Lattice Decoding Attacks on Binary LWE

Shi Bai and Steven D. Galbraith

Department of Mathematics,
University of Auckland,
New Zealand.
S.Bai@auckland.ac.nz
S.Galbraith@math.auckland.ac.nz

Abstract. We consider the binary-LWE problem, which is the learning with errors problem when the entries of the secret vector are chosen from $\{0, 1\}$ or $\{-1, 0, 1\}$ (and the error vector is sampled from a discrete Gaussian distribution). Our main result is an improved lattice decoding algorithm for binary-LWE which first translates the problem to the inhomogeneous short integer solution (ISIS) problem, and then solves the closest vector problem using a re-scaling of the lattice. We also discuss modulus switching as an approach to the problem. Our conclusion is that binary-LWE is easier than general LWE. We give experimental results and theoretical estimates that can be used to choose parameters for binary-LWE to achieve certain security levels.

Keywords: lattice decoding attacks, learning with errors, closest vector problem.

1 Introduction

The learning with errors problem is: Given an $m \times n$ matrix $\mathbf{A}$ and a vector $\mathbf{b} \equiv \mathbf{A}\mathbf{s}+\mathbf{e} \pmod{q}$, where $\mathbf{e} \in \mathbb{Z}_q^m$ is a "short" error vector, to compute $\mathbf{s} \in \mathbb{Z}_q^n$. This is a computational problem of major current importance in cryptography. Recently, Brakerski, Langlois, Peikert, Regev and Stehlé [8] and Micciancio and Peikert [21] have considered variants of this problem where the secret vectors are chosen uniformly from the set $\{0, 1\}^n$ (or $\{-1, 0, 1\}^n$), rather than from $\mathbb{Z}_q^n$. These variants of the problem are called binary-LWE.

It is natural to expect that the binary-LWE problem is easier than the standard LWE problem, but it is an open question to determine how much easier. Both papers [8, 21] give reductions that imply that binary-LWE is hard, but those results require increasing the parameter n to approximately $n \log_2(q)) = O(n \log_2(n))$ (it is usually the case that q is a low-degree polynomial in n). An interesting problem is to determine whether these results are optimal. As an example, taking $n = 256$ for standard LWE would lead to a parameter of at least $n \log_2(n) = 2048$ for binary LWE, which seems excessive.

Our goal is to develop and analyse improved algorithms for the binary-LWE problem. We first translate the problem to a related problem called the inhomogeneous short integer solution problem (ISIS). Our main tool is to *rescale* the lattice so that the standard lattice methods to solve the closest vector problem are more effective. We also consider other approaches to the problem, such as modulus switching. We show that modulus switching is not a helpful tool in this setting, which may be counter-intuitive. We also give theoretical and experimental analysis of the algorithm.

Experimental results with the new algorithm do confirm that the parameter n needs to be increased when using binary-LWE. Returning to the example of $n = 256$, our results suggest that a parameter around 440 may be sufficient to achieve the same security level as standard LWE with parameter 256. This is much smaller and therefore more practical than using parameter 2048.

Our approaches are all based on lattice decoding attacks. There is another class of algorithms for LWE that are more combinatorial, originating with Blum, Kalai and Wasserman [6, 1]. However, these algorithms require an extremely large number of samples from the LWE distribution, which may not be realistic in certain applications.

The paper is organised as follows. Sections 2 and 3 give precise definitions for the LWE and binary-LWE problems. Section 4 recalls the current state-of-the-art for lattice attacks on LWE. Section 5 describes modulus switching and evaluates its performance. Section 6 contains our algorithm and its analysis, specifically the description of the rescaling in Section 6.1 and the discussion of why modulus switching is unhelpful in Section 6.3. Some experimental results, that confirm our improvement over previous methods, are given in Section 7.

2 LWE

Let $\sigma \in \mathbb{R}_{>0}$. Define $\rho_\sigma(x) = \exp(-x^2/(2\sigma^2))$ and $\rho_\sigma(\mathbb{Z}) = 1 + 2\sum_{x=1}^{\infty} \rho_\sigma(x)$. The discrete Gaussian distribution D_σ on $\mathbb{Z}$ with standard deviation σ is the distribution that associates to $x \in \mathbb{Z}$ the probability $\rho_\sigma(x)/\rho_\sigma(\mathbb{Z})$.

Fix parameters (n, m, q, σ). Typical choices of parameters are $(n, m, q, \sigma) = (256, 640, 4093, 32)$. Let $\mathbf{A}$ be a uniformly chosen $m \times n$ matrix with entries in $\mathbb{Z}_q$. Let $\mathbf{s}$ and $\mathbf{e}$ be integer vectors of lengths n and m respectively whose entries are sampled independently from the Gaussian distribution on $\mathbb{Z}$ with standard deviation σ (this is the case of LWE with secrets chosen from the error distribution, which is no loss of generality [3]). We call $\mathbf{s}$ the "secret vector" and $\mathbf{e}$ the "error vector". The LWE distribution is the distribution on $(\mathbb{Z}_q^{m\times n}, \mathbb{Z}_q^m)$ induced by pairs $(\mathbf{A}, \mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q})$ sampled as above. The search-LWE problem is: Given $(\mathbf{A}, \mathbf{b})$ chosen from the LWE distribution, to compute the pair $(\mathbf{s}, \mathbf{e})$. The search-LWE problem is well-defined if there is one pair $(\mathbf{s}, \mathbf{e})$ satisfying $\mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q}$ that is significantly more likely (with respect to the distributions on $(\mathbf{s}, \mathbf{e})$) to have be chosen than any other solution.

The $(m, n, q, \mathcal{B})$-SIS problem is: Given an $n \times m$ integer matrix $\mathbf{A}'$ (where typically m is much bigger than n) and an integer q to find a vector $\mathbf{y} \in \mathbb{Z}^m$, if it exists, such that $\mathbf{A}'\mathbf{y} \equiv 0 \pmod{q}$ and $\mathbf{y} \in \mathcal{B}$. Here $\mathcal{B}$ is a set of vectors that are "short" in some sense (e.g., $\mathcal{B} = \{\mathbf{y} \in \mathbb{Z}^m : \|\mathbf{y}\| \le B\}$ for some bound B, or $\mathcal{B} = \{-1, 0, 1\}^m$). One can also define an inhomogeneous version of the

SIS problem (ISIS): Given $\mathbf{A}'$ and $\mathbf{v}$ find $\mathbf{y} \in \mathcal{B}$, if it exists, such that $\mathbf{A}'\mathbf{y} \equiv \mathbf{v} \pmod{q}$.

The LWE problem can be rephrased as inhomogenous-SIS: Given $(\mathbf{A}, \mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q})$ one can form the ISIS instance

$$(\mathbf{A}|\mathbf{I}_m) \begin{pmatrix} \mathbf{s} \\ \mathbf{e} \end{pmatrix} \equiv \mathbf{b} \pmod{q}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix. An alternative transformation of LWE to ISIS is mentioned in Remark 1 of Section 4.3. Conversely, ISIS can be translated to LWE, for details see Lemmas 9 and 10 of Micciancio and Mol [20]. However, it is notable that the (I)SIS problem has often been considered in the case when the solution vector $\mathbf{y}$ might lie in $\{0,1\}^m$ or $\{-1,0,1\}^m$ and might not be uniquely determined, whereas for LWE the focus has always been on vectors sampled from discrete Gaussians and there being a unique most likely solution.

