Multi-ciphertext security degradation for lattices

Daniel J. Bernstein[1,2]

[1] Department of Computer Science, University of Illinois at Chicago, USA
[2] Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
djb@cr.yp.to

Abstract. Typical lattice-based cryptosystems are commonly believed to resist multi-target attacks. For example, the New Hope proposal stated that it avoids "all-for-the-price-of-one attacks". An ACM CCS 2021 paper from Duman–Hövelmanns–Kiltz–Lyubashevsky–Seiler stated that "we can show that $\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}} \approx \mathrm{Adv}^{(n,q_C)\text{-IND-CPA}}_{\text{PKE}}$" for "lattice-based schemes" such as Kyber, i.e. that one-out-of-many-target IND-CPA is as difficult to break as single-target IND-CPA, assuming "the hardness of MLWE as originally defined for the purpose of worst-case to average-case reductions". Meanwhile NIST expressed concern regarding multi-target attacks against non-lattice cryptosystems.

This paper quantifies the asymptotic impact of multiple ciphertexts per public key upon standard analyses of known primal lattice attacks, assuming existing heuristics. The qualitative conclusions are that typical lattice PKEs asymptotically degrade in heuristic multi-ciphertext IND-CPA security as the number of ciphertexts increases. These PKE attacks also imply multi-ciphertext IND-CCA2 attacks against typical constructions of lattice KEMs. Quantitatively, the asymptotic heuristic security degradation is exponential in $\Theta(n)$ for decrypting many ciphertexts, cutting a constant fraction out of the total number of bits of security, and exponential in $\Theta(n/\log n)$ for decrypting one out of many ciphertexts, for conservative cryptosystem parameters.

This shows a contradiction between the existing heuristics and the idea that multi-target security matches single-target security. Also, whether or not the existing heuristics are correct, (1) there are flaws in the claim of an MLWE-based proof of tight multi-target security, and (2) there is a 2^{88}-guess attack breaking one out of 2^{40} ciphertexts for a FrodoKEM-640 public key, disproving FrodoKEM's claim that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin".

Keywords: algorithm analysis, multi-target attacks, lattices

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the Excellence Strategy of the German Federal and State Governments—EXC 2092 CASA—390781972 "Cyber Security in the Age of Large-Scale Adversaries"; by the U.S. National Science Foundation under grant 1913167; by the Taiwan's Executive Yuan Data Safety and Talent Cultivation Project (AS-KPQ-109-DSTCP); and by the Cisco University Research Program. "Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation" (or other funding agencies). Permanent ID of this document: cff870df5aeaaa8a4fd37778bac69658864897ef. Date: 2023.03.17.

1 Introduction

Given $(\mathrm{AES}_k(0), \mathrm{AES}_k(1))$ for an AES-128 key k, brute-force search finds k with at most 2^{128} AES-128 computations, about 2^{143} bit operations. Given $(\mathrm{AES}_k(0), \mathrm{AES}_k(1))$ for each k in a list of 2^{40} AES-128 keys, a batch version of brute-force search finds the entire list with at most 2^{128} AES-128 computations. The obvious batch attack is bottlenecked by memory access rather than computation, but the Hellman–Rivest–van Oorschot–Wiener parallel batch attack uses essentially the same amount of computation and much less communication; see generally [24].

These batch algorithms will find *one of the keys in the list* with only about 2^{103} bit operations. To put this in perspective, Bitcoin currently carries out roughly 2^{111} bit operations per year. The original 2^{143} bit operations sounded comfortably out of reach, but 2^{103} is certainly feasible for a large-scale attacker.

After these 2^{103} bit operations, the attacker has acquired a secret key that the attacker was not supposed to have, perhaps exposing other critical information—e.g., a password that can be used to launch further attacks. Any particular user is likely to be safe, but the full multi-user cryptographic system has been broken.

There are more examples in the literature of cryptographic systems for which known multi-target attacks are much more efficient than state-of-the-art single-target attacks. For example, state-of-the-art non-quantum algorithms to compute discrete logarithms in the group $(\mathbb{Z}/p)^*$, where p is a prime chosen in the usual way, can compute many discrete logarithms for only slightly higher total cost; see, e.g., [2]. More subtly, work can be shared across many primes p; see [20, Chapter 7]. Finding *one of many discrete logarithms* for a single p is as expensive as attacking a single prespecified target, but the batch attacks have lower per-target cost and undermine a common argument for taking small p; see [25, "difficulties for the 'attacker economist' philosophy"].

Given the tremendous diversity of ways that multi-target security has been shown to fail, some cryptosystem proposals (see, e.g., [5, Section 8.3]) presume that there will be $\lg T$ bits of security loss from T-target attacks—for example, 40 bits of security loss from 2^{40}-target attacks—unless proven otherwise. However, one can easily find many other cryptosystem proposals that choose low single-target security levels, implicitly or explicitly presuming that there is *no* multi-target security loss. For those systems, multi-target attacks can easily make the difference between being safe and being broken in the real world.

1.1. Bleeding-edge lattice systems. The current version of Kyber-512 was proposed in [17] in October 2020. The same document presented a new security analysis [17, Section 5] and concluded that "this preliminary analysis gives a cost of 2^{151} gates" to break Kyber-512. This might sound safely beyond 2^{143}, but there are at least three reasons for concern:

- The same document [17, page 26] said that "this number could be affected by a factor of up to 2^{16} in either direction"—i.e., possibly as low as 2^{135} bit operations—because of "known unknowns".

- Subsequent work has added knowledge regarding the "known unknowns" but has also found exploitable gaps in the analysis in [17]. See, e.g., the attacks in [53] and [78], the latter estimating 2^{137} bit operations; some newer papers claim that this is an underestimate, but others claim further speedups.
- These are single-target security analyses, leaving open the possibility of much more efficient multi-target attacks. Even if there are no further single-target improvements, why should users believe that multi-target attacks against Kyber-512 are not already feasible with current technology?

Regarding the third point, the literature quoted in Appendix A gives the impression that typical lattice systems such as Kyber are safe from all-for-the-price-of-one attacks, one-out-of-many attacks, and other types of multi-target attacks. For example, an ACM CCS 2021 paper [47, page 3] stated that "we can show that $\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}} \approx \mathrm{Adv}^{(n,q_C)\text{-IND-CPA}}_{\text{PKE}}$" for "lattice-based schemes", meaning that single-target and one-out-of-many IND-CPA security are approximately the same for these PKEs. This statement relied on a hypothesis, but [47] argued plausibility of the hypothesis.

1.2. Contributions of this paper. This paper has three main contributions:

- Asymptotic analysis and optimization of parameters for well-known single-target lattice attacks, *assuming* accuracy of the existing heuristics regarding those attacks. See Section 2.
- Asymptotic analysis of an attack breaking many ciphertexts for one public key, *assuming* the existing heuristics. This attack has total heuristic cost asymptotically similar to the single-target message-recovery attacks, i.e., much lower heuristic cost per ciphertext. This does not rely on key recovery: it applies even if the cryptosystem is modified to put extra defenses around the secret key. See Section 3.
- Asymptotic analysis of an attack breaking one out of many ciphertexts for one public key, *assuming* the existing heuristics. For a wide range of cryptosystem parameters, this attack has heuristic cost beating the usual single-target message-recovery attack by a factor $2^{\Theta(n/\lg n)}$, contrary to the statement "$\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}} \approx \mathrm{Adv}^{(n,q_C)\text{-IND-CPA}}_{\text{PKE}}$". See Section 4.

This paper also identifies logical gaps in how the previous literature arrived at various statements contrary to this paper's calculations. See Appendix A. Furthermore, this paper presents a particularly easy attack specifically against FrodoKEM, disproving the claim in [12] that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin". See Appendix B. The gap identification and the FrodoKEM attack apply even if the existing heuristics are so inaccurate as to undermine the other analyses.

1.3. Caveats. In the literature on algorithms, papers introducing asymptotic speedups are often followed by papers doing more work to show corresponding concrete speedups. Sometimes, however, concrete input sizes of interest turn out to be too small for the asymptotic speedups to apply, even after further optimization effort. See, e.g., [52].

Establishing Kyber-512's *single-target* security level remains a challenging research problem today; see, e.g., the open problems listed in [78]. Establishing Kyber-512's T*-target* security level is even more challenging. Perhaps Kyber-512 is too small for the heuristic multi-target speedups to be applicable.

Furthermore, relying on heuristics is error-prone, as illustrated by previous lattice-attack literature endorsing application of the same heuristics to S-unit lattices, where the heuristics were later shown to be highly inaccurate; see [32]. Typical techniques to detect errors in analyses, such as checking proofs and carrying out experiments, are inherently unreliable for heuristic asymptotic analyses: saying that the analyses are heuristic implies that the results are not completely proven, and saying that the analyses are asymptotic implies that discrepancies from experiments are to be expected.

Sections 2, 3, and 4 of this paper are saying that two ideas visible in the literature—the idea that the existing heuristics accurately predict lattice security via the standard analysis, and the idea that multi-target lattice security matches single-target lattice security—cannot both be correct. This paper is *not* saying that one idea is accurate and the other idea is inaccurate; it seems more likely that there are serious inaccuracies in *both* ideas.

Yet another caveat is that, like many previous papers on this topic, this paper relies on unrealistic models of computation where arbitrarily large arrays can be accessed for free. The multi-ciphertext attack analyzed here is not exceptionally memory-intensive—it handles one ciphertext at a time, and, for each ciphertext, uses memory in similar ways to the usual single-target attacks—but the issue is that all of these attacks could be outperformed by lower-memory approaches in more realistic models of computation.

Finally, the importance of multi-target attacks rests on the cryptographic community's habit of taking narrow security margins. See generally [16] and [28]. This habit is not a rule; in particular, at the time of this writing, NIST has not committed to standardizing Kyber-512.[3] However, NIST's latest report claims that Kyber-512 is harder to break than single-target AES-128,[4] and there are ongoing large-scale deployment experiments with Kyber-512, such as [39].

[3] For example, NIST's announcement [84] of the Kyber selection includes a statement that "NIST will seek input on specific parameter sets to include" in the resulting standard. NIST stated in November 2022 [90] that it wanted "a broader range of perspectives on whether our current plan to standardize Kyber512 is a good one".

[4] [4, page 8] says "Figure 1 shows [performance] for Kyber, NTRU, and Saber for security categories 1 and 3". The Kyber parameter sets in [4, Figure 1] are Kyber-512 and Kyber-768, evidently assigning Kyber-512 to "category 1" and Kyber-768 to "category 3". [4, page 6] says "category 1" means "the best attack violating the security definition of a parameter set should cost more than a brute-force key search attack on a single instance of AES-128, according to any plausible assumption regarding the relative cost of the various computational resources involved in a real-world attack".

2 Asymptotic heuristic cost of single-target attacks

The starting point for this paper is the standard heuristic analysis of the usual single-target message-recovery and key-recovery attacks against lattice PKEs. This analysis was introduced in the original New Hope paper [13, Section 6.3] in 2015, building on various earlier components, and was then copied into many newer cryptosystem proposals. Recent literature continues to report results of the same analysis; see, e.g., [12, Section 5.2.2]. There have been subsequent speedups and corrections, but the literature generally portrays these adjustments as minor; see, e.g., the recent paper [7], starting with the "history of refinements" title.

As a specific target for attacks, consider the Lyubashevsky–Peikert–Regev PKE [76, eprint version, page 4], using the ring $(\mathbb{Z}/q)[x]/(x^n+1)$ and sampling each error position from distribution χ. The standard heuristic analysis cares only about (n, q, s), where s is the standard deviation of χ. As illustrated by the tables in [6], the same heuristic analysis also applies to a wide range of further cryptosystems; all the analysis needs to know about each cryptosystem is (n, q, s), without regard to complications such as the error-correcting codes in New Hope or the matrices in Kyber.

A critical parameter in the usual attacks is a "block size" β. The standard heuristic analysis includes a standard choice of β, namely the smallest β for which the analysis says that the attacks succeed. The point of this section is to calculate how the standard choice of β scales with the PKE parameters (n, q, s).

Specifically, fix real numbers Q_0, Q_1, S_0, S_1. To simplify the analysis, assume $0 \le S_0 \le 1/2 < Q_0 - S_0$. Consider an infinite sequence of (n, q, s) where

- n runs through a subsequence of $2, 3, 4, \ldots$;
- $\lg q \in Q_0 \lg n + Q_1 + o(1)$, or equivalently $q \in n^{Q_0}(2^{Q_1} + o(1))$, where $o(1)$ means the set of functions of n that converge to 0 as $n \to \infty$; and
- $\lg s \in S_0 \lg n + S_1 + o(1)$.

The conclusion of this section, in short, is that the standard block size β has $\beta/n \in z_0 + (z_1 + o(1))/\lg n$ where $z_0 = 2Q_0/(Q_0 - S_0 + 1/2)^2$ and

$$z_1 = \left(2S_1 + \lg z_0 - \left(S_0 - Q_0 + \frac{3}{2}\right) \lg \frac{z_0}{2\pi \exp 1} - \frac{Q_1(Q_0 + S_0 - \frac{1}{2})}{Q_0}\right) \frac{2Q_0}{(Q_0 - S_0 + \frac{1}{2})^3}.$$

Unsurprisingly, this conclusion applies equally to (1) the usual key-recovery attack and (2) the usual single-target message-recovery attack. This makes the key-recovery attack more attractive, since key recovery trivially implies a multi-ciphertext attack that breaks *all* of the ciphertexts, with just minor cost per ciphertext to run the same decryption algorithm as the legitimate user.

Presumably the use of key-recovery attacks as multi-ciphertext attacks is the motivation for, e.g., [74, Section 6] saying "arguably, the secret key ought to be better-protected than any individual ciphertext". One can easily modify the LPR cryptosystem to use one error distribution for encryption and a larger error distribution for key generation. This section's asymptotics for β/n quantify the impact of moving from one pair (S_0, S_1) for encryption to a (lexicographically) larger pair (S_0, S_1) for key generation.

2.1. Comparison to the structure of concrete analyses. What this section does is conceptually similar to what the literature already does in evaluating attack costs for, e.g., New Hope or Kyber: the same heuristic analyses of the same attacks are used to predict which attack parameters will succeed; these predictions are then used to optimize attack parameters.

The big difference is that the previous predictions and optimizations are concrete while this section is asymptotic. Consider, e.g., [13, Table 1] saying the optimal β/n is $967/1024$ for the original version of New Hope, while this section says the optimal β/n is $270/289 + o(1)$ if the parameters (Q_0, S_0) are $(1.35, 0.15)$. These statements about β might seem similar, especially since in this example the numbers $967/1024 = 0.944\ldots$ and $270/289 = 0.934\ldots$ are close; but $967/1024$ is a statement about one parameter set, while $270/289 + o(1)$ is a statement about an infinite family of parameter sets.

This change forces calculations to be carried out symbolically, rather than through simple enumeration of all possible parameter choices. The compensating advantage is that a reader can use asymptotics to see "big" costs and "big" optimizations, such as changes in algorithm exponents, without having to worry about "small" costs and "small" optimizations, such as polynomial-factor speedups to an exponential-time algorithm. The literature contains many "small" improvements in lattice attacks, and it is useful to be able to skip those in seeing the "big" improvements—such as multi-target improvements.

2.2. Block size versus cost. The main bottleneck in the usual attacks is a BKZ-β computation. BKZ-β is a family of algorithms, not a single algorithm; its cost has dropped over the years, in part because underlying subroutines have been improved, notably for SVP-β, and in part because the use of those subroutines inside BKZ-β has been improved.

For example, Becker–Ducas–Gama–Laarhoven [21] reported heuristic costs $(3/2+o(1))^{\beta/2} = 2^{(\log_4(3/2)+o(1))\beta}$ for a non-quantum[5] algorithm to solve SVP-β. The exponent $\log_4(3/2) = 0.292\ldots$ was smaller than in previous papers.

To first order, the heuristic BKZ-β cost is the same as the heuristic SVP-β cost: e.g., $2^{(\log_4(3/2)+o(1))\beta}$ using [21]. Combining this with the standard heuristic analysis of the required β, and with the first-order asymptotics $\beta/n \in z_0 + o(1)$ from this paper where z_0 is defined above, gives an overall heuristic attack cost of $2^{(z_0 \log_4(3/2)+o(1))n}$, showing that n must be at least $\lambda/(z_0 \log_4(3/2) + o(1))$ to reach λ bits of heuristic security.

This paper computes more complicated second-order asymptotics $\beta/n \in z_0 + (z_1 + o(1))/\lg n$ so that attack improvements that change block sizes by $\Theta(n/\lg n)$ become visible in the asymptotics. Examples of such improvements are "dimensions for free" from [45], heuristically reducing block size β to

[5] The literature reports somewhat lower exponents for quantum algorithms, including improvements in 2021; see [37] and [56]. This directly produces better heuristic exponents for quantum lattice attacks. However, quantum algorithms in this context are typically dismissed as not producing enough speedup to overcome quantum overhead for realistic sizes (see, e.g., [8]) or as not being competitive with quantum speedups against AES. This paper focuses on non-quantum algorithms.

$\beta-(\lg(4/3)+o(1))\beta/\lg\beta = \beta-(0.415\ldots+o(1))\beta/\lg\beta$; the improved techniques from [44], heuristically improving $\lg(4/3)$ to $\lg(13/9) = 0.530\ldots$; and Section 4 of this paper.

Beware that the first-order asymptotics stated in the literature, such as $(3/2+o(1))^{\beta/2}$ from [21], are logically insufficient to conclude that a second-order improvement in β such as [45] produces a corresponding improvement in attack cost. To see the issue, imagine a version of BKZ-β in which the cost depends only on $x = \lceil 0.01(\log\beta)^2\rceil$. This is compatible with the cost having the form $2^{(c+o(1))\beta}$: for example, $10\sqrt{x} - \log\beta \in o(1)$, so $\exp(10\sqrt{x}) \in (1+o(1))\beta$, so $2^{0.1\exp(10\sqrt{x})} \in 2^{(0.1+o(1))\beta}$. However, decreasing β to $\beta - \beta/\lg\beta$ changes $0.01(\log\beta)^2$ by only about $0.02\log 2 = 0.0138\ldots$, and thus changes costs for under 1.4% of the values of β, perhaps missing the occasional values of β that appear, asymptotically, in a particular cryptosystem of interest.

It would be interesting to see second-order asymptotics for various algorithms for SVP-β and BKZ-β. For purposes of this paper, it suffices to observe that if costs are $2^{(c+o(1))\beta}$ then the total impact of $\Theta(\lg\beta)$ reductions in β, where each reduction replaces β with $\beta-(d+o(1))\beta/\lg\beta$, is an improvement in c. Even if this improvement is not divided evenly across the reductions in β, it is useful on average across any sufficiently large interval of β, and is thus useful on average across the broad spectrum of parameter choices considered in this paper.

2.3. Review of the LPR cryptosystem. This PKE has three parameters: an integer $n \geq 2$; an integer $q \geq 2$; and a probability distribution χ supported on a finite set of integers. Assume for simplicity that the average of χ is 0. Write R for the ring $\mathbb{Z}[x]/(x^n+1)$.

Key generation works as follows. Generate uniform random $G \in R/q$. Generate $a, e \in R$ with coefficients drawn independently at random from χ. Compute $A = aG + e \in R/q$. The secret key is (a, e). The public key is $(G, A) \in (R/q)^2$.

The set of messages is the set of elements of R with coefficients in $\{0, \lceil q/2\rceil\}$. Encryption of a message M to a public key (G, A) works as follows. Generate $b, c, d \in R$ with coefficients drawn independently at random from χ. Compute $B = Gb+d \in R/q$ and $C = M+Ab+c \in R/q$. The ciphertext is $(B, C) \in (R/q)^2$.

Decryption of a ciphertext (B, C) works as follows. Compute $X = C - aB \in R/q$. Round each coefficient of X to the closest element of $\{0, \lceil q/2\rceil\}$ in $\mathbb{Z}/q$, specifically 0 if both elements are equally close.[6]

The above PKE definition skips two requirements from the LPR paper, namely that n is a power of 2 and that q is a prime congruent to 1 modulo $2n$; see [76, Section 1.1]. Cryptosystems after [76] loosened the restrictions on q; for example, Kyber's current prime 3329 is $1 + 13 \cdot 256$. As for n, readers concerned about attacks enabled by factors of $x^n + 1$ in $\mathbb{Z}[x]$ should feel free to substitute the marginally larger polynomial $x^n - x - 1$, as in [31]; this is orthogonal to the topic of this paper.

