A Wunderer-style analysis of the hybrid attack on FHE-style LWE parameters

Rachel Player

rachel.player@rhul.ac.uk

The hybrid lattice reduction and meet-in-the-middle attack (referred to simply as the hybrid attack) was introduced by Howgrave-Graham [How07] as an efficient lattice attack on the NTRU [HPS98] cryptosystem. The hybrid attack can be applied to LWE [BGPW16]. A recent analysis of the hybrid attack can be found in Wunderer's thesis [Wun18] who gives under- and overestimates for the total cost of the hybrid attack (including the precomputation part, which consists of performing a strong lattice reduction). We state these under- and overestimates in Equations 7 and 8 respectively. To determine the cost $T_{\text{hyb}}(\beta, r)$ of the hybrid attack itself (excluding precomputation), together with its success probability $p_{\text{succ}}(\beta, r)$, for a given scheme, the analysis can be structured as follows [Wun18]:

- Constructing an appropriate lattice;
- Determining the attack parameters, including c_i, k, y, Y;
- Determining the success probability p_{succ}; and
- Optimising the runtime.

1 Constructing the lattice

The hybrid attack as presented by Wunderer [Wun18] returns a unique shortest vector $\mathbf{w}$ in the dimension-d lattice Λ given a basis of Λ in the following form:

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{I}_r \end{pmatrix} \in \mathbb{Z}^{d \times d}, \tag{1}$$

where the submatrix $\mathbf{B}$ is a basis of the sublattice in which lattice problems will be solved and r is the guessing dimension. Note that for a q-ary lattice for prime q, we can always construct a basis of this form. When the hybrid attack is applied in the LWE setting, the embedding of Kannan [Kan87], or respectively Bai and Galbraith [BG14], can be used to transform an LWE instance $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ into a uSVP instance with $\mathbf{w} = (\mathbf{e}, 1)$, or respectively $\mathbf{w} = (\mathbf{s}, \mathbf{e}, 1)$.

We will use a Bai and Galbraith [BG14] embedding. This enables us to guess in the $\mathbf{s}$ part, and take advantage of the sparseness of $\mathbf{s}$. It is not clear how to take advantage of the sparseness of $\mathbf{s}$ when using Kannan's embedding.

We split $\mathbf{s}$ into a guessing part and a decoding part as $\mathbf{s} = (\mathbf{s}_g, \mathbf{s}_l)$, and split $\mathbf{A} = (\mathbf{A}_1 || \mathbf{A}_2)$ with $\mathbf{A}_1 \in \mathbb{Z}^{m \times r}$ and $\mathbf{A}_2 \in \mathbb{Z}^{m \times (n-r)}$. It can be verified that the lattice with basis

$$\mathbf{M} = \begin{pmatrix} \mathbf{I}_r & \mathbf{0} \\ -\mathbf{A}_1 & \mathbf{M}' \end{pmatrix}$$

where

$$\mathbf{M}' = \begin{pmatrix} \mathbf{I}_{n-r} & \mathbf{0} & \mathbf{0} \\ -\mathbf{A}_2 & q\mathbf{I}_m & \mathbf{b} \\ \mathbf{0} & \mathbf{0} & 1 \end{pmatrix}$$

contains the short vector $(\mathbf{s}_g, \mathbf{s}_l, \mathbf{e}, \mathbf{1})$. Denote $\mathbf{w} = (\mathbf{w}_g, \mathbf{w}_l)$ where $\mathbf{w}_g = \mathbf{s}_g$ and $\mathbf{w}_l = (\mathbf{s}_l, \mathbf{e}, \mathbf{1})$. We can see that for some $\mathbf{x} \in \mathbb{Z}^m$,

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

Hence $-\mathbf{A}_1\mathbf{s}_g$ is close to the lattice $\Lambda(\mathbf{M}')$, the offset being the short vector $\mathbf{w}_l$. If we can guess $\mathbf{w}_g$ correctly, we can hope to recover $\mathbf{w}_l$ as the output of Babai's Nearest Plane algorithm [Bab85].

