A Note on Parameter Choices of `Round5`

Yongha Son

August 21, 2019

1 Overview

We examine the current parameter choice of `Round5`, and rectify its consideration of the improved dual attack due to Albrecht [Alb17]: there is one significant optimization of Albrecht's dual attack, which was not reflected to `Round5` parameter choices. By taking this into consideration, some parameter choices of `Round5` cannot enjoy the claimed security level.

For simplicity, we omit the details for the dual attack (and primal attack) and assume the readers are already familiar with the attacks. Please refer to Section 2.7.4 and 2.7.5 of original specification document [BBF$^+$19] for the detailed explanation for the attacks.

2 Sparse-secrets Attacks

We briefly review the current consideration for sparse secret of [BBF$^+$19]. Albrecht [Alb17] observed that one can reduce the LWE sample dimension by ignoring randomly chosen k columns of A while expecting all the corresponding components of secret are zero. In this case, the total complexity is estimated by

$$\min_k \frac{T_{dual}(n-k)}{p_{k,h}}$$

where $p_{k,h}$ is the probability that all the k ignored components of secret are zero where the Hamming weight of secret key is h, computed by

$$p_{k,h} = \frac{\binom{n-h}{k}}{\binom{n}{k}}.$$

One can apply the dual attack (or the primal attack) for the lower-dimensional LWE sample, and this corresponds to *Sparse-secrets attack* of the current `Round5` parameter choice table; for instance, Table 3 in page 45 of [BBF$^+$19].

3 Albrecht's Dual Attack

Indeed, Albrecht [Alb17] further observed that, unlike the primal attack, the dual attack for the sparse secret can be further improved by admitting some failures on guessing zero components with some post-process: We may assume that one ignores the first k columns of A, and denote $A = [A_1|A_2]$. Then the dual attack applied on $(A_2, \vec{b})$ finds short vectors $\vec{y}_i \in L_q^{\perp}(A_2)$. It gives

$$\begin{aligned}
\langle \vec{y}_i, \vec{b} \rangle &= \langle \vec{y}_i, A\vec{s} + \vec{e} \rangle \\
&= \langle \vec{y}_i, A_1\vec{s}_1 + A_2\vec{s}_2 + \vec{e} \rangle \\
&= \vec{y}_i^T A_1\vec{s}_1 + \vec{y}_i^T A_2\vec{s}_2 + \langle \vec{y}_i, \vec{e} \rangle \\
&= \vec{y}_i^T A_1\vec{s}_1 + \langle \vec{y}_i, \vec{e} \rangle
\end{aligned}$$

By writing a matrix Y consisting of rows $\vec{y}_i^T$, we have

$$Y\vec{b} = YA_1\vec{s}_1 + Y\vec{e}$$ [1],

from which $(YA_1, Y\vec{b})$ can be viewed as another LWE-instance with secret $\vec{s}_1$ and error $Y\vec{b}$. Now by exhaustively searching some candidates for $\vec{s}_1$ to some extent, the dual attack for the sparse secret succeeds even if all the guessing are not correct.

By putting ℓ by the (expected) Hamming weight of $\vec{s}_1$, we conclude the total complexity by

$$\min_k \frac{T_{dual}(n-k) + T_{exh}(k, \ell)}{\sum_{i=0}^{\ell} p_{k,i}}$$

where $p_{k,h,i}$ is the probability that $\mathsf{HW}(\vec{s}_1) = i$, easily computed by

$$p_{k,h,i} = \frac{\binom{n-k}{h-i} \cdot \binom{k}{i}}{\binom{n}{k}}$$

4 Applying Albrecht's Dual Attack

By taking this improvement of the dual attack for sparse secret, we observe that some parameter sets of [BBF+19] fail to have the claimed security level. For that, we modified the `LWE-estimator` [APS15] to follow the cost estimation of Section 2.7.5 in [BBF+19] for the dual attack (without `postprocess` improvement), and it almost reproduces the bit-security of the dual attack estimated in [BBF+19]. In Appendix A, we specify the changes in code line compared to the original `LWE-estimator`.

Since the postprocess strategy has already been considered and implemented in `LWE-estimator`, we just toggle `postprocess` option to estimate the bit-security of Albrecht's dual attack, which gives Table 1, 2 and 3.