[6] This rounding detail is not specified in [76]; also, [76] says $\lfloor q/2\rceil$ without specifying whether the rounding rounds 0.5 up or rounds to even. These details do not affect the standard analysis; they are specified here so as to have a complete PKE definition.

Correct decryption requires $X = C - aB = M + eb + c - ad$ to round to M, i.e., requires each coefficient of $eb + c - ad$ to be smaller than about $q/4$. If χ is, e.g., the uniform distribution on $\{-1, 0, 1\}$ then each coefficient of eb is a sum of n products where one expects about $4/9$ to be nonzero, evenly balanced between 1 and -1, so typically the coefficient will be on the scale of $\sqrt{n}$, with considerable variation in the exact size. There are various proposals to reduce q close to this scale, and to avoid frequent decryption failures by applying an error-correcting code to M. In the opposite direction, sometimes cryptosystems take larger χ and correspondingly larger q; sometimes cryptosystems pack more message bits into each coefficient, again taking larger q. To cover many different cases, this paper considers a spectrum of possibilities for the asymptotic sizes of q and s.

2.4. Review of the usual key-recovery attack. Consider the problem of recovering the private key $(a, e) \in R^2$ from the public key $(G, A) \in (R/q)^2$. Recovering a suffices, since $e = A - aG$ by definition. The usual "primal" attack works as follows.

There is an attack parameter $\kappa \leq n$. Define a function $\text{First}_\kappa : R \to \mathbb{Z}^\kappa$ that extracts the first κ coefficients from its input. This induces a function, also written First_κ, from R/q to $(\mathbb{Z}/q)^\kappa$.

Define L as the set of all $(\alpha, \epsilon, \beta) \in R \times \mathbb{Z}^\kappa \times \mathbb{Z}$ such that $\text{First}_\kappa(\beta A - \alpha G)$ is the same as ϵ modulo q. This is a lattice of full rank $d = n + \kappa + 1$ and determinant q^κ. Note that $\pm(a, \text{First}_\kappa(e), 1)$ are elements of this lattice.

There is another attack parameter β. The attack writes down a basis for L, applies BKZ-β to reduce this basis, and hopes that BKZ-β outputs one of the short nonzero vectors $\pm(a, \text{First}_\kappa(e), 1)$, in particular revealing a.

The problem being attacked here, the problem of finding a, e given a random G and $aG+e$, is typically called "Ring-LWE", specifically "normal-form 1-sample search Ring-LWE", where "normal form" refers to the secret a being small. The Ring-LWE problem is typically credited to [98] and [76]. However, this problem was already attacked in the 1998 Hoffstein–Pipher–Silverman NTRU paper, both for the homogeneous case $A = 0$ (see [57, Section 3.4.1]) and for general A (see [57, Section 3.4.2]). The problem statements in [98] and [76] merely generalize to more "samples": e.g., finding a, e_1, e_2 given random $G_1, G_2, aG_1 + e_1, aG_2 + e_2$, or equivalently replacing $G \in R/q$ and $e \in R$ with row vectors $(G_1\ G_2) \in (R/q)^2$ and $(e_1\ e_2) \in R^2$ respectively.

The attack in [57] has $\kappa = n$. May–Silverman [80] generalized the attack to any $\kappa \leq n$. In the original 1996 NTRU handout [58], various concrete examples chose different sizes for a and e, motivating another generalization from Coppersmith and Shamir [41] to set up a lattice with, e.g., short vector $(3.14a, e, 1)$ rather than $(a, e, 1)$; this paper focuses on cryptosystems that take a and e of the same size, such as the LPR system. There can still be a tiny improvement from setting up a lattice with, e.g., short vector $(a, e, 3.14)$; this paper ignores this improvement for simplicity.

2.5. Review of the standard analysis. Beyond asking about the cost of BKZ-β (see Section 2.2), the standard analysis of Section 2.4's attack asks whether BKZ-β succeeds at finding the target vector. The standard heuristic conclusion is that BKZ-β succeeds if and only if the 2-norm of the target vector is below $(d/\beta)^{1/2}\delta^{2\beta-d-1}q^{\kappa/d}$, where $\delta = (\beta(\pi\beta)^{1/\beta}/(2\pi\exp 1))^{1/2(\beta-1)}$. The rationale for this inequality is as follows:

- One heuristic says that, for a "random" lattice L of rank d, BKZ-β finds a nonzero vector of length close to $\delta^{d-1}(\det L)^{1/d}$, with δ defined as above.
- Another heuristic says that the Gram–Schmidt lengths of the BKZ-β output are close to a geometric series. Combining this with the shortest length being close to $\delta^{d-1}(\det L)^{1/d}$ and the product of the lengths being $\det L$ says approximately how large each length is. In particular, the length at position $d-\beta+1$ is close to $\delta^{2\beta-d-1}(\det L)^{1/d}$, which for this lattice is $\delta^{2\beta-d-1}q^{\kappa/d}$. The rationale treats this approximation as an equation.
- Another heuristic says that if the target vector has length t then its projection onto the space spanned by the last β Gram–Schmidt vectors has length approximately $t\sqrt{\beta/d}$. The rationale also treats this approximation as an equation.
- If the latter length $t\sqrt{\beta/d}$ is below the previous length $\delta^{2\beta-d-1}q^{\kappa/d}$ then the above heuristics seem to contradict each other, since the last SVP-β call in each "tour" of BKZ-β guarantees that the projection of the vector at position $d-\beta+1$ is a minimum-length nonzero vector in the projection of L. Note, however, that the first heuristic was only for a "random" lattice. Another heuristic says that this seeming contradiction occurs if and only if BKZ-β detects the non-"randomness" of the lattice by finding the projection of v.
- A further heuristic says that BKZ-β finds the projection of v if and only if BKZ-β finds v. A slightly different statement appears in [9], which says that if BKZ-β finds the projection of v then BKZ-β finds v with "high probability" for large β.

The following paragraphs use the inequality as a black box without regard to the rationale, but the rationale matters for Section 3.

For the LPR PKE, the first $n+\kappa$ entries in the target vector $(a, \mathrm{First}_\kappa(e), 1)$ are drawn independently and uniformly at random from χ. Each entry has square $\sum_i \chi_i i^2 = s^2$ on average (since χ has average 0 and standard deviation s), so the squared 2-norm of $(a, \mathrm{First}_\kappa(e), 1)$ is $(n+\kappa)s^2+1$ on average. The standard heuristic analysis treats the squared 2-norm as being exactly its average, concluding for this PKE that BKZ-β works if and only if $((n+\kappa)s^2+1)^{1/2} < (d/\beta)^{1/2}\delta^{2\beta-d-1}q^{\kappa/d}$; i.e., if and only if $\mathrm{StandardRatio}(n,q,s,\kappa,\beta) < 1$, with the notation of Definition 2.5.1.

Other PKEs do not necessarily choose (a,e) this way: consider, e.g., a PKE that chooses a as a fixed-weight ternary vector. To apply the standard analysis to such cases, the literature calculates s so that $(n+\kappa)s^2+1$ is a reasonable estimate of the squared 2-norm of $(a, \mathrm{First}_\kappa(e), 1)$, and concludes heuristically that BKZ-β works if and only if $\mathrm{StandardRatio}(n,q,s,\kappa,\beta) < 1$.

Definition 2.5.1 (the standard ratio). *Let n, q, s, κ, β be real numbers such that $2 \le n$; $2 \le q$; $0 < s$; $1 \le \kappa$; and $2 \le \beta$. Then* StandardRatio(n, q, s, κ, β) *is defined as $((n+\kappa)s^2+1)^{1/2}/(d/\beta)^{1/2}\delta^{2\beta-d-1}q^{\kappa/d}$ where $d = n+\kappa+1$ and $\delta = (\beta(\pi\beta)^{1/\beta}/(2\pi\exp 1))^{1/2(\beta-1)}$.*

2.6. A known flaw in the standard analysis. Note that δ increases with β until β reaches 36, contrary to ample evidence that, e.g., BKZ-20 usually finds shorter vectors than BKZ-10 does. As an extreme case, if one takes $\beta = 2$ (or any $\beta < 13$), then $\delta < 1$, so the first heuristic says that BKZ-β finds a nonzero vector of length exponentially below $(\det L)^{1/d}$. In fact, for most lattices, such vectors do not even exist, so certainly BKZ-β will not find them.

The standard heuristic conclusion says that, for any particular (q, s), BKZ-2 breaks LPR for all n above an easily calculated bound. Choosing (q, s), calculating that bound, and simply trying BKZ-2 shows that, no, BKZ-2 does not in fact do this. The standard patch[7] for this flaw is to simply disallow small values of β: for example, require $\beta \ge 60$.

For some of this paper's calculations, it suffices to assume $\beta \ge 2$, ensuring that the exponent $1/2(\beta-1)$ is defined. At many points in the logic, β/n is known to grow asymptotically as $Y_0 + o(1)$ for some positive real number Y_0, implying $\beta \ge 60$ for all sufficiently large n. However, Theorem 2.7.1(2) does not assume any particular asymptotic growth of β/n, and the conclusion of Theorem 2.7.1(2) would be incorrect if the hypothesis $\beta \ge 60$ were weakened to $\beta \ge 2$.

2.7. Asymptotics for the standard key-recovery attack. The following theorem pinpoints how the standard block size β grows asymptotically with n when the target cryptosystem has $\lg q \in Q_0 \lg n + Q_1 + o(1)$ and $\lg s \in S_0 \lg n + S_1 + o(1)$. See Appendix C for the proof.

Theorem 2.7.1 (asymptotic growth of the standard block size). *Let Q_0, Q_1, S_0, S_1 be real numbers such that $0 \le S_0 \le 1/2 < Q_0 - S_0$. Let N be an infinite subset of $\{2, 3, 4, 5, \ldots\}$. Let $n \mapsto q$ and $n \mapsto s$ be functions from N to $\mathbb{R}$ such that*

$$2 \le q, \qquad \lg q \in Q_0 \lg n + Q_1 + o(1),$$
$$0 < s, \qquad \lg s \in S_0 \lg n + S_1 + o(1).$$

Define $x_0 = (Q_0 + S_0 - 1/2)/(Q_0 - S_0 + 1/2)$; $z_0 = 2Q_0/(Q_0 - S_0 + 1/2)^2$; and

$$z_1 = \left(2S_1 + \lg z_0 - \left(S_0 - Q_0 + \frac{3}{2}\right)\lg\frac{z_0}{2\pi\exp 1} - \frac{Q_1(Q_0+S_0-\frac{1}{2})}{Q_0}\right)\frac{2Q_0}{(Q_0-S_0+\frac{1}{2})^3}.$$

(1) There are functions $n \mapsto \kappa$ and $n \mapsto \beta$ from N to $\mathbb{Z}$ such that

$1 \le \kappa \le n$ *for all* n, $\quad \kappa/n \in x_0 + o(1)/\lg n$,

$2 \le \beta \le n + \kappa + 1$ *for all* n, $\quad \beta/n \in z_0 + (z_1 + o(1))/\lg n$, *and*

StandardRatio$(n, q, s, \kappa, \beta) < 1$ *for all sufficiently large* n.

[7] Some parts of the literature propose other estimates for the BKZ-β behavior and give reasons to believe that these estimates are more accurate than the standard estimate for concrete values of β. Perhaps these estimates have different second-order asymptotics from the standard estimate. This paper focuses on the standard analysis.

(2) Let $n \mapsto \kappa$ and $n \mapsto \beta$ be functions from N to $\mathbb{R}$ such that

$$1 \le \kappa \le 100n \text{ for all } n,$$
$$60 \le \beta \le n + \kappa + 1 \text{ for all } n, \quad \text{and}$$
$$\text{StandardRatio}(n, q, s, \kappa, \beta) \le 1 \text{ for all sufficiently large } n.$$

Then $\beta \ge \ell$ for some function $n \mapsto \ell$ with $\ell/n \in z_0 + (z_1 + o(1))/\lg n$.

2.8. Asymptotics for the standard message-recovery attack. Consider now the problem of recovering the encryption secrets (b, d) from (G, B) where $B = Gb + d$. The standard analysis handles this exactly the same way as the key-recovery attack, except for starting with the distribution of (b, d) rather than the distribution of (a, e).

In the case of the LPR PKE, the distributions are the same, so the conclusions are the same. The attacker will then prefer to carry out the key-recovery attack since it breaks many ciphertexts; this can be understood as motivation to modify the PKE to take a larger distribution for (a, e) than for (b, d), as noted above.

The second-order asymptotics make it easy to see the effect of making errors 1 bit larger, i.e., increasing S_1 by 1: this increases z_1 by $4Q_0/(Q_0 - S_0 + 1/2)^3$, increasing the standard β by $(4Q_0/(Q_0 - S_0 + 1/2)^3 + o(1))n/\lg n$. This will be important in Section 4.

2.9. Further attacks. The standard analysis also considers the problem of recovering (b, c, d) given G, A, $B = Gb + d$, and $C = Ab + c$, i.e., from a public key and an encryption of 0. A successful recovery attack immediately gives an IND-CPA attack.

Structurally, this problem provides more "samples" to the attacker, allowing κ to be chosen as large as $2n$. However, Theorem 2.7.1(2) ends up with $\kappa/n \in x_0 + o(1)$ with $x_0 \le 1$, even if κ/n is initially allowed to be much larger than 1.

The situation would change if S_0 were allowed to be above $1/2$: the same optimizations would then produce $x_0 > 1$. For essentially the same reason, a close look at [6] finds, e.g., "Frodo-0640" listed as 2^{142} on [6, page 29] for "n LWE samples" but as only 2^{141} on [6, page 35] for "$2n$ LWE samples".[8]

The literature often reports that a "dual" attack allows marginally smaller β than the primal attack. However, [17, page 26, "Primal attack only"] argues that the dual attack is actually much slower than the primal attack. On the other hand, recent attack papers cited in Section 1 introduce new dual attacks that are claimed to be noticeably faster than primal attacks. Since single-target dual attacks are such an unstable topic at the moment, trying to optimize

[8] This should already have raised questions regarding a common notion that releasing more samples preserves security levels. This notion is made explicit in, e.g., [47, page 3], which says "we believe that in practice the MLWE problem with k samples is *no easier* than with 1 sample" (emphasis added). This notion is also implicitly assumed by an argument in the literature that GAM/LPR cryptosystems are "at least as hard" to break as NTRU; see generally [86, Sections 3.14, 5.1, and 5.2] for analysis of that argument and further references.

multi-target dual attacks would be premature. More to the point, this paper is investigating how multiple ciphertexts asymptotically affect the standard analysis; the standard analysis does not include the new dual attacks.

IND-CCA2 attacks against lattice KEMs can be much easier than the usual lattice attacks against the underlying PKEs. See, e.g., the Round2 break in [22], or the FrodoKEM attack in Appendix B. However, lattice attacks against PKEs seem to be the top threat for most lattice proposals.

3 Asymptotic heuristic cost of breaking many ciphertexts

This section analyzes what the existing heuristics say about another attack that breaks many ciphertexts. There is nothing new in the attack algorithm—in short, it plugs [46] into the algorithm briefly outlined in [48, Section 5, third paragraph], and applies this to multiple ciphertexts—but the impact on PKE security and KEM security was not clear from the literature.

Quantitatively, say cryptosystem parameters are chosen so that the usual single-target message-recovery attack reviewed in Section 2.8 has heuristic cost $2^{(1+o(1))\lambda}$. This section shows that total heuristic cost $2^{(1+o(1))\lambda}$ suffices to decrypt T ciphertexts, for any $T \leq 2^{(0.19\ldots+o(1))\lambda}$. The heuristic per-ciphertext cost is just $2^{(1+o(1))\lambda}/T$, as low as $2^{(0.80\ldots+o(1))\lambda}$. Because this is a first-order change in the heuristic per-ciphertext cost exponent as soon as $T \in 2^{\Theta(\lambda)}$, there is no need for this section to consider second-order terms in the asymptotics.

This is, at this level of asymptotic detail, an "all-for-the-price-of-one" attack heuristically breaking many ciphertexts. This is already achieved by the usual key-recovery attack from Section 2.4 *if* key generation and encryption use the same error distribution, as in the LPR PKE; but it is important to realize that modifying key generation to use a larger error distribution, so as to increase the heuristic cost of the key-recovery attack, fails to heuristically protect the ciphertexts. This section is also a stepping-stone to the one-out-of-many analysis in Section 4.

"Multi-ciphertext security degradation" in the title of this paper refers to the fact that the existing lattice heuristics claim less security against these multi-ciphertext attacks—both in the all-for-price-of-one scenario of this section, and the one-out-of-many scenario in Section 4—than they claim against the usual single-ciphertext attacks.

Note that improved attacks could easily change the single-target/multi-target cost ratio upwards or downwards. Certainly there is no proof to the contrary: for example, no proof rules out the extreme possibility of an attack breaking typical lattice-based cryptosystems in time $n^{O(1)}$, in which case attacking $2^{\Theta(n)}$ ciphertexts would certainly not produce a $2^{\Theta(n)}$ speedup.

3.1. High-level structure of multi-target lattice attacks. In 2007, Howgrave-Graham [61] introduced a "hybrid" attack algorithm against lattice problems. "Hybrid" here refers to a mixture of combinatorial searches and lattice-basis reduction. The algorithm was stated as an attack against NTRU but also applies to LWE, Ring-LWE, etc.

Howgrave-Graham's algorithm starts with a single attack target. It carries out a brute-force search for some error positions. Each guess of those error positions produces a lower-dimensional target; note that there are many of these targets. Each of these lower-dimensional targets is handled by an algorithm typically called the "nearest-plane algorithm", after a preliminary *target-independent* reduction of the relevant lattice.

The nearest-plane algorithm, also known as "weak reduction", finds lattice vectors close to a given target. It takes a lattice basis as input, and the resulting closeness depends on how well the lattice basis has been reduced. An analysis of the nearest-plane algorithm appears implicitly as part of the analysis in the LLL paper [**73**], and explicitly in a paper by Babai [**19**].

The previous two paragraphs oversimplify Howgrave-Graham's algorithm in two ways:

- [**61**, Section 4] presented a more sophisticated meet-in-the-middle search, after presenting a brute-force search ("an algorithm that enumerates all possible v") as a warmup.[9]
- [**61**, Section 7] briefly suggested "using a better CVP algorithm than Babai's closest plane algorithm (e.g. mixing Babai's CVP (which is essentially blocksize 1) with searching in higher blocksizes $2, 3, \ldots$)", while keeping the total time under control.

This paper attacks errors of any size, and follows the standard analysis of lattice security in ignoring the possibility of combinatorially searching for small errors.[10] Consequently, the distinction between a brute-force search and a meet-in-the-middle search is not relevant here. What matters is simply the portion of Howgrave-Graham's algorithm that searches for lattice vectors close to a batch of targets, after precomputation of short lattice vectors.

The question of what block size β to use—and how to handle β-dimensional close-vector computations after precomputation—remains important in this context. The nearest-plane algorithm, the case $\beta = 1$, runs in polynomial time, but will it find vectors at the necessary distance, after a given amount of effort

[9] A paper in 2010 from Lindner–Peikert [**74**] claimed a "new" lattice attack on LWE, a "decoding" attack that "combines basis reduction with an enumeration algorithm"; [**74**] does not explain any differences between this decoding attack and the warmup in [**61**].

[10] If the error width s is $\Theta(1)$ then, heuristically, hybrid attacks provide an exponential speedup, reducing the asymptotic attack exponent; even for somewhat larger s, the hybrid speedup should be visible in second-order asymptotics. This raises questions regarding the wisdom of choosing parameters based on a standard attack analysis that ignores hybrid attacks. The recent perception that dual attacks outperform primal attacks, based in particular on the speeds reported in [**78**], could be explained by the use of a brute-force search in [**78**]; it would be interesting to disentangle the primal-vs.-dual question from the non-hybrid-vs.-hybrid question, and also to understand the impact of more sophisticated combinatorial searches, such as the meet-in-the-middle search from [**61**]. Quantifying the hybrid improvement is outside the scope of this paper.

precomputing short lattice vectors? The necessary distance could be too large for the nearest-plane algorithm to find; perhaps a larger β succeeds within the same total time budget for precomputations and main computations.