2.1 Size of the error vector

Let D_σ be the discrete Gaussian distribution on $\mathbb{Z}$ with standard deviation σ. Let $\mathbf{e}$ be sampled from D_σ^m, which means that $\mathbf{e} = (e_1, \dots, e_m)$ is formed by taking m independent samples from D_σ. We need to know the distribution of $\|\mathbf{e}\|$. If the entries e_i were chosen from a true Gaussian with standard deviation σ then $\|\mathbf{e}\|^2$ comes from the chi-squared distribution, and so has mean $m\sigma^2$. Since our case is rather close, we assume that $\|\mathbf{e}\|^2$ is also close to a chi-squared distribution, and we further assume that the expected value of $\|\mathbf{e}\|$ is close to $\sqrt{m}\sigma$. Lyubashevsky (Lemma 4.4(3) of the full version of [18]) shows that

$$\Pr\left(\|\mathbf{e}\| \le k\sigma\sqrt{m}\right) \ge 1 - \left(ke^{\frac{1-k^2}{2}}\right)^m$$

for $k > 0$. This supports our assumption that $\|\mathbf{e}\| \approx \sqrt{m}\sigma$. To achieve overwhelming probability, we may use $k \approx 2$. In practice, this bound is quite useful for $k \gtrapprox 1$. In practice, we can easily estimate the expected value of $\|\mathbf{e}\|$ for any fixed parameters by sampling.

3 Binary LWE and related work

We now restrict the LWE problem so that the secret vector $\mathbf{s}$ is chosen to lie in a much smaller set. Fix (n, m, q, σ). To be compatible with Regev's results (e.g., see Theorem 1.1 of [24]), we usually take $\sigma \approx 2\sqrt{n}$. Let $\mathbf{A}$ be a uniformly chosen $m \times n$ matrix with entries in $\mathbb{Z}_q$. Let $\mathbf{s} \in \mathbb{Z}^n$ have entries chosen independently and uniformly from $\{0,1\}$. Let $\mathbf{e} \in \mathbb{Z}^m$ have entries sampled independently from the discrete Gaussian distribution on $\mathbb{Z}$ with standard deviation σ. The binary-LWE distribution is the distribution on $(\mathbb{Z}_q^{m\times n}, \mathbb{Z}_q^m)$ induced by pairs $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q})$ sampled as above. The search-binary-LWE problem is: Given $(\mathbf{A}, \mathbf{b})$ chosen from the binary-LWE distribution, to compute the pair

$(\mathbf{s}, \mathbf{e})$. One can also consider a decisional problem, but in this paper we focus on the search problem.

The binary-LWE problem (where secret vectors $\mathbf{s}$ are from $\{0, 1\}^n$) has been considered in work by Brakerski, Langlois, Peikert, Regev and Stehlé [8]. The main focus of their paper is to prove hardness results for LWE in the classical setting (i.e., without using quantum algorithms as in Regev's original result). They use modulus switching, which is a tool to transform an LWE instance modulo q to an LWE instance modulo a different prime q'. Micciancio and Peikert [21] have considered the binary-LWE problem where $\mathbf{s} \in \{-1, 0, 1\}^n$. Their main result is a hardness result for the case where not only the secrets are small but even the errors are small. Of course, due to the Arora-Ge attack [4] this is only possible if one makes the (realistic) assumption that one has access to a very restricted number of samples from the LWE distribution. This problem is closely relevant to the ISIS problem, since the ISIS problem is always stated in terms of a fixed number of samples.

It is worth noting that there is a standard reduction [3] from LWE to the case of LWE where the secret is chosen from the error distribution. But there is not a general reduction from LWE instances to ones whose error is chosen from the secret's distribution (apart from the naive case of $n \times n$ LWE instances $(\mathbf{A}, \mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q})$ giving $(\mathbf{A}' \equiv \mathbf{A}^{-1} \pmod{q}, \mathbf{b}' \equiv \mathbf{A}^{-1}\mathbf{b} \equiv \mathbf{A}'\mathbf{e} + \mathbf{s} \pmod{q})$).

Both papers [8, 21] give reductions that imply that binary-LWE is hard, assuming certain other lattice problems are hard. Essentially, the papers relate (n, q)-binary-LWE to $(n/t, q)$-LWE (where $t = O(\log(n)) = O(\log(q))$). In other words, we can be confident that binary-LWE is hard as long as we increase the parameter n by a factor of $\log(n)$. For example, taking $n = 256$ as a reasonably hard case for standard LWE, we can be confident that binary-LWE is hard for $n = 256 \log_2(256) = 2048$. Our feeling is that these reductions are too conservative, and that binary-LWE is harder than these results would suggest.

The main goal of our paper is to study the LWE problem where the secret vector is binary, but the errors are still discrete Gaussians. We focus on the case $\mathbf{s} \in \{-1, 0, 1\}^n$, but our methods are immediately applicable to the case $\mathbf{s} \in \{-B, \ldots, -1, 0, 1, \ldots, B\}$ for any $B < \sigma$.

It is clear that one can solve the binary-LWE problem in $O(3^n)$ operations (or $O(2^n)$ when entries are in $\{0, 1\}$), by trying all choices for $\mathbf{s}$ and testing whether $\mathbf{b} - \mathbf{A}\mathbf{s} \pmod{q}$ is a short vector. There is also a meet-in-the-middle attack that requires $\tilde{O}(3^{n/2})$ (respectively, $\tilde{O}(2^{n/2})$) space and time. One can also convert to ISIS and apply algorithms due to Howgrave-Graham and Joux [13] and Becker, Coron and Joux [5]. All such attacks can be defeated by taking $n \geq 200$ (the storage requirement is a serious constraint).

4 Standard lattice attack on LWE

We recall the standard lattice decoding attack on LWE, and its analysis. Let $L = \Lambda_q(\mathbf{A}^T) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{v} \equiv \mathbf{A}\mathbf{s} \pmod{q}, \mathbf{s} \in \mathbb{Z}^n\}$. This is a lattice of rank m.

Typically the rank of $\mathbf{A}$ will be n, and so L has volume q^{m-n}. Suppose one can solve the closest vector problem (CVP) instance $(L, \mathbf{b})$. Then one finds a vector $\mathbf{v} \in L$ such that $\|\mathbf{b} - \mathbf{v}\|$ is small. Writing $\mathbf{e} = \mathbf{b} - \mathbf{v}$ and $\mathbf{v} \equiv \mathbf{A}\mathbf{s} \pmod{q}$ for some $\mathbf{s} \in \mathbb{Z}^n$ (it is easy to solve for $\mathbf{s}$ using linear algebra when $m \geq n$), then

$$\mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q}.$$

Hence, if we can solve CVP then we have a chance to solve LWE.

The CVP instance can be solved using the embedding technique [14] (reducing CVP to SVP in a lattice of dimension one larger) or an enumeration algorithm (there are several such algorithms, but Liu and Nguyen [16] argue that all variants can be considered as cases of pruned enumeration algorithms). For the complexity analysis here we use the embedding technique, so we recall this now. Some discussions of enumeration algorithms will be given in Section 7.3.