In fact, we will use the rescaling idea of Bai and Galbraith [BG14] to balance the contribution of $\mathbf{s}_l$ and $\mathbf{e}$. This increases the determinant of the lattice, making the attack easier. Concretely, we will find the short vector $\mathbf{w}_l = (\nu\,\mathbf{s}_l, \mathbf{e}, \mathbf{1})$ contained in the lattice with basis

$$\mathbf{M}'' = \begin{pmatrix} \nu\mathbf{I}_{n-r} & \mathbf{0} & \mathbf{0} \\ -\mathbf{A}_2 & q\mathbf{I}_m & \mathbf{b} \\ \mathbf{0} & \mathbf{0} & 1 \end{pmatrix}$$

This lattice has determinant $\nu^{n-r}q^m$.

2 Determining the attack parameters

With a basis of the form in Equation 1 we split $\mathbf{w} = (\mathbf{w}_l, \mathbf{w}_g)$ into a short vector $\mathbf{w}_l \in \mathbb{Z}^{m-r}$ to be recovered by solving lattice problems and a short vector $\mathbf{w}_g \in \mathbb{Z}^r$ to be recovered by guessing. The guessing part will be sped up using a meet-in-the-middle process. Because of the form of the basis $\mathbf{B}'$, we know that for some $\mathbf{x} \in \mathbb{Z}^{m-r}$,

$$\mathbf{w} = \begin{pmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{I}_r \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{w}_g \end{pmatrix} = \begin{pmatrix} \mathbf{B}\mathbf{x} + \mathbf{C}\mathbf{w}_g \\ \mathbf{w}_g \end{pmatrix} = \begin{pmatrix} \mathbf{w}_l \\ \mathbf{w}_g \end{pmatrix}.$$

Hence $\mathbf{C}\mathbf{w}_g = -\mathbf{B}\mathbf{x} + \mathbf{w}_l$; that is, the vector $\mathbf{C}\mathbf{w}_g$ is close to the lattice $\Lambda(\mathbf{B})$, the offset being the short vector $\mathbf{w}_l$. If we can guess $\mathbf{w}_g$ correctly, we can hope to recover $\mathbf{w}_l$ as the output of Babai's Nearest Plane algorithm [Bab85]. We denote this as $\mathrm{NP}_{\mathbf{B}}(\mathbf{C}\mathbf{w}_g) = \mathbf{w}_l$.

The parameters k and c_i. Denote by k an upper bound on the infinity norm of the guessed part $\mathbf{w}_g$, so that $\|\mathbf{w}_g\|_\infty \leq k$. This means in particular that all entries of $\mathbf{w}_g$ are in $\{-k, \ldots, -1, 0, 1, \ldots k\}$. Denote by X the set of vectors from which we guess $\mathbf{w}_g$. Wunderer [Wun18] defines X to be the set of all vectors such that there are a fixed number $2c_i$ of each of the nonzero entries $i \in \{\pm 1, \ldots, \pm k\}$. Informally, $2c_i$ is the 'Hamming weight' of entries of $\mathbf{w}_g$ taking the values $\pm i$.

In fact, it is sufficient for the analysis that $2c_i$ is the expected number of entries equal to i for each $i \in \{\pm 1, \ldots, \pm k\}$.

In our setting, we are guessing $\mathbf{w}_g \in \mathbb{Z}^r$ which we know has components in $\{-1, 0, 1\}$ and hence can set $k = \|\mathbf{w}_g\|_\infty = 1$. We expect that $\frac{r}{n} \cdot h$ of the nonzero vectors are in $\mathbf{w}_g$, assuming independence of the coordinates of $\mathbf{w}_g$. Of these, we expect the same number of 1s as -1s hence we set $2c_1 = \frac{rh}{2n}$ and $2c_{-1} = \frac{rh}{2n}$.