In 2015, the original New Hope proposal [13, page 5] briefly considered the following attack against cryptosystems that share a lattice across many ciphertexts: "finding *once* a good enough basis of the lattice" and then compromising "*all* communications, using for example Babai's decoding algorithm". This is the case $\beta = 1$ of the same multi-ciphertext attack. This again begs the question of whether this attack is competitive with the usual single-target attacks. Each ciphertext is handled efficiently, but one also has to quantify the time spent on precomputation and analyze whether the ciphertexts are decoded successfully.

In 2020, Espitau and Kirchner [48] introduced the name "nearest-colattice algorithm" for the block-size-β generalization of the nearest-plane algorithm, wrote "we believe that this algorithm has been in the folklore for some time", and analyzed the asymptotic performance of this generalization. Specifically, [48, Theorem 3.3] states, assuming heuristics and assuming $d > 2\beta$, that the nearest-colattice algorithm finds a vector in a d-dimensional lattice L at distance $\Theta(\beta)^{d/2\beta}(\det L)^{1/d}$ from any given target. The algorithm uses a series of closest-vector computations in β-dimensional lattices derived from L; the lattices are independent of the target, so one can "use CVP algorithms after precomputations", as noted in [48, Section 3].

Espitau and Kirchner also briefly outlined a variant of this algorithm [48, Section 5, third paragraph] that, heuristically, finds an element of L when it is known that the element has distance below $(d/\beta)^{1/2}\Theta(\beta)^{1-d/2\beta}(\det L)^{1/d}$ from a given target vector, asymptotically the same inequality as in Section 2.5:

- Start by finding a "highly reduced basis" of the lattice L. Presumably this means applying BKZ-β.
- Compute a "CVP on the tail of the basis". This is just one closest-vector computation (rather than a series of closest-vector computations) for the given target, finding a vector in the β-dimensional projection of L closest to the projection of the target.
- "Finish with Babai's algorithm", hopefully recovering the closest vector in L to the target.

The rationale for claiming distance $(d/\beta)^{1/2}\Theta(\beta)^{1-d/2\beta}(\det L)^{1/d}$ is not spelled out in [48], but presumably is intended to be as follows, by analogy to the rationale reviewed in Section 2.5:

- The δ heuristic and the geometric-series heuristic imply that the shortest nonzero vector in the projection has length approximately $\delta^{2\beta-d-1}(\det L)^{1/d}$.
- Another existing heuristic says that, for "random" lattices and "random" targets, the distance to the closest vector is approximately the length of the shortest nonzero vector, hence approximately $\delta^{2\beta-d-1}(\det L)^{1/d}$.
- If the target is at distance t from L then its projection is, heuristically, at distance approximately $t\sqrt{\beta/d}$ from the β-dimensional projection of L.

- The approximations are again treated as equations, giving a contradiction if $t\sqrt{\beta/d} < \delta^{2\beta-d-1}(\det L)^{1/d}$. Heuristically, this contradiction occurs if and only if the computation finds the target.
- Finally, δ is $\Theta(\beta)^{1/2\beta}$ as $\beta \to \infty$, so $\delta^{2\beta-d-1}$ is $\Theta(\beta)^{1-d/2\beta}$.

The $\sqrt{\beta/d}$ factor is claimed in [48] to not be "significant" under the assumption $d > 2\beta$, and is thus suppressed in [48]. However, if $d = 2\beta + 1$, then $\Theta(\beta)^{1-d/2\beta}$ is $\Theta(\beta)^{-1/2\beta}$, which is in $1 + o(1)$, while $\sqrt{\beta/d} = \sqrt{1/2} + o(1)$. To derive the asymptotics stated in [48] from the existing heuristics, one needs to make a slightly stronger assumption, such as $d > 2.01\beta$, or modify the asymptotics to include the $\sqrt{\beta/d}$ factor. When the $\sqrt{\beta/d}$ factor is included, there is nothing in the existing heuristic analysis of this algorithm that relies on the assumption $d > 2\beta$, so this paper does not make that assumption.[11]

If the closest-vs.-shortest ratio is heuristically assumed to be $1+o(1)$, as in [48, Section 3.3, last sentence], then this rationale says that the algorithm works when the distance is below $(1+o(1))(d/\beta)^{1/2}\delta^{2\beta-d-1}(\det L)^{1/d}$. This is within a factor $1 + o(1)$ of the inequality in Theorem 2.7.1. The second-order asymptotics there generalize immediately to this situation, since $(\lg(1 + o(1)))/\lg n$ is $o(1)/\lg n$. For deriving first-order asymptotics, the weaker $\Theta(\beta)$ statement in [48] suffices.

3.2. CVP after precomputation. The next step in the analysis is to see how precomputation changes the exponent of CVP algorithms, algorithms to find *closest* vectors, since these algorithms are used as subroutines above.

The basic idea used in the literature is to precompute a database of all short nonzero vectors in L, and then apply the following greedy reduction algorithm:[12] start from a target vector t; repeatedly replace t with the shortest vector having the form $t - ju$ where u is a database entry and j is an integer; stop when no $t - ju$ is shorter than t. The algorithm output is the sum of vectors ju that were used, i.e., the original t minus the final t. This is a lattice vector close to t, hopefully the closest vector.

Sommer, Feder, and Shalvi [96] showed that a database of $2^{d+1} - 2$ vectors suffices for perfect reduction. Laarhoven [72] argued heuristically that $2^{(1/2+o(1))d}$ vectors suffice. The literature presents two important ways to further reduce the heuristic cost exponent:

- Use a smaller database, i.e., precompute fewer short vectors in L. This reduces the heuristic probability of finding the closest vector, but hopefully saves more time. The literature then amplifies the heuristic probability close to 1 by repeating the algorithm many times, each time adding a random lattice vector to t.

[11] The optimal d/β in Theorem 2.7.1 is $Q_0 - S_0 + 1/2 + o(1)$. Readers who wish to restrict attention to d/β being asymptotically above 2, so as to be able to more directly apply [48], can restrict attention to parameter choices with $Q_0 - S_0 + 1/2 > 2$.

[12] Often this greedy algorithm, the "iterative slicer", is credited to [96]. However, the algorithm had already appeared in a 2000 textbook by Cohen [40, pages 375–376]. See [32, Appendix D.3] for the relevant quotes.

- Hash the database into buckets using a "locality-sensitive hash function", and search only buckets close to the hash of t. This again trades heuristic probability for time. The literature then amplifies the heuristic probability close to 1 by trying many hash functions.

The latter "near-neighbor" techniques are also used in, e.g., the aforementioned paper [21] to heuristically save time in building the database of short vectors in the first place.

Ducas–Laarhoven–van Woerden [46], heuristically improving on results from Laarhoven [72] and Doulgerakis–Laarhoven–de Weger [43], concluded[13] that one can "heuristically solve CVPP instances in $2^{0.234d+o(d)}$ amortized time, for batches of size at least $2^{0.058d+o(d)}$". Here $(0.234, 0.058)$ actually means an approximation to a pair of real numbers with sum $\log_4(3/2)$. Formulas for the real numbers can in principle be calculated from [46]; this paper takes 0.058 as axiomatic and does not review the calculations.

The minimum batch size comes from the fact that there is an initial heuristic cost of $(3/2 + o(1))^{d/2} = 2^{(0.292\ldots+o(1))d}$ for computing the database of all short vectors. The algorithm also works for smaller batches, but then the total heuristic cost is dominated by this initial heuristic cost, so the per-target heuristic cost is this initial heuristic cost divided by the number of targets.

3.3. Breaking many ciphertexts. The results cited above are used essentially as black boxes in the following analysis of the heuristic cost of a multi-ciphertext attack that breaks *all* of the ciphertexts.

By definition, the jth ciphertext is $(B_j, C_j) = (Gb_j + d_j, M_j + Ab_j + c_j)$; all of b_j, c_j, d_j are small, M_j is the jth message, and the public key reveals G and A. The attack will use G and B_j to find b_j, then subtract Ab_j from C_j to find $M_j + c_j$, then round to find M_j as desired. The critical step is finding b_j, which works as follows.

As in Section 2.4, there is an attack parameter κ, and a map $\mathrm{First}_\kappa : R \to \mathbb{Z}^\kappa$ that selects the first κ coefficients of its input, inducing a map $R/q \to (\mathbb{Z}/q)^\kappa$.

Lift $\mathrm{First}_\kappa(B_j) \in (\mathbb{Z}/q)^\kappa$ to $\mathbb{Z}^\kappa$: i.e., choose a vector $v_j \in \mathbb{Z}^\kappa$ such that v_j is the same modulo q as $\mathrm{First}_\kappa(B_j)$, i.e., as $\mathrm{First}_\kappa(Gb_j) + \mathrm{First}_\kappa(d_j)$.

Define L as the set of all $(\alpha, \epsilon) \in R \times \mathbb{Z}^\kappa$ such that $\mathrm{First}_\kappa(G\alpha)$ is the same as ϵ modulo q. This is a lattice of full rank $d = n + \kappa$ and determinant q^κ.

This lattice contains a vector close to $(0, v_j)$, namely $(b_j, v_j - \mathrm{First}_\kappa(d_j))$, since $\mathrm{First}_\kappa(Gb_j)$ is the same as $v_j - \mathrm{First}_\kappa(d_j)$ modulo q. To quantify "close":

[13] The same paper [46, Section 7] also considered the special case of "bounded-distance decoding" (BDD), the case that the target is guaranteed to have a particular closeness to the lattice, as in the problem of decrypting a ciphertext. The paper found that the algorithm obtained only limited benefit from this guarantee as the distance bound decreases, and said "An open problem would be to adapt the iterative slicer to make better use of this guarantee". One solution to this problem is as follows: use the slicer inside the CVP algorithm from [46], plug the CVP algorithm into the algorithm from [48, Section 5, third paragraph], and then allow the block size in that algorithm to drop in the usual way with the distance bound.

the difference is $(b_j, -\mathrm{First}_\kappa(d_j))$, a vector consisting of coefficients that were generated independently at random from χ.

Now use the algorithm from [48, Section 5, third paragraph], with the CVP subroutine from [46], to try to find this lattice vector:

- As precomputation for [48], apply BKZ-β to a basis for L. This costs, heuristically, $(3/2 + o(1))^{\beta/2}$. This work is shared across all j.
- Apply the precomputation step from [46] to the β-dimensional projection of L, computing a database of all short vectors in that projection. (Some versions of BKZ-β produce this database as a side effect.) This also has heuristic cost $(3/2 + o(1))^{\beta/2}$, again shared across all j.
- For each j, apply the main step from [46] to find a vector in the projection of L close—hopefully closest—to the projection of $(0, v_j)$. The heuristic analysis of [46] says that this costs just $2^{(0.234+o(1))\beta}$; this is not a bottleneck if there are fewer than $2^{(0.058+o(1))\beta}$ targets.
- "Finish with Babai's algorithm" as in [48], hopefully finding the desired $(b_j, v_j - \mathrm{First}_\kappa(d_j))$.

The existing heuristics say that this works if the difference $(b_j, -\mathrm{First}_\kappa(d_j))$ has length below $(d/\beta)^{1/2}\Theta(\beta)^{1-d/2\beta}(\det L)^{1/d}$; again, this is already stated in [48], modulo the issues noted above regarding d/β.

At the level of detail of first-order asymptotics, this is the same inequality as in Section 2.8, so the smallest β that heuristically works here is again $(2Q_0/(Q_0 - S_0 + 1/2)^2 + o(1))n$. The heuristic cost is the same $(3/2+o(1))^{\beta/2} = 2^{((\log_2(3/2))Q_0/(Q_0-S_0+1/2)^2+o(1))n}$, if there are at most $2^{(0.058+o(1))\beta}$ targets. The critical difference is that this attack heuristically breaks *all* of the targets for this total cost, whereas Section 2.8 was for just one target.

In particular, if n is chosen so that the usual single-target message-recovery attack has heuristic cost $2^{(1+o(1))\lambda}$, then this attack breaks T targets, for any $T \le 2^{(0.058+o(1))\beta} = 2^{(0.19\ldots+o(1))\lambda}$, with total heuristic cost $2^{(1+o(1))\lambda}$.

4 Asymptotic heuristic cost of breaking one out of many ciphertexts

This section analyzes what the existing heuristics say about an attack that decrypts one out of many ciphertexts.[14] The conclusion, in short, is that one can heuristically reduce the block size β from Section 3.3 by $\Theta(n/\lg n)$ for breaking one out of T ciphertexts, provided that $\lg T \in \Theta(n)$.

The Θ constant in $\Theta(n/\lg n)$ depends on how many ciphertexts are being attacked and, more subtly, on details of how the error distribution χ is chosen. As a numerical example, consider the following conservative choices:

[14] A decoding-one-out-of-many attack against code-based cryptography was dubbed a "DOOM" attack in [93], but using the title "Lattice DOOM" for this paper would be unnecessarily theatrical.

- Choose χ as the binomial distribution on $\{-16,\ldots,16\}$, exactly as in the original New Hope paper. Then $S_0 = 0$ and $S_1 = 3/2$.
- Since the literature says that smaller moduli are safer, and since any Q_0 above $1/2$ together with error correction suffices to asymptotically eliminate decryption failures (given that $S_0 = 0$), choose $Q_0 = 0.501$.

One can then heuristically reduce the block size β by $(0.60537\ldots + o(1))\beta/\lg\beta$ for breaking one out of $2^{(0.0574+o(1))\lg\beta}$ ciphertexts. This number of ciphertexts fits inside the per-ciphertext budget from Section 3, since $0.0574 < 0.058$. The calculation of $0.60537\ldots$ is explained in detail below.

This is a larger heuristic improvement than the heuristic "dimensions for free" improvements reported in [45] and [44] for SVP-β. Consequently, the standard message-recovery attack *plus* the improvements from [45] and [44] is, for this conservative (χ, Q_0), heuristically outperformed by this one-out-of-many attack.

Qualitatively, compared to the security consequences of an attack decrypting many ciphertexts, there are different, often stronger, security consequences of a faster attack decrypting one out of many ciphertexts. For example, consider what the attacks say regarding security levels:

- Section 3 illustrates the danger of the "attacker economist" philosophy, choosing the security level λ so that 2^λ is slightly beyond the attacker's benefit of breaking *one* ciphertext.
- This section illustrates the danger of choosing the security level λ so that 2^λ is slightly beyond a feasible computation.

As another example of the consequences, consider again the statement that "$\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}} \approx \mathrm{Adv}^{(n,q_C)\text{-IND-CPA}}_{\text{PKE}}$". If the time limit on attacks is too low for a single-target attack but is high enough for a faster one-out-of-q_C attack, then the first Adv is essentially 0 and the second Adv is far above 0 (where exact numbers depend on probabilistic aspects of the attacks), contradicting the statement. For comparison, an attack breaking *all* targets is at least as expensive as a single-target attack, so it cannot similarly contradict the same statement.

4.1. Designing one-out-of-many algorithms. There is a large overlap from the algorithm designer's perspective between solving many problems and solving one of many problems, despite the differences in security consequences.

In particular, starting from an algorithm to solve many problems—especially an algorithm that performs a shared precomputation and then handles each problem separately—one can usually save time by simply reducing algorithm parameters. There is then nothing new to design; what matters is the algorithm analysis. Often it turns out that the reduced algorithm continues to successfully break one or more of the problems, because of randomness in the problems, randomness in the algorithm, or both. In the simple AES-key-guessing example from Section 1, guessing only 2^{88} keys has a good chance (more precisely, probability close to $1 - \exp(-1) = 0.632\ldots$) of colliding with one of the 2^{40} keys chosen at random by users.

In the T-target attack from Section 3.3, reducing the block size β correspondingly reduces the heuristic cost of the initial BKZ-β computation and

the precomputation of a database of short vectors in a β-dimensional lattice; under a reasonable assumption on T, other steps are not heuristic bottlenecks. The critical question is then how far one can reduce β so that the resulting algorithm, when applied to one target, still has success probability at least $1/T$.

There are large gaps in the literature at this point. It is clear from small-scale experiments that there are probabilistic effects in lattice attacks;[15] however, for many of the necessary subroutines, the heuristic analyses in the literature consider only the time taken for high success probability, rather than the whole time-probability tradeoff curve. This is a defensible simplification, but it poses an obstacle for the analysis of multi-target attacks.[16] It seems necessary to revisit a large part of the lattice-attack literature, systematically filling in every missing probability analysis and building new heuristics accordingly.

Fortunately, the standard analysis has one step with two features that are important for this paper:

- A probabilistic analysis of that step can be rigorously carried out as a separate module. This is the main technical content of this section.
- Plugging the results for probability $1/2^{(c+o(1))n}$ into the rest of the standard analysis obtains a block size $\Theta(n/\lg n)$ smaller than before: more precisely, $(d+o(1))n/\lg n$ smaller, with d quantified below.

This makes no changes in the existing heuristics. The analysis here applies the existing heuristics to a modified PKE; each ciphertext in the original PKE has a noticeable chance, rigorously quantified below, of matching a ciphertext in the modified PKE.

4.2. Computing 2-norm distributions from error distributions. Take any probability distribution χ supported on a finite set of integers such that 0 and 1 each have nonzero probability and the average of χ is 0. Let n be a positive integer. Let v be an element of $\mathbb{Z}^n$ with each entry drawn independently at random from χ.

The goal here is to analyze the chance that v has a particularly small 2-norm. Section 4.4 will replace n with $n+\kappa$ to see the chance of a ciphertext being particularly weak in the context of attacks against the LPR PKE.

Define g as the polynomial $\sum_{i\in\mathbb{Z}} \chi_i z^{i^2} \in \mathbb{R}[z]$. Then, for each j, the generating function of v_j^2 is g. This means that the chance of $v_j^2 = \ell$ is the coefficient of z^ℓ in g. See [**101**] for an introduction to generating functions.

One of the useful properties of generating functions is that adding independent random variables corresponds to multiplying generating functions. In particular,

[15] Consider, e.g., the first batch of experiments in [**9**, Table 1], reporting that probabilities dropped from 93.3% to 52.8% to 4.8% as β dropped from the cutoff mandated by the standard analysis to 5 or 10 lower.

[16] Even for single-target attacks, understanding the cost of lower-probability attacks is important. Presumably many users would be unhappy to hear that a feasible attack succeeds with probability 10%. Furthermore, attacks with much lower probability, such as 2^{-64}, arise in the context of applying loose theorems.

the generating function of $\sum_j v_j^2$ is g^n: the chance of $\sum_j v_j^2 = \ell$ is the coefficient of z^ℓ in g^n.

Fix a positive rational number y below the degree of g. If $yn \in \mathbb{Z}$ then the coefficient of z^{yn} in g^n is $(c+o(1))n^{-1/2}(g(\rho)/\rho^y)^n$; here ρ is the unique positive real number satisfying $\rho g'(\rho) = yg(\rho)$, and

$$c = \left(2\pi\left(\frac{g(\rho)g''(\rho) - g'(\rho)^2}{g(\rho)^2} + \frac{y}{\rho^2}\right)\right)^{-1/2}.$$

The previous sentence is an example of [49, Proposition VIII.8], which in turn is an application of the saddle-point method in analytic combinatorics. For the asymptotics in this paper, the important factor is $(g(\rho)/\rho^y)^n$; but the extra precision of $(c+o(1))n^{-1/2}(g(\rho)/\rho^y)^n$ is used in the sanity check described below.

Take, for example, χ as the binomial distribution on $\{-16,\ldots,16\}$, and take $y = 8$. Then $g(1) = 1$, $g'(1) = 8$, and $g''(1) = 180$; $\rho = 1$ is a root of $\rho g'(\rho) = yg(\rho)$, the unique positive root; $g(\rho)/\rho^y = 1$; and $c = (248\pi)^{-1/2} = 0.035826\ldots$. The coefficient of z^{8n} in g^n, the chance of $\sum_j v_j^2$ being exactly its average value, is $((248\pi)^{-1/2} + o(1))n^{-1/2}$.