Let $L \subseteq \mathbb{Z}^m$ be a lattice of rank m with (column) basis matrix $\mathbf{B}$, and suppose $\mathbf{b} \in \mathbb{Z}^m$ is a target vector. We wish to find $\mathbf{v} = \mathbf{B}\mathbf{u} \in L$ such that $\mathbf{e} = \mathbf{v} - \mathbf{b} = \mathbf{B}\mathbf{u} - \mathbf{b}$ is a short vector. The idea is to consider the basis matrix, where $M \in \mathbb{N}$ is chosen appropriately (e.g., $M \approx \sqrt{m}\sigma$),

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{b} \\ 0 & M \end{pmatrix}. \tag{1}$$

This is the basis for a lattice L' of rank $d = m + 1$ and volume $M \cdot \mathrm{vol}(L)$. Note that

$$\mathbf{B}' \begin{pmatrix} \mathbf{u} \\ -1 \end{pmatrix} = \begin{pmatrix} \mathbf{B}\mathbf{u} - \mathbf{b} \\ -M \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ -M \end{pmatrix}.$$

Hence, the (column) lattice generated by $\mathbf{B}'$ contains a short vector giving a potential solution to our problem. One therefore applies an SVP algorithm (e.g., LLL or BKZ lattice basis reduction).

Lyubashevsky and Micciancio (Theorem 1 of [17]) argue that the best choice for M above is $\|\mathbf{e}\|$, which is approximately $\sqrt{m}\sigma$ in our case. However, in our experiments $M = 1$ worked fine (and leads to a more powerful attack [2] in practice).

4.1 Unique-SVP

Gama and Nguyen [11] have given a heuristic approach to estimate the capability of lattice basis reduction algorithms. Consider a lattice basis reduction algorithm that takes as input a basis for a lattice L of dimension d, and outputs a list of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$. Gama and Nguyen define the root Hermite factor of such an algorithm to be $\delta \in \mathbb{R}$ such that

$$\|\mathbf{b}_1\| \leq \delta^d \mathrm{vol}(L)^{1/d}$$

for all d and almost all lattices L.

The standard LLL algorithm corresponds to $\delta = 1.021$. The paper [11] argues that $\delta = 1.01$ is about the limit of practical algorithms (i.e., variants of BKZ using extreme pruning and large block size). Chen and Nguyen [10] extended this analysis to algorithms with greater running time. Their heuristic argument is that a Hermite factor corresponding to $\delta = 1.006$ might be reachable with an algorithm performing around 2^{110} operations.

In Section 3.3 of [11], Gama and Nguyen turn their attention to the unique-SVP problem. One seeks a short vector in a lattice L when one knows that there is a large gap $\gamma = \lambda_2(L)/\lambda_1(L)$, where $\lambda_i(L)$ denotes the i-th successive minima of the lattice. The unique-SVP problem arises when solving CVP using the embedding technique. The standard theoretical result is that if one is using a lattice reduction algorithm with Hermite factor δ, then the algorithm outputs the shortest vector if the lattice gap satisfies $\gamma > \delta^{2m}$. However, Gama and Nguyen observe that practical algorithms will succeed as long as $\gamma > c\delta^m$ for some small constant c (their paper gives $c = 0.26$ and $c = 0.45$ for different families of lattices). Moreover, Luzzi, Stehlé and Ling [19] gave some theoretical justification that the unique-SVP problem is easier to solve when the gap is large.

4.2 Application to LWE

Consider running the embedding technique on an LWE instance, using the lattice L' given by the matrix $\mathbf{B}'$ from equation (1). We have a good chance of getting the right answer if the error vector $\mathbf{e}$ is very short compared with the second shortest vector in the lattice L', which we assume to be the shortest vector in the original lattice L.

The Gaussian heuristic suggests that the shortest vector in a lattice L of rank d has Euclidean norm about $\frac{1}{\sqrt{\pi}}\Gamma(1+\frac{d}{2})^{1/d}\mathrm{vol}(L)^{1/d}$ which is approximately $\sqrt{\frac{d}{2\pi e}}\mathrm{vol}(L)^{1/d}$. In lattice L (of rank m), this is $\sqrt{\frac{m}{2\pi e}}q^{(m-n)/m}$. Note also that our lattices contain known vectors of Euclidean length equal to q. Hence, our estimate of the Euclidean length of known short vectors is

$$\lambda_2(L') \approx \lambda_1(L) \approx \min\left\{q, \sqrt{\frac{m}{2\pi e}}\, q^{\frac{m-n}{m}}\right\}.$$

In contrast, the vector $\mathbf{e}$ has Euclidean length around $\sqrt{m}\sigma$ on average (see Section 2.1), and so the vector $\binom{\mathbf{e}}{M}$ has length approximately $\sqrt{2m}\sigma$ when $M = \sqrt{m}\sigma$. In our experiments we take $M = 1$ and so assume that $\lambda_1(L') \approx \sqrt{m}\sigma$. Hence the gap is

$$\gamma(m) = \frac{\lambda_2(L')}{\lambda_1(L')} \approx \frac{\min\{q, \frac{1}{\sqrt{\pi}}\Gamma(1+\frac{m}{2})^{1/m}q^{\frac{m-n}{m}}\}}{\sqrt{m}\sigma} \approx \frac{\min\{q, \sqrt{\frac{m}{2\pi e}}q^{\frac{m-n}{m}}\}}{\sqrt{m}\sigma}. \quad (2)$$

For a successful attack we want this gap to be large, so we will need

$$\sigma \ll q^{\frac{m-n}{m}} < \frac{q}{\sqrt{m}}.$$

To determine whether an LWE instance can be solved using the embedding technique and a lattice reduction algorithm with a given (root) Hermite factor δ, one can choose a suitable subdimension m and verify that the corresponding gap satisfies the condition $\gamma = \gamma(m) > c\delta^m$ for a suitable value c. Since the constant c is unknown, we can maximize $\min\{q, q^{(m-n)/m}\}/\delta^m$ for fixed n, q, δ to get the "optimal" sub-dimension (which maximizes the success probability of the algorithm) to be

$$m = \sqrt{\frac{n\log(q)}{\log(\delta)}}, \tag{3}$$

where δ is the Hermite factor of the lattice basis reduction algorithm used.

Furthermore, we may assume c is upper bounded by 1 due to Gama and Nguyen [11]. Hence, for fixed $n, q, \sigma = 2\sqrt{n}$, we can easily compute values (m, δ) satisfying the constraint $\gamma^{1/m} \geq \delta$ and such that δ is maximal. These values have lattice dimension m as in equation (3). By doing this we obtained Table 1 (for $n \geq 160$ the length of the second shortest vector is taken to be q and this leads to very large dimensions; enlarging q to around 13000 in the case $n = 300$ leads to $m = 1258$ and $\delta \approx 1.002$). The last row consists of the estimated time

$$\log(T_{BKZ}) = \frac{1.8}{\log_2(\delta)} - 110 \tag{4}$$

for running the BKZ lattice basis reduction algorithm, based on Lindner and Peikert's work [15]. Note that we do not know the value of the constant c for our lattices, only the experimental results by Gama and Nguyen [11]. There is no known sharp theoretical bound for it. Hence the running time in Table 1 may not be the optimal embedding attack for the LWE problem with parameter n.