The parameters y and Y. For the meet-in-the-middle speed-up we will guess vectors $\mathbf{w}'_g$ and $\mathbf{w}''_g$ from a set W such that $\mathbf{w}'_g + \mathbf{w}''_g = \mathbf{w}_g$, for vectors $\mathbf{w}_g \in X$. The set W is defined as the set of all vectors such that there are (expected to be) c_i of each of the nonzero entries $i \in \{\pm 1, \ldots, \pm k\}$. Denote $\mathrm{NP}_{\mathbf{B}}(\mathbf{C}\mathbf{w}'_g) = \mathbf{w}'_l$ and $\mathrm{NP}_{\mathbf{B}}(\mathbf{C}\mathbf{w}''_g) = \mathbf{w}''_l$. We need to be able to recognise pairs $\mathbf{w}'_g$ and $\mathbf{w}''_g$ such that $\mathbf{w}'_g + \mathbf{w}''_g = \mathbf{w}_g$ and $\mathbf{w}'_l + \mathbf{w}''_l = \mathbf{w}_l$. In order to do so, we store guesses $\mathbf{w}'_g$ in hash boxes whose addresses depend on $\mathbf{w}'_l$. Suitable addresses are obtained from the observation that in the case of such a collision, $\mathbf{w}'_l$ and $\mathbf{w}''_l$ differ only by a vector of infinity norm y where $\|\mathbf{w}_l\|_\infty = y$. We denote the expected Euclidean norm of $\mathbf{w}_l$ as $\|\mathbf{w}_l\| = Y$.

In our setting, we hope to find $\mathbf{w}_l = (\nu\,\mathbf{s}_l, \mathbf{e}, 1)$ using Babai's Nearest Plane algorithm in a lattice of dimension $m+n-r$, ignoring the component introduced by the embedding. We first determine an appropriate ν so that $\|\nu\,\mathbf{s}_l\| \approx \|\mathbf{e}\|$. We choose $\nu = \sqrt{\frac{n-r}{h}}\sigma$, so that the expected squared norm of $\mathbf{w}_l$ is given by

$$\begin{aligned}
\|\mathbf{w}_l\|^2 &= \|\nu\,\mathbf{s}_l\|^2 + \|\mathbf{e}\|^2 + 1^2 \\
&\approx \nu^2 \|\,\mathbf{s}_l\|^2 + \|\mathbf{e}\|^2 \\
&= \frac{n-r}{h}\sigma^2 \cdot h \cdot 1 + m\sigma^2 \\
&= \sigma^2(m+n-r)\,,
\end{aligned}$$

and hence $Y = \|\mathbf{w}_l\| = \sigma\sqrt{m+n-r}$. We set $y = \|\mathbf{w}_l\|_\infty = \|\mathbf{e}\|_\infty = 6\sigma$ as a reasonable upper bound on the infinity norm of $\mathbf{w}_l$.

The set S. We define the set S as the set of all nonzero lattice vectors $\mathbf{w} = (\mathbf{w}_l, \mathbf{w}_g) \in \Lambda$ such that $\mathbf{w}_l \in \mathbb{Z}^{m-r}$ with $\|\mathbf{w}_l\|_\infty \leq y$ and $\|\mathbf{w}_l\| \approx Y$; $\mathbf{w}_g \in \mathbb{Z}^r$ with $\|\mathbf{w}_g\|_\infty \leq k$ and exactly $2c_i$ entries equal to i for all $i \in \{\pm 1, \ldots, \pm k\}$; and $\mathbf{w}_l = \mathrm{NP}_{\mathbf{B}}(\mathbf{C}\mathbf{w}_g)$. The analysis of the hybrid attack assumes that the set S is non-empty: the parameters y, k, and c_i must be such that this is likely to be the case. When the hybrid attack is applied to an LWE instance, it is further assumed that if the attack is successful then $|S| = 1$; that is, the vector $\mathbf{w} = (\mathbf{e}, 1)$[1] is the only vector that can be found by the attack. When applied in the NTRU setting, rotations of a short vector may also be short vectors in the lattice and hence it is possible that $|S| > 1$ (see for example [Wun18, Section 5.4.2]).

[1] or $\mathbf{w} = (\mathbf{s}, \mathbf{e}, 1)$, depending on the embedding

The probability p. For the meet in the middle part to be successful, we need that $\mathbf{w}'_l + \mathbf{w}''_l = \mathbf{w}_l$ holds for the correct pair $\mathbf{w}'_g$ and $\mathbf{w}''_g$. This is equivalent to requiring that the vector $\mathbf{C}\mathbf{w}'_g$ is $\mathbf{w}_l$-*admissible* [How07,Wun18]. We are interested in guessing from the subset $V \subseteq W$ defined as

$$V = \{\mathbf{v} \in W | \mathbf{w}_g - \mathbf{v} \in W \text{ and } \mathbf{C}\mathbf{v} \text{ is } \mathbf{w}_l\text{-admissible for some } (\mathbf{w}_l, \mathbf{w}_g) \in S\} \ .$$

