Estimating the Security of Lattice-based Cryptosystems

Abstract. Encryption and signature schemes based on worst-case lattice problems are promising candidates for the post-quantum era, where classic number-theoretic assumptions are rendered false. Although there have been many important results and breakthroughs in lattice cryptography, the questions of how to systematically evaluate their security in practice and how to choose secure parameters are still open. This is mainly due to the fact that most security proofs are essentially asymptotic statements. In addition, the hardness of the underlying complexity assumption is controlled by several interdependent parameters rather than just a simple bit length as in classic schemes.

With our work, we close this gap by providing a handy framework that (1) distills a hardness estimate out of a given parameter set and (2) relates the complexity of practical lattice-based attacks to symmetric "bit security" for the first time. Our approach takes various security levels, or attacker types, into account. Moreover, we use it to predict long-term security in a similar fashion as the results that are collected on `www.keylength.com`. In contrast to the experiments by Gama and Nguyen (Eurocrypt 2008), our estimates are based on precisely the family of lattices that is relevant in cryptography.

Our framework can be applied in two ways: Firstly, to assess the hardness of the (few) proposed parameter sets so far and secondly, to propose secure parameters in the first place. Our methodology is applicable to essentially all lattice-based schemes that are based on the learning with errors problem (LWE) or the small integer solution problem (SIS) and it allows us to compare efficiency and security across different schemes and even across different types of cryptographic primitives.

Keywords. Lattice-based cryptography, post-quantum cryptography, Lenstra Heuristic

1. Introduction

Lattice-based cryptography has received a lot of attention in the last couple of years. Not only because Gentry solved the long-standing problem of fully homomorphic encryption [Gen09], but mainly because people were, for the first time, able to base security on worst-case assumptions rather than on average-case assumptions. This was first pointed out by Ajtai [Ajt96] in a worst-case to average-case reduction. In other words, successfully attacking a random instance of a cryptosystem immediately implies being able to solve *all* instances of the underlying problem, such as finding short vectors in *all* lattices.

In addition, these lattice problems are considered to withstand quantum-computer attacks, whereas factoring or discrete-logarithm-based systems are rendered insecure by the work of Shor [Sho97]. Another desirable trait of lattice problems is that they, unlike factoring, withstand subexponential-time attacks.

However, the above advantages come at a price. Usually, the bit lengths of the involved keys are at least $\mathcal{O}(n^2 \log(n))$, where n is the natural system parameter. Fortunately, we can use ideal lattices, introduced by Micciancio [Mic07] as well as Peikert and Rosen [PR06], that reduce the key size to $\mathcal{O}(n \log(n))$ bits. Thus, in practice, choosing n as small as possible is crucial. To the best of our knowledge, there is no work that systematically deals with selecting secure parameters or analyzing the hardness of the employed assumptions. Indeed, the task is more involved than in the case of, say, RSA.

Lattice cryptosystems have numerous parameters that affect security and dealing with n alone is not sufficient.

So far, only Micciancio and Regev [MR08], Lyubashevky [Lyu09], as well as Lyubashevky and Micciancio [LM08] have proposed parameters for their schemes. In [MR08, Lyu09], this choice is based on an interesting observation by Gama and Nguyen [GN08b]. They consider the Hermite Short Vector Problem HSVP with parameter $\delta > 0$ in lattices L of dimension d. There, the task is to find a vector $\mathbf{v}$ with $0 < \|\mathbf{v}\|_2 \leq \delta^d D(L)^{1/d}$, where $D(L)$ is a lattice constant. In [GN08b], the authors analyze "random lattices" according to the Goldstein-Mayer distribution [GM03] that are considered to provide hard instances of HSVP. Their observation is that δ is the dominating parameter and d only plays a minor role. They conjecture that HSVP seems reachable for $\delta \approx 1.01$ and "totally out of reach" for $\delta < 1.005$ in dimensions $d \geq 500$ if the lattice does not have a special structure.

The good news is that, given d, the hardness estimate δ could be determined from the security proof for the cryptosystem. The bad news is that *cryptographic*, typically called *q-ary*, lattices have a particular structure that can be exploited in attacks. E.g., Micciancio and Regev describe this sublattice attack in [MR08]. The bottom line is that solving δ-HSVP in q-ary lattices of dimension m is only as hard as solving δ'-HSVP in dimension $d < m$ and $\delta' > \delta$. Thus, HSVP becomes strictly easier in q-ary lattices because there is a certain "slack" in the required attack dimension.

With this knowledge, two unsatisfying options remain. The *first* involves Ajtai's worst-case to average-case reduction or its improvements [MR07, GPV08]. One could interpret the results of Gama and Nguyen as observations about the worst-case problem. Ajtai's worst-case problems are in dimension n, while the typical attack against the cryptosystem needs to work in dimension $\Omega(\sqrt{n \log(n)})$. Hence, this approach would work but it is overly conservative and the resulting parameters would be impractical. The *second* possibility is using the results of Gama and Nguyen in dimension d, while demanding that $\delta < 1.01$ for security against current means. Basically, this is the methodology in [MR08, Lyu09] but it only offers a *yes/no* certificate, i.e., the parameter set is either secure or insecure. In particular, it does not offer security levels, such as 100 bits, meaning that the attack effort should be close to 2^{100} storage times computation units.

With our work, we intend to provide a *third* option, with a focus on lattice-based encryption [Reg09, GPV08, Pei09, SSTX09, LPR10] and signature schemes [GPV08, SSTX09, Lyu09, LM08, CHKP10, Boy10] because they are the main building blocks of public-key cryptography. Nevertheless, our results can be easily applied to more advanced schemes, such as identity-based encryption [GPV08], oblivious transfer [PW08, PVW08], collision resistant hashing [LM06, ADL$^+$08], secret key delegation [CHKP10], and others.

We do not consider ad-hoc constructions like NTRU [HPS98] that fall outside the category of schemes motivated by Ajtai's work. The lattices that correspond to attacks on NTRU have a particular structure and contain *essentially* one unusually short "trapdoor" vector. Random Ajtai, or q-ary, lattices do not admit such a structure.

Apart from choosing secure parameters, we often wish to compare schemes with regard to their security level. Say, we have scheme X and a new scheme Y, which is more efficient than X in the sense that its public key is smaller, but at the expense of a stronger assumption. For a fair comparison, we first need a methodology to generate parameter sets that yield comparable security levels. Now, two things could happen: (1) the improvements in Y are still noticeable or (2) due to the stronger assumption, Y requires, say, a larger dimension that effectively nullifies the proposed improvement.

Our Contribution. Inspired by the works of Lenstra and Verheul [LV01] and the subsequent update by Lenstra [Len05], we propose a unified methodology for estimating security and selecting secure parameters for *all* modern lattice-based cryptography. To this end, we adopt the handy notion of dollar-days, i.e., equipment cost in dollar times attack time in days, as introduced in [Len05]. Our methodology also includes 3 different attacker types, ranging from a resource-constrained "Hacker" to an all-powerful "Intelligence agency".