Table 1: Theoretical prediction of (optimal) root Hermite factor δ and running time T of the standard embedding technique algorithm using BKZ for LWE instances with $q = 4093$, $\sigma = 2\sqrt{n}$ for the given values for n. The lattice dimension $d = m + 1$ is calculated using equation (3) and the running time T is estimated using equation (4).

n	30	40	50	60	70	100	150	200	250	300
d	110	151	194	239	284	425	673	1144	1919	3962
$\delta \approx$	1.0208	1.0147	1.0111	1.0088	1.0072	1.0046	1.0028	1.0013	1.0006	1.0002
$\log(T) \approx$	0	0	3	33	63	161	343	872	2100	7739

The running times and values for δ in Table 1 are worse than those reported in some other papers on LWE. This is because we consider rather large values $\sigma = 2\sqrt{n}$ for the error distribution, instead of very small values like $\sigma = 3$. Since LWE can always be reduced to the case where the secrets are chosen from the error distribution, the question of the hardness of binary-LWE is most interesting when the error distribution itself is not very small.

4.3 How to solve ISIS

Recall the inhomogeneous-SIS (ISIS) problem: Given $(\mathbf{A}', \mathbf{v})$ to find a short vector $\mathbf{y} \in \mathbb{Z}^m$ such that $\mathbf{v} \equiv \mathbf{A}'\mathbf{y} \pmod{q}$. It is standard that ISIS is also attacked by reducing to CVP: One considers the lattice $L' = \Lambda_q^{\perp}(\mathbf{A}') = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{A}'\mathbf{y} \equiv 0 \pmod{q}\}$, finds any vector (not necessarily short) $\mathbf{w} \in \mathbb{Z}^m$ such that $\mathbf{A}'\mathbf{w} \equiv \mathbf{v} \pmod{q}$, then solves CVP for $(L', \mathbf{w})$ to find some $\mathbf{y}$ close to $\mathbf{w}$ and so returns $\mathbf{w} - \mathbf{y}$ as the ISIS solution.

We sketch the details of solving LWE (in the case of short secrets) by reducing to ISIS and then solving by CVP (more details are given in Section 6). Given $(\mathbf{A}, \mathbf{b})$ we define $\mathbf{A}' = (\mathbf{A}|\mathbf{I}_m)$ to get an ISIS instance $(\mathbf{A}', \mathbf{b})$. Choose any vector $\mathbf{w} \in \mathbb{Z}^{n+m}$ such that $\mathbf{A}'\mathbf{w} \equiv \mathbf{b} \pmod{q}$. Then the lattice $L' = \Lambda_q^{\perp}(\mathbf{A}') = \{\mathbf{y} \in \mathbb{Z}^{n+m} : \mathbf{A}'\mathbf{y} \equiv 0 \pmod{q}\}$ is seen to have rank $m' = n + m$ and (assuming the rank of $\mathbf{A}'$ is n) determinant $q^m = q^{m'-n}$ (the determinant condition can be seen by considering the index of the subgroup $q\mathbb{Z}^{n+m}$ in the additive group L'). The condition for success in the algorithm is $\sigma \ll q^{m/(n+m)}$. Writing $m' = n+m$ this is $q^{(m'-n)/m'}$, which is the same as the LWE condition above.

Remark 1. We can also reduce LWE to ISIS using the approach of Micciancio and Mol [20]. In particular, one can construct a matrix $\mathbf{A}^{\perp} \in \mathbb{Z}_q^{(m-n)\times m}$ such that $\mathbf{A}^{\perp}\mathbf{A} \equiv 0 \pmod{q}$. The LWE problem $(\mathbf{A}, \mathbf{b})$ is therefore transformed into the ISIS instance $(\mathbf{A}^{\perp}, \mathbf{A}^{\perp}\mathbf{b} \equiv \mathbf{A}^{\perp}\mathbf{e} \pmod{q})$. It follows that a solution to the ISIS problem gives a value for $\mathbf{e}$ and hence solves the LWE problem. It is easy to see that this approach is equivalent to the previous one in the case where the secret vector $\mathbf{s}$ is chosen from the error distribution. However, since this reduction eliminates the vector $\mathbf{s}$, we are no longer able to take advantage of the "smallness" of $\mathbf{s}$ compared with $\mathbf{e}$, as we will do in the following sections. So we do not consider this approach further.

4.4 Distinguishing attack

One can also study the decisional variant of the LWE problem: Given a pair $(\mathbf{A}, \mathbf{b})$ to decide if it has been sampled uniformly at random from $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$, or from the LWE distribution. There is a standard distinguishing algorithm based on finding short vectors in the lattice $\{\mathbf{v} \in \mathbb{Z}^m : \mathbf{v}\mathbf{A} \equiv \mathbf{0} \pmod{q}\}$. The idea is that if $\mathbf{v}$ is such a lattice point and if $\mathbf{b} \equiv \mathbf{A}\mathbf{s}+\mathbf{e} \pmod{q}$ then $\mathbf{v}\mathbf{b} \equiv \mathbf{v}\mathbf{e} \pmod{q}$ may be a small integer. Linder and Peikert [15] have argued that this approach is generally less effective than the decoding attack on the computational variant of LWE. We remark that one can define a decisional variant of binary-LWE (where the LWE distribution is defined by choosing both $\mathbf{s}$ and $\mathbf{e}$ to be small), but the above distinguishing attack no longer solves this problem as it only tests that $\mathbf{e}$ is small. Hence, we do not consider the distinguishing attack in this paper.

5 Modulus switching

Modulus switching was first proposed by Brakerski and Vaikuntanathan [7], in the context of homomorphic encryption. Write the LWE instance $(\mathbf{A}, \mathbf{b} \equiv \mathbf{A}\mathbf{s}+\mathbf{e}$

(mod q)) as

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

for some $\mathbf{u} \in \mathbb{Z}^m$. Now suppose q' is another integer and define $\mathbf{A}' = [\frac{q'}{q}\mathbf{A}]$ and $\mathbf{b}' = [\frac{q'}{q}\mathbf{b}]$, where the operation [ ] applied to a vector or matrix means rounding each entry to the nearest integer. Write $\mathbf{A}' = \frac{q'}{q}\mathbf{A} + \mathbf{W}$ and $\mathbf{b}' = \frac{q'}{q}\mathbf{b} + \mathbf{w}$ where $\mathbf{W}$ is an $m \times n$ matrix with entries in $[-1/2, 1/2]$ and $\mathbf{w}$ is a length m vector with entries in $[-1/2, 1/2]$. One can now verify that

$$\begin{aligned} \mathbf{b}' - \mathbf{A}'\mathbf{s} &= \tfrac{q'}{q}\mathbf{b} + \mathbf{w} - (\tfrac{q'}{q}\mathbf{A} + \mathbf{W})\mathbf{s} \\ &= \tfrac{q'}{q}(\mathbf{A}\mathbf{s} + \mathbf{e} + q\mathbf{u} - \mathbf{A}\mathbf{s}) + \mathbf{w} - \mathbf{W}\mathbf{s} \\ &= \tfrac{q'}{q}\mathbf{e} + \mathbf{w} - \mathbf{W}\mathbf{s} + q'\mathbf{u}. \end{aligned}$$