For all $\mathbf{w} = (\mathbf{w}_l, \mathbf{w}_g) \in S$ we define $p(\mathbf{w})$ to be the probability, taken over the choice of $\mathbf{v} \leftarrow V$, that $\mathbf{C}\mathbf{v}$ is $\mathbf{w}_l$-admissible. It is assumed that for all $\mathbf{w} \in S$, $p(\mathbf{w}) \approx p$, where the probability p is defined as follows

$$p = \Pr_{\mathbf{x} \leftarrow \mathcal{P}(\mathbf{B}^*), \mathbf{y} \leftarrow S_{m-r}(Y)} [\mathbf{x} \text{ is } \mathbf{y}\text{-admissible}] \tag{2}$$

The probability p can be calculated numerically. The assumptions underlying the definition of p are firstly that $\mathbf{C}\mathbf{v}$ is randomly distributed modulo the parallelepiped $\mathcal{P}(\mathbf{B}^*)$ defined by the Gram-Schmidt vectors $\mathbf{B}^*$; and secondly that $\mathbf{w}_l$ is randomly distributed on the surface of the sphere $S_{m-r}(Y)$ in $\mathbb{R}^{m-r}$ of radius Y centred at the origin.

Estimating the cost of the hybrid attack $T_{hyb}(\beta, r)$. Putting all of this together, Wunderer gives under- and overestimates the cost of the hybrid attack $T_{\text{hyb}}(\beta, r)$ as follows:

$$T_{\text{hyb,under}}(\beta, r) = \frac{m-r}{2^{1.06}} L \, . \tag{3}$$

$$T_{\text{hyb,over}}(\beta, r) = \frac{(m-r)^2}{2^{1.06}} L \, . \tag{4}$$

The cost of the hybrid attack is assumed to be dominated by the cost of the Nearest Plane computations for many candidate halves $\mathbf{w}'_g$ of the guessed part until the correct pair $\mathbf{w}'_g + \mathbf{w}''_g = \mathbf{w}_g$ is found. This is stated as [Wun18, Assumption 5.3] and was previously claimed in [How07]. Both Equations 3 and 4 are estimates based on the cost of one call to Nearest Plane for each of the L loops in the algorithm. In dimension d, the cost of one call to Nearest Plane is estimated as $\frac{d}{2^{1.06}}$ for the under estimates and $\frac{d^2}{2^{1.06}}$ for the over estimates. These costs are based on experiments in [HHHW09]. The expected number of loops L in the hybrid attack is estimated as follows:

$$L = \binom{r}{c_{-k}, \ldots, c_k} \cdot \left(p \cdot |S| \cdot \prod_{i \in \{\pm 1, \ldots, \pm k\}} \binom{2c_i}{c_i} \right)^{-\frac{1}{2}} \, . \tag{5}$$

3 Determining the success probability $p_{\text{succ}}(\beta, r)$

The presentation in this subsection assumes for simplicity that $|S| = 1$, as is done for example when the hybrid attack is applied to an LWE instance. [2]

[2] When applied in the NTRU setting, the (expected) number of elements in S must be determined, and the probability that the desired vector is in S must be accounted for when determining the success probability (see for example [Wun18, Section 5.4.2]).

Let $\mathbf{w} = (\mathbf{w}_l, \mathbf{w}_g) \in S$. We denote by p_c the probability that $\mathbf{w}_g$ has exactly $2c_i$ entries equal to i for all $i \in \{\pm 1, \ldots, \pm k\}$. We denote by p_{NP} the probability that $\mathbf{w}_l = \mathrm{NP}_{\mathbf{B}}(\mathbf{C}\mathbf{w}_g)$. Assuming independence of these two events, the success probability can be estimated as

$$p_{\mathrm{succ}} = p_c \cdot p_{\mathrm{NP}} \,. \tag{6}$$

In our setting, p_c is the probability that $\mathbf{w}_g$ has exactly $2c_1 = \frac{rh}{2n}$ entries equal to 1 and $2c_{-1} = \frac{rh}{2n}$ entries equal to -1. The probability p_{NP} can be calculated via [Wun18, Equation 5.6], which depends on the Gram-Schmidt vectors $\mathbf{b}_i^*$ and generalises a result of Lindner and Peikert [LP11]. We use Wunderer's code to calculate the probability p_{NP}.