We follow a modular three-tier approach: core analysis, experiments, and application.
Tier 1: At the core, there are our conjectures and observations about how the various parameters for LWE and SIS influence the hardness of these problems in Section 3. In addition, via the duality of LWE and SIS, we translate LWE instances into the language of SIS. Here, we manage to distill the hardness into one single parameter.
Tier 2: Then, we establish a relation between the attack effort in practice and this single hardness parameter by running a large number of experiments. In particular, this relation offers a way to determine the equivalent symmetric bit-security. This is done by running practical attacks on feasible instances of SIS, followed by a conservative extrapolation in Section 3. Like Gama and Nguyen [GN08b] did in a different context, we observe that the complexity of lattice-based attacks is mainly governed by δ. Therefore, we propose a function $T(\delta)$ that estimates the attack complexity in dollar-days for $\delta \in (1, 1.02]$ in Section 3. There, we also demonstrate that current records in practical lattice basis reduction support our findings. The underlying experiments can be easily replaced as soon as there are more powerful algorithms. The other two tiers stay unchanged. Notice that new experiments are *not* required if the algorithmic improvements are already covered by our double-Moore Law, i.e., we already anticipate new attacks. Interestingly, our estimation shows that, today, $\delta = 1.009$ is potentially reachable with an effort of 40 million dollar-days. However, even a powerful intelligence agency with over 100 billion dollar-days of resources should not be able to reach $\delta = 1.005$ before the year 2050.
Tier 3: The third part is the application of our framework to cryptographic schemes in Section 4. There are various potential applications. We can evaluate the security of proposed parameter sets in the literature and find that some do not provide sufficient security. Similarly, we can use our formulae in the reverse direction to output parameter sets for a given security level. Thus, we can make absolute statements about individual cryptosystems, saying that schemes X with parameter set $P(X)$ is secure against a certain type of attacker until the year 2030. In addition, we can also make relative statements across different SIS- and LWE-based schemes. For example, saying that SIS scheme X with parameters $P(X)$ is *more*, *less*, or *as* as secure as LWE scheme Y with parameters $P(Y)$. This allows a fair and easy comparison, especially when new schemes are presented, and it also allows us to match the security level of various primitives when used in a more complex protocol.

As an aside, we show a couple of interesting ideal (or ring) variants that have not been written down explicitly before in the Appendix. In our opinion, three findings are particularly interesting. The first is regarding ring-LWE, due to Lyubashevsky et al. [LPR10]. Using ideal lattices typically improves bandwidth but our multi-bit ring-LWE and dual ring-LWE schemes demonstrate that ideal lattices make the ciphertext larger and, when using hybrid encryption, they may waste space because the plaintext space is larger than necessary. Also, when using ideal lattices in LWE, one requires a significantly larger modulus. The second observation is that signature and encryption schemes that require a short trapdoor-basis are rather impractical, mainly due to their huge, often gigabyte-sized secret key. The result of Stehlé et al. [SSTX09] can improve this situation to some extent. However, one needs to keep in mind that the signing procedure [GPV08, Pei10] for GPV, Bonsai, Ideal-GPV, and Ideal-Bonsai is rather inefficient as it involves a Gram-Schmidt orthogonalization of the secret trapdoor matrix in high dimensions. Finally, we would like to remark that when combining [SSTX09] and [LPR10] to obtain an ideal version of trapdoor-LWE [GPV08], where the decision-LWE problem is hard[1], there is a caveat. The parameter relations required for [LPR10] are within the worst-case for the trapdoor generation algorithm in [SSTX09]. As a result, one needs to resort to a sub-optimal setup for trapdoor generation with rather large dimensions.

[1] The trapdoor-LWE construction in [SSTX09] only offers hardness of the search-LWE problem, making it necessary for them to use generic hardcore bits and a subexponential-time reduction.

2. Preliminaries

We denote with log the logarithm to base e, all other logarithms are specified, e.g., $\log_2$. Vectors and matrices are written in boldface, e.g., $\mathbf{v}$ and $\mathbf{M}$. The norm of a matrix $\mathbf{M}$ is defined to be $\|\mathbf{M}\| = \max_i \|\mathbf{m}_i\|$, with $\mathbf{m}_i$ being the columns of $\mathbf{m}$. We write $\|\mathbf{v}\|$ for the Euclidean norm.

2.1. Lattices

In this work, we only require full-dimension lattices. A (full-dimensional) lattice in $\mathbb{R}^n$ is a discrete subgroup $\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i \,|\, x_i \in \mathbb{Z}\}$, typically represented by a matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{n \times n}$ of $\mathbb{R}$-linearly independent vectors. The matrix $\mathbf{B}$ is a basis of the lattice Λ and we write $\Lambda = \Lambda(\mathbf{B})$. The number of linearly independent vectors in any such basis is the dimension $\dim(\Lambda)$ of the lattice. Given any basis $\mathbf{B}$ of the lattice Λ, the determinant $\det(\Lambda)$ of the lattice is $\sqrt{\det(\mathbf{B}^t\mathbf{B})}$. It is an invariant of the lattice. Another set of invariants is the successive minima. The i-th successive minimum $\lambda_i(\Lambda)$ is the smallest radius of a sphere that contains i linearly independent vectors in Λ. For a lattice $\Lambda(\mathbf{B})$ with $\mathbf{B} \in \mathbb{R}^{n \times n}$ define the (full-dimensional) dual lattice as the set of all $\mathbf{x} \in \mathbb{R}^n$ with $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$ for all $\mathbf{y} \in \Lambda(\mathbf{B})$.

Problems. One of the main computational problems in lattices is the approximate shortest vector problem (SVP). Given a basis $\mathbf{B}$ of Λ and an approximation factor $\gamma \geq 1$, the task is to find a non-zero vector $\mathbf{v} \in \Lambda$ with $\|\mathbf{v}\|_2 \leq \gamma \lambda_1(\Lambda)$. A related problem is the approximate shortest independent vector problem (SIVP), where given a basis $\mathbf{B}$ of Λ and an approximation factor γ, one is supposed to find a set $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of linearly independent vectors in Λ such that $\max_i \|\mathbf{v}_i\|_2 \leq \gamma \lambda_n$. For approximation factors exponential in $\dim(\Lambda)$, the problem is solvable in polynomial time (in $\dim(\Lambda)$) by the LLL algorithm [LLL82] for approximation factors bigger than $(4/3)^{\dim(\Lambda)}$. Using the block-wise algorithms of [Sch87, GHGKN06, GN08a], even sub-exponential approximation factors are reachable in polynomial time.

For polynomial approximation factors, which are relevant for cryptography, the best known algorithms are exponential (space and time) [AKS01, MV10]. The algorithm mostly used in practice is the BKZ algorithm [SE94]. Unfortunately, there is no theoretical average-case analysis of BKZ that could be used for determining its complexity.