The script in Figure 4.2.1 uses the Sage [92] mathematics system to carry out the same computation for the same χ but with $y = 21/4 = 5.25$. The script uses Sage's built-in interval arithmetic to track rounding errors, printing question marks for unknown digits. The output of the script indicates that $\rho = 0.96713\ldots$, $g(\rho)/\rho^y = 2^{-0.057241\ldots}$, and $c = 0.054093\ldots$. The coefficient of z^{yn} in g^n is thus $2^{(-0.057241\ldots+o(1))n}$ if yn is an integer.

For comparison, naively modeling the distribution of the squared 2-norm of v as a normal distribution with average $8n$ and variance $124n$ would say that squared 2-norm $\leq(21/4)n$ appears with probability $(1-\operatorname{erf}((11/4)\sqrt{n/248}))/2 \in \exp((-(11/4)^2/248 + o(1))n) = 2^{(-(11/4)^2/(248\log 2)+o(1))n} = 2^{(-0.043993\ldots+o(1))n}$, overestimating the actual $2^{(-0.057241\ldots+o(1))n}$ chance by an exponential factor.

As a sanity check, the script also runs through various small n where yn is an integer, printing out tuples $(n, yn, (g^n)_{8n}, (g^n)_{8n}/cn^{-1/2}(g(\rho)/\rho^y)^n$. The ratio here, the last tuple element, is $1 + o(1)$ according to the proposition cited above, meaning that *if* the script is performing the right computations then the printed ratios are also $1 + o(1)$. Concrete examples of the printed ratios include $0.87716\ldots$ for $n = 8$; $0.99683\ldots$ for $n = 16$; $0.99442\ldots$ for $n = 32$; $0.99718\ldots$ for $n = 64$; $0.99859\ldots$ for $n = 128$; $0.99929\ldots$ for $n = 256$; and $0.99964\ldots$ for $n = 512$. A proper test, rather than just a sanity check, would require explicit bounds on the $o(1)$ in $c + o(1)$.

Changing y to $5511515/1048576 = 5.2561\ldots$ in the script produces $\rho = 0.96724\ldots$, $g(\rho)/\rho^y = 2^{-0.056943\ldots}$, and $c = 0.054030\ldots$; unsurprisingly, these are close to the results for $y = 5.25$. For this value of y, the coefficient of z^{yn} in g^n is $2^{(-0.056943\ldots+o(1))n}$ if yn is an integer. The script skips the sanity check in this case given the size of 1048576.

What follows is a simpler example that can be checked by hand for variable y—assuming Stirling's well-known asymptotic formula for factorials. Stirling's

```
QQx.<x> = QQ[]

s2 = 16
B = (1+x)^(2*s2)/2^(2*s2)
g = sum(B[i+s2]*x^(i^2) for i in range(-s2,s2+1))

y = 21/4

assert all(gi >= 0 for gi in list(g))
assert g[0] > 0
assert g[1] > 0
assert y in QQ
assert y > 0
assert y < g.degree()

g1 = g.diff()
g2 = g1.diff()
foundrho = False
for rho in (x*g1-y*g).roots(RIF,multiplicities=False):
  if rho > 0:
    foundrho = True
    break
assert foundrho
base = g(rho)/rho^y
xi = (g(rho)*g2(rho)-g1(rho)^2)/g(rho)^2+y/rho^2
c = 1/(rho*sqrt(2*RIF(pi)*xi))

print('y',y)
print('rho',rho)
print('base',base)
print('lgbase',log(base,2.0))
print('lgnaive',-RIF((g1(1)-y)^2/(2*(g2(1)+g1(1)-g1(1)^2)*log(2))))
print('c',c)
sys.stdout.flush()

D = QQ(y).denominator()
if D < 1024:
  g = g.change_ring(RIF)
  gD = g^D
  n,gn = D,gD
  while n < 1024:
    est = c*base^n/sqrt(RIF(n))
    print('n',n,'yn',y*n,'coeff',gn[y*n],'coeff/est',gn[y*n]/est)
    sys.stdout.flush()
    n,gn = n+D,gn*gD
```

Fig. 4.2.1. Computing the asymptotic probability of a specific 2-norm of an error vector; and computing the exact probability for various n as a sanity check. See text for details.

formula can be proven as an example of the saddle-point method, but can also be proven more directly as in [91].

Take χ as the uniform distribution on $\{-1,0,1\}$, and take any rational number y with $0<y<1$. Then $g=(1+2z)/3$. The equation $\rho g'(\rho)=yg(\rho)$ is $\rho(2/3)=y(1+2\rho)/3$, i.e., $2(1-y)\rho=y$, which has a unique positive root $\rho=y/2(1-y)$. Here $g(\rho)/\rho^y=((1+y/(1-y))/3)/(y/2(1-y))^y=2^y/3y^y(1-y)^{1-y}$, so the coefficient of z^{yn} in g^n is $(2^y/3y^y(1-y)^{1-y})^n/n^{1/2+o(1)}$ when yn is an integer.

The check, given that this g is linear, is to use the binomial formula for g^n to see that the coefficient of z^i in g^n is $\binom{n}{i}2^i/3^n$. Equivalently, observe that there are exactly $\binom{n}{i}2^i$ ways for a vector $v\in\{-1,0,1\}^n$ to have i nonzero positions, i.e., squared 2-norm i. By Stirling's formula, the chance $\binom{n}{i}2^i/3^n$ is $(2^y/3y^y(1-y)^{1-y})^n/n^{1/2+o(1)}$ where $y=i/n$.

4.3. A modified cryptosystem. Fix a probability distribution χ. Assume as above that χ is supported on a finite set of integers, that 0 and 1 each have nonzero probability, and that the average of χ is 0. Fix a rational number y with $0<y<s^2$, where s is the standard deviation of χ. Also fix a rational number $Q_0>1/2$, and define $x_0=(Q_0-1/2)/(Q_0+1/2)$; then $0<x_0<1$.

Consider a cryptosystem SqueezedLPR that is identical to the LPR PKE with distribution χ except for two restrictions (to simplify the analysis) and one critical change. The two restrictions are that x_0n and $y(1+x_0)n$ are integers. The change is that encryption rejects (b,d) and starts over[17] unless the sum of the squares of the first $(1+x_0)n$ components of the $2n$-component vector (b,d) is exactly $y(1+x_0)n$. (Non-rejected vectors exist for all sufficiently large n.)

Now apply the standard heuristic analysis to SqueezedLPR, specifically the analysis of the usual single-target message-recovery attack, specifically with the attack parameter κ chosen as x_0n.

For LPR, the squared 2-norm of the target vector $(b,\mathrm{First}_\kappa(d),1)$ was estimated as $(n+\kappa)s^2+1$. For SqueezedLPR, the squared 2-norm of the target vector $(b,\mathrm{First}_\kappa(d),1)$ is exactly $y(1+x_0)n+1=(n+\kappa)y+1$, in effect replacing s^2 with y. The standard analysis proceeds identically aside from this difference.

Now fix Q_1, and fix a function $n\mapsto q$ with $q\in\mathbb{Z}$, $q\geq 2$, and $\lg q\in Q_0\lg n+Q_1+o(1)$. Asymptotically, SqueezedLPR is just like LPR with cryptosystem parameters (n,q,χ) in that the asymptotics of the standard choice of β are covered by Theorem 2.7.1, specifically with $S_0=0$ since the error distribution is independent of n. The only difference is that S_1 drops from $\lg s$ to $(\lg y)/2$.

The optimized attack parameters in Theorem 2.7.1 have $z_0=2Q_0/(Q_0+1/2)^2$ and $z_1=(2S_1+\cdots)2Q_0/(Q_0+1/2)^3$. The reduction of S_1 by $\lg s-(\lg y)/2$ reduces z_1 by $(2\lg s-\lg y)2Q_0/(Q_0+1/2)^3$, and thus reduces β by

$$\frac{((2\lg s-\lg y)2Q_0/(Q_0+1/2)^3+o(1))n}{\lg n}=\frac{((2\lg s-\lg y)/(Q_0+1/2)+o(1))\beta}{\lg\beta};$$

[17] There are much more efficient ways to sample the resulting distribution, but the speed of encryption has no relevance to the standard attack analysis.

the equation comes from $\beta \in (z_0 + o(1))n$ and $2Q_0/(Q_0+1/2)^3 = z_0/(Q_0+1/2)$. See Section 5 for a sanity check on this formula for the change in β.

For the multi-ciphertext attack in Section 3.3, the existing heuristics say that the requirement on β is the same inequality—aside from a $1+o(1)$ factor, which has no effect on the first-order or second-order asymptotics. Again this change of cryptosystem reduces the block size β by $((2\lg s - \lg y)/(Q_0+1/2)+o(1))\beta/\lg\beta$.

4.4. Attacks against the modified cryptosystem as attacks against the original cryptosystem. Consider again LPR with distribution χ, and assume that $x_0 n$ and $y(1+x_0)n$ are integers.

An LPR ciphertext has a chance of being a SqueezedLPR ciphertext; this chance is quantified in Section 4.2. In a large enough collection of LPR ciphertexts, one expects to find some SqueezedLPR ciphertexts. The standard attack analysis, applied to SqueezedLPR, says, under the existing heuristics, that a smaller block size β for the multi-ciphertext attack breaks those SqueezedLPR ciphertexts; this is quantified in Section 4.3. Putting this together gives a one-out-of-many-ciphertexts attack against LPR, heuristically faster than the original multi-ciphertext attack.

Here is a numerical example of the asymptotic impact:

- Take $Q_0 = 0.501$. Then $x_0 = 1/1001$ and $z_0 = 1002000/1002001$.
- Take χ as the binomial distribution on $\{-16,\ldots,16\}$. This has standard deviation $s = 2^{3/2}$.
- Take $y = 5511515/1048576$. Restrict attention to n such that $(1+x_0)n$ and $y(1+x_0)n$ are integers.
- The chance of the first $(1+x_0)n$ entries of (b,d) having squared 2-norm exactly $y(1+x_0)n$ is $2^{(-0.056943\ldots+o(1))(1+x_0)n} = 2^{(-0.056999\ldots+o(1))n} = 2^{(-0.056999\ldots+o(1))\beta}$. This is simply replacing n with $(1+x_0)n$ in one of the examples from Section 4.2, and then rewriting n as $(1/z_0+o(1))\beta$.
- In a collection of, say, $2^{(0.0574+o(1))\beta}$ LPR ciphertexts, there will be, with probability $1-o(1)$, at least one ciphertext meeting this condition—in other words, a SqueezedLPR ciphertext.[18] This is also a small enough number of ciphertexts to not be a heuristic bottleneck in the attack analyzed in Section 3, since 0.0574 is safely below 0.058.
- Finally, $(2\lg s - \lg y)/(Q_0+1/2)$ is $0.60537\ldots$. The analysis from Section 4.3 says that reducing β by $(0.60537\ldots + o(1))\beta/\lg\beta$ suffices to break the SqueezedLPR ciphertexts, according to the existing heuristics.

The gap between $0.056999\ldots$ and the 0.058 from [46] means that $0.60537\ldots$ can be increased, but finding the exact limit would require a tedious recalculation of the 0.058. The choice of 0.0574 here is designed to protect this paper against the possibility that 0.058 actually means something as small as 0.0575. As a separate matter, readers who prefer n to be a power of 2 can instead take $Q_0 = 1025/2046$ with $x_0 = 1/1024$; this marginally changes the other numbers listed above.

[18] In fact, one expects an exponential number of SqueezedLPR ciphertexts, so this is a some-out-of-many attack. More precisely, one expects to find more than $2^{(0.0004+o(1))\beta}$ SqueezedLPR ciphertexts.

The possibility of distance variations being important for security was already pointed out in the NTRU Prime proposal [31, page 41] in 2019: "The analysis does not take into account the variations in the 2-norm of g; this simplification could overestimate or underestimate security". However, that was not the same as pointing out the possibility of exploiting these variations for an efficient multi-ciphertext attack, let alone analyzing the resulting impact. See also the 2021 survey [7, page 13], which states the usual "expected norm $\sqrt{(n+m)\cdot\sigma^2+t^2}$" without mentioning the possibility of vectors being smaller and easier to find than this formula indicates.

4.5. Countermeasures. Switching from the binomial distribution on $\{-16,\ldots,16\}$ to the binomial distribution on $\{-8,\ldots,8\}$, as in newer versions of New Hope, and adjusting y appropriately, reduces the above $0.60537\ldots$ to $0.59739\ldots$. Switching from 8 to 3, one of the distributions used in the latest version of Kyber, produces $0.56984\ldots$. Switching from 3 to 2, another distribution used in the latest version of Kyber,[19] produces $0.54668\ldots$. These numbers were calculated by the Sage script in Figure 4.5.1.

More interestingly, switching to the uniform distribution on $\{-1,0,1\}$ produces just $0.32860\ldots$, smaller than the constants from [45] and [44]. Compared to New Hope, this distribution has much smaller squared 2-norms on average, but considerably less variation *relative to the average*: i.e., the variation in heuristic security from one ciphertext to another is lower.

There is even less 2-norm variation in cryptosystem proposals that take, e.g., fixed-weight ternary vectors b and d. More generally, starting from cryptosystems with any χ, one can modify vector generation to efficiently force a narrow range of 2-norms (e.g., a specific 2-norm), limiting the size variations exploited in this section. There is still *some* variation: $\mathrm{First}_\kappa(d)$ selects some positions from d, and the attacker can vary which positions are taken.

Ternary distributions with a prespecified number of 1 entries and a prespecified number of -1 entries were already highlighted in the 1998 NTRU proposal [57] as an attractive choice. Subsequent analysis consistently indicated that proposed wider-than-ternary distributions damaged security against known attacks for any given key size, as noted in, e.g., [31, Section 4.6].

New Hope's use of a wide close-to-normal distribution on $\{-16,\ldots,16\}$ can be traced to the emphasis upon "worst-case-to-average-case reductions" for Ring-LWE in [76]. On the other hand, New Hope is still not large enough for the worst-case-to-average-case reductions to apply; see generally [70]. Descendants of New Hope typically returned to smaller distributions, putting more emphasis

[19] A full analysis of Kyber would also have to account for the rounding in Kyber ciphertexts, which complicates single-target and multi-target attacks. This rounding is mostly in the C component but also a little in the B component, slightly increasing the effective size of d. The latest Kyber documentation claims that the rounding in the latest version of Kyber-512 gains several bits of security for attacks against the ciphertexts. The concrete question—not addressed by this paper's asymptotics; see Section 1.3—is then the extent to which security is damaged by variations in the effective size of (b,d).

```
batchtarget = 0.057 # safely below 0.058
QQz.<z> = QQ[]

def reldrop(g,Q0):
  W0 = (Q0-1/2)/(Q0+1/2)
  Z0 = 2*Q0/(Q0+1/2)^2
  g = QQz(g)
  assert all(gi >= 0 for gi in list(g))
  assert g[0] > 0 and g[1] > 0 and g(1) == 1
  g1 = g.diff()
  g2 = g1.diff()
  V = g1(1)

  def rhoy(y):
    return max((z*g1-y*g).roots(ring=RR,multiplicities=False))

  def grhorhoy(y):
    rho = rhoy(y)
    assert rho > 0
    return g(rho)/rho^y

  def yevaluation(y):
    # require y(n+kappa) = sum of squares of n+kappa positions
    lgbasenk = -log(grhorhoy(y))/log(2.0)
    # probability is 2^((-lgbasenk+o(1))*(n+kappa))
    lgbasen = (1+W0)*lgbasenk # n+kappa is (1+W0+o(1))n
    # probability is 2^((-lgbasen+o(1))*n)
    lgbasebeta = lgbasen/Z0 # beta is (Z0+o(1))n
    # probability is 2^((-lgbasebeta+o(1))*beta)
    return lgbasebeta-batchtarget

  y = find_root(yevaluation,0.001,V)
  y = RR(ceil(y*1048576)/1048576)
  return log(V/y,2.0)*Z0^2/(1+W0)

for Q0 in 0.501,1:
  for c in 16,8,3,2:
    g = sum(binomial(2*c,j)*z^((j-c)^2)
            for j in range(2*c+1))/4^(c)
    print('binomial -%d,...,%d Q0 %.8f reldrop %.8f'
          %(c,c,Q0,reldrop(g,Q0)))

  g = (1+2*z)/3
  print('uniform -1,0,1 Q0 %.8f reldrop %.8f'%(Q0,reldrop(g,Q0)))
```

Fig. 4.5.1. A script to calculate the asymptotic drops of the heuristically required β, relative to $\beta/\lg\beta$, for various χ and $Q_0 \in \{0.501, 1\}$, with $S_0 = 0$, with batch size fitting into at most $2^{(0.057+o(1))\beta}$ ciphertexts.

on security against known attacks—but often avoiding ternary distributions, for example because of a common misimpression that hybrid attacks do not apply to wider-than-ternary distributions.

Changing error distributions is not the only way to reduce the frequency of particularly weak ciphertexts. Increasing Q_0, while leaving the vector distribution unchanged, generally reduces the variations that one can expect to see within $2^{(0.058+o(1))\beta}$ ciphertexts. Some cryptosystem proposals already choose larger q (and correspondingly larger n to reach the same heuristic security level), asymptotically with $Q_0 = 1$, for another reason: namely, to provide proofs that there are no decryption failures.

Instead of, or as a supplement to, trying to reduce variations, one can modify the standard analysis to account for those variations, increasing parameter sizes to protect the weakest ciphertexts. The obvious approach is as follows: decide a limit on the number of ciphertexts, decide a limit on the acceptable success probability of a multi-ciphertext attack, use generating-function techniques to calculate the smallest 2-norm that will be found with that probability among that number of ciphertexts, and use this 2-norm in place of an "expected" 2-norm.

5 A sanity check on block sizes

The critical formula from Section 4.3 says that the standard choice of β drops by $((2 \lg s - \lg y)/(Q_0 + 1/2) + o(1))\beta/\lg \beta$ when the variance drops from s^2 to y. This formula relies on Theorem 2.7.1, which in turn relies on a chain of asymptotic calculations.

As a sanity check on these calculations (analogous to the sanity check in Section 4.2 on the calculations of how often the squared 2-norm is yn), this section asks what the same drop in variance does to concrete values of β.

5.1. Using an existing estimator. The obvious way to see the change in β for concrete sizes is to use an estimator already in the literature that calculates β for concrete (n, q, s).

Specifically, consider [6], which reported security estimates for a wide range of NIST submissions, specifically by plugging a variety of estimates for the cost of BKZ-β into an estimator using the standard heuristic analysis. It is straightforward to run the same estimator with different choices of variance to see how the variance affects the choice of β for concrete examples of (n, q).

Take, in particular, $n \in \{2^5, 2^{10}, 2^{15}, 2^{20}\}$ and $q \in \{\lfloor 64n^{0.501} \rfloor, 64n\}$. These are realistic sizes of q for $n = 1024$:

- FrodoKEM-976 uses the modulus $64 \cdot 1024 = 65536$.
- Back-of-the-envelope calculations suggest that using BCH codes as in LAC [75] should make a modulus as small as $64\sqrt{1024} = 2048$ work with the New Hope error distribution.

Extrapolating from these sizes of q for $n = 1024$ to formulas for q that differ only in the exponent of n, namely $64n^{0.501}$ and $64n$, corresponds to purely changing

n	q	β_0/n	κ_0/n	β_1/n	κ_1/n	reldrop
32	363	1.25000	0.90625	1.25000	0.75000	0.00000
1024	2062	1.22461	1.03809	1.14941	1.02930	0.63198
32768	11706	1.25000	0.72910	1.20096	0.69708	0.60113
1048576	66450	1.22636	0.55642	1.19031	0.53552	0.59660
32	2048	1.25000	0.53125	1.25000	0.46875	0.00000
1024	65536	0.76465	1.08887	0.72949	1.03320	0.44197
32768	2097152	0.86414	0.83578	0.83987	0.81613	0.41523
1048576	67108864	0.89010	0.71349	0.87176	0.69796	0.40855

Table 5.1.1. The impact of changes in variance upon block size for various pairs (n, q), according to the estimator used in [6]. The β_0/n and κ_0/n columns are for variance 8. The β_1/n and κ_1/n columns are for variance 5511515/1048576. The "reldrop" column is $(\beta_0 - \beta_1)/(\beta_0/\lg \beta_0)$. The last five columns are rounded to 5 digits after the decimal point.

