FAQ on Kyber512

NIST

December 2023

Q: What is the current understanding of the security of Kyber512 (i.e., ML-KEM-512)?

A: Here is a brief summary, current as of December 2023. In the Crystals-Kyber 3rd round submission document [1], Kyber512 claims category 1 security as defined in section 4.A.5 of the Call For Proposals (CFP) of the NIST PQC standardization process [2]. Broadly speaking, this means Kyber512 is claimed to be at least as hard to break as AES-128. Since the quantum speedup of brute force attacks against AES-128 via Grover's algorithm is widely acknowledged to be more powerful than the quantum speedups known for the best-known classical lattice attacks[1], classical attack costs are the most relevant for assessing the security of Kyber512.

The CFP mentions assessing security relative to AES-128 according to "a wide variety of metrics that NIST deems potentially relevant to practical security." One such metric highlighted by the CFP is known as "gate-count," which counts operations acting on one or two classical bits at a time. For classical attacks, this is generally applied in the RAM (Random Access Machine) model of computation, where the cost to access a bit of memory is independent of the total size of the memory and the location of the bit within the memory. With this fact in mind, we summarize the current understanding of the security of Kyber512 with regard to the "gate-count" metric:

1. The attack analyzed in the Kyber specification [1] claiming a gate-count of 2^{151} (as compared to about 2^{143} for a brute force attack against AES-128) is still generally acknowledged to be the best attack against Kyber512 (modulo some minor tweaks and optimizations). The specification itself cites earlier published research, including [6].

2. The Kyber512 security estimate of 2^{151} in the gate-count metric came with some uncertainties analyzed in section 5.2.1 of the Kyber specification [1].

[1] Known speedups to sieving-based algorithms e.g. [3] all achieve significantly less than the quadratic speedup achieved by Grover's algorithm and require quantum access to exponentially large memories (QRAM) to achieve the claimed speedup. Known speedups to enumeration-based algorithms achieve a speedup more comparable to Grover's algorithm, but enumeration algorithms e.g. [4] are asymptotically more expensive than sieving algorithms and, when considering the baseline classical attacks, do not appear competitive even for parameters which are small enough to be attacked in practice see e.g. [5].

The sources of uncertainty are labeled Q1-Q8 (excepting Q6 "beyond the gate metric" which concerns corrections to the gate count metric based on memory access costs, and which is not quantified). If these uncertainties are considered together, the uncertainty for the gate count would range from 2^{135} to 2^{165}.

3. An improved attack against Kyber512 (and other lattice schemes) was claimed by Matzov [7]. This attack claimed to reduce the gate count to 2^{137} but the main result was brought into question by Ducas and Pulles [8], which also brought into question earlier claimed improvements by Guo and Johansson [9] along the same lines (this approach is referred to as the dual-sieve attack.) It has been speculated the main Matzov result may be saved, for example, using techniques similar to those of [10], but thus far, this has not been demonstrated. Further discussion of the issues involved in rescuing the dual-sieve attack are discussed in [11, 12, 13] as well as section 6.3 of [8].

4. About 6 out of the 14 bits worth of the security loss claimed by the Matzov paper comes from tweaks and optimizations to sieving algorithms (described in section 6 of their paper), which don't depend on the correctness of the main result.

5. The lattice estimator software of [14] has been updated to include the tweaks and optimizations detailed in section 6 of the Matzov paper, as well as some further improvements corresponding to Q7 in the Kyber specification's analysis. Running "LWE.estimate(schemes.Kyber512, red_shape_model = Simulator.CN11, red_cost_model = RC.MATZOV)" gives the complexity for the attack including both improvements (marked bdd) as $2^{142.2}$. Note: the most recent commit when NIST obtained this number was 564470e. Martin Albrecht, the owner of this Github repository, remarks that the estimator software has not been peer reviewed in detail, and would welcome additional code review.

6. However, this $2^{142.2}$ number does not include hidden overheads (corresponding to Q2 in the Kyber specification) analyzed by [15]. In the context of the relevant class of progressive sieve/progressive BKZ type attacks, this increases the cost of attacks by about a factor of 2^5 which can be reduced to about 2^3 if the memory size is drastically increased (by a factor of 2^{10} or more). NIST internal analysis, subsequently published on the PQC forum, [16] suggests that extrapolating from current technology, this larger memory size is probably beyond the capabilities of a category 1 attacker to build and maintain. The estimated maximum memory size for a category 1 attacker was $2^{96.5}$.

7. Thus, our best estimate for the gate cost of attacking Kyber512 using known techniques is about 2^{147} (or 2^{145} if we consider memories in excess of 2^{105} bits feasible for a category 1 attacker). Considering Q2 to be

resolved and Q7 to be partially resolved would result in an uncertainty window of 2^{135} to 2^{158} (or 2^{133} to 2^{156}).