In cryptography, we use lattices of a special form, which we call *q-ary*: let $q \in \mathbb{N}$, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A}\,\mathbf{v} \equiv \mathbf{0} \pmod{q}\}$. Its, up to scaling, dual lattice $\Lambda_q(\mathbf{A})$ is defined as $\{\mathbf{w} \in \mathbb{Z}^n : \exists \mathbf{e} \in \mathbb{Z}^m\, \mathbf{A}^t\mathbf{e} \equiv \mathbf{w} \pmod{q}\}$, i.e., we have $1/q \cdot \Lambda_q^{\perp}(\mathbf{A}) = (\Lambda_q(\mathbf{A}))^*$. For a randomly chosen $\mathbf{A}$, prime q, and $m > n$, the determinant of the corresponding q-ary lattice is q^n with high probability and typically, we have $m = \Omega(n \log(n))$. A second type of cryptographic lattices are ideal lattices, which can also be represented as a q-ary lattice.

The main computational problem in a q-ary lattice $\Lambda_q^{\perp}(\mathbf{A})$ is the "short integer solution" problem (SIS): given $n, m, q, \mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and a norm bound $1 \leq \nu < q$, find $\mathbf{v} \in \Lambda_q^{\perp}(\mathbf{A})$ with $0 < \|\mathbf{v}\|_2 \leq \nu$.[2] Basically, the SIS was introduced and analyzed by Ajtai [Ajt96] but there are numerous improvements to the analysis in, e.g., [MR07, GPV08]. For $\Lambda_q(\mathbf{A})$, we consider the "learning with errors“ problem (LWE): given $n, m, q, \mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and m "noisy“ inner products $\mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e} \bmod q$, where the components of $\mathbf{e}$ are chosen from a centered, discretized normal distribution χ_α over $\mathbb{Z}_q$ with standard deviation $\alpha q/\sqrt{2\pi}$. The task is to recover $\mathbf{s} \in \mathbb{Z}_q^n$. Stated differently, given $\mathbf{A}, \mathbf{b}$, solve the bounded distance decoding problem that is similar to finding the closest lattice vector to $\mathbf{b}$ because $\mathbf{w} = \mathbf{A}^t s$ is a lattice vector that is close to $\mathbf{b}$. Given $\mathbf{w}$, one can easily recover $\mathbf{s}$ by linear algebra. This search version of LWE is at least as hard as solving the decision problem, i.e., distinguish $(\mathbf{A}, \mathbf{b})$ from uniform. The problems for ideal lattices are defined analogously.

Algorithmic View. In order to grasp lattice reduction algorithmically, the notion of Hermite-SVP (HSVP) approximation seems more adequate than that of approximate SVP. In practice, it is unlikely

[2] We can restrict the problem to $\nu < q$ because the length-q vector $(0, \ldots, 0, q, 0, \ldots, 0)$ is always in the lattice.

that λ_1 is known, therefore it is impossible to check the SVP-condition $\|\mathbf{v}\|_2 \leq \gamma\lambda_1(\Lambda)$. HSVP asks for a non-zero vector that satisfies $\|\mathbf{v}\|_2 \leq \delta^{\dim(\Lambda)} \det(\Lambda)^{1/\dim(\Lambda)}$ for a given $\delta > 0$, which can be easily verified without knowing λ_1.

Concerning the hardness of this problem, the lattice dimension certainly plays a role but Nguyen and Gama show that δ is the dominating parameter. For random Goldstein-Mayer lattices, Gama and Nguyen argue that $\delta = 1.01$ seems to be an approximate limit for today's lattice basis reduction algorithms, even in high dimensions. For significantly smaller δ, the problem is intractable. This shows that, from a theoretical point of view, δ can be considered to be the main parameter controlling the hardness of HSVP. However, in cryptanalysis, we do not deal with random Goldstein-Mayer lattice bases that have very large entries of bit length $\Omega(\dim(\Lambda))$, that are usually used to analyze lattice reduction algorithms. We rather have bases with entries of bit length $\log_2(q) = \Omega(\log_2(n))$. Here, lattice reduction is potentially easier as we will discuss in the following.

Average-case Hardness. Both, LWE and SIS, are treated as average-case problems that are directly related to cryptographic schemes with a randomly chosen matrix $\mathbf{A}$. By a worst-case to average-case reduction, they are provably at least as hard as *all* instances of SIVP in dimension n. In Section 3.2, we discuss how LWE can be interpreted as SIS in a related lattice.

Each instance of SIS can be naturally interpreted as an instance of the Hermite-SVP. Given SIS with (n, m, q, ν), we compute $\delta = \sqrt[m]{\nu/q^{n/m}}$ and ask the Hermite-SVP solver to find $\mathbf{v}$ with $0 < \|\mathbf{v}\|_2 \leq \delta^m q^{n/m}$. However, this direct translation is not the best possible attack. In [MR08], Micciancio and Regev point out that one can solve the same problem in a significantly lower lattice dimension. They assume the existence of a δ-HSVP solver for a fixed δ. Then, they argue that the optimum dimension for solving SIS with (n, m, q) with this solver is $d = \min\{\sqrt{n \log(q)/\log(\delta)}, m\}$. Now, one removes $m - d$ random columns from $\mathbf{A}$ to obtain $\mathbf{A}'$, reduce the d-dimensional lattice bases of $\Lambda_q^\perp(\mathbf{A}')$, and pad a short vector therein with zeros. The result is a rather sparse vector of norm $\leq \delta^d q^{n/d}$ in $\Lambda_q^\perp(\mathbf{A})$.

Unfortunately, this approach is not directly applicable to cryptography because in practice, when attacking a cryptosystem, the attacker will also take ν into account and employ stronger and stronger HSVP solvers until a sufficiently short vector is found. Therefore, we need a re-interpretation of the approach taken in [MR08] that involves ν instead of δ. This re-interpretation allows us to *normalize* SIS(n, m, q, ν) by removing the "slack“ in the dimension parameter m. The resulting distribution of lattices is what we will analyze by directly applying lattice basis reduction. We defer the details to Section 3.

Notice that the bases of ideal lattices have essentially the same structure and there is no lattice basis reduction algorithm that can take significant advantage of the ideal structure. Therefore our analysis carries over.

Worst-case Hardness. One might argue that, since there is a worst-case to average-case reduction, one might simply treat Goldstein-Mayer lattices as worst-case lattices, apply the reduction, and analyze the hardness of HSVP in dimension n in Goldstein-Mayer lattices with an appropriate δ. However, this leads to security estimates that are too conservative because the worst-case to average-case reduction seems far from tight, with respect to the involved lattice dimension and the approximation factor. Nevertheless, the worst-case to average-case reduction helps in choosing sensible parameters for the analyzed cryptosystems.