One sees that $(\mathbf{A}', \mathbf{b}')$ is an LWE instance modulo q', with the same secret vector, and that the "error vector" has length

$$\|\tfrac{q'}{q}\mathbf{e} + \mathbf{w} - \mathbf{W}\mathbf{s}\| \leq \tfrac{q'}{q}\|\mathbf{e}\| + \|\mathbf{w}\| + \|\mathbf{W}\mathbf{s}\|.$$

Note that the final term $\|\mathbf{W}\mathbf{s}\|$ has the potential to be small only when $\mathbf{s}$ has small entries, as is the case for binary LWE. The term $\|\mathbf{w}\|$ is bounded by $\frac{1}{2}\sqrt{m}$. The term $\|\mathbf{W}\mathbf{s}\|$ is easily bounded, but it is more useful to determine its expected value. Each entry of the vector $\mathbf{W}\mathbf{s}$ is a sum of n (or around $n/2$ in the case where $\mathbf{s} \in \{0, 1\}^n$) rational numbers in the interval $[-1/2, 1/2]$. Assuming the entries of $\mathbf{W}$ are uniformly distributed then the central limit theorem suggests that each entry of $\mathbf{W}\mathbf{s}$ has absolute value roughly $\frac{1}{4}\sqrt{n/2}$. Hence, it seems plausible to think that $\|\mathbf{W}\mathbf{s}\|$ can be as small as $\frac{1}{4}\sqrt{nm}$.

Modulus switching was originally proposed to control the growth of the noise under homomorphic operations. The standard scenario is that if $\|\mathbf{e}\|$ becomes too large then, by taking q' much smaller than q, one can reduce the noise by the factor $\frac{q'}{q}$ while only adding a relatively small additional noise. However, the idea is also interesting for cryptanalysis: One can perform a modulus switching to make the error terms smaller and hence the scheme more easily attacked. We will consider such an attack in the case of binary LWE in the next section.

We now give a back-of-the-envelope calculation that shows modulus switching can be a useful way to improve lattice attacks on LWE. Note that modulus switching reduces the error vector by a factor of $\frac{q'}{q}$, as long as the other terms (dominated by $\frac{1}{4}\sqrt{nm}$) introduced into the noise are smaller than $\frac{q'}{q}\sigma\sqrt{m}$. However, note that the volume of the lattice is also reduced, since it goes from $q^{(m-n)/m}$ to $q'^{(m-n)/m}$. Let us write ϵ for the reduction factor $\frac{q'}{q}$. All other parameters remaining the same, the lattice gap $\gamma = \lambda_2/\lambda_1 \approx q^{(m-n)/m}/(\sigma\sqrt{2\pi e})$ changes to

$$\gamma' \approx (\epsilon q)^{(m-n)/m}/(\epsilon\sigma\sqrt{2\pi e}) = (\epsilon^{1-n/m}/\epsilon)\gamma = \epsilon^{-n/m}\gamma. \qquad (5)$$

Now, $0 < \epsilon < 1$ and so this is a positive improvement to the lattice gap (and hence the Hermite factor).

For LWE we usually have errors chosen from a discrete Gaussian with standard deviation at most $2\sqrt{n}$, and so $\|\mathbf{e}\|$ is typically $O(\sqrt{mn})$. As discussed above, the additional noise introduced by performing modulus reduction (from the $\mathbf{Ws}$ term) will typically be around $\frac{1}{4}\sqrt{nm}$. Hence, it seems the best we can hope for is $q'/q \approx \frac{1}{8}$ giving an error vector of norm reduced by a factor of approximately $\frac{1}{4}$ (from $2\sqrt{mn}$ to $\sqrt{mn}/2$). This does give a modest improvement to the performance of lattice decoding algorithms for LWE.

6 New attacks on binary-LWE

We now present our original work. We want to exploit the fact that $\mathbf{s}$ is small. The standard lattice attack on LWE (reducing to CVP) cannot use this information. However, going via ISIS seems more appropriate.

6.1 Reducing binary-LWE to ISIS and then rescaling

Let $(\mathbf{A}, \mathbf{b})$ be the (n, m, q, σ)-LWE instance. We may discard rows to reduce the value for m. We write $m' = n+m$. Write $\mathbf{A}' = (\mathbf{A}|\mathbf{I}_m)$, being an $m \times m'$ matrix, and consider the ISIS instance

$$\mathbf{b} \equiv \mathbf{A}'\left(\begin{smallmatrix}\mathbf{s}\\ \mathbf{e}\end{smallmatrix}\right) \pmod{q}$$

where the target short vector is $\left(\begin{smallmatrix}\mathbf{s}\\ \mathbf{e}\end{smallmatrix}\right)$.

The next step is to reduce this ISIS instance to CVP in a lattice. So define the vector $\mathbf{w} = (0, \mathbf{b}^T)^T$. Clearly $\mathbf{A}'\mathbf{w} \equiv \mathbf{b} \pmod{q}$. We now construct a basis matrix $\mathbf{B}$ for the lattice $L' = \{\mathbf{v} \in \mathbb{Z}^{m'} : \mathbf{A}'\mathbf{v} \equiv 0 \pmod{q}\}$. This can be done as follows: The columns of the $(n+m) \times (m+2n)$ matrix

$$\mathbf{M} = \left(\begin{array}{c|c} \mathbf{I}_n & \\ & q\mathbf{I}_{n+m} \\ -\mathbf{A} & \end{array}\right)$$

span the space of all vectors $\mathbf{v}$ such that $\mathbf{A}'\mathbf{v} \equiv 0 \pmod{q}$. Computing the column Hermite normal form of $\mathbf{M}$ gives an $m' \times m'$ matrix $\mathbf{B}$ whose columns generate the lattice L'.

One can confirm that $\det(\mathbf{B}) = q^m = q^{m'-n}$. As before, we seek a vector $\mathbf{v} \in \mathbb{Z}^{m'}$ such that $\mathbf{Bv} \equiv 0 \pmod{q}$ and $\mathbf{v} \approx \mathbf{w}$. We hope that $\mathbf{w} - \mathbf{v} = \left(\begin{smallmatrix}\mathbf{s}\\ \mathbf{e}\end{smallmatrix}\right)$ and so $\mathbf{v} = \left(\begin{smallmatrix}\mathbf{s}\\ *\end{smallmatrix}\right)$, where $*$ is actually going to be $\mathbf{b} - \mathbf{e}$. Our main observation is that $\|\mathbf{s}\| \ll \|\mathbf{e}\|$ and so the CVP algorithm is trying to find an unbalanced solution. It makes sense to try to rebalance things.

Our proposal is to multiply the first n rows of $\mathbf{B}$ by σ (or some other appropriate scaling factor). This increases the volume of the lattice, without significantly increasing the norm of the error vector in the CVP instance. As a result, the

Hermite factor of the problem is increased and hence the range of the lattice attack for a given security level is increased.

