetermined sequences over F_2 with solutions in F_2. Technical Report 5049, INRIA (December 2003), `http://www.inria.fr/rrrt/rr-5049.html`

[3] However, further improved embedding attacks may enable larger values of n to be attacked using only LLL, but we do not deal with this here.

7. Bardet, M., Faugère, J.-C., Salvy, B.: On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In: Proc. International Conference on Polynomial System Solving (ICPSS), pp. 71–75 (2004)
8. Berbain, C., Gilbert, H., Patarin, J.: QUAD: A multivariate stream cipher with provable security. J. Symb. Comput. 44(12), 1703–1723 (2009)
9. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of Learning with Errors. To appear STOC 2013 (2013)
10. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Ostrovsky, R. (ed.) IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, pp. 97–106. IEEE (2011)
11. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. PhD thesis, University of Innsbruck (1965)
12. Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, pp. 75–83. ACM, New York (2002)
13. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
15. G.: GMP: The GNU multiple precision arithmetic library, `http://gmplib.org/`
16. Goldstein, D., Mayer, A.: On the equidistribution of hecke points (2003)
17. Huang, Y.-J., Liu, F.-H., Yang, B.-Y.: Public-key cryptography from new multivariate quadratic assumptions. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 190–205. Springer, Heidelberg (2012)
18. Kannan, R.: Minkowski's convex body theorem and integer programming. Mathematics of Operations Research 12(3), 415–440 (1987)
19. Khot, S.: Hardness of approximating the shortest vector problem in lattices. J. ACM 52(5), 789–808 (2005)
20. Lovász, L., Lenstra Jr., H.W., Lenstra, A.K.: Factoring polynomials with rational coefficients. Mathematische Annalen 261, 515–534 (1982)
21. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. IACR Cryptology ePrint Archive, 592 (2010)
22. Liu, Y.-K., Lyubashevsky, V., Micciancio, D.: On Bounded Distance Decoding for General Lattices. In: Díaz, J., Jansen, K., Rolim, J.D.P., Zwick, U. (eds.) APPROX 2006 and RANDOM 2006. LNCS, vol. 4110, pp. 450–461. Springer, Heidelberg (2006)
23. Lyubashevsky, V., Micciancio, D.: On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 577–594. Springer, Heidelberg (2009)
24. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
25. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a cryptographic perspective. The Kluwer International Series in Engineering and Computer Science, vol. 671. Kluwer Academic Publishers, Boston (2002)
26. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 147–191. Springer, Heidelberg (2009)

27. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6) (2009)
28. Shoup, V.: NTL: A library for doing number theory, http://shoup.net/ntl/
29. Stéhle, D., et al.: fpLLL 4.0.4. fpLLL Development Team (2013), http://perso.ens-lyon.fr/damien.stehle/fplll/
30. Stein, W.A., et al.: Sage Mathematics Software (Version 5.2). The Sage Development Team (2012), http://www.sagemath.org
31. von Zur Gathen, J., Gerhard, J.: Modern computer algebra, 2nd edn. Cambridge University Press (2003)