We can boost the overall success probability to close to one by repeating $1/p_{\mathrm{succ}}$ times. This is exactly what is done for the overestimate (see Equation 8). For the under estimate (see Equation 7), it is additionally assumed that we can amortise the cost of the expensive lattice reduction precomputation step, and hence we only need to repeat the hybrid attack step $1/p_{\mathrm{succ}}$ times. A similar assumption for amortising the cost of an expensive lattice reduction was made in [Alb17].

4 Optimising the runtime of the overall attack

In the precomputation step, a strong lattice reduction is performed to compute a suitable basis $\mathbf{B}$ to be used as an input to Nearest Plane. For a fixed guessing dimension r, the quality of the output basis can be characterised by the root-Hermite factor δ_0 or, equivalently, the blocksize β required to achieve this δ_0. A stronger lattice reduction would mean more time is taken for the precomputation, but the output basis would be of higher quality and hence the Nearest Plane algorithm is more likely to correctly return $\mathbf{w}_l$. The value β therefore determines the trade-off between the runtime of the precomputation and the actual hybrid attack, and should be optimised. We omit Wunderer's estimates $T_{\mathrm{red}}(\beta, r)$ for the cost of the lattice reduction precomputation step as the lattice reduction cost model used by Wunderer is different to that used the HE standard.

As well as optimising the choice of the blocksize β we need to optimise the guessing dimension r. On the one hand, increasing r increases the complexity of the guessing part, since more entries of the secret have to be guessed. On the other hand, increasing r decreases the dimension of the lattice, making the decoding part of the attack easier. Since there are only finitely many choices for r, the optimal choice can be found numerically.

Wunderer's under- and overestimates are obtained by optimising the following functions, giving the total running time, including the hybrid attack itself as well as the lattice reduction precomputation, as a function the BKZ blocksize β and the guessing dimension r:

$$T_{\mathrm{total,under}}(\beta, r) = T_{\mathrm{red,under}}(\beta, r) + \frac{T_{\mathrm{hyb,under}}(\beta, r)}{p_{\mathrm{succ}}(\beta, r)} \,. \tag{7}$$

$$T_{\text{total,over}}(\beta, r) = \frac{T_{\text{red,over}}(\beta, r) + T_{\text{hyb,over}}(\beta, r)}{p_{\text{succ}}(\beta, r)} \,. \tag{8}$$

The optimisation of the functions $T_{\text{total}}(\beta, r)$, in order to find the lowest cost, is done as follows:

- fix a guessing dimension $r \in R = \{1, \ldots, n-1\}$
- find the optimal β (call this β_r) minimising $T_{\text{total}}(\beta, r)$
- pick the smallest among the set $\{T_{\text{total}}(\beta_r, r) \,|\, r \in R\}$

References

[Alb17] Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 103–129. Springer, Heidelberg, April / May 2017.

[Bab85] László Babai. On Lovász' lattice reduction and the nearest lattice point problem (shortened version). In Kurt Mehlhorn, editor, *STACS '86*, volume 82 of *Lecture Notes in Computer Science*, pages 13–20. Springer, 1985.

[BG14] Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In Willy Susilo and Yi Mu, editors, *ACISP 14*, volume 8544 of *LNCS*, pages 322–337. Springer, Heidelberg, July 2014.

[BGPW16] Johannes A. Buchmann, Florian Göpfert, Rachel Player, and Thomas Wunderer. On the hardness of LWE with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 16*, volume 9646 of *LNCS*, pages 24–43. Springer, Heidelberg, April 2016.

[HHHW09] Philip S. Hirschhorn, Jeffrey Hoffstein, Nick Howgrave-Graham, and William Whyte. Choosing NTRUEncrypt parameters in light of combined lattice reduction and MITM approaches. In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *ACNS 09*, volume 5536 of *LNCS*, pages 437–455. Springer, Heidelberg, June 2009.

[How07] Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 150–169. Springer, Heidelberg, August 2007.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe P. Buhler, editor, *ANTS-III*, pages 267–288, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, August 1987.

[LP11] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, February 2011.

[Wun18] Thomas Wunderer. *On the Security of Lattice-Based Cryptography Against Lattice Reduction and Hybrid Attacks*. PhD thesis, Technische Universität, Darmstadt, 2018.