A further trick, when $\mathbf{s} \in \{0,1\}^n$, is to rebalance $\mathbf{s}$ so that it is symmetric around zero. In this case we rescale by multiplying the first n rows of $\mathbf{B}$ by 2σ and then subtract $(\sigma, \ldots, \sigma, 0, \ldots, 0)^T$ from $\mathbf{w}$. Now the difference $\mathbf{w} - \mathbf{v}$ is of the form

$$(\pm\sigma, \ldots, \pm\sigma, \mathbf{e}_1, \ldots, \mathbf{e}_m)^T$$

which is more balanced.

6.2 Gap in the Unique-SVP

The determinant has been increased by a factor of σ^n (or $(2\sigma)^n$ in the $\{0,1\}$ case). So the gap in the re-scaled lattice is expected to be larger compared to the original lattice. In the embedded lattice formed by the standard attack, $\lambda_1(L') \approx \sqrt{m} \cdot \sigma$ and $\lambda_2(L') \approx q^{(m-n)/m}\sqrt{\frac{m}{2\pi e}}$ where m is the subdimension being used. In the embedded lattice formed by the new attack, $\lambda_1(L') \approx \sqrt{m+n} \cdot \sigma$ and $\lambda_2(L') \approx (q^m \sigma^n)^{1/(m+n)}\sqrt{\frac{m+n}{2\pi e}}$ where m is the number of LWE samples being used. Hence the new lattice gap is $\gamma = \lambda_2(L')/\lambda_1(L')$ and so we will need to use lattice reduction algorithms with Hermite factor $\delta \leq \gamma^{1/(m+n)}$.

Lemma 1. *Let q, n, σ and δ be fixed. Let $m' \approx m + n$ be the dimension of the embedded lattice in the new attack. For a given Hermite factor δ, the optimal value for m' is approximately*

$$\sqrt{\frac{n(\log q - \log \sigma)}{\log \delta}}. \tag{6}$$

Proof. The goal is to choose m' (and hence m) to minimize the function $f(m') = q^{(m'-n)/m'}\sigma^{n/m'}\delta^{-m'}$. It suffices to find a minimum for the function $F(x) = \log(f(x)) = ((x-n)/x)\log(q) + (n/x)\log(\sigma) - x\log(\delta)$. Differentiating gives $n(\log(q) - \log(\sigma)) = x^2\log(\delta)$ and the result follows.

Table 2: Theoretical prediction of (optimal) root Hermite factor δ and running time T of embedding technique for rescaled binary-LWE instances $\mathbf{s} \in \{-1,0,1\}^n$ with $q = 4093$, $\sigma = 2\sqrt{n}$ for the given values for n. The lattice dimension d' ($\approx m'$) is calculated using equation (6) and the running time T is estimated using equation (4).

n	30	40	50	60	70	100	150	200	250	300
d'	78	105	132	160	187	271	414	558	799	1144
δ	1.0296	1.0212	1.0164	1.0132	1.0111	1.0073	1.0045	1.0032	1.0019	1.0011
$\log(T)$	0	0	0	0	3	63	169	280	545	1031

Given n, q and σ, we use Lemma 1 to obtain Table 2 of optimal subdimensions m' and values for δ. Comparing this table with Table 1 one sees that the lattice dimensions m' and the Hermite factors δ are all much improved.

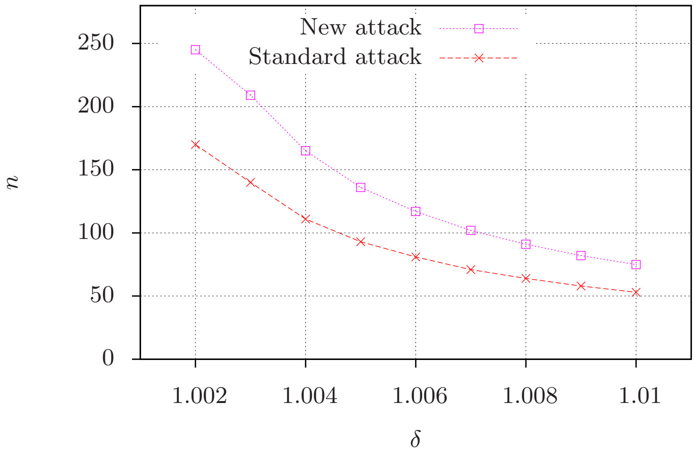

Fig. 1: Theoretical prediction of the largest binary-LWE parameter n that can be solved using an algorithm with the given root Hermite factor.

By fixing a lattice reduction algorithm that has the ability to produce some fixed Hermite factor δ, we can compare the maximum n that this algorithm can attack, based on the standard attack or our new attack. Figure 1 indicates that, for instance, the binary LWE with secret in $\{-1, 0, 1\}$ and $n \approx 100$ provides approximately the same security as the regular LWE with $n \approx 70$.

6.3 Using modulus switching

It is natural to consider applying modulus switching before performing the improved lattice attack. We now explain that this is not a good idea in general.

As discussed in Section 5, the best we can try to do is to have $q'/q \approx 1/8$ and the error vector is reduced in size from elements of standard deviation σ to elements of standard deviation approximately $\sigma/4$.

Consider the desired Hermite factor $\delta = \gamma^{1/m'}$ to attack a lattice with gap

$$\gamma = (\sigma^{n/m'} q^{(m'-n)/m'} / (\sigma\sqrt{2\pi e}))^{1/m'}$$

as in our improved lattice attack using rescaling. Applying this attack to the lattice after modulus switching gives Hermite factor

$$\left((\tfrac{1}{4}\sigma)^{n/m'} (\tfrac{1}{8}q)^{(m'-n)/m'} / (\tfrac{1}{4}\sigma\sqrt{2\pi e}) \right)^{1/m'} = \delta \left(\frac{1}{2^{(m'-n)/m'}} \right)^{1/m'} \tag{7}$$

which is strictly smaller than δ. Hence, the instance after modulus switching is harder than the instance before modulus switching. Intuitively, the problem is this: Modulus switching reduces the size of q and also the size of the error.

But it reduces q by a larger factor than it reduces the size of the error (due to the additional error arising from the modulus switching process). When we do the rescaling, we are also rescaling by a smaller factor relative to q. Hence, the crucial lattice gap property is weakened by modulus switching.

6.4 Combining the lattice attack with exhaustive search

A natural extension is to first guess k bits of the secret $\mathbf{s}$ and then apply the lattice attack to the remaining problem. Since this reduces n it also reduces the optimal choice for m, leading to a simpler problem.

For example, attacking an instance with $n = 100$ one could repeat the attack 2^{25} times, trying all possibilities for the first 25 entries of $\mathbf{s}$, where the lattice attack is now applied to binary-LWE instances having $n = 75$, which seems quite practical compared with the 2^{64} time for $n = 100$ predicted in Table 2. We do not consider this further in the paper.

7 Experiments

Our theoretical analysis (Figure 1) indicates that our new algorithm is superior to previous methods when solving CVP using the embedding technique. In this section we give experimental evidence that confirms these theoretical predictions. However, the state-of-the-art for solving CVP is not to use the embedding technique, but to use enumeration methods with suitable pruning strategies. Hence, in this section we also report some predictions based on experiments of using enumeration algorithms to solve binary-LWE using the standard method and our new method. For full details on enumeration algorithms in lattices see [10–12].