Q_0. Further motivations for considering $Q_0 = 0.501$ and $Q_0 = 1$ are mentioned elsewhere in this paper. More to the point, these two values of Q_0 are separated by enough to easily see the impact on $1/(Q_0 + 1/2)$.

Table 5.1.1 reports, for each of these pairs (n, q), what happens to β when variance drops from 8 to 5511515/1048576, according to the estimator used in [6], with κ allowed to be as large as $2n$. The "reldrop" column in the table is $(\lg \beta)/\beta$ times the drop in β. For comparison, the asymptotics say that "reldrop" is asymptotically $0.605\ldots + o(1)$ for $Q_0 = 0.501$, and about 2/3 as large, asymptotically $0.403\ldots + o(1)$, for $Q_0 = 1$. The concrete numbers in the table are not far from this.

The numbers in the table are printed out by the Sage script in Figure 5.1.2. As a check that it is using the estimator correctly, the script also prints out (β, κ) for the original New Hope parameters $(n, q, s) = (1024, 12289, \sqrt{8})$ and the revised new Hope parameters $(n, q, s) = (1024, 12289, 2)$, namely $(968, 1071)$ and $(886, 1019)$ respectively. This matches the $(886, 1019)$ produced by the the "LWE $2n$ samples" script provided by [6] for "NewHope" with $n = 1024$, and the $(968, 1071)$ produced if the script is modified to say `sd = sqrt(8)` instead of `sd = 2`.

A slightly smaller $\beta = 967$ is claimed for the original New Hope in [13, Table 1] (as mentioned in Section 2.1) and in [11, Table 12], but those attack parameters do not satisfy the inequality stated in [13]. When asked about this error, the New Hope authors reported that their software to find β was defining d as $n + \kappa$ rather than $n + \kappa + 1$. More broadly, readers attempting to verify calculations in the literature should be alert for small variations. Consider, e.g., the $(a, e, 3.14)$ variant mentioned in Section 2.4; as another example, the literature sometimes replaces δ with $\delta^{d/(d-1)}$.

5.2. Larger sizes. Table 5.1.1 also reports the values of β/n and κ/n from the estimator used in [6]. For comparison, the asymptotics say that β/n and κ/n converge to the aforementioned 1002000/1002001 and 1/1001 respectively for

```
load('estimator.py')

def bkzmodel(beta,d,B):
  return (3/2)^RR(beta/2)

def betakappa(n,q,V):
  alpha = sqrt(2*pi*V)/RR(q)
  samplesavailable = 2*n # limiting kappa
  est = primal_usvp(n,alpha,q,m=samplesavailable,
                    success_probability=0.99,
                    secret_distribution='normal',
                    reduction_cost_model=bkzmodel)
  return est['beta'],est['m']

for n,q,V in (1024,12289,8),(1024,12289,4):
  beta,kappa = betakappa(n,q,V)
  print('n',n,'q',q,'V',V,'kappa',kappa,'beta',beta)
  sys.stdout.flush()

for j in 5,10,15,20:
  n = 2^j
  for q in floor(64*n^0.501),64*n:
    betas = []
    for V in 8,5511515/1048576:
      beta,kappa = betakappa(n,q,V)
      print('n',n,'q',q,'V',V,'kappa/n',RR(kappa/n),'beta/n',RR(beta/n))
      sys.stdout.flush()
      betas += [beta]
    reldrop = RR((betas[0]-betas[1])/(betas[0]/log(betas[0],2.0)))
    print('n',n,'q',q,'reldrop',reldrop)
    sys.stdout.flush()
```

Fig. 5.1.2. A Sage script that calculated the numbers in Table 5.1.1, using the same estimator used in [6].

$Q_0 = 0.501$, and to 8/9 and 1/3 respectively for $Q_0 = 1$. The concrete numbers in the table for both β/n and κ/n seem to be considerably above these limits (except that β/n seems close for $Q_0 = 1$), motivating the following investigation of larger n.

The estimators in the literature become extremely slow as n increases. However, for any particular (n, q, s), one can instead minimize β on the curve of *real* pairs $(\kappa, \beta) \in \mathbb{R} \times \mathbb{R}$ for which $\text{StandardRatio}(n, q, s, \kappa, \beta) = 1$. Rewriting the equation in the form $\varphi(\kappa, \beta) = 0$ and applying Lagrange multipliers converts this minimization problem into the problem of solving the two equations $\varphi(\kappa, \beta) = 0$ and $\varphi_x(\kappa, \beta) = 0$ where φ_x is the derivative of φ with respect to its first input. Figure 5.2.1 is a Sage script that finds, to reasonably high precision, a solution at reasonable speed by well-known equation-solving techniques.

```
x,y = var('x y')
for Q0 in 1,0.501:
  for V in 8,5511515/1048576:
    for j in 10,15,20,25,32,64,128,256,512,1024,2048:
      RR = RealField(2*j+50)
      n = 2^j
      q = floor(64*n^Q0)
      d = n+x+1
      delta = (y*(pi*y)^(1/y)/(2*pi*exp(1)))^(1/(2*(y-1)))
      rho = ((n+x)*V+1)^(1/2)
      rho /= (d/y)^(1/2)*delta^(2*y-d-1)*q^(x/d)
      g = log(rho)
      gx = g.derivative(x)
      gy = g.derivative(y)
      gxx = gx.derivative(x)
      gxy = gx.derivative(y)

      ok = False
      while not ok:
        a = RR.random_element(1/10,19/10)*n
        b = RR.random_element(n/10,9*n/10+a)

        for loop in range(30):
          gab = RR(g(x=a,y=b))
          gxab = RR(gx(x=a,y=b))
          if abs(gab) < 1/2^30 and abs(gxab) < 1/2^30:
            ok = True
            break
          gyab = RR(gy(x=a,y=b))
          gxxab = RR(gxx(x=a,y=b))
          gxyab = RR(gxy(x=a,y=b))
          J = matrix([[gxab,gyab],[gxxab,gxyab]])
          d,e = J.solve_right(vector([gab,gxab]))
          step = 1
          while True:
            a1,b1 = a-step*d,b-step*e
            if a1 > 0 and b1 > 0 and a1 < 2*n and b1 < a1+n:
              a,b = a1,b1
              break
            step *= RR(1/2)

      print('n 2^%d Q0 %.8f V %s beta/n %.8f kappa/n %.8f'
            % (j,Q0,V,b/n,a/n))
```

Fig. 5.2.1. A script that, for various (n, q, s), finds real numbers (β, κ) to approximately locally minimize β subject to $((n+\kappa)s^2+1)^{1/2} = (d/\beta)^{1/2}\delta^{2\beta-d-1}q^{\kappa/d}$.

For, e.g., $(n, q, s) = (1024, 65536, \sqrt{8})$, the output of this script says that $\beta/n \approx 0.76508$ and $\kappa/n \approx 1.10203$. This is a slightly larger value of β/n than the 0.76465 (i.e., $\beta = 783$) reported in Table 5.1.1 from the estimator used in [6], whereas a minimum over $\mathbb{R} \times \mathbb{R}$ cannot be larger than a minimum of the same function over $\mathbb{Z} \times \mathbb{Z}$. One possible explanation would be that the local minimum found by Figure 5.2.1 is not a global minimum; a simpler explanation is the small variations mentioned above in the inequalities considered in the literature.

For $n = 2^{2048}$, again with $q = 64n$ and $s = \sqrt{8}$, the same script outputs $(\beta/n, \kappa/n) = (0.88945\ldots, 0.33709\ldots)$, much closer to the $(8/9, 1/3)$ limit. For $n = 2^{2048}$, $q = \lfloor 64n^{0.501} \rfloor$, and $s = \sqrt{8}$, the script outputs $(\beta/n, \kappa/n) = (1.00343\ldots, 0.00665\ldots)$, almost as close to the $(0.99999\ldots, 0.00099\ldots)$ limit. Finally, the relative drops in β for $n = 2^{2048}$ are $0.40394\ldots$ for $Q_0 = 1$ and $0.60513\ldots$ for $Q_0 = 0.501$, very close to the limits.

References

[1] — (no editor), *IEEE international symposium on information theory, ISIT 2007, Nice, France, June 24–29, 2007*, IEEE, 2007. URL: https://ieeexplore.ieee.org/xpl/conhome/4497218/proceeding. See [96].

[2] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella Béguelin, Paul Zimmermann, *Imperfect forward secrecy: how Diffie-Hellman fails in practice*, Communications of the ACM **62** (2019), 106–114. URL: https://weakdh.org. Citations in this document: §1.

[3] Gorjan Alagic, *FrodoKEM in the third round* (2021). URL: https://nist.pqcrypto.org/foia/20221107/Frodo-final.pptx. Citations in this document: §B.5.

[4] Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, Yi-Kai Liu, *Status report on the third round of the NIST Post-Quantum Cryptography Standardization Process* (2022). NISTIR 8413. URL: https://csrc.nist.gov/publications/detail/nistir/8413/final. Citations in this document: §4, §4, §4, §B, §B.5, §B.5.

[5] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, Wen Wang, *Classic McEliece: conservative code-based cryptography* (2020). URL: https://classic.mceliece.org/nist/mceliece-20201010.pdf. Citations in this document: §1, §A.3.

[6] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, Thomas Wunderer, *Estimate all the {LWE, NTRU} Schemes!*, in SCN 2018 [**36**] (2018), 351–367. URL: https://eprint.iacr.org/2018/331. Citations in this document: §2, §2.9, §2.9, §2.9, §5.1, §5.1, §5.1.1, §5.1.1, §5.1, §5.1.2, §5.1.2, §5.2, §5.2.

[7] Martin R. Albrecht, Léo Ducas, *Lattice attacks on NTRU and LWE: a history of refinements* (2021). URL: https://eprint.iacr.org/2021/799. Citations in this document: §2, §4.4, §C.2.

[8] Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, John M. Schanck, *Estimating quantum speedups for lattice sieves*, in Asiacrypt 2020 [**82**] (2020), 583–613. URL: https://eprint.iacr.org/2019/1161. Citations in this document: §5.

[9] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, Thomas Wunderer, *Revisiting the expected cost of solving uSVP and applications to LWE*, in Asiacrypt 2017 [**99**] (2017). URL: https://eprint.iacr.org/2017/815. Citations in this document: §2.5, §15.

[10] Erdem Alkim, Roberto Avanzi, Joppe Bos, Leo Ducas, Antonio de la Piedra, Thomas Poppelmann, Peter Schwabe, Douglas Stebila, *NewHope: algorithm specifications and supporting documentation* (2017). URL: https://web.archive.org/web/20190411045044/https://newhopecrypto.org/data/NewHope_2017_12_21.pdf. Citations in this document: §A.2, §A.2.

[11] Erdem Alkim, Roberto Avanzi, Joppe Bos, Leo Ducas, Antonio de la Piedra, Thomas Poppelmann, Peter Schwabe, Douglas Stebila, Martin R. Albrecht, Emmanuela Orsini, Valery Osheter, Kenneth G. Paterson, Guy Peer, Nigel P. Smart, *NewHope: algorithm specifications and supporting documentation* (2019). URL: https://web.archive.org/web/20191015124615/https://newhopecrypto.org/data/NewHope_2019_04_10.pdf. Citations in this document: §5.1, §A.2, §A.2.

[12] Erdem Alkim, Joppe W. Bos, Léo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, Douglas Stebila, *FrodoKEM: Learning With Errors key encapsulation: algorithm specifications and supporting documentation* (2021). URL: https://web.archive.org/web/20220119174856/https://frodokem.org/files/FrodoKEM-specification-20210604.pdf. Citations in this document: §1.2, §2, §B, §B.2, §B.2, §B.3, §B.3, §B.3, §B.4, §B.5, §B.5, §B.5, §B.5.

[13] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, Peter Schwabe, *Post-quantum key exchange—a new hope*, in USENIX Security 2016 [**60**] (2016), 327–343. URL: https://eprint.iacr.org/2015/1092. Citations in this document: §2, §2.1, §3.1, §5.1, §5.1, §A.1, §20, §A.2, §A.2, §A.2.

[14] Daniel Apon, *Re: Looseness, security risks, and LWR vs. LWE* (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/Yx0wZuZP6ag/m/CqxTSn3TCAAJ. Citations in this document: §B.5.

[15] Benny Applebaum, David Cash, Chris Peikert, Amit Sahai, *Fast cryptographic primitives and circular-secure encryption based on hard learning problems*, in Crypto 2009 [**54**] (2009), 595–618. URL: https://www.wisdom.weizmann.ac.il/~/bennyap/pubs/kdm-learning.pdf. Citations in this document: §A.4, §A.4.

[16] Jean-Philippe Aumasson, *Too much crypto* (2019). URL: https://eprint.iacr.org/2019/1492. Citations in this document: §1.3.

[17] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé, *CRYSTALS-Kyber: Algorithm specifications and supporting documentation* (2020). URL: https://web.archive.org/web/20211007045636/https://pq-crystals.org/kyber/data/kyber-specification-round3.pdf. Citations in this document: §1.1, §1.1, §1.1, §1.1, §2.9.

[18] Roberto Avanzi, Howard M. Heys (editors), *Selected areas in cryptography—SAC 2016—23rd international conference, St. John's, NL, Canada, August 10–12, 2016, revised selected papers*, Lecture Notes in Computer Science, 10532, Springer, 2017. ISBN 978-3-319-69452-8. See [72].

[19] László Babai, *On Lovász' lattice reduction and the nearest lattice point problem*, Combinatorica **6** (1986), 1–13. Citations in this document: §3.1.

[20] Razvan Barbulescu, *Algorithms of discrete logarithm in finite fields* (2013). URL: https://tel.archives-ouvertes.fr/tel-00925228/file/these%5Favec%5Fresume.pdf. Citations in this document: §1.

[21] Anja Becker, Léo Ducas, Nicolas Gama, Thijs Laarhoven, *New directions in nearest neighbor searching with applications to lattice sieving*, in SODA 2016 [**71**] (2016), 10–24. URL: https://eprint.iacr.org/2015/1128. Citations in this document: §2.2, §2.2, §2.2, §3.2.

[22] Mihir Bellare, Hannah Davis, Felix Günther, *Separate your domains: NIST PQC KEMs, oracle cloning and read-only indifferentiability*, in Eurocrypt 2020 [**35**] (2020), 3–32. URL: https://eprint.iacr.org/2020/241. Citations in this document: §2.9.

[23] Mihir Bellare, Phillip Rogaway, *The exact security of digital signatures: how to sign with RSA and Rabin*, in Eurocrypt 1996 [**79**] (1996), 399–416. URL: https://cseweb.ucsd.edu/~mihir/papers/exactsigs.html. Citations in this document: §B, §B.5, §24.

[24] Daniel J. Bernstein, *Understanding brute force* (2005). ECRYPT STVL Workshop on Symmetric Key Encryption. URL: http://cr.yp.to/papers.html#bruteforce. Citations in this document: §1.

[25] Daniel J. Bernstein, *Break a dozen secret keys, get a million more for free* (2015). URL: https://blog.cr.yp.to/20151120-batchattacks.html. Citations in this document: §1.

[26] Daniel J. Bernstein, *OFFICIAL COMMENT: Frodo* (2018). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/rJYnyTEi92E/m/l5xBpeTpBQAJ. Citations in this document: §B.5.

[27] Daniel J. Bernstein, *Comparing proofs of security for lattice-based encryption* (2019). Second PQC Standardization Conference. URL: https://cr.yp.to/papers.html#latticeproofs. Citations in this document: §A.4.

[28] Daniel J. Bernstein, *Cryptographic competitions* (2020). URL: https://cr.yp.to/papers.html#competitions. Citations in this document: §1.3.

[29] Daniel J. Bernstein, *Re: Looseness, security risks, and LWR vs. LWE* (2021). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/YxOwZuZP6ag/m/GXIwomNsCAAJ. Citations in this document: §B.5.

[30] Daniel J. Bernstein, *Re: Was: eddsa (un)suited for mandatory to implement ciphersuite?* (2022). URL: https://mailarchive.ietf.org/arch/msg/cfrg/GRigAYvZ8-Z8qmxJ1jOiKR8eLyQ/. Citations in this document: §23.

[31] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal, *NTRU Prime: round 2* (2019). URL: https://ntruprime.cr.yp.to/nist.html. Citations in this document: §2.3, §4.4, §4.5.

[32] Daniel J. Bernstein, Tanja Lange, *Non-randomness of S-unit lattices* (2021). URL: https://cr.yp.to/papers.html#spherical. Citations in this document: §1.3, §12.

[33] Joe P. Buhler (editor), *Algorithmic number theory, third international symposium, ANTS-III, Portland, Oregon, USA, June 21–25, 1998, proceedings*, Lecture Notes in Computer Science, 1423, Springer, 1998. ISBN 3-540-64657-4. See [57].

[34] Peter Campbell, Michael Groves, Dan Shepherd, *Soliloquy: a cautionary tale* (2014). URL: https://web.archive.org/web/20211022132624/https://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf. Citations in this document: §A.1.

[35] Anne Canteaut, Yuval Ishai (editors), *Advances in cryptology—EUROCRYPT 2020—39th annual international conference on the theory and applications of cryptographic techniques, Zagreb, Croatia, May 10–14, 2020, proceedings, part II*, Lecture Notes in Computer Science, 12106, Springer, 2020. ISBN 978-3-030-45723-5. See [22].

[36] Dario Catalano, Roberto De Prisco (editors), *Security and cryptography for networks—11th international conference, SCN 2018, Amalfi, Italy, September 5–7, 2018, proceedings*, Lecture Notes in Computer Science, 11035, Springer, 2018. See [6].

[37] André Chailloux, Johanna Loyer, *Lattice sieving via quantum random walks* (2021). URL: https://eprint.iacr.org/2021/570. Citations in this document: §5.

[38] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, Peter Schwabe, *MQDSS specifications* (2019). URL: https://web.archive.org/web/20220120235746/https://mqdss.org/files/MQDSS_Ver2.pdf. Citations in this document: §B.5.

[39] Cloudflare Research, *Post-quantum key agreement* (2022). URL: https://pq.cloudflareresearch.com. Citations in this document: §1.3.

[40] Henri Cohen, *Advanced topics in computational number theory*, Springer, 2000. ISBN 0-387-98727-4. Citations in this document: §12.

[41] Don Coppersmith, Adi Shamir, *Lattice attacks on NTRU*, in Eurocrypt 1997 [**50**] (1997), 52–61. URL: https://link.springer.com/content/pdf/10.1007/3-540-69053-0_5.pdf. Citations in this document: §2.4.

[42] Jintai Ding, Rainer Steinwandt (editors), *Post-quantum cryptography—10th international conference, PQCrypto 2019, Chongqing, China, May 8–10, 2019, revised selected papers*, 11505, Springer, 2019. ISBN 978-3-030-25509-1. See [43].

[43] Emmanouil Doulgerakis, Thijs Laarhoven, Benne de Weger, *Finding closest lattice vectors using approximate Voronoi cells*, in PQCrypto 2019 [**42**] (2019), 3–22. Citations in this document: §3.2.

[44] Emmanouil Doulgerakis, Thijs Laarhoven, Benne de Weger, *Sieve, enumerate, slice, and lift: hybrid lattice algorithms for SVP via CVPP*, in Africacrypt 2020 [**85**] (2020), 301–320. URL: https://eprint.iacr.org/2020/487. Citations in this document: §2.2, §4, §4, §4.5.

[45] Léo Ducas, *Shortest vector from lattice sieving: A few dimensions for free*, in Eurocrypt 2018 [**83**] (2018), 125–145. URL: https://eprint.iacr.org/2017/999. Citations in this document: §2.2, §2.2, §4, §4, §4.5.

[46] Léo Ducas, Thijs Laarhoven, Wessel P. J. van Woerden, *The randomized slicer for CVPP: sharper, faster, smaller, batchier*, in PKC 2020 [**67**] (2020), 3–36. URL: https://eprint.iacr.org/2020/120. Citations in this document: §3, §3.2, §13, §13, §3.2, §3.3, §3.3, §3.3, §3.3, §4.4.

[47] Julien Duman, Kathrin Hövelmanns, Eike Kiltz, Vadim Lyubashevsky, Gregor Seiler, *Faster lattice-based KEMs via a generic Fujisaki-Okamoto transform using prefix hashing*, in CCS 2021 [**68**] (2021), 2722–2737. URL: https://eprint.iacr.org/2021/1351. Citations in this document: §1.1, §1.1, §8, §A, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4, §A.4.