2.2. Lenstra's Heuristic

The authors of [ECR09] describe an attacker model with attacker classes according to [BDR+96]; a subset of these classes is shown in Table 1. We add an attacker called “Lenstra”, with an amount of 40M dollar-days, which was the value for a suitable attacker proposed by Lenstra in [Len05]. Following the work of A.K. Lenstra and Verheul in [LV01], A.K. Lenstra proposed a slightly simplified framework to choose secure cryptographic parameters in [Len05]. Let k be the security parameter and assume the

Attacker class	Budget	Time	Dollar-days
Hacker	\$400	1 d	400 DD
Lenstra			40M DD
Intelligence agency	\$300M	360 d	108B DD

Table 1: Attacker classes and corresponding budget for each attacker.

best attack against a given cryptosystem takes $t(k)$ seconds on a machine that costs d dollars. Then, the total "cost" of the attack is $T(k) = d\,t(k)/(3600 \cdot 24)$ dollar-days (DD). This notion is particularly interesting when estimating attack cost against lattice cryptography, where attacks may be parallelized with a time-money tradeoff.

Assume we have an estimate for the function $T(k)$ for attacks against lattice-based cryptosystems. Then, we can find the optimum k^* such that $T(k^*) \geq T_{2009}$, where T_{2009} is chosen according to the last column of Table 1. *We choose 2009 as a reference date here because the employed compute server was bought in that year.*

Estimating Future Developments. First of all, we consider Moore's Law, which states that computing power doubles every 18 months. Secondly, we want to take cryptanalytic developments against asymmetric primitives into account. Thus, we apply a combined degradation function $2^{-12/9}$ that Lenstra calls "double Moore Law". This is motivated by the algorithmic progress in the area of integer factorization. As for lattice basis reduction, the algorithmic progress for practical strong algorithms, such as BKZ, is hard to judge. While, there are recent results [GHGKN06, GN08a, GNR10] showing that progress is indeed possible, there are no public implementations that beat BKZ in practice.

The above condition only yields secure parameters for the year 2009. For year y, k needs to satisfy the inequality $T(k) \geq T_{2009} \cdot 2^{(y-2009)\cdot 12/9}$ to be secure until year y.

Asymmetric primitives are often combined with symmetric ones. Hash functions are necessary to sign long documents and block ciphers allow efficient hybrid encryption. We assume that these primitives are available at any given time in the future and that they are only affected by Moore's Law. Unlike public-key primitives, block ciphers and hash functions can easily be replaced if there is a new attack.

3. Analysis

Let us first restrict our analysis to signature schemes, i.e., SIS-based schmes. The best known attacks against these schemes involve a q-ary lattice $\Lambda = \Lambda_q^\perp(\mathbf{A})$ of dimension $m = \Omega(n \log(n))$ and a scheme-specific norm bound ν, which can be obtained by studying the security reductions. Later on, in Section 3.2, we will see that attacking LWE-based encryption is quite naturally done by expressing it as an SIS problem as well.

Thus, the main goal of this section is to determine the effort T_{2009} (in dollar-days) that is required today for mounting attacks on SIS. From there, we can apply Lenstra's Heuristic to estimate parameters for the future.

In order to grasp the hardness of most of these problems, we have conducted experiments on 10-100 random q-ary lattices per dimension $m \in \{100, 125, 150, 175, 200, 225, 250, 275, 300\}$ and exponent $c \in \{2, 3, 4, 5, 6, 7, 8\}$ for the relation $q \geq n^c$. The number of experiments per dimension has been chosen adaptively to focus on the interesting invervals. These parameters also determine n if we demand that $m > n \log_2(q)$. This setting covers even the hardest instances of SIS, where we demand the solution to be binary, i.e., $\nu = \sqrt{m}$. The existence of such vectors can be verified with a pidgeonhole argument because the function $f_{\mathbf{A}}(\mathbf{v}) = \mathbf{A}\mathbf{v} \bmod q$ admits a collision $(\mathbf{v}, \mathbf{v}') \in (\{0,1\}^m)^2$ if $q^n/2^m < 1$. Such a collision yields $\mathbf{v} - \mathbf{v}' \in \Lambda_q^\perp(\mathbf{A})$ with $\|\mathbf{v} - \mathbf{v}'\|_2 \leq \sqrt{m}$.

As mentioned earlier, we need to take attacks into account that do not require the full lattice dimension m but rather work in a sub-dimension d. In Section 2, we have already explained that we require a re-interpretation of the approach taken in [MR08]. There, the sub-dimension d is determined by the *fixed* capability δ of the employed HSVP solver, namely $d = \sqrt{n\log(q)/\log(\delta)}$, without taking ν into account. We need the following approach and let d be determined only via n, q, and ν.

Proposition 3.1 *Let $n \geq 128$, $q \geq n^2$, and $\nu < q$. Let S be a δ-HSVP solver for variable δ. The optimal dimension for solving $\mathsf{SIS}(n, m, q, \nu)$ with S is $d = \min\{x \in \mathbb{N} : q^{2n/x} \leq \nu\}$.*

Proof. Notice that when removing $m - d$ random columns from $\mathbf{A}$ to form a matrix $\mathbf{A}' \in \mathbb{Z}_q^{n\times d}$, the resulting q-ary lattice $\Lambda_q^{\perp}(\mathbf{A}')$ still has determinant q^n with high probability. Observe that $d > 2n$ as otherwise $q^{2n/d} \geq q > \nu$. Let $k = d - n$. Then, the probability that $\mathbf{A}'$ generates $\mathbb{Z}_q^n$ is $\prod_{i=1}^{n-1}(1 - q^{i-d}) \approx e^{-\frac{q^{-k}-q^{-d}}{q-1}} \approx e^{-\frac{q^{-k}}{q-1}}$, which is already > 0.999999 for $k \geq 1$.

The solver S finds lattice vectors of norm at most $\delta^d q^{n/d}$ in dimension d. Given δ, the minimum of this function is obtained for $d = \sqrt{n\log_2(q)/\log_2(\delta)}$ (cf. [MR08]). Equivalently, this means that, given d, one can solve HSVP for $\delta = 2^{n\log_2(q)/d^2}$. In consequence, a sufficiently good HSVP solver in dimension d can find vectors for length $\delta^d q^{n/d} = 2^{n\log_2(q)/d} q^{n/d} = q^{2n/d}$. Hence, we merely need to ensure that $q^{2n/d} \leq \nu$ and that the solver S works for $\delta \leq \sqrt[d]{\nu/q^{n/d}}$. □

Note that, as mentioned in the above proof, the minimum attack dimension is $d > 2n \geq 256$. Hence, special algorithms that efficiently reach smaller δ in dimensions < 256, do not contradict our analysis. To sum up, our analysis is based on the following conjecture.