The binary LWE problem considered in this section has secret vectors $\mathbf{s} \in \{-1, 0, 1\}^n$ (i.e., it follows Micciancio and Peikert's definition [21]). Thus our results are more conservative compared to the case where $\mathbf{s} \in \{0, 1\}^n$. In the experiments, we fix parameters $q = 4093$ and vary $n \in [30, 80]$. We use $\sigma = 2\sqrt{n}$.

7.1 Embedding

We first consider the embedding technique with $M = 1$ to solve the CVP problems (we used FPLLL [9] on a 2.4G desktop). In Tables 1 and 2, we have determined the optimal (root) Hermite factor and subdimension that maximize the success probability using the embedding technique. However, when (the Hermite factor of) a lattice reduction algorithm is fixed (call it δ), the optimal subdimension m is the one that minimizes the running time while satisfying the lattice gap argument: $\gamma(m) > c\delta^m$ for some constant c (where $\gamma(m)$ is defined in equation (2)).

For a successful attack we want the lattice gap $\gamma(m)$ to be larger than δ^m which is to assume c is upper bounded by 1. As long as this condition is satisfied, we can reduce m in order to minimize the running time.

In the meantime, we want to maintain a certain success probability. In the LWE problem, the norm of the error vector is unknown to the attacker, so we guess that its value is equal to the average norm of 10^4 randomly sampled vectors from the error distribution. We choose a bound for the norm of the error vector so that the expected success probability is $\geq 1/2$. In this way, we can decide an optimal m. Also in our experiments, we restrict to $m \geq n$. On the other hand, if $\gamma(m) < \delta^m$ for all m, we set $m \approx \sqrt{n \log q / \log \delta}$ which maximizes $\gamma(m)/\delta^m$ for given δ. Of course, the reduction algorithm is likely to fail in such cases.

Table 3: Results of the embedding technique using BKZ for binary-LWE using the standard approach and the new lattice rescaling (with and without modulus switching). The columns m_i are the number of LWE samples used for the experiments (the value in parenthesis is the theoretical value for m_i from equation (3) or equation (6) as appropriate). The lattice dimensions are $d_1 = m_1 + 1$, $d_2 = m_2 + n + 1$ and $d_3 = m_3 + n + 1$. The lattice gap γ_i is estimated as in equation (2) and the corresponding Hermite factor is $\delta_i = \gamma_i^{1/d_i}$. Column Succ is the success probability observed from 10 trials (where – denotes no success at all).

	Standard embedding attack				New attack				New attack with modulus switching			
n	m_1	γ_1^{1/d_1}	Succ	Time	m_2	γ_2^{1/d_2}	Succ	Time	m_3	γ_3^{1/d_3}	Succ	Time
30	68 (151)	1.013	1.0	$0.83s$	30 (97)	1.027	1.0	$0.32s$	53(90)	1.023	1.0	$3.76s$
40	105 (174)	1.012	1.0	$6.70s$	40 (105)	1.019	1.0	$1.30s$	67(96)	1.018	1.0	$6.29s$
50	195 (195)	1.011	0.5	$61.71s$	50 (111)	1.015	1.0	$3.61s$	84(101)	1.014	1.0	$10.58s$
60	214 (214)	1.009	–	$90.20s$	115 (115)	1.013	1.0	$27.83s$	104(104)	1.011	0.4	$17.17s$
70	231 (231)	1.007	–	$127.82s$	117 (117)	1.011	0.5	$42.41s$	105(105)	1.010	–	$29.11s$
80	247 (247)	1.005	–	$189.25s$	119 (119)	1.009	–	$56.54s$	106(106)	1.009	–	$43.88s$

In Table 3, we use BKZ-60 with pruned enumeration [11]. To decide the optimal subdimension as described above, we assume the Hermite factor $\delta \gtrapprox 1.011$. This is verified experimentally in Table 3 and in [11]. Note that using a smaller dimension than the "optimum" may be slightly faster. In the standard attack, the optimal subdimension is m_1 and the lattice dimension is $d_1 = m_1 + 1$. In the new attack, the re-scaled lattice has dimension $d_2 = m_2 + n + 1$. We record the average running time for ten instances. The values for $\gamma_i = \lambda_2/\lambda_1$ are computed by assuming λ_1 is the length of the error vector and that λ_2 is given by the Gaussian heuristic. The success probability reflects the fact that we are using BKZ for the embedding technique, and so for larger n the shortest vector in the reduced basis is not the desired target vector. To get a higher success probability one uses enumeration, as discussed in Section 7.3.

7.2 Modulus switching

We also experimented with modulus switching for the new algorithm. We confirm our theoretical analysis that the performance is worse. As mentioned in Section 5, the best choice for modulus switching is to use q' such that $q'/q \approx 1/8$. The third block in Table 3 records the running time and success probability of the new attack based on modulus switching. Note that we use $q' = 512$. The table shows that the success probability is worse than the new attack without modulus switching.

7.3 Enumeration

When solving CVP for practical parameters the state of the art method [15, 16] is to use BKZ pre-processing of the lattice basis followed by pruned enumeration. This is organised so that the time spent on pre-processing and enumeration is roughly equal. We consider these algorithms here. Note that one can expect a similar speedup from our lattice rescaling for the binary-LWE problem, since the volume of the lattice is increased, which creates an easier CVP instance.

We give predictions of the running time for larger parameters using Chen, Liu and Nguyen's methods [10, 16]: we first preprocess the CVP basis by BKZ-β for some large β and then enumerate on the reduced basis.

Write $\delta(\beta)$ for the Hermite factor achieved by BKZ with blocks of size β. Given a target $\delta(\beta)$ and dimension m, Chen and Nguyen [10] described an algorithm to estimate the BKZ time. It is observed that a small number of calls to the enumeration routine (for each block reduction in the BKZ-β) is often sufficient to achieve the targeted δ. It boils down to estimating the enumeration time (either for the local basis within BKZ or the full enumeration later), which depends on the number of nodes visited in the enumeration. We use the approach of [10, 16] to estimate the enumeration time, which assumes the Gaussian heuristic and the Geometric Series Assumption (GSA) [25]. Following this approach, and under those assumptions, we estimate the running time for solving binary-LWE with $n = 128, q = 4093$ in Table 4.

Table 4: Predictions of the running time for solving binary-LWE with $(n, q, \sigma) = (128, 4093, 22.6)$ using BKZ lattice reduction followed by pruned enumeration. Columns d_i are the lattice dimensions. The BKZ reduction (preprocessing) achieves the targeted Hermite factor δ_i. Column T_{Red} is an estimate of the BKZ reduction time (in seconds). Column $\#E$ denotes the estimated number of nodes in the enumeration. Column T denotes the estimated total running-time in seconds.

Standard attack					New attack				
δ_1	d_1	$\log(T_{Red})$	$\log(\#E)$	$\log(T)$	δ_2	d_2	$\log(T_{Red})$	$\log(\#E)$	$\log(T)$
1.008	366	42.94	197.96	175	1.009	273	29.35	57.22	34
1.007	391	59.13	152.99	130	1.0085	280	34.27	48.07	35
1.0065	405	76.82	129.54	107	1.008	289	42.61	39.19	43
1.006	422	93.04	105.71	94	1.007	309	58.74	23.09	59

8 Conclusion

We have described a lattice rescaling approach to the binary-LWE problem, and we have given theoretical and experimental results that confirm its superiority to the standard approach. These results are most interesting when the standard deviation of the error distribution is large.