[48] Thomas Espitau, Paul Kirchner, *The nearest-colattice algorithm: time-approximation tradeoff for approx-CVP*, in ANTS 2020 [**51**] (2020), 251–266. URL: https://eprint.iacr.org/2020/694. Citations in this document: §3, §3.1, §3.1, §3.1, §3.1, §3.1, §3.1, §3.1, §3.1, §11, §3.1, §3.1, §13, §3.3, §3.3, §3.3, §3.3.

[49] Philippe Flajolet, Robert Sedgewick, *Analytic combinatorics*, Cambridge University Press, 2009. ISBN 978-0-521-89806-5. URL: https://ac.cs.princeton.edu/home/. Citations in this document: §4.2.

[50] Walter Fumy (editor), *Advances in cryptology—EUROCRYPT '97, international conference on the theory and application of cryptographic techniques, Konstanz, Germany, May 11–15, 1997*, Lecture Notes in Computer Science, 1233, Springer, 1997. See [41].

[51] Steven Galbraith (editor), *ANTS XIV: proceedings of the fourteenth algorithmic number theory symposium, Auckland 2020*, Open Book Series, 4, Mathematical Sciences Publishers, 2020. ISBN 978-1-935107-07-1. See [48].

[52] Michael T. Goodrich, *Zig-zag sort: a simple deterministic data-oblivious sorting algorithm running in $O(n \log n)$ time*, in STOC 2014 [**94**] (2014), 684–693. URL: https://arxiv.org/abs/1403.2777. Citations in this document: §1.3.

[53] Qian Guo, Thomas Johansson, *Faster dual lattice attacks for solving LWE with applications to CRYSTALS*, in Asiacrypt 2021 [**100**] (2021), 33–62. Citations in this document: §1.1.

[54] Shai Halevi (editor), *Advances in cryptology—CRYPTO 2009, 29th annual international cryptology conference, Santa Barbara, CA, USA, August 16–20, 2009. proceedings*, 5677, Springer, 2009. ISBN 978-3-642-03355-1. See [15].

[55] John Harrison, *HOL Light: A tutorial introduction*, in FMCAD 1996~[**97**] (1996), 265–269. Citations in this document: §C.

[56] Max Heiser, *Improved quantum hypercone locality sensitive filtering in lattice sieving* (2021). URL: https://eprint.iacr.org/2021/1295. Citations in this document: §5.

[57] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, *NTRU: a ring-based public key cryptosystem*, in ANTS 1998 [**33**] (1998), 267–288. URL: https://ntru.org/f/hps98.pdf. Citations in this document: §2.4, §2.4, §2.4, §4.5.

[58] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, *NTRU: a new high speed public key cryptosystem* (2016). Circulated at Crypto 1996, put online in 2016. URL: https://ntru.org/f/hps96.pdf. Citations in this document: §2.4.

[59] Dennis Hofheinz, Kathrin Hövelmanns, Eike Kiltz, *A modular analysis of the Fujisaki-Okamoto transformation*, in TCC 2017-1 [**64**] (2017), 341–371. URL: https://eprint.iacr.org/2017/604. Citations in this document: §B.1.

[60] Thorsten Holz, Stefan Savage (editors), *25th USENIX security symposium, USENIX Security 16, Austin, TX, USA, August 10–12, 2016*, USENIX Association, 2016. URL: https://www.usenix.org/conference/usenixsecurity16. See [13].

[61] Nick Howgrave-Graham, *A hybrid lattice-reduction amd meet-in-the-middle attack against NTRU*, in Crypto 2007 [**81**] (2007), 150–169. URL: https://www.iacr.org/archive/crypto2007/46220150/46220150.pdf. Citations in this document: §3.1, §3.1, §9, §3.1, §10.

[62] Yuval Ishai, Vincent Rijmen (editors), *Advances in cryptology—EUROCRYPT 2019—38th annual international conference on the theory and applications of cryptographic techniques, Darmstadt, Germany, May 19–23, 2019, proceedings, part II*, Springer, 2019. ISBN 978-3-030-17655-6. See [87].

[63] Thomas Johansson, Fredrik Jönsson, *On the complexity of some cryptographic problems based on the general decoding problem*, IEEE Transactions on Information Theory **48** (2002), 2669–2678. Citations in this document: §A.3.

[64] Yael Kalai, Leonid Reyzin (editors), *Theory of cryptography—15th international conference, TCC 2017, Baltimore, MD, USA, November 12–15, 2017, proceedings, part I*, Lecture Notes in Computer Science, 10677, Springer, 2017. ISBN 978-3-319-70499-9. See [59].

[65] Daniel Kales, Greg Zaverucha, *Forgery attacks on MQDSSv2.0* (2019). URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/official-comments/MQDSS-round2-official-comment.pdf. Citations in this document: §B.5.

[66] Aggelos Kiayias (editor), *Topics in cryptology—CT-RSA 2011—the cryptographers' track at the RSA Conference 2011, San Francisco, CA, USA, February 14–18, 2011, proceedings*, Lecture Notes in Computer Science, 6558, Springer, 2011. ISBN 978-3-642-19073-5. See [74].

[67] Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, Vassilis Zikas (editors), *Public-key cryptography—PKC 2020—23rd IACR international conference on practice and theory of public-key cryptography, Edinburgh, UK, May 4–7, 2020, proceedings, part II*, 12111, Springer, 2020. ISBN 978-3-030-45387-9. See [46].

[68] Yongdae Kim, Jong Kim, Giovanni Vigna, Elaine Shi (editors), *CCS '21: 2021 ACM SIGSAC conference on computer and communications security, virtual event, Republic of Korea, November 15–19, 2021*, ACM, 2021. ISBN 978-1-4503-8454-4. See [47].

[69] Neal Koblitz, Alfred Menezes, *Critical perspectives on provable security: fifteen years of "another look" papers*, Advances in Mathematics of Communications **13** (2019), 517–558. URL: https://eprint.iacr.org/2019/1336. Citations in this document: §B.5, §24.

[70] Neal Koblitz, Subhabrata Samajder, Palash Sarkar, Subhadip Singha, *Concrete analysis of approximate Ideal-SIVP to decision Ring-LWE reduction* (2022). URL: https://eprint.iacr.org/2022/275. Citations in this document: §4.5.

[71] Robert Krauthgamer (editor), *Proceedings of the twenty-seventh annual ACM-SIAM symposium on discrete algorithms, SODA 2016, Arlington, VA, USA, January 10–12, 2016*, SIAM, 2016. ISBN 978-1-61197-433-1. See [21].

[72] Thijs Laarhoven, *Sieving for closest lattice vectors (with preprocessing)*, in SAC 2016 [**18**] (2016), 523–542. Citations in this document: §3.2, §3.2.

[73] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., László Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), 515–534. ISSN 0025-5831. MR 84a:12002. URL: https://openaccess.leidenuniv.nl/bitstream/handle/1887/3810/346_050.pdf. Citations in this document: §3.1.

[74] Richard Lindner, Chris Peikert, *Better key sizes (and attacks) for LWE-based encryption*, in CT-RSA [**66**] (2011), 319–339. URL: https://eprint.iacr.org/2010/613. Citations in this document: §2, §9, §9.

[75] Xianhui Lu, Yamin Liu, Zhenfei Zhang, Dingding Jia, Haiyang Xue, Jingnan He, Bao Li, *LAC: practical Ring-LWE based public-key encryption with byte-level modulus* (2018). URL: https://eprint.iacr.org/2018/1009. Citations in this document: §5.1.

[76] Vadim Lyubashevsky, Chris Peikert, Oded Regev, *On ideal lattices and learning with errors over rings*, Journal of the ACM **60** (2013), Article 43, 35 pages. URL: https://eprint.iacr.org/2012/230. Citations in this document: §2, §6, §6, §2.3, §2.3, §2.4, §2.4, §4.5.

[77] Mitsuru Matsui (editor), *Advances in cryptology—ASIACRYPT 2009, 15th international conference on the theory and application of cryptology and information security, Tokyo, Japan, December 6–10, 2009. proceedings*, 5912, Springer, 2009. ISBN 978-3-642-10365-0. See [98].

[78] MATZOV, *Report on the security of LWE: improved dual lattice attack* (2022). URL: https://doi.org/10.5281/zenodo.6412487. Citations in this document: §1.1, §1.3, §10, §10.

[79] Ueli M. Maurer (editor), *Advances in cryptology—EUROCRYPT '96: proceedings of the fifteenth international conference on the theory and application of cryptographic techniques held in Saragossa, May 12–16, 1996*, Lecture Notes in Computer Science, 1070, Springer, 1996. ISBN 3-540-61186-X. MR 97g:94002. See [23].

[80] Alexander May, Joseph H. Silverman, *Dimension reduction methods for convolution modular lattices*, in [**95**] (2001), 110–15. URL: https://www.cits.ruhr-uni-bochum.de/personen/may/publications.html. Citations in this document: §2.4.

[81] Alfred Menezes (editor), *Advances in cryptology—CRYPTO 2007, 27th annual international cryptology conference, Santa Barbara, CA, USA, August 19–23, 2007, proceedings*, Lecture Notes in Computer Science, 4622, Springer, 2007. ISBN 978-3-540-74142-8. See [61].

[82] Shiho Moriai, Huaxiong Wang (editors), *Advances in cryptology—ASIACRYPT 2020—26th international conference on the theory and application of cryptology and information security, Daejeon, South Korea, December 7–11, 2020, proceedings, part II*, 12492, Springer, 2020. ISBN 978-3-030-64833-6. See [8].

[83] Jesper Buus Nielsen, Vincent Rijmen (editors), *Advances in cryptology—EUROCRYPT 2018—37th annual international conference on the theory and applications of cryptographic techniques, Tel Aviv, Israel, April 29–May 3, 2018 proceedings, part I*, 10820, Springer, 2018. ISBN 978-3-319-78380-2. See [45].

[84] NIST PQC team, *Announcement: The end of the 3rd round - the first PQC algorithms to be standardized* (2022). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkGPk/m/f3Hl0sh3AgAJ. Citations in this document: §3.

[85] Abderrahmane Nitaj, Amr M. Youssef (editors), *Progress in cryptology—AFRICACRYPT 2020—12th international conference on cryptology in Africa, Cairo, Egypt, July 20–22, 2020, proceedings*, 12174, Springer, 2020. ISBN 978-3-030-51937-7. See [44].

[86] NTRU Prime Risk-Management Team, *Risks of lattice KEMs* (2021). URL: https://ntruprime.cr.yp.to/warnings.html. Citations in this document: §8.

[87] Alice Pellet-Mary, Guillaume Hanrot, Damien Stehlé, *Approx-SVP in ideal lattices with pre-processing*, in Eurocrypt 2019 [**62**] (2019), 685–716. URL: https://eprint.iacr.org/2019/215. Citations in this document: §A.1.

[88] Ray Perlner, *Re: post-quantum benchmarking and RNGs* (2017). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/PW6GF-wHGFE/m/nnAnuUAUAAAJ. Citations in this document: §B.4.

[89] Ray Perlner, *Requirements for security against multi-target attacks, for McEliece and other code-based cryptosystems?* (2021). URL: https://web.archive.org/web/20210717065041/https://crypto.stackexchange.com/questions/92073/requirements-for-security-against-multi-target-attacks-for-mceliece-and-other-c. Citations in this document: §A.3.

[90] Ray Perlner, *Re: Parameter selection for the selected algorithms* (2022). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/4MBurXr58Rs/m/xHojUDCaBAAJ. Citations in this document: §3.

[91] Herbert Robbins, *A remark on Stirling's formula*, The American Mathematical Monthly **62** (1955), 26–29. Citations in this document: §4.2.

[92] The Sage Developers (editor), *SageMath, the Sage Mathematics Software System (Version 9.5)*, 2022. URL: https://www.sagemath.org. Citations in this document: §4.2.

[93] Nicolas Sendrier, *Decoding one out of many*, in PQCrypto 2011 [**102**] (2011), 51–67. URL: https://eprint.iacr.org/2011/367. Citations in this document: §14, §A.3.

[94] David B. Shmoys (editor), *Symposium on theory of computing, STOC 2014, New York, NY, USA, May 31–June 03, 2014*, ACM, 2014. ISBN 978-1-4503-2710-7. See [52].

[95] Joseph H. Silverman (editor), *Cryptography and lattices: proceedings of the 1st International Conference (CaLC 2001) held in Providence, RI, March 29–30, 2001*, Lecture Notes in Computer Science, 2146, Springer, 2001. ISBN 3-540-42488-1. MR 2002m:11002. See [80].

[96] Naftali Sommer, Meir Feder, Ofir Shalvi, *Finding the closest lattice point by iterative slicing*, in [**1**] (2007), 206–210. Citations in this document: §12, §3.2.

[97] Mandayam K. Srivas, Albert John Camilleri (editors), *Formal methods in computer-aided design, first international conference, FMCAD '96, Palo Alto, California, USA, November 6–8, 1996, proceedings*, Lecture Notes in Computer Science, 1166, Springer, 1996. ISBN 3-540-61937-2. See [55].

[98] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, Keita Xagawa, *Efficient public key encryption based on ideal lattices*, in Asiacrypt 2009 [**77**] (2009), 617–635. URL: https://eprint.iacr.org/2009/285. Citations in this document: §2.4, §2.4.

[99] Tsuyoshi Takagi, Thomas Peyrin (editors), *Advances in cryptology—ASIACRYPT 2017—23rd international conference on the theory and applications of cryptology and information security, Hong Kong, China, December 3–7, 2017, proceedings, part II*, Lecture Notes in Computer Science, 10625, Springer, 2017. ISBN 978-3-319-70696-2. See [9].

[100] Mehdi Tibouchi, Huaxiong Wang (editors), *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*, 13093, Springer, 2021. ISBN 978-3-030-92067-8. See [53].

[101] Herbert S. Wilf, *generatingfunctionology*, Academic Press, 1994. URL: https://www2.math.upenn.edu/~wilf/DownldGF.html. Citations in this document: §4.2.

[102] Bo-Yin Yang (editor), *Post-quantum cryptography—4th international workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. proceedings*, 7071, Springer, 2011. ISBN 978-3-642-25404-8. See [93].

A Reconciliation with previous work

This appendix traces various quotes from the literature that seem likely to have led many readers to believe that multi-target attacks against lattice KEMs are no more efficient than single-target attacks. To the extent that the sources appear

to be stating arguments for this belief, this appendix explains how this paper's analysis exploits gaps in those arguments. The deepest part of this appendix is identifying flaws in a provable-security argument in [47]; the other gaps are easier to see.

A.1. New Hope 2015. A subsection labeled "all-for-the-price-of-one attacks" in the New Hope paper [13] expressed concern about the possibility[20] of attacks that break T targets ("compromise *all* communications") for the cost of breaking 1 target. The paper continued by saying that "all those pitfalls can be avoided by having the communicating parties generate a fresh **a** at each instance of the protocol (as we propose)".

The argument that there are no all-for-the-price-of-one attacks was not stated explicitly, but appears to be the following:

- Multi-target attacks exploit the fact that multiple ciphertexts share one lattice.
- An instance of the 2015 New Hope protocol, by definition, involves just one ciphertext. Generating a fresh **a** means generating a fresh public key—a fresh lattice—to be used for a single ciphertext. Ergo, multiple ciphertexts do not share one lattice.

This paper's attack fits the first part of the argument, but straightforwardly dodges the second part by attacking KEMs that are designed for IND-CCA2 security, KEMs that allow many ciphertexts for a single public key; the second part of the argument would fail if it were applied to those KEMs.

Note that the first part of the argument is merely a prediction, not a proof, so it could fail too. For example, multiple New Hope public keys share a ring, and thus share an S-unit lattice. There are some other lattice problems that appear to be vulnerable to S-unit precomputations (see, e.g., [34] and [87]); perhaps there are similar vulnerabilities in New Hope. However, this possibility is not what this paper's attack exploits. This paper is not attacking KEMs that support only one ciphertext per public key, such as the 2015 version of New Hope.

A.2. New Hope 2017. A revised version [10] of New Hope was submitted to the NIST Post-Quantum Cryptography Standardization Project in 2017. In particular, this revision included two cryptosystems: NewHope-CPA for one ciphertext per key, and NewHope-CCA claiming IND-CCA2 security.

The text "all those pitfalls can be avoided by having the communicating parties generate a fresh **a** at each instance of the protocol (as we propose)" was copied from [13], where it was correct, into [10], where it was incorrect: NewHope-CCA was not limited to a single ciphertext per key.

The erroneous description was eliminated in the 2019 version of the New Hope submission [11, page 18], which instead made the following claim: "all those pitfalls can be avoided by having the communicating parties generate a

[20] See Section 3.1 regarding questions left open by the specific multi-target attack strategy outlined in [13].

fresh **a** for each public key (as we propose)". It is not clear that the consequences of this change were ever considered.

"All those pitfalls" in [11] appears to be referring to "all-for-the-price-of-one attacks", as in [13]. Saying that the pitfalls "can be avoided" appears to be saying that there are no all-for-the-price-of-one attacks. But the argument that there are no all-for-the-price-of-one attacks is again not stated explicitly.

The argument stated in Appendix A.1 relies on each public key having only one ciphertext. This argument is broken by the subsequent change from "each instance of the protocol" (i.e., each ciphertext) to "each public key". Each public key has its own lattice, but that lattice is shared by all of the ciphertexts for that public key.

New Hope has similar heuristic security levels for key-recovery attacks and message-recovery attacks. This opened up key recovery as an obvious "all-for-the-price-of-one attack" against multiple ciphertexts, given that New Hope was modified to allow multiple ciphertexts per key. Similar comments apply to many other lattice proposals that claim IND-CCA2 security and that do not try to make the usual key-recovery attacks more difficult than the usual message-recovery attacks.

One could try to argue that, quantitatively, attacks against many ciphertexts for one public key are a smaller problem than attacks against many ciphertexts for many public keys. This is abandoning the claim that the pitfalls are *avoided*, and is instead arguing that the pitfalls are quantitatively *reduced*. This begs further questions, such as the following:

- How many ciphertexts are sent to the most popular public keys?
- What security level is achieved by that number of ciphertexts?
- If Kyber-512 is not simply discarded, should users deploying Kyber-512 be told that Kyber-512 keys are not safe to use for more than, say, 2^{10} ciphertexts per key?

The New Hope documentation already stated in 2015 [13, page 5] that it is "rather uncomfortable to have the security of all connections rely on a single instance of a lattice problem". Reusing a lattice for all connections for one public key is quantitatively different from reusing a lattice for all connections for all public keys, but still creates the same basic issue of multi-target attacks being more efficient than single-target attacks.

A.3. NIST 2021. A 2021 NIST posting "Requirements for security against multi-target attacks, for McEliece and other code-based cryptosystems?" [89] said "it seems reasonable to be worried about 2^{64} target ciphertexts" and criticized code-based cryptosystems for allowing "decoding one out of many" attacks.

As background, the Classic McEliece submission [5, Section 8.3] had cited examples of multi-ciphertext attacks from 2002 [63] and 2011 [93] giving roughly a $T^{1/2}$ speedup for breaking one out of T ciphertexts, and had handled those attacks as follows:

> Rather than analyzing multi-message security in detail, we rely on the general fact that attacking T targets cannot gain more than a factor T. For example, with our recommended `6688/6960/8192` parameter sets, one ciphertext is expected to be secure against an attacker without the resources to find an AES-256 key, and 2^{64} ciphertexts are expected to all be secure against an attacker without the resources to find an AES-192 key.

Taking a very high single-target security level straightforwardly protects against multi-target attacks, whether all-for-the-price-of-one attacks or one-out-of-many attacks or something intermediate.

Instead of acknowledging the submission for this protection against multi-target attacks, NIST criticized the submission for the fact that multi-target attacks existed in the first place. Meanwhile NIST did not criticize lattice submissions that (1) did not recommend such high security levels and (2) did not cite any multi-target cryptanalysis. In essence, one area of cryptography was punished for having investigated and proactively eliminated a threat, while another area of cryptography was rewarded for not even being aware of the threat, never mind protecting against it.