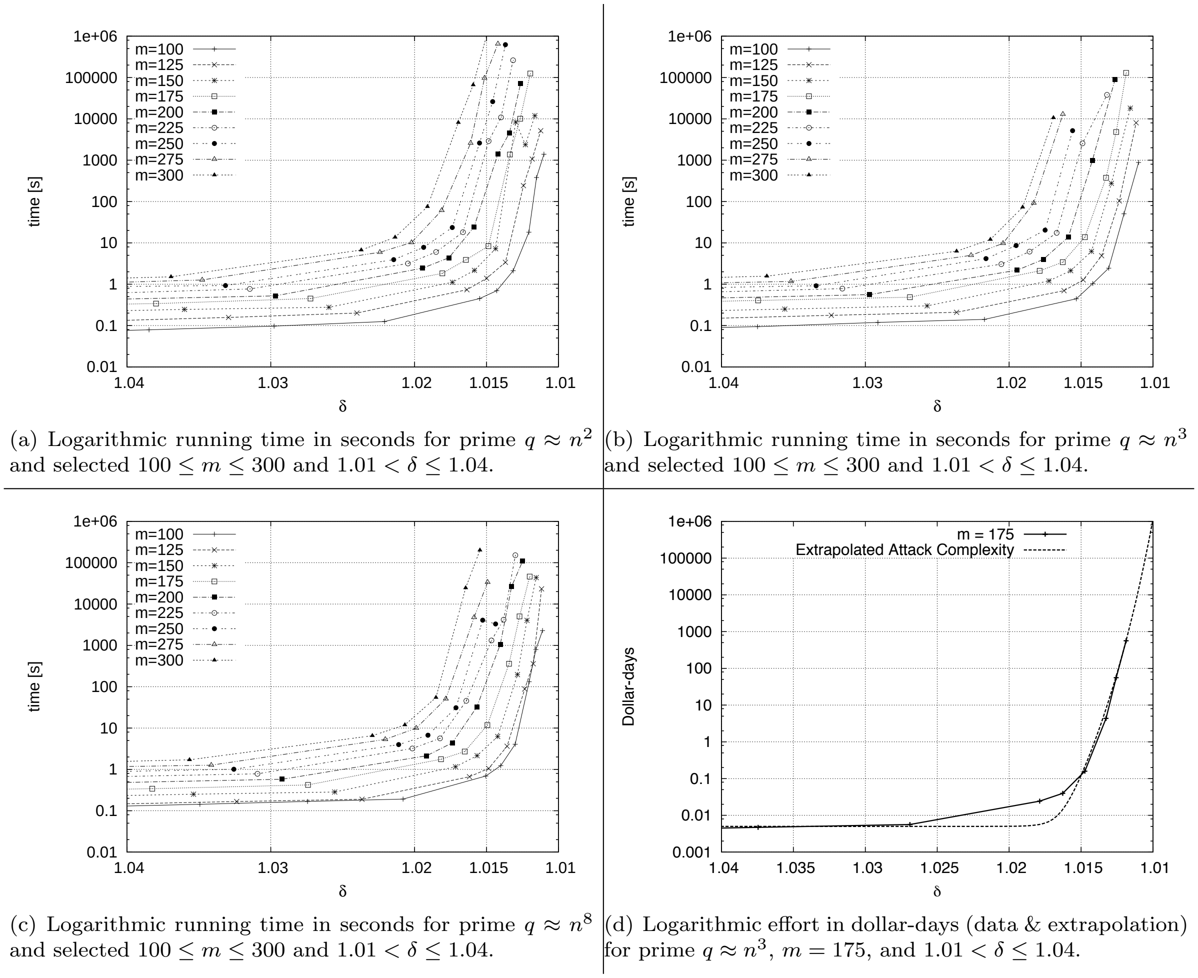
Conjecture 1 *For every $n > 128$, constant $c \geq 2$, prime $q \geq n^c$, $m = \Omega(n\log_2(q))$, and $\nu < q$, the best known approach to solve SIS with parameters (n, q, m, ν) involves solving δ-HSVP in dimension $d = \min\{x \in \mathbb{N} : q^{2n/x} \leq \nu\}$ with $\delta = \sqrt[d]{\nu/q^{n/d}}$.*

3.1. Experimental Data

In our experiments, we have analyzed the running time of BKZ [SE94] with double floating-point precision, a scalable HSVP-solver, as implemented in Shoup's NTL [Sho] on a \$1,000 machine (AMD Opteron CPU, running at 2.4 GHz). We apply BKZ in the sub-dimension d with an increasing block size parameter, i.e., with decreasing δ, until a vector of the desired length is found. Mark that our experiments only involve block size parameters ≤ 30 in order to avoid a known erratic behavior of the implementation in NTL. Also, performing the experiments in rather small dimensions, we give quite a conservative hardness estimate. Our first observation is that q plays a minor role if $\delta \in (1, 1.02]$. To see this, compare Figures 1(a) ($q \approx n^2$) and 1(c) ($q \approx n^8$). For $\delta \leq 1.02$, the graphs show the same shape. This also holds for $n^2 \leq q \leq n^8$. Observe that the timings are in log-scale. The impact of the dimension m is noticeable, but the slope of all graphs seems to be the same. The interesting part of the figures is where δ is smaller than 1.015, i.e., the right side of the graphs. Here, the impact of the parameter δ is compelling, and much more noticeable than the impact of the dimension m. Thus, we can consider δ to be the main security parameter.

Figure 1(b) shows the averaged samples for $q \approx n^3$ that were used for the interpolation. The fitting in Figure 1(d) was used to determine the hardness of attacks against lattice-based cryptosystems. For the interesting area where $\delta < 1.015$, the "extrapolated attack complexity" function nicely approximates the data samples.

To arrive at very conservative estimates, we use SIS instances with a fix $m = 175$ and n, q accordingly as our reference. For similar reasons, we choose a fix relation $q \approx n^3$ because all cryptosystems in Appendix A require $q > n^2$. Thus, from now on, we can treat δ as the main security parameter and consider the cost function in dollar-days to be $T(\delta) = a2^{1/(\log_2(\delta)^b)} + c$, for real constants a, b, c. We use the (averaged) data samples in Figure 1(d) to find parameters a, b, c for the above function $T(\delta)$ by a least-squares approximation. The resulting parameter $c = 0.005$ can be neglected for small δ. Now, we can draw our main conjecture, where $n \geq 128$ rules out unnaturally easy cases in small lattice dimensions $d < 256$.



(a) Logarithmic running time in seconds for prime $q \approx n^2$ and selected $100 \leq m \leq 300$ and $1.01 < \delta \leq 1.04$.

(b) Logarithmic running time in seconds for prime $q \approx n^3$ and selected $100 \leq m \leq 300$ and $1.01 < \delta \leq 1.04$.

(c) Logarithmic running time in seconds for prime $q \approx n^8$ and selected $100 \leq m \leq 300$ and $1.01 < \delta \leq 1.04$.

(d) Logarithmic effort in dollar-days (data & extrapolation) for prime $q \approx n^3$, $m = 175$, and $1.01 < \delta \leq 1.04$.

Figure 1: Logarithmic time complexity for solving δ-HSVP in different dimensions and for different moduli q. The x-axis corresponds to the Hermite factor δ.

Conjecture 2 *Let all other parameters and relations as in Conjecture 1. For any $\delta \in (1, 1.015]$, solving δ-HSVP (in normalized q-ary lattices) of dimension d involves an effort of at least $T(\delta) = 10^{-15}2^{1/(\log_2(\delta)^{1.001})}$ dollar-days.*

Extrapolating T for smaller δ yields Figure 2. The horizontal bars correspond to today's capabilities of the attacker types in Table 1. Notice that the extrapolation has moderate slope for $\delta < 1.01$ when compared to the actual data.

3.2. Attacking LWE

In contrast to lattice signatures that rely on (search) SIS, lattice-based encryption schemes are usually based on the decision LWE problem. While solving the search LWE problem also immediately solves the corresponding decision problem, the reverse direction only holds via a polynomial-time reduction. Thus, we choose to attack the decision problem because it presents the easier problem.

The most natural approach to distinguish $(\mathbf{A}, \mathbf{v})$ from uniform seems to be solving an instance of the SIS problem. Evidence for this connection can be found in [MR08] and [SSTX09]. We can interpret the decision-LWE problem as an instance of SIS in the dual lattice $1/q\Lambda_q^{\perp}(\mathbf{A})$ because finding a short vector

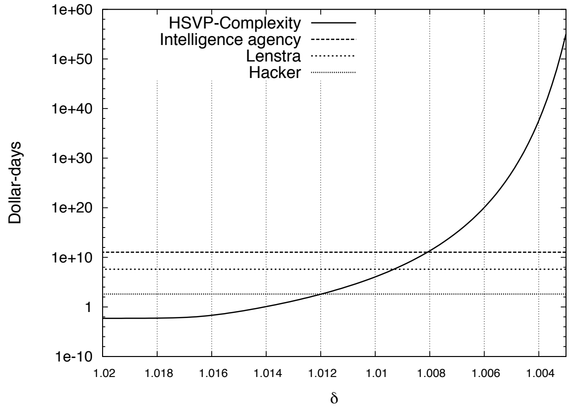


Figure 2: Estimated time complexity of δ-HSVP for $\delta \in [1.003, 1.02]$. The plots include horizontal lines, illustrating today's power of different attacker types.

$\mathbf{w} \in 1/q\Lambda_q^{\perp}(\mathbf{A})$ and checking whether $\langle \mathbf{v}, \mathbf{w} \rangle$ is close to $\mathbb{Z}$ solves the decision problem. Note that an alternative interpretation is transforming an instance of the "bounded-distance decoding" problem in the LWE-lattice into an instance of the approximate shortest vector problem via a well-known embedding method [GGH97]. If $\mathbf{v}$ is close to $\Lambda_q(\mathbf{A})$, its inner product with $\mathbf{w}$ will be close to an integer. To see this, consider $\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{A}^t\mathbf{s} + \mathbf{e}, \mathbf{w} \rangle = \langle \mathbf{A}^t\mathbf{s}, \mathbf{w} \rangle + \langle \mathbf{e}, \mathbf{w} \rangle$. Now, the first part of the sum is an integer because $\mathbf{A}\mathbf{w} \equiv \mathbf{0} \pmod{q}$. As for the second part, we have to consider $|\langle \mathbf{e}, \mathbf{w} \rangle|$. The length of $\mathbf{e}$ in the direction of $\mathbf{w}$ is short by design because we need to be able to decode and because it is drawn from a relatively tight Gaussian with standard deviation $\alpha q/\sqrt{2\pi}$ in each direction. However, the attack only works if both vectors are short. The length of $\mathbf{w}$ depends on how well we can cryptanalyze the lattice $1/q\Lambda_q^{\perp}(\mathbf{A})$. Following the reasoning in [MR08], we require $\|\mathbf{w}\| \geq 1.5\sqrt{2\pi}/(\alpha q)$ for the attack to fail as it makes the distribution of $\langle \mathbf{e}, \mathbf{w} \rangle \bmod 1$ essentially uniform.

In consequence, we can phrase decision-LWE in the language of SIS with with $\nu = 1.5\sqrt{2\pi}/\alpha$ and re-use the hardness estimates from Section 3.

3.3. Applying Lenstra's Heuristic

Fix an attacker type $\mathcal{A}$ and let $\delta_{\mathcal{A}}$ be infeasible for $\mathcal{A}$ today. Assuming the Lentra Heuristic in conjunction with the "double Moore Law", which takes algorithmic and technological advancement into account, the inequality $T(\delta) \geq T_{2009} \cdot 2^{12(y-2009)/9}$ for $T_{2009} = T(\delta_{\mathcal{A}})$ can be used in both directions, i.e., compute a δ such that it is infeasible until the end of a given year y and vice versa. Note that the inverse function is $T^{-1}(t) = 2^{(1/(\log_2(t-0.005)\cdot 10^{15}))^{1/1.001}}$, where t is the amount of dollar days available. For example, let $\mathcal{A} =$ "Int. agency". Compared with the year 2009, it can manage $t = 108 \cdot 2^{124/3}$ billion dollar-days in 2040. Thus, we require $\delta \leq T^{-1}(t) = 1.00548$ for infeasibility until the end of 2040. Vice versa, if an attack requires $\delta \leq 1.00548$, the corresponding lattice problem is at least intractable until the end of 2040. Table 2 provides an overview of hard values for δ for the different attacker types until 2100. This table also allows a mapping between symmetric security and security parameters for lattice cryptography. In addition, we include a column "standard" for a standard hash function (SHA-1) and a standard block cipher (AES-128). The resulting parameter sets can be considered secure against *non-quantum* adversaries until 2018.

year	Standard (2018)	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
bit security	SHA/AES	75	82	88	95	102	108	115	122	128	135
λ	160	225	246	264	285	306	324	345	366	384	405
κ	128	150	164	176	190	204	216	230	244	256	270
Hacker	1.00993	1.01177	1.00965	1.00808	1.00702	1.00621	1.00552	1.00501	1.00458	1.00419	1.00389
Lenstra	1.00803	1.00919	1.00785	1.00678	1.00602	1.00541	1.00488	1.00447	1.00413	1.00381	1.00356
Int. agency	1.00710	1.00799	1.00695	1.00610	1.00548	1.00497	1.00452	1.00417	1.00387	1.00359	1.00336

Table 2: Infeasible parameters δ for HSVP. The upper rows present recommended post-quantum secure symmetric key size κ and hash function length λ. Each of the lower cells contains an upper bound for the HSVP-parameter δ, such that this problem is computationally hard for the given attacker (row) until the end of a given year (column). *According to Proposition 3.1 solving δ-HSVP needs to be infeasible in dimensions $d \geq 256$.*

3.4. Post-quantum Secure Hash Functions and Symmetric Key Size

Encryption schemes and hash functions are rarely used without block ciphers and collision resistant hash functions, respectively. Since we want to propose parameters for the post-quantum era, we also want the symmetric ciphers and hash functions to be secure in this setting. In consequence, we need to take Grover's search algorithm for quantum computers into account [Gro96]. Basically, its effect is that we have to double the key length of block ciphers that would be required in the non-quantum setting for symmetric ciphers. The output length of hash functions has to be multiplied with $3/2$. According to the recommendations in [Len05] in conjunction with this doubling-law, we use the following formula that computes the required key length for security until the end of a given year y. As a simplification, we choose the symmetric parameters independently of the attacker type. A natural extension of our work would be to let λ and κ be functions of the attacker's resources. Here, we use the simple Moore Law and the assumption that DES was secure in the year 1982, even against the strongest attacker. Then, $\kappa \geq 2 \lceil 56 + 12(y - 1982)/18 \rceil$ is the proposed symmetric key length and $\lambda \geq 3\kappa/2$ is the proposed output length for hash functions. Using these formulae, we obtain the recommendations in Table 2. Notice that some of the schemes require the hash function to act as a random oracle. One scheme [Lyu09] even relies on "rewinding" the adversary to extract the solution to a hard problem. Generally, this is not possible with quantum adversaries due to the no-cloning theorem. Hence, we implicitly assume a stronger, quantum definition of the random oracle model or restrict the adversary to classical random oracle queries.

This concludes the analysis. Table 2 and Conjecture 2 provide all the necessary tools for estimating the security of all SIS and LWE-based cryptosystems. It also shows the equivalent level of symmetric security, sometimes referred to as "bit security". In the next section, we analyze the security of parameter sets proposed in literature. In Appendix A, we apply our framework to propose secure parameter sets for essentially all modern lattice-based signature and encryption schemes.

3.5. Comparison with Known Records in Lattice Reduction

There are currently two "lattice challenges" available online. The SVP challenge[3] corresponds to Goldstein-Mayer lattices that are well-suited to benchmark strong (non approximate) SVP solvers in rather small dimensions < 200. The best participants found a vector of ℓ_2-norm 2781 in dimension $d = 112$, which corresponds to $\delta = 1.009$. However, a success in this challenge does not have any immediate implication in our context, as the type of lattices differs and the dimensions involved are smaller than 256.

The Ajtai challenge[4] corresponds to lattices that are very similar to the ones we are studying here

[3] `http://www.latticechallenge.org/svp-challenge`

[4] `http://www.latticechallenge.org`

n	136	166	192	214	233	233
q	2003	4093	8191	16381	32749	32749
α	0.0065	0.0024	0.0009959	0.00045	0.000217	0.000217
ν	5.8e2	1.6e3	3.8e3	8.4e3	1.7e4	1.7e4
d	326	376	421	460	497	497
δ	1.0098	1.0098	1.0099	1.0099	1.0099	1.0099
year	2006	2006	2005	2005	2005	2005
bit	72	72	72	72	72	72

n	512	512	512	1024
q	$2^{31.727}$	$2^{59.748}$	$2^{95.747}$	$2^{95.872}$
m	4	5	8	8
ν	5.6e8	1.3e10	2.6e10	6.4e10
d	1118	1823	2835	5471
δ	1.0091	1.0064	1.0042	1.0023
year	2010	2035	2077	2180
bit	75	92	120	188

Table 3: Parameters given in [MR08] (left) and [Lyu09] (right), optimal attack dimension d, and hardness estimate δ. "year" denotes the expiration year of the parameter set and "bit" denotes the corresponding "bit security".

and there are challenges up to dimension 2000. More precisely, the Ajtai challenge asks to solve $\mathsf{SIS}(n, m, q, \nu)$ in a simplified setup where $q = n$, $m \approx 2n\log(n)$, and $\nu < q$. For the same n, this setup yields slightly easier instances than for the setup in this paper. Here, the best participants found a vector of length ≈ 107 in dimension $m = 725$. This corresponds to $\delta = 1.0103$ and an optimal attack dimension of $d = 229$ (cf. Proposition 3.1). Even though this result is still outside the relevant range for our analysis, it confirms our estimates, saying that today, a "Hacker" should be able to solve the problem for $\delta \approx 1.011$ and the adversary "Lenstra" might even solve it for $\delta \approx 1.009$ in 2010 (cf. Table 2).

4. Applying the Framework

There are essentially two "directions" for applying our analytic framework. In the "forward" direction, we can take a cryptographic parameter set and an attacker type as input and output an equivalent security level or even a prediction of how long this parameter set can be considered secure.

When working in the "reverse" direction, we analyze a given schemes parameters and their relations as well as the corresponding worst-case to average-case reduction and, on input a year and an attacker type, output a set of concrete parameters that can be considered secure against the given attacker type until the given year.

As mentioned before, we can easily make relative statements as well: Given SIS scheme X with parameters (n, q, m, ν) and LWE scheme Y with parameters (n, q, m, α), we can compute their hardness parameters δ_X and δ_Y. If $\delta_X < (>) \delta_Y$, the instance of X is more (less) secure than the instance of Y.

In this section, we will only apply our framework in the first sense, i.e., to analyze the (few) parameter sets that have been proposed in literature so far, regarding their exact security level. More concretely, we estimate the security of the parameters presented for LWE encryption in [MR08], to Lyubashevsky's Fiat-Shamir signature scheme in [Lyu09] (cf. Table 3), and to the one-time signature scheme due to Lyubashevsky and Micciancio [LM08]. *Mark that neither of these authors make claims about the exact security of there proposals because, prior to this work, there was no way of telling.*

For SIS-based schemes [LM08, Lyu09], we analyze the corresponding security proof to determine the relevant SIS instances. For LWE [MR08], we compute the corresponding SIS-parameters as outlined in Section 3.2.

Since the parameter sets given in [MR08] (see Table 3) were specifically chosen to be secure against attackers that can solve HSVP for $\delta \geq 1.01$, they do not provide sufficient security against the medium adversary "Lenstra" and even the "Hacker" should be able to break them by 2020. In Appendix A, we propose various parameter sets for LWE that provide more security. There, we also take the appropriate message length for hybrid encryption into account.

For the Fiat-Shamir-type signature scheme in [Lyu09], we compute the SIS norm parameter $\nu = 2\sqrt{mn}nmd_sd_c$, where d_s is the norm bound for signing keys and d_c controls the hashed message length. The values differ from the ones in [Lyu09] because we express it in ℓ_2-norm, whereas they are given in

ℓ_∞-norm in [Lyu09]. Note that the $\sqrt{mn}$ factor corresponds to the dimension of ideal lattices, which is typically denoted with nm as opposed to m in q-ary lattices.

The parameters in [Lyu09] are based on an assumed hash length of 160 bit, therefore the underlying hash functions would only be secure until year 2018 (without taking quantum adversaries into account). However, the lattice parameters are quite reasonable as shown in Table 3. All but the first parameter set provide some security margin and with our framework, we can actually estimate how large it is.