Another well-known metric is the widely used "coreSVP methodology" laid out in section 6.1 of [17]. This methodology uses simplified formulas to estimate the bit security of lattice-based cryptosystems. Despite progress in lattice cryptanalysis since 2015, and the availability of more refined estimates for concrete gate count, such as that used in the Kyber specification, the coreSVP metric is widely used because it is relatively easy to calculate, and because it allows for the comparison of the security of contemporary parameter sets with those of parameter sets published many years in the past.

The coreSVP formulae are believed to be mostly monotonically related to the real security level of lattice schemes (i.e., a parameter set with higher coreSVP will usually be more secure than one with lower coreSVP). Nonetheless, when applied to parameters in the range relevant to cryptography, the simplifications in the complexity formulae used to compute coreSVP produce numbers that are significantly smaller than those produced by more refined methods for estimating gate counts. For example, the coreSVP formula would only give 2^{118} as the security for Kyber512, which is several orders of magnitude less than more rigorous estimates of the gate count of the attacks being analyzed. For this reason, coreSVP estimates should not be interpreted as concrete gate counts.

An additional important consideration in assessing the real-world security of Kyber512 is the cost of memory access in lattice reduction algorithms. The best known attacks against Kyber512 all involve sieving, which requires exponentially many queries to an exponentially large memory. For the most efficient variants of sieving, there is no known way to use methods like caching to reduce the effective cost of these queries without a significant increase in other computational costs. State of the art implementations of sieving, e.g., [18], trade memory access cost against local computation by adjusting the bucket-size parameter. Using a larger bucket size increases computation but reduces memory access costs. The optimal bucket size is determined by the cost of random-access queries given the total memory size required by the sieving algorithm. E.g., if the cost of random-access queries is assumed to scale like the square root of the size of the memory, then the bucket size should be proportional to the square root of the total memory size.

Asymptotic cost estimates for sieving-based attacks generally are of the form $\log_2(\text{cost}) = \text{constant} \times \text{sieving dimension} + o(\text{sieving dimension})$. In its "beyond-core-SVP" methodology, the Kyber specification [1] estimates the sieving dimension for an attack on Kyber to be 375. Ignoring memory costs, and using the best known sieving techniques (with a bucket size exponentially smaller than the total memory size), results in the constant 0.292..., (equivalent to the BDGL algorithm [19]) which is the constant used in coreSVP. In [18], asymptotic cost estimates are given for several variants with a larger bucket size (paremetrized by k):

- When $k = 1$, (equivalent to the BGJ1 algorithm – a simplified version,

described in [20], of the BGJ algorithm [21]) the bucket size scales with the square root of the memory size, and the resulting constant is 0.349....

- When $k = 2$, the bucket size scales with the cube root of the memory size, and the resulting constant is 0.3294....
- When $k = 3$, the bucket size scales with the fourth root of the memory size, and the resulting constant is 0.3198....

The most widely used assumption (aside from the unrealistic assumption that memory access always has unit cost) is to assume that the cost of random access to memory scales like the square root of the memory size, which would imply the 0.349 constant ($k = 1$). The justification for this assumption is that, under the assumption that a computing system is made up of components, each of which dissipates a heat at a constant rate, then the system can only grow indefinitely in 2 dimensions without overheating. The auxiliary assumption that computing components, especially those devoted to long distance communication and those devoted to the passive storage of memory, will dissipate heat at a constant rate as an attack is scaled up is somewhat dubious – long distance communication technologies like fiber optic cables dissipate far less heat per unit length than, for example, wires used for local communication on a microchip. Also, it may be advantageous to use a more compact arrangement than a 2D array, and to slow down the computation to compensate for the slower heat dissipation. This can be modeled by assuming a smaller exponent for memory costs. NIST's analysis from [16] suggests that, extrapolating current technology to the scale relevant for an attack on Kyber512, that an assumption of memory access cost scaling as the cube root of the size of the memory (implying a constant of 0.3294... ($k = 2$)) might be fairly close to the real cost (although it could be an underestimate or an overestimate.) Given this, it seems reasonable to treat an estimate based on $k = 2$ as a best guess for the real cost of attacking Kyber512 with $k = 3$ and $k = 1$ providing more conservative and more optimistic estimates of the security of Kyber512.

Naively ignoring the o(sieving dimension) term in the asymptotic formulas would suggest a gain of 21 bits of security if $k = 1$, 14 bits of security if $k = 2$ and 10 bits of security if $k = 3$ relative to attack cost models that ignore the cost of accessing memory. While NIST is not aware of concrete estimates for $k = 1, k = 2$, and $k = 3$, which take into account many of the refinements considered for the gate counts based on the RAM model, there are some older works which imply that ignoring subexponential factors may not result in figures that are particularly far off. In particular the tables[2] in [6] give a cost of 2^{158} for an AllPairSearch algorithm equivalent to $k = 1$ at sieving dimension 376. This compares to the gate count of 2^{137} (in sieving dimension 375) quoted by the Kyber specification ignoring memory costs.