Readers who know the context of NIST comparing submissions to a competition, and who see NIST arguing that one-out-of-many attacks are of concern for code-based cryptography, can easily leap to the conclusion that there are no one-out-of-many attacks against lattice-based cryptography. It is easy to conflate lack of *knowledge* of such lattice attacks with lack of *existence* of those attacks; however, until a problem has survived extensive cryptanalysis, there is little reason to believe that the problem is hard to break.

A.4. ACM CCS 2021. A typical statement that cryptosystem X is "provably secure" communicates two ideas to the informed reader. The first idea is that there is a proof saying that there cannot be an attack against X *unless* there is an attack against some underlying problem P. The second idea is that there are good reasons to think that there are no attacks against P.

If a cryptanalysis paper then claims attacks against X, obviously the inconsistency requires explanation. Here are three possibilities to consider:

- Perhaps the attack analysis is incorrect. This is a cryptanalysis failure.
- Perhaps the proof is incorrect. This is a provable-security failure.
- Perhaps the attack is correct and the proof is correct—so the idea that there are no attacks against P has to be wrong; the proof is actually an attack tool, converting the attack against X into an attack against P. This is a different type of provable-security failure, where a problem P without sufficient cryptanalysis was being used to portray X as safe.

The case considered below is that X is one-out-of-many-ciphertext lattice security and P is many-sample Module-LWE.

Here is the context. Duman–Hövelmanns–Kiltz–Lyubashevsky–Seiler [47] gave a proof, not the proof at issue here, relating multi-target ROM IND-CCA2 security of a KEM to multi-target IND-CPA security of an underlying PKE.

They described this proof as "tighter" than previous proofs. However, a closer look shows a potentially very important exception: *for PKEs that suffer a $T\times$ loss from one-out-of-T attacks*, the quantitative results of [47] are no better than simpler, older proofs. To quantitatively compare [47] to previous work, one has to quantify the amount of damage that one-out-of-T attacks do to commonly considered PKEs.

The same paper gave a provable-security argument that for "lattice-based schemes" there is *no* loss from one-out-of-T attacks. Specifically, the paper said for these systems that "we can show that $\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}} \approx \mathrm{Adv}^{(n,q_C)\text{-IND-CPA}}_{\text{PKE}}$". This is not an unconditional statement: it assumes "the hardness of MLWE as originally defined for the purpose of worst-case to average-case reductions [28, 30, 32] where the number of samples (using the same secret) is unlimited".

The quantity $\mathrm{Adv}^{(n,q)\text{-IND-CPA}}_{\text{PKE}}$ is defined in [47, Figure 2] as the attacker's advantage at guessing a bit r in the following scenario. There are n legitimately generated public keys. The attacker carries out at most q queries (j, M_0, M_1). The response to query (j, M_0, M_1) is a legitimately generated ciphertext for M_r under the jth public key. The quantity $\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}}$ is defined the same way for the single-target case $n = q = 1$.

The class of "lattice-based schemes" is not defined in [47], but presumably includes LPR, New Hope, Kyber, and further noisy-ElGamal schemes surveyed in [27, Section 8]. The features of these schemes that matter for the following analysis are as follows: key generation generates G uniformly at random, generates small a, e at random, computes $A = aG + e$, and outputs public key (G, A); encryption of message M generates small b, c, d at random, computes $B = Gb + d$, computes $C = M + Ab + c$, and outputs ciphertext (B, C).

The conventional single-target "security proof" for such a scheme is as follows. Assume that $(G, aG+e)$ is indistinguishable from a uniform random pair (G, A'); this is an example of a 1-sample small-secret Module-LWE indistinguishability assumption.[21] Also assume that $(G, A', Gb + d, A'b + c)$ is indistinguishable from a uniform random tuple (G, A', B', X'); this is an example of a 2-sample small-secret Module-LWE indistinguishability assumption. Then (G, A, B, C) is indistinguishable from $(G, A', B, M + A'b + c)$, which is indistinguishable from $(G, A', B', M + X')$. Finally, one cannot distinguish $M_0 + X'$ from $M_1 + X'$ when X' is chosen uniformly at random.

Sometimes s-sample Module-LWE is defined only for large secrets a. One might think that an attack against, e.g., 3-sample small-secret Module-LWE can be trivially converted into an attack against 4-sample large-secret Module-LWE as follows:

- We are given 4 Module-LWE samples for a uniform random secret r (or a secret r from any distribution; the distribution will not matter). In other words, we are given uniform random G_0, G_1, G_2, G_3, and given A_0, A_1, A_2, A_3 where $A_j = rG_j + e_j$ for small secrets e_0, e_1, e_2, e_3.

[21] It can also be an example of a small-secret LWE indistinguishability assumption, and an example of a small-secret Ring-LWE indistinguishability assumption. The naming depends on details of the algebraic structures containing G, A, etc.

- Compute $G'_j = -G_0^{-1}G_j$ and $A'_j = A_0G'_j + A_j$. The point of this computation is that $A'_j = e_0G'_j + e_j$.
- We now have uniform random G'_1, G'_2, G'_3, along with A'_1, A'_2, A'_3 where $A'_j = e_0G'_j + e_j$. These are 3 Module-LWE samples for the small secret e_0. Apply the 3-sample small-secret Module-LWE attack.

Caution is required here: what happens if G_0 is not invertible? One can try to dodge this difficulty by searching for an invertible G_j, but in general this can use more samples, possibly many more if invertibility is infrequent. For the case of LWE, [15, Lemma 2] takes as many samples as necessary to build an invertible matrix.

Now consider what happens if one replaces the ratios $G_0^{-1}G_1$, $G_0^{-1}G_2$, $G_0^{-1}G_3$ with the ratios $G_0^{-1}G_1$, $G_2^{-1}G_3$, and adjusts the A_j calculations analogously, assuming invertibility of G_0 and G_2. Instead of obtaining 3 samples with a small secret e_0, one obtains a batch of 2 independent problems with small secrets e_0 and e_2, each with 1 sample. More generally, aside from the invertibility issue, $2s$ samples reduce to a batch of s independent 1-sample problems; $3s$ samples reduce to a batch of s independent 2-sample problems; etc.

This generalized reduction appears to be the point of the brief proof outline in [47, page 3, bottom of second column]. However, the conclusions drawn in [47] go beyond this in three ways that are not justified in [47]:

- The many-sample Module-LWE problem is claimed to be difficult: "we believe that in practice the MLWE problem with k samples is no easier than with 1 sample". However, the Arora–Ge attack cited in [47, footnote 2] is already known to break the Module-LWE problem for (e.g.) the Kyber error distribution with a feasible number of samples, showing that a proof based on that problem would be vacuous for attacks against a feasible number of Kyber targets.
- Single-target and multi-target IND-CPA advantages are claimed to be proven to be close. However, this does not logically follow from a one-way conversion of IND-CPA attacks into Module-LWE attacks. The obvious way to close this gap would be to prove a conversion the other way (presumably also breaking the many-ciphertext case via the Arora–Ge attack)—but the words "plausibly much harder problem" in [47, footnote 2] indicate that a conversion is not known the other way.
- The type of multi-target attack for which [47] claims provable security is not merely an attack against many independent keys, but an attack against many ciphertexts per key. However, if one wants to convert an attack against s ciphertexts for one key (with many ciphertexts multiplied by the same public G, making G a natural target for precomputation) into an attack against $\Theta(s)$ Module-LWE samples, it is not obviously sufficient to convert an attack against s independent keys into an attack against $\Theta(s)$ Module-LWE samples.

Perhaps the second and third items have been addressed by a full proof, but the proof outline in [47] does not seem to have been written as a summary of a

full proof. For example, the stated number of samples, "$\max(n, q_C)$", does not account for losing at least n samples as denominators, and for the possibility of needing more samples for the necessary invertibility. What is cited here in [47], namely the transformation from [15], does not count the number of samples, and does not handle Ring-LWE or Module-LWE.

The set of T for which this paper gives heuristic one-out-of-T-ciphertext speedups is not covered by the set of T for which there is a $\Theta(T)$-sample Arora–Ge attack, so the fact that the proof outlined in [47] is vacuous for the number of samples covered by the Arora–Ge attack is not logically sufficient to reconcile [47] with this paper. However, if there *is* a conversion from a multi-ciphertext attack into a multi-sample Module-LWE attack, the simplest explanation is not "these attacks do not exist, except that the Arora–Ge attack suddenly breaks Module-LWE with enough samples", but rather "Module-LWE security degrades with the number of samples".

More fundamentally, it is not clear that such a conversion exists. There is a proof gap at this point in [47]. It is clear how the proof techniques mentioned in [47] apply to multiple keys, but it is not clear how those techniques justify claiming multi-ciphertext security for a single key.

These flaws were disclosed privately (in considerable detail) on 4 November 2022, and publicly (in more detail, in the first version of this paper) on 14 November 2022. At the time of this writing, there has been no response from the authors of [47]; the claim that "we can show that $\mathrm{Adv}_{\mathrm{PKE}}^{\text{IND-CPA}} \approx \mathrm{Adv}_{\mathrm{PKE}}^{(n,q_C)\text{-IND-CPA}}$" assuming "the hardness of MLWE" still appears in the latest version of [47].

B The curious case of FrodoKEM

> *We reiterate the crucial point: if the reduction proving security is "loose," like the one above, the efficiency of the scheme is impacted, because we must move to a larger security parameter.* —1996 Bellare–Rogaway [23]
>
> *In terms of security, Frodo's conservative design choices are laudable.* —2022 NIST [4, Section 4.3.1]

This appendix focuses specifically on FrodoKEM. Unusual features of the FrodoKEM design might at first seem to make FrodoKEM quantitatively more resistant to one-out-of-many-ciphertext attacks than many lattice systems are (although perhaps not quite as resistant as fixed-weight ternary systems; see Section 4.5). However, it turns out that the same design approach opens up FrodoKEM to a trivial, well-known attack strategy.

The current version of FrodoKEM includes three proposed parameter sets:[22] FrodoKEM-640, FrodoKEM-976, and FrodoKEM-1344, targeting the security

[22] Actually six, since each parameter set has an AES-based version and a SHAKE-based version. The difference has a marginal effect on the performance of the attack here.

levels of brute-force single-key search for AES-128, AES-192, and AES-256 respectively. For each parameter set, the attack highlighted in this appendix is *almost* below the target security level. This disproves the claim in [12, page 44] that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin".

When T ciphertexts are sent to one FrodoKEM public key, the same well-known attack strategy immediately produces an all-for-the-price-of-one attack, and a one-out-of-many attack that is approximately $T\times$ faster. These attacks against FrodoKEM are valid even if the existing lattice heuristics are incorrect.

B.1. Multiple independent ciphertexts. Say E is a PKE with message space X. Write E^j for the PKE with message space X^j that encrypts $(M_1,\ldots,M_j)$ by using E to independently encrypt each of $M_1,\ldots,M_j$.

An attack that completely decrypts jT ciphertexts for E is also an attack that completely decrypts T ciphertexts for E^j. But decrypting *some* of the jT ciphertexts for E is, as j increases, less and less effective as an attack against T ciphertexts for E^j.

Quantitatively, if each of the jT ciphertexts for E is decrypted with probability p, then each ciphertext for E^j is decrypted with probability just p^j. For essentially the same reason, as j increases, ROM IND-CPA for a tweak of E^j becomes provably almost as hard to break as OW-CPA for E; see [59, Section 3.4].

It is hard to imagine how this could be the best way to defend against one-out-of-many attacks (or against IND-CPA attacks). The problem is that E^2 is twice as expensive as E. Normally, for the same cost as E^2, one can instead take E with larger parameters, aiming at a much higher security target, presumably making *all* attacks more difficult. On the other hand, perhaps one is forced to use E^2 for other reasons, as in Appendix B.2.

B.2. PKEs with small message spaces. Say one is trying to build an IND-CCA2 KEM out of a limited PKE E that sends only 32-bit messages.

The limited message space does not stop E from achieving IND-CPA security: the encryption procedure can generate many more bits of randomness. However, if one builds a KEM by applying any common variant of the Fujisaki–Okamoto transform to E, then the FO derandomization step eliminates the additional randomness. Ciphertexts and session keys for any particular public key have just 32 bits of entropy and are easily breakable.

A straightforward fix, at the cost of making everything $8\times$ less efficient, is to instead apply the FO transform to E^8. Ciphertexts and session keys then have 256 bits of entropy.

This is essentially how FrodoKEM and its underlying PKE, FrodoPKE, are built. See, e.g., [12, page 17] saying "Several ($\overline{m}$) ciphertexts are generated at once". The "$\overline{m}$" parameter *could* be chosen as 1, but is instead chosen as 8 for performance reasons derived from a design goal of minimizing "the sum of the bit lengths of FrodoPKE's ciphertext and its public key"; see [12, page 23].

Concretely, FrodoKEM-1344 can be viewed as sending 8 independent ciphertexts. Each ciphertext, in turn, communicates just 4 message bits at each position in a C vector of length 8 (while B is much larger), for a total of 32 message bits in each ciphertext, or 256 bits across the 8 ciphertexts.

B.3. Multi-ciphertext attacks on FrodoKEM. One might think that this 8-ciphertext structure reduces the quantitative impact of attacks trying to decrypt one out of many FrodoKEM ciphertexts for the same public key.

Concretely, to exploit the shared lattice to break one of 2^{40} ciphertexts, one would want to break $1/2^5$ of the underlying length-8 ciphertexts. This factor 2^5 is much less concerning than the original 2^{40}, especially since FrodoKEM is a "conservative" lattice design claiming that its lattice problems have even more security than needed. However, the same performance pressure that drove FrodoKEM to use 8 ciphertexts turns out to create an even bigger security problem for FrodoKEM.

Consider FrodoKEM-640, for which [12, Table 11] estimates "LWE hardness" of $2^{175.1}$ "gates" and $2^{110.4}$ "memory in bits". This appears to be the basis of the claim in [12, page 44] that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin"—but FrodoKEM has an even larger attack surface than the underlying LWE problem does. It is an error to conflate LWE hardness with FrodoKEM security.

Like FrodoKEM-1344, FrodoKEM-640 sends 8 ciphertexts, each having 8 positions, but—again, because of performance pressure—FrodoKEM-640 has just 2 message bits per position, a total of 128 message bits.

This limited message space, combined with FO derandomization, means that there are only 2^{128} possible FrodoKEM-640 ciphertexts for any particular public key. Consequently, FrodoKEM-640 is vulnerable to a trivial brute-force search through messages.

A full search through all 2^{128} messages is much more efficient than the $2^{175.1}$ "gates" and $2^{110.4}$ "memory in bits" estimated in [12] for FrodoKEM's "LWE hardness".[23] An attacker faced with 2^{40} ciphertexts for a FrodoKEM-640 public key will successfully decrypt one of the ciphertexts after trying about 2^{88} messages, a feasible computation.

There is nothing new about this attack: it is an example of one of the workhorses of symmetric cryptanalysis, namely "guess possibilities for secret x, checking each guess against published $F(x)$ values". But the literature does not seem to have pointed out that FrodoKEM-640 opens up this attack with

[23] FrodoKEM encryption is slow, but not *that* slow. Also, computing well under 1% of each ciphertext, using well under 1% of the multiplications in $\mathbb{Z}/q$, is sufficient for the attack. Readers interested in more detailed optimization should note that the IND-CCA2 definition provides a session key to the attacker, so one can skip all the multiplications and simply check the session-key hash. Furthermore, most of the hash computation for the session key can be precomputed, since FrodoKEM's hash input puts the secret message *after* the public ciphertext. For comparison, one rule of thumb for the order of hash inputs—see, e.g., [30]—is that whatever is least likely to be predictable by the attacker should come first.

a 128-bit x; that this contradicts the claim that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin"; and that F depends only on the public key, allowing a $2^{128}/T$ attack against T FrodoKEM-640 ciphertexts for that key.

The performance of this FrodoKEM attack was announced on 31 October 2022. A request for an erratum regarding the claim quoted above was sent to the FrodoKEM team on 1 November 2022. Full attack details were provided in the first public version of this paper on 14 November 2022. At the time of this writing, there has been no response from the FrodoKEM team; the claim still appears in the latest FrodoKEM documentation.

B.4. Broader impacts. It is interesting to compare this FrodoKEM-640 attack to Appendix A.3. Recall that, in 2021, NIST criticized code-based proposals on the basis of known one-out-of-T-ciphertext attacks that save roughly $\sqrt{T}$ from the starting cost 2^b of decoding—even when the code-based proposals cited the attacks, presumed that $2^b/T$ is possible, and recommended large b as a defense. The presumed $2^b/T$ is far beyond the $2^{128}/T$ cost of the one-out-of-T-ciphertext attack against FrodoKEM-640, because b has intentionally been chosen much larger than 128.

It is also interesting to compare this FrodoKEM-640 attack to NIST's stated reason [88] for continuing to recommend AES-128 in the context of post-quantum KEMs:

> Finally, you seem to be advocating that NIST respond to the possibility of multikey attacks by withdrawing AES128, rather than by advocating for modes of operation that have built-in multi-key security. Given that
> 1) AES-128 is the most widely used block cipher at present, and it has never come even close to being practically attacked based on an insufficiently large key size.
> 2) Most widely-used high-volume protocols, where multi-key security is actually a concern (e.g. TLS and IPSec) already have built-in protections against multi-key attacks.
>
> It seems premature to pull AES128.

The protections mentioned here are randomizing the inputs to various protocols using AES so that any specific input is not encrypted under many AES keys. This fails to protect a broader system that uses FrodoKEM-640 to communicate AES-128 keys in the first place: the attack directly breaks FrodoKEM-640 ciphertexts, independently of how the resulting session keys are used.

Trying to review multi-target security of every aspect of a cryptographic system, and trying to patch every multi-target attack that is found, is much more complicated and much less robust than systematically requiring higher security levels. This particular attack is very easy, but it still appears to have been missed in the FrodoKEM design. Note that multi-target security was within scope for the FrodoKEM proposal: see, e.g., [12, page 33], claiming that a tweak elsewhere in FrodoKEM "has the potential to provide stronger multi-target security".

Moving up to FrodoKEM-976 makes these multi-target attacks infeasible today. But this is not the end of the story: FrodoKEM-976 claims a correspondingly higher security level, and claims to have "a large margin" beyond that security level. The claim of a large margin is false: FrodoKEM-976's performance-driven choice to use just 192-bit messages exposes FrodoKEM-976 to a straightforward 2^{192}-guess single-target brute-force attack, and to correspondingly more efficient multi-ciphertext attacks.

Most other noisy-DH lattice proposals (e.g., Kyber) have less performance pressure on the size of the message space and simply take 256-bit messages for all security levels. When the target security level is AES-256, the same easy attack of searching through messages means that none of these proposals have large security margins; and this attack becomes $T\times$ faster for decrypting one out of T ciphertexts to the same public key. Of course, $2^{256}/T$ is of vastly less real-world concern than $2^{128}/T$.

B.5. The importance of accounting for looseness. FrodoKEM was not the only submission to the NIST Post-Quantum Cryptography Standardization Project putting heavy emphasis on proofs. See, e.g., [38, Section 14]: "MQDSS is the first multivariate signature scheme that is provably secure, and whose security relies solely on the MQ problem."

The MQDSS proofs said that a ROM attack against MQDSS could be converted into an attack against the MQ problem, a simpler problem that had already attracted cryptanalysis. However, these proofs were *loose*. The proofs did not say that the cost of the resulting MQ attack was *close to* the cost of the MQDSS attack; the proofs merely put some sort of quantitative limit on the change in attack costs. The proofs thus allowed MQDSS to be much easier to break than the underlying MQ problem.

As Bellare and Rogaway noted in [23], applying a loose proof means that, logically, one "must move to a larger security parameter" to compensate for the looseness. But MQDSS chose cryptosystem parameters without regard to the looseness. MQDSS then had its security claims disproven by an attack [65] that exploited the looseness of the proofs. See also [69] for a survey of many more provable-security failures, including failures directly attributable to looseness.