The authors of [LM08] propose an exemplary parameter set for their one-time signature scheme. They let $n = 512$, $q \approx 2^{27}$, and $m = 9$. This leads to $\nu = 20q^{1/m} n \log^2(n)\sqrt{mn} \approx 2.2 \cdot 10^8$. Using this parameter set, the attack dimension would be $d = 999$ and the hardness estimate is $\delta = 1.0097$. Hence, it would be insecure against the attacker "Lenstra" and the "Hacker" is expected to break it in the year 2020. Secure parameters can be found in the appendix.

Applying the framework in the other direction, i.e., estimating secure parameters sets based on the constraints given in a cryptographic scheme, is also possible, as demonstrated in Appendix A.

5. Conclusions

With our framework to analyze the SIS and LWE problems, we have established a connection between lattice problems and symmetric "bit security" for the first time. While our analysis reveals certain weaknesses in the way parameters for lattice-based cryptosystems are currently proposed, it also provides the tools to systematically do so for various levels of security.

We propose that the presented methodology should be used whenever a new cryptographic primitive is presented to ensure that, concerning efficiency and security, it actually presents an improvement over known work. Furthermore, our work can be used to compare the security levels of parameter sets for entirely different cryptographic primitives, e.g., encryption and signature schemes. This is important when both are used in a more complex protocol, where all components should provide approximately the same level of security.

An additional application of our work is the proposition of parameters for lattice-based signature and encryption schemes for which there were no known concrete parameter sets. Doing so has revealed that all schemes that require trapdoor matrices (short bases) are far from practical and seem to require an enormous effort to become so. On the other hand, we have also seen that there are quite competitive signature and encryption schemes already, especially those working in ideal lattices. Refer to Appendix A.3 for details.

To conclude, with our work we would like to draw renewed interest to the development of practical, strong lattice basis reduction algorithms for large dimensions as well as to further optimizing the parameter constraints for known lattice-based cryptosystems, which have mainly been of theoretic interest so far.

References

[ADL+08] Y. Arbitman, G. Dogon, V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFTX: A proposal for the SHA-3 standard, 2008. In the First SHA-3 Candidate Conference.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108. ACM, 1996.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610. ACM, 2001.

[AP09] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In Susanne Albers and Jean-Yves Marion, editors, *STACS*, volume 09001 of *Dagstuhl Sem-*

inar Proceedings, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2009.

[Bab86] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[BDR$^+$96] Matt Blaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener. Minimal key lengths for symmetric ciphers to provide adequate commercial security. A Report by an Ad Hoc Group of Cryptographers and Computer Scientists, 1996.

[Boy10] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517. Springer, 2010.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis, 2010. to appear in EUROCRYPT 2010.

[Dwo08] Cynthia Dwork, editor. *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008.* ACM, 2008.

[ECR09] ECRYPT2. Yearly report on algorithms and keysizes — report D.SPA.7, 2009. available at `http://www.ecrypt.eu.org/documents/D.SPA.7.pdf`.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Mitzenmacher [Mit09], pages 169–178.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *LNCS*, pages 112–131. Springer, 1997.

[GHGKN06] Nicolas Gama, Nick Howgrave-Graham, Henrik Koy, and Phong Q. Nguyen. Rankin's constant and blockwise lattice reduction. In *CRYPTO*, volume 4117 of *LNCS*, pages 112–130. Springer, 2006.

[GM03] Daniel Goldstein and Andrew Mayer. On the equidistribution of Hecke points. *Forum Mathematicum 2003, 15:2*, pages 165–189, 2003.

[GN08a] Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within mordell's inequality. In Dwork [Dwo08], pages 207–216.

[GN08b] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *LNCS*, pages 31–51. Springer, 2008.

[GNR10] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. Lattice enumeration using extreme pruning. In *EUROCRYPT*, LNCS, pages 257–278. Springer, 2010.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Dwork [Dwo08], pages 197–206.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219. ACM, 1996.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In Joe Buhler, editor, *ANTS*, volume 1423 of *LNCS*, pages 267–288. Springer, 1998.

[KTX07] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit cryptosystems based on lattice problems. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 315–329. Springer, 2007.

[Len05] Arjen Lenstra. *The Handbook of Information Security*, chapter 114 — Key Lengths. Wiley, 2005. available at `http://www.keylength.com/biblio/Handbook_of_Information_Security_-_Keylength.pdf`.

[LLL82] Arjen Lenstra, Hendrik Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *LNCS*, pages 144–155. Springer, 2006.

[LM08] Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In Ran Canetti, editor, *TCC*, volume 4948 of *LNCS*, pages 37–54. Springer, 2008.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, LNCS, pages 1–23. Springer, 2010.

[LV01] Arjen Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *J. Cryptology*, 14(4):255–293, 2001.

[Lyu09] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009.

[Mer89] Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *LNCS*, pages 218–238. Springer, 1989.

[Mic01] Daniele Micciancio. Improving lattice based cryptosystems using the hermite normal form. In Joseph H. Silverman, editor, *CaLC*, volume 2146 of *LNCS*, pages 126–145. Springer, 2001.

[Mic07] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Prelim. in FOCS 2002.

[Mit09] Michael Mitzenmacher, editor. *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009.* ACM, 2009.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[MR08] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Bernstein, Johannes A. Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer, 2008.

[MV10] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *STOC*. ACM, 2010.

[Pei07] Chris Peikert. Limits on the hardness of lattice problems in ℓ_p norms. In *IEEE Conference on Computational Complexity*, pages 333–346. IEEE Computer Society, 2007.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Mitzenmacher [Mit09], pages 333–342.

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 80–97. Springer, 2010.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Dwork [Dwo08], pages 187–196.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[Reg10] Oded Regev. Regularity lemma for ring-lwe, 2010. Personal communication, June 2010.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In Omer Reingold, editor, *TCC*, volume 5444 of *LNCS*, pages 419–436. Springer, 2009.

[Rüc10] Markus Rückert. Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In Nicolas Sendrier, editor, *PQCrypto*, volume 6061 of *LNCS*, pages 182–200. Springer, 2010.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[SE94] Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.

[Sho] Victor Shoup. Number theory library (NTL) for C++. `http://www.shoup.net/ntl/`.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, 2009.

A. Selecting Secure Parameters

Here, we apply our framework in the "reverse" direction, i.e., we generate secure parameter sets out of their original description, such that they require a small enough δ when attacked with an HSVP solver. The resulting parameters are merely exemplary because most schemes allow various trade-offs.

We cover essentially every published lattice-based signature and encryption scheme and also some unpublished variants. Moreover, due to our modular three-tier approach, it is easy to include new schemes in the future, that use LWE or SIS as their security assumption. For each scheme, one needs to figure out the exact (not asymptotic) parameter relations and constraints as functions of the main security parameter n. In addition, we let the worst-case to average-case reduction be a guiding principle for choosing the modulus q. In conjunction with the average-case reduction from SIS (signatures) or LWE (encryption), these parameter relations specify the type of lattice that needs to be "attacked" in order to break the scheme. For signature schemes, the resulting instance of SIS immediately yields the hardness estimate δ via Conjecture 1. As for encryption schemes, we