Figure 2 plots (the comparison of) the running time of our attack (using the embedding technique) for binary LWE and standard LWE. This graph should only be interpreted as a very rough approximation to the truth, but it allows us to compare the relative security. The papers [8, 21] have shown that to match the hardness of standard LWE for parameter n one can use binary-LWE with parameter $n\log(n)$. Figure 2 suggests that this is overkill and that even $n\log(\log(n))$ may be more than sufficient. However, it seems to be not sufficient to take parameter cn where c is a constant.

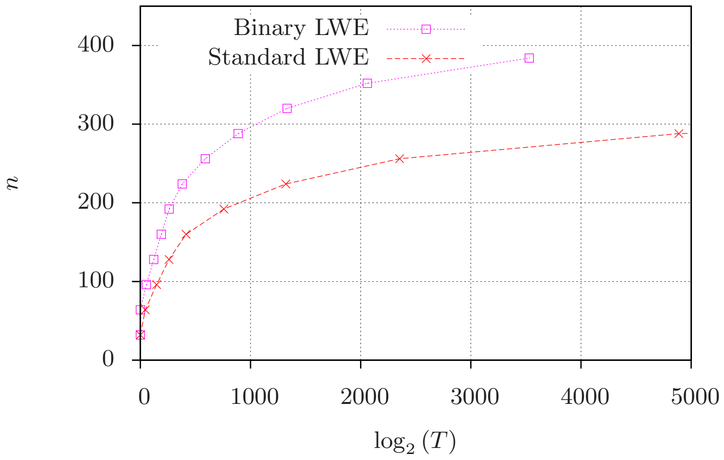

Fig. 2: Plot of predicted running time with respect to LWE parameter n for embedding attack on standard LWE and binary LWE.

Acknowledgements

The authors are grateful to Chris Peikert for suggesting the lattice re-scaling idea. The authors would also like to thank Robert Fitzpatrick, Mingjie Liu, Phong Q. Nguyen and the Program Chairs for helpful comments and discussions.

The authors wish to acknowledge NeSI (New Zealand eScience Infrastructure) and the Centre for eResearch at the University of Auckland for providing CPU hours and support.

References

1. Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick and Ludovic Perret, On the Complexity of the BKW Algorithm on LWE, to appear in Designs, Codes and Cryptography. Published online 19 July 2013.
2. Martin R. Albrecht, Robert Fitzpatrick, and Florian Göpfert, On the Efficacy of Solving LWE by Reduction to Unique-SVP, to appear in proceedings of 2013 International Conference on Information Security and Cryptology.
3. Benny Applebaum and David Cash and Chris Peikert and Amit Sahai, Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems, in S. Halevi (ed.), CRYPTO 2009, Springer LNCS 5677 (2009) 595–618.
4. Sanjeev Arora and Rong Ge, New Algorithms for Learning in Presence of Errors, in L. Aceto, M. Henzinger and J. Sgall (eds), ICALP, Springer LNCS 6755 (2011) 403–415.
5. Anja Becker, Jean-Sébastien Coron and Antoine Joux, Improved Generic Algorithms for Hard Knapsacks, in K. G. Paterson (ed.), EUROCRYPT 2011, Springer LNCS 6632 (2011) 364–385.
6. Avrim Blum, Adam Kalai and Hal Wasserman, Noise-tolerant learning, the parity problem, and the statistical query model, Journal of ACM, **50**, no. 4 (2003) 506–519.
7. Zvika Brakerski and Vinod Vaikuntanathan, Efficient Fully Homomorphic Encryption from (Standard) LWE, in R. Ostrovsky (ed.), FOCS 2011, IEEE (2011) 97–106.
8. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev and Damien Stehlé, Classical hardness of learning with errors, in D. Boneh, T. Roughgarden and J. Feigenbaum (eds.), STOC 2013, ACM (2013) 575–584.
9. David Cadé, Xavier Pujol and Damien Stehlé, FPLLL, http://perso.ens-lyon.fr/damien.stehle/fplll, 2013.
10. Yuanmi Chen and Phong Q. Nguyen, BKZ 2.0: Better Lattice Security Estimates, in D. H. Lee and X. Wang (eds.), ASIACRYPT 2011, Springer LNCS 7073 (2011) 1–20.
11. Nicolas Gama and Phong Q. Nguyen, Predicting Lattice Reduction, in N. P. Smart (ed.), EUROCRYPT 2008, Springer LNCS 4965 (2008) 31–51.
12. Nicolas Gama, Phong Q. Nguyen and Oded Regev, Lattice enumeration using extreme pruning, in H. Gilbert (ed.), EUROCRYPT 2010, Springer LNCS 6110 (2010) 257–278.
13. Nick Howgrave-Graham and Antoine Joux, New Generic Algorithms for Hard Knapsacks, in H. Gilbert (ed.), EUROCRYPT 2010, Springer LNCS 6110 (2010) 235–256.
14. Ravi Kannan, Minkowski's convex body theorem and integer programming, Mathematics of Operations Research **12**, no. 3 (1987) 415–440.
15. Richard Lindner and Chris Peikert, Better key sizes (and attacks) for LWE-based encryption, in A. Kiayias (ed.), CT-RSA'11, Springer LNCS 6558 (2011) 319–339.
16. Mingjie Liu and Phong Q. Nguyen, Solving BDD by Enumeration: An Update, in E. Dawson (ed.), CT-RSA 2013, Springer LNCS 7779 (2013) 293–309.
17. Vadim Lyubashevsky and Daniele Micciancio, On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem, in S. Halevi (ed.), CRYPTO 2009, Springer LNCS 5677 (2009) 577–594.
18. Vadim Lyubashevsky, Lattice signatures without trapdoors, in D. Pointcheval and T. Johansson (eds.), EUROCRYPT 2012, Springer LNCS 7237 (2012) 738–755.

19. Laura Luzzi, Damien Stehlé and Cong Ling, Decoding by Embedding: Correct Decoding Radius and DMT Optimality, IEEE Transactions on Information Theory, 59(5), (2013) 2960–2973.
20. Daniele Micciancio and Petros Mol, Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions, in P. Rogaway (ed.), CRYPTO 2011, Springer LNCS 6841 (2011), 465–484.
21. Daniele Micciancio and Chris Peikert, Hardness of SIS and LWE with Small Parameters, in R. Canetti and J. A. Garay (eds.), CRYPTO 2013, Springer LNCS 8042 (2013) 21–39.
22. Daniele Micciancio and Oded Regev, Lattice-based cryptography, in D. J. Bernstein, J. Buchmann, and E. Dahmen (eds.), Post Quantum Cryptography, Springer (2009) 147–191.
23. Oded Regev, On lattices, learning with errors, random linear codes, and cryptography, in H. N. Gabow and R. Fagin (eds.), STOC 2005, ACM (2005) 84–93.
24. Oded Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM 56(6), article 34, 2009.
25. Claus P. Schnorr, Lattice reduction by random sampling and birthday methods, In Proc. STACS 2003, Eds. H. Alt and M. Habib, LNCS 2607 (2003), 145–156.