Similarly, various loose theorems are falsely advertised in [12, Section 5] as "supporting the security of FrodoKEM". To the extent that the security of the underlying problems has been studied, these theorems would support the security of a hypothetical CarefulFrodoKEM with parameters chosen to systematically account for looseness; but it is an error to conflate the security of FrodoKEM with the security of CarefulFrodoKEM. This error has been remarkably persistent:

- I had, in [26], already pointed out this error in an earlier version of [12], and yet the same error remained in [12].
- In [29], I recommended that NIST add "procedural protections against loose proofs". In [14], a NIST team member replied "Stop propagandizing".
- Secret NIST discussions included, e.g., slides [3] from another NIST team member repeating the false advertising of FrodoKEM's proofs, categorically dismissing objections as "a lot of DJBFUD about the relevance of asymptotic

security reductions to security",[24] and not rationally addressing the objections. What makes this particularly bizarre is that the same slides also admit, regarding the advertising, that "not all of this applies for practically relevant parameters".

- A public NIST report [4] also admitted that "these theorems do not hold for the concrete parameter choices used in Frodo" but tried to defend the false advertising as follows: (1) labeling this as "typical"; (2) claiming that the theorems "do indicate some fundamental soundness in the core idea underlying the Frodo approach"—which is not the question at hand.

NIST was supposed to be asking whether FrodoKEM is secure. Merely asking whether there is "some fundamental soundness in the core idea" is a poor substitute for this; the history of cryptography is littered with the corpses of cryptosystems that have "some fundamental soundness in the core idea". NIST was also supposed to be evaluating "the quality of the analysis provided by the submitter"; having loose theorems incorrectly labeled as "supporting the security of FrodoKEM" should have been treated negatively, not positively.

To see the relevance of this error to the sudden collapse of FrodoKEM's "large margin", consider the fact that CarefulFrodoKEM would be forced to account for, among other things,

- $2q + 1$ divided by the size of the message space in [12, Lemma 5.3], where q is the number of hash calls in an attack; and
- the looseness factor T in the generic proof that a T-target attack implies a 1-target attack.

The trivial FrodoKEM attack highlighted in this appendix exploits exactly these two looseness issues. For a single ciphertext, the attack searches the message space (which is dangerously small for FrodoKEM-640, and does not have a "large margin" for any of the parameter sets), checking message guesses via hash calls; for T ciphertexts, the attack gains a factor T. The success of this attack in disproving one of FrodoKEM's security claims is thus directly attributable to FrodoKEM's failure to account for proof looseness, a failure endorsed by NIST. CarefulFrodoKEM would be bigger and slower than FrodoKEM, but it would not have this type of security failure.

According to [4, page 38], the only reason that NIST did not select FrodoKEM was performance. Other organizations are continuing to consider FrodoKEM on the basis of its "conservative" design. It is deeply concerning to see that the evaluations of FrodoKEM are based primarily upon the volume of allegedly related proofs, with no insistence upon investigating what—if anything—those proofs actually say about FrodoKEM. Much more work needs to be done to systematically study the FrodoKEM attack surface.

[24] One might wonder why there is no credit to, e.g., [23] for the BRFUD and [69] for the KMFUD.

C Proofs

As a supplement to this paper, https://cr.yp.to/2023/lprrr-20230317.ml presents computer-verified proofs for a generalization of Theorem 2.7.1. These proofs are written in the HOL Light [55] language and have been verified by the HOL Light verifier.

This appendix reports how the proofs were developed, compares the computer-verified theorem statement to the statement of Theorem 2.7.1, and explains how to re-run the verifier.

C.1. Proof development. The first public version of this paper contained 2.5 pages with a theorem and proof—at a normal mathematical level of formality, not computer-verified—determining the second-order asymptotics of StandardRatio for any particular growth of (n, q, s, κ, β). It also contained 1.5 pages informally optimizing (κ, β) and sketching how this optimization could be proven; and a separate sanity check on the optimizations.

The theorem, proof, informal optimization, and optimization-proof sketch were then upgraded to a total of 13 pages of theorems and proofs, culminating in a formal statement of the asymptotics of the optimal (κ, β).

The 13 pages of theorems and proofs were then upgraded to computer-verified proofs in the HOL Light language—along with proofs of many necessary lemmas. For example, the proofs in the 13 pages (and in the original 2.5 pages) had commented that "$1/(A_0 + (A_1 + o(1))/\lg n)$ is $1/A_0 + (-A_1/A_0^2 + o(1))/\lg n$ if $A_0 \neq 0$", where the $o(1)$ is as $n \to \infty$. Inside `lprrr-20230317.ml`, the computer-verified proof of this comment

- starts from the mean-value theorem (which is already proven in the HOL Light library);
- spends 25 lines stating and proving a more suitable two-sided version of the mean-value theorem;
- spends 74 lines stating and proving that if f is continuously differentiable at A_0 then anything in $f(A_0+(A_1+o(1))/X)$ is in $f(A_0)+(A_1f'(A_0)+o(1))/X$ where the $o(1)$ is as $X \to \infty$;
- spends 39 lines specializing the statement and proof to inversion; and
- spends 17 lines specializing the statement and proof to $X = \lg n$.

Overall `lprrr-20230317.ml` occupies 345KB, nearly 10000 lines. Some of the proofs were generated by ad-hoc scripts. There are a few comments, partly for tracking the internal organization of `lprrr-20230317.ml` and partly about proofs of one of the lemmas (Bolzano's theorem). The time for writing `lprrr-20230317.ml` was an unrecorded fraction of a 3.5-week period. No claims of optimality are made for the numbers 345, 10000, and 3.5.

In this version of the paper, the formal theorem statement is Theorem 2.7.1, the computer-verified proofs are supplied separately, and the separate sanity check is Section 5.

C.2. The value of computer verification. Given that the literature presents merely heuristic arguments that the standard block size scales similarly to the actual block size required for attacks, the reader might be wondering why this paper puts so much effort into eliminating risks of error in this paper's statements about the standard block size.

One answer is that the literature often describes the standard block size as an accurate approximation to the actual block size. Consider, e.g., [7] saying that the existing heuristics were "empirically investigated and confirmed" and that various discrepancies disappear as problem sizes increase. Readers who trust the existing heuristics, on the basis of current evidence or evidence collected in the future, can—thanks to the computer-verified proofs—place the same trust in conclusions obtained by combining the heuristics with Theorem 2.7.1.

Another answer is as follows. Consider a reader faced with papers saying (1) that the existing heuristics are accurate, (2) that multi-target security matches single-target security, and (3) that #1 contradicts #2. A reader who sees that there is just one paper saying #3, and that the paper relies on pages of calculations, can easily hypothesize that #3 was spoiled by a mistake somewhere in those calculations, while #1 and #2 are fine. This concern is directly addressed by computer-verified proofs.

Note that computer-verified proofs do not remove the value of the sanity check in Section 5. The sanity check provides information regarding the behavior of concrete block sizes, and tools for scalable numerical optimization of block sizes.

C.3. The statement of the computer-verified theorem. There are two main theorems in `lprrr-20230317.ml`: `forward_main` is a generalization of Theorem 2.7.1(1), and `converse_main` is a generalization of Theorem 2.7.1(2).

A reader checking that Theorem 2.7.1 has been computer-verified must check the statements of `forward_main` and `converse_main`, along with the underlying definitions. The following paragraphs review the definitions and the theorem statements, without assuming familiarity with HOL Light.

```
let log2 = new_definition `
  log2 x = (ln x) / (ln (&2))
`;;
```

The HOL Light library already defines a function `ln`; these lines define a function `log2`. In the HOL Light language, natural numbers such as `2` are distinguished from real numbers such as `&2`. Parentheses can often be omitted but are included here for clarity.

```
let ceil = new_definition `
  ceil x = -- floor(--x)
`;;
```

This defines `ceil` on top of the function `floor` defined in the HOL Light library: $\lceil x \rceil = -\lfloor -x \rfloor$. In the HOL Light language, `--` is negation.

```
let o1_seq = new_definition `
  o1_seq (f:num->real) <=>
    !e:real. &0 < e ==>
      ?m:num.
        !i:num. m <= i ==> abs(f(i)) <= e
`;;
```

Let f be a function from $\{0, 1, 2, \ldots\}$ to $\mathbb{R}$. This definition says that `o1_seq` f is the following statement: for every $e \in \mathbb{R}$ with $0 < e$, there exists $m \in \{0, 1, 2, \ldots\}$ such that every $i \in \{0, 1, 2, \ldots\}$ with $m \leq i$ has $|f(i)| \leq e$. This is one of the traditional ways to say that f converges to 0, i.e., that $f \in o(1)$.

This is equivalent to a special case of a concept `tends_num_real` in the HOL Light library. There is only a small overlap between `o1_seq` theorems in `lprrr-20230317.ml` and `tends_num_real` theorems already in the library.

As this definition illustrates, in the HOL Light language, "`!x.`" means "for every `x` we have"; "`?x.`" means "there exists `x` such that"; "`:num`", "`:real`", and "`:num->real`" specify types. Often HOL Light can deduce types automatically, but including the types can still add clarity.

```
parse_as_infix("powreal",(24,"left"));;
let powreal = new_definition `
  x powreal y = exp(y * ln x)
`;;
```

This defines x `powreal` y as $\exp(y \ln x)$, i.e., x^y.

```
let bkzdelta = new_definition `
  bkzdelta x =
    (x * ((pi*x) powreal (&1 / x))/(&2 * pi * exp(&1)))
    powreal (&1 / (&2 * (x - &1)))
`;;
```

For comparison, $\delta = (\beta(\pi\beta)^{1/\beta}/(2\pi \exp 1))^{1/2(\beta-1)}$ inside Definition 2.5.1.

```
let standardratio = new_definition `
  standardratio n q s k x =
     ( ((n + k)*(s pow 2) + &1) powreal (&1 / &2)
     ) / ( (((n + k + &1)/x) powreal (&1 / &2))
           * ((bkzdelta x) powreal (&2 * x - (n + k + &1) - &1))
           * (q powreal (k/(n + k + &1)))
         )
`;;
```

Compare StandardRatio$(n, q, s, \kappa, \beta) = ((n+\kappa)s^2+1)^{1/2}/(d/\beta)^{1/2}\delta^{2\beta-d-1}q^{\kappa/d}$ in Definition 2.5.1, with $d = n+\kappa+1$ and δ as above. In the HOL Light language, `pow` is exponentiation with a natural-number exponent; there is also a `sqrt(...)` that could be used in place of `(...) powreal (&1 / &2)`.

```
let forward_main = prove(`
  !n:num->real q:num->real s:num->real
   Q0:real Q1:real S0:real S1:real
   x0:real z0:real z1:real.
```

This is the start of the first main theorem statement. In general, the statement looks like "`!X. A /\ B ==> ?Y. C /\ D`", meaning that, for every `X` where the hypotheses `A` and `B` hold, there exists `Y` where the conclusions `C` and `D` hold. In the HOL Light language, `==>` is implication, and `/\` is conjunction.

In Theorem 2.7.1, n runs through a specified infinite subset N of $\{2, 3, \ldots\}$. To match this up to the more general `n:num->real` allowed in `forward_main`, define n_0 as the smallest element of N, define n_1 as the next element of N, etc.

In Theorem 2.7.1, q and s are determined by n. The setting of `forward_main` is more general, allowing $n_i = n_j$ with $q_i \neq q_j$ or $s_i \neq s_j$.

```
(!i. &1 < n(i))
/\ (!i. &1 < q(i))
/\ (!i. &0 < s(i))
/\ o1_seq (\i. &1 / n(i))
/\ o1_seq (\i. (log2(q(i))/log2(n(i)) - Q0) * log2(n(i)) - Q1)
/\ o1_seq (\i. (log2(s(i))/log2(n(i)) - S0) * log2(n(i)) - S1)
```

These hypotheses say that each n_i is larger than 1; each q_i is larger than 1; each s_i is larger than 0; $1/n_i \in o(1)$ as $i \to \infty$; $((\lg q_i)/\lg n_i - Q_0) \lg n_i - Q_1 \in o(1)$, i.e., $\lg q_i \in Q_0 \lg n_i + Q_1 + o(1)$; and $\lg s_i \in S_0 \lg n_i + S_1 + o(1)$.

If n_i runs through the elements of N in order, with N as in Theorem 2.7.1, then $n_i \to \infty$ as $i \to \infty$, and $1/n_i \in o(1)$. Also, Theorem 2.7.1 assumes $\lg q \in Q_0 \lg n + Q_1 + o(1)$ and $\lg s \in S_0 \lg n + S_1 + o(1)$.

```
/\ -- &1 / &2 < S0
/\ S0 <= &1 / &2
/\ &1 / &2 < Q0 - S0
/\ &1 / &2 < Q0 + S0
```

These hypotheses say $-1/2 < S_0 \leq 1/2$, $1/2 < Q_0 - S_0$, and $1/2 < Q_0 + S_0$. All of these are satisfied in Theorem 2.7.1, which requires $0 \leq S_0 \leq 1/2 < Q_0 - S_0$.

```
/\ x0 = (Q0 + S0 - &1 / &2)/(Q0 - S0 + &1 / &2)
/\ z0 = &2 * Q0/((Q0 - S0 + &1 / &2) pow 2)
/\ z1 =   (&2 * S1 + log2(z0)
           - (S0 - Q0 + &3 / &2) * (log2(z0) - log2(&2 * pi * exp(&1)))
           - Q1 * (Q0 + S0 - &1 / &2)/Q0
          )
        * (&2 * Q0) / ((Q0 - S0 + &1 / &2) pow 3)
```

For comparison, Theorem 2.7.1 says $x_0 = (Q_0 + S_0 - 1/2)/(Q_0 - S_0 + 1/2)$; $z_0 = 2Q_0/(Q_0 - S_0 + 1/2)^2$; and

$$z_1 = \left(2S_1 + \lg z_0 - \left(S_0 - Q_0 + \frac{3}{2}\right) \lg \frac{z_0}{2\pi \exp 1} - \frac{Q_1(Q_0 + S_0 - \frac{1}{2})}{Q_0}\right) \frac{2Q_0}{(Q_0 - S_0 + \frac{1}{2})^3}.$$

The constant `pi` is provided by the HOL Light library.

```
==> ?k:num->real b:num->real.
```

This says that, if the above hypotheses are satisfied, then there exist functions k, b from $\{0, 1, 2, \ldots\}$ to $\mathbb{R}$ satisfying the conclusions that follow.

If n is an injective function on $\{0, 1, 2, \ldots\}$ then the functions $i \mapsto k_i$ and $i \mapsto b_i$ are determined by functions $n \mapsto k$ and $n \mapsto b$; Theorem 2.7.1 is phrased in terms of the latter functions.

```
(!i. integer(k(i)))
/\ (!i. &0 < k(i))
/\ (!i. k(i) <= ceil(n(i)))
/\ o1_seq (\i. (k(i)/n(i) - x0) * log2(n(i)) - &0)
```

This says that each k_i is an integer, that $0 < k_i \leq \lceil n_i \rceil$, and that $k_i/n_i \in x_0 + o(1)/\lg n_i$. In particular, $1 \leq k_i$, and $k_i \leq n_i$ if n_i is an integer, the situation of Theorem 2.7.1.

```
/\ (!i. integer(b(i)))
/\ (!i. &1 < b(i))
/\ (!i. b(i) <= ceil(n(i) + k(i) + &1))
/\ o1_seq (\i. (b(i)/n(i) - z0) * log2(n(i)) - z1)
```

This says that each b_i is an integer, that $1 < b_i \leq \lceil n_i + k_i + 1 \rceil$, and that $b_i/n_i \in z_0 + (z_1 + o(1))/\lg n_i$. In particular, $2 \leq b_i$, and $b_i \leq n_i + k_i + 1$ if n_i is an integer (since k_i is also an integer).

```
/\ ?m. !i. m <= i ==>
   standardratio (n(i)) (q(i)) (s(i)) (k(i)) (b(i)) < &1
`,
...
```

This covers the last conclusion of Theorem 2.7.1(1): there is some m such that every $i \geq m$ has StandardRatio$(n_i, q_i, s_i, k_i, b_i) < 1$.

A proof in `lprrr-20230317.ml` has been replaced with `...` here. The main point of computer verification is that the reader does not need to check the proof.

```
let converse_main = prove(`
  !n:num->real q:num->real s:num->real
   k:num->real b:num->real
   Q0:real Q1:real S0:real S1:real
   x0:real z0:real z1:real.
```

This starts the other main theorem statement, generalizing Theorem 2.7.1(2). Note the extra `k` and `b` here.

```
(!i. &1 < n(i))
/\ (!i. &1 < q(i))
/\ (!i. &0 < s(i))
/\ o1_seq (\i. &1 / n(i))
/\ o1_seq (\i. (log2(q(i))/log2(n(i)) - Q0) * log2(n(i)) - Q1)
/\ o1_seq (\i. (log2(s(i))/log2(n(i)) - S0) * log2(n(i)) - S1)
```

This is exactly the same as in the first main theorem statement.

```
/\ -- &1 / &2 < S0
/\ S0 <= &1 / &2
/\ -- &1 / &2 < Q0 - S0
/\ &1 / &2 < Q0 + S0
```

This is more generous than in the first main theorem statement: this requires merely $-1/2 < Q_0 - S_0$, not $1/2 < Q_0 - S_0$.

```
/\ x0 = (Q0 + S0 - &1 / &2)/(Q0 - S0 + &1 / &2)
/\ z0 = &2 * Q0/((Q0 - S0 + &1 / &2) pow 2)
/\ z1 =   (&2 * S1 + log2(z0)
           - (S0 - Q0 + &3 / &2) * (log2(z0) - log2(&2 * pi * exp(&1)))
           - Q1 * (Q0 + S0 - &1 / &2)/Q0
          )
        * (&2 * Q0) / ((Q0 - S0 + &1 / &2) pow 3)
```

This is again exactly the same as in the first main theorem statement.

```
/\ (!i. &0 < k(i))
/\ (!i. k(i) <= &100 * n(i))
/\ (!i. &60 <= b(i))
/\ (!i. b(i) <= ceil(n(i) + k(i) + &1))
/\ (?m. !i. m <= i ==>
    standardratio (n(i)) (q(i)) (s(i)) (k(i)) (b(i)) <= &1)
```

This says $0 < k_i \le 100n_i$ and $60 \le b_i \le \lceil n_i + k_i + 1 \rceil$. These inequalities are satisfied if $1 \le k_i \le 100n_i$ and $60 \le b_i \le n_i + k_i + 1$, as in Theorem 2.7.1(2).

This also says that, for all suficiently large i, StandardRatio$(n_i, q_i, s_i, k_i, b_i) \le 1$. This is assumed by Theorem 2.7.1(2).

```
==>
?L. (!i. L(i) <= b(i))
/\ o1_seq (\i. (L(i)/n(i) - z0) * log2(n(i)) - z1)
`,
...
```

For comparison, the conclusion of Theorem 2.7.1(2) is that $\beta \ge \ell$ for some function $n \mapsto \ell$ with $\ell/n \in z_0 + (z_1 + o(1))/\lg n$.

C.4. Redoing the computer verification. Readers are cautioned that, beyond the portion of HOL Light responsible for verifying theorems, there are many more lines of code in the HOL Light library providing proof tools and specific proofs—and perhaps doing something else, since all of this is written in a general-purpose programming language. Malicious code in HOL Light or in this paper's `lprrr-20230317.ml` could exfiltrate secret files, install ransomware, or, perhaps most terrifyingly, output a "`thm`" that has not, in fact, been proven.

The following commands have been tested on an Ubuntu 22.04 system (which requires the `--disable-sandboxing`) and on a Debian Bookworm system. These commands download the HOL Light development package (rather than using the HOL Light package built into Bookworm), and should work on a wider range of Linux distributions, as long as the `apt` line is adapted appropriately.

```
sudo apt install opam wget -y

time opam init -a --disable-sandboxing
time opam switch create 4.05.0
eval `opam env`
time opam pin add camlp5 7.10 -y
time opam install num camlp-streams ocamlfind -y

git clone https://github.com/jrh13/hol-light
cd hol-light
git checkout 1a1de6ce7a6e9f60bec8bc501c426836d0e6b231
make

wget https://cr.yp.to/2023/lprrr-20230317.ml
time ocaml -I `camlp5 -where` camlp5o.cma -init hol.ml \
< lprrr-20230317.ml > lprrr-20230317.out
```

On the Ubuntu 22.04 system (with an AMD FX-8350 CPU), the timed commands were observed to take 39 seconds, 314 seconds, 72 seconds, 187 seconds, and 365 seconds respectively. The reader can check that the resulting `lprrr-20230317.out` file includes the definitions and theorems shown above, each certified by HOL Light to be a `thm`.