Remark. To reproduce the tables, run the following with our Sage module;

[1] Note that the original strategy simply expects $\vec{s}_1 = \vec{0}$ to have $YA_1\vec{s}_1 = \vec{0}$.

```
sage: n = 490; q = 2**10; h = 162; stddev = 2.29
sage: alpha = stddev * sqrt(2*pi) / q
sage: #‘Dual’ in Round5 specification document
sage: dual_scale(n, alpha, q, secret_distribution=((-1,1),h))
sage: #‘Primal’ in Round5 specification document
sage: primal_usvp(n, alpha, q, secret_distribution=((-1,1),h))
sage: #Alb’s Dual
sage: duald = partial(drop_and_solve, dual_scale)
sage: duald(n, alpha, q, secret_distribution=((-1,1),h), postprocess=True)
```

Table 1: Solving costs for $\lambda = 128$ parameters in [BBF$^+$19].

(Claimed to be) 128 bit-security						
n	$\log q$	h	σ	Primal	Hybrid	**Alb's Dual**[Alb17]
490	10	162	2.29	130	128	**123.6**
508	10	136	2.29	132	128	**124.6**
586	13	182	4.61	130	128	**124.3**
618	11	104	2.29	144	128	131.8

Table 2: Solving costs for $\lambda = 192$ parameters in [BBF$^+$19].

(Claimed to be) 192 bit-security						
n	$\log q$	h	σ	Primal	Hybrid	**Alb's Dual**[Alb17]
756	12	242	4.61	194	192	**183.2**
786	13	384	4.61	192	194	**184.6**
852	12	212	2.29	199	192	**186.6**

Table 3: Solving costs for $\lambda = 256$ parameters in [BBF$^+$19].

(Claimed to be) 256 bit-security						
n	$\log q$	h	σ	Primal	Hybrid	**Alb's Dual**[Alb17]
940	12	414	4.61	256	263	**243.6**
946	11	388	2.29	256	263	**245.2**
1018	14	428	9.23	256	261	**245.3**
1170	13	222	4.61	281	257	256.3

References

[Alb17] Martin R Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In *Proc. of EUROCRYPT '17*, pages 103–129. Springer, 2017.

[APS15] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[BBF$^+$19] Hayo Baan, Sauvik Bhattacharya, Scott Fluhrer, Oscar Garcia-Morchon, Thijs Laarhoven, Rachel Player, Ronald Rietman, Markku-Juhani O Saarinen, Ludo Tolhuizen, José Luis Torre-Arce, and Zhenfei Zhang. Round5:kem and pke based on (ring) learning with rounding. Technical report, 2019.

A Code changes

A.1 BKZ cost model

The default BKZ cost model of `LWE-estimator` is different from the cost model of [BBF$^+$19]. We change the line 1496 of the original code by

```
1497: reduction_default_cost = BKZ.ADPS16
```

A.2 Number of repeats

In the dual attack, finding only one short vector $\vec{y} \in L_q^{\perp}(A)$ is not enough, as the resulting distinguishing advantage from $\langle \vec{y}, \vec{e} \rangle$ and $\langle \vec{y}, \vec{u} \rangle$ is too small. Thus a lots of short vectors in $L_q^{\perp}(A)$ are required to have sufficient advantage, and `LWE-estimator` and `Round5` differently estimates the number of such required vectors. Moreover, `Round5` also accepts a heuristic that one sieving oracle call gives $2^{0.2075\beta}$ numbers of short vectors at once, and hence the number of SVP oracle calls is estimated by the number of required vectors divided by $2^{0.2075\beta}$, which is not considered in `LWE-estimator`.

To relfect this, we change the line 2530 of the original code by

```
2531: vecnum = RR(exp(RR(2*pi**2 * RR(v*stddev/q)**2))/sqrt(2))**2
2532: repeat = max(1, vecnum / (2**(0.2075*ret["beta"])))
```

To evaluate the postprocess cost, we need to memorize the number of required vectors. Thus we add new lines

```
536: u"vecnum": False
2541: ret[u"vecnum"] = vecnum
```

and change line 1801 of the original code by

```
1802: repeat = current["vecnum"]
```

A.3 Repeating BKZ cost

As mentioned above, the dual attack requires a lots of short vectors in $L_q^{\perp}(A)$. In this regard, LWE-estimator accepts a heuristic due to [Alb17] that says, after one (expensive) BKZ, the LLL algorithm followed by re-randomizing also outputs sufficiently short vector. Align with this, LWE-estimator measures the costs for finding short vectors in the same lattice by one BKZ cost plus several LLL costs.

Since Round5 does not consider this heuristic, we switch-off this option by changing the line 2438 of the original code by

```
2439: c=None, use_lll=False):
```