Combining the above estimates of the cost of memory access, together with the more refined gate counts from the RAM model (discussed earlier), NIST's

[2]The tables are also available at `https://github.com/jschanck/eprint-2019-1161/blob/main/data/cost-estimate-random_buckets-classical.csv`

best guess for the realistic cost of attacking Kyber512 is the equivalent of about 2^{160} bit operations/ gates, with a plausible range of uncertainty being something like 2^{140} to 2^{180}.

While the current state of research leaves some uncertainty about the precision of security estimates of Kyber512 (and other lattice schemes) against known attacks, the most plausible values for the practical security of Kyber512 against known attacks are significantly higher than that of AES128, and NIST deems it highly unlikely that the known sources of uncertainty are large enough to make Kyber512 significantly less secure than AES128. Without further cryptanalytic advances, this level of security is sufficient for any category 1 application of Kyber512 — a fact which makes NIST comfortable in standardizing it.

Acknowledgement

NIST would like to acknowledge feedback on a preliminary version of this document from Martin Albrecht, Léo Ducas, and John Schanck.

References

[1] Robert Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber algorithm specifications and supporting documentation. Third-round submission to the NIST's post-quantum cryptography standardization process, 2020. `https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

[2] National Institute of Standards and Technology. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`.

[3] André Chailloux and Johanna Loyer. Lattice sieving via quantum random walks. In *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27*, pages 63–91. Springer, 2021. See also `https://eprint.iacr.org/2021/570.pdf`.

[4] Martin R Albrecht, Shi Bai, Pierre-Alain Fouque, Paul Kirchner, Damien Stehlé, and Weiqiang Wen. Faster enumeration-based lattice reduction: Root hermite factor time. In *Annual International Cryptology Conference*, pages 186–212. Springer, 2020. See also `https://eprint.iacr.org/2020/707`.

[5] TU Darmstadt Lattice Challenge website. See https://latticechallenge.org/.

[6] Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck. Estimating quantum speedups for lattice sieves. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 583–613, Cham, 2020. Springer International Publishing. See also https://jmschanck.info/papers/20201127-sieve.pdf.

[7] The Center of Encryption and Information Security – MATZOV IDF. Report on the security of LWE: Improved dual lattice attack, April 2022. Available at https://doi.org/10.5281/zenodo.6412487.

[8] Léo Ducas and Ludo N Pulles. Does the dual-sieve attack on learning with errors even work? In *Annual International Cryptology Conference*, pages 37–69. Springer, 2023. See also https://eprint.iacr.org/2023/302).

[9] Qian Guo and Thomas Johansson. Faster dual lattice attacks forÂ solving lwe withÂ applications toÂ crystals. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 33–62, Cham, 2021. Springer International Publishing.

[10] Charles Meyer-Hilfiger and Jean-Pierre Tillich. Rigorous foundations for dual attacks in coding theory. In *Theory of Cryptography Conference*, pages 3–32. Springer, 2023. See also https://eprint.iacr.org/2023/1460.pdf].

[11] Andreas Wiemers and Stephan Ehlen. A remark on the independence heuristic in the dual attack. Cryptology ePrint Archive, Paper 2023/1238, 2023. https://eprint.iacr.org/2023/1238.

[12] Léo Ducas and Ludo N. Pulles. Accurate score prediction for dual-sieve attacks. Cryptology ePrint Archive, Paper 2023/1850, 2023. https://eprint.iacr.org/2023/1850.

[13] Kévin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Reduction from sparse lpn to lpn, dual attack 3.0. Cryptology ePrint Archive, Paper 2023/1852, 2023. https://eprint.iacr.org/2023/1852.

[14] 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. See malb/lattice-estimator: An attempt at a new LWE estimator (github.com).

[15] Léo Ducas. Estimating the hidden overheads in the bdgl lattice sieving algorithm. In *International Conference on Post-Quantum Cryptography*, pages 480–497. Springer, 2022. See also http://eprint.iacr.org/2022/922.pdf.

[16] Ray Perlner. Nist pqc-forum post "update on category 1 security", August 2020. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/GR3plWpSV4U/m/NajnbRmxBQAJ.

[17] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key {Exchange—A} new hope. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 327–343, 2016. See also https://eprint.iacr.org/2015/1092.pdf.

[18] Léo Ducas, Marc Stevens, and Wessel van Woerden. Advanced lattice sieving on gpus, with tensor cores. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 249–279. Springer, 2021. See also https://eprint.iacr.org/2021/141).

[19] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 10–24. SIAM, 2016. See also https://eprint.iacr.org/2015/1128.

[20] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 717–746, Cham, 2019. Springer International Publishing. See also https://eprint.iacr.org/2019/089.

[21] Anja Becker, Nicolas Gama, and Antoine Joux. Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search. *Cryptology ePrint Archive*, 2015. Available at https://eprint.iacr.org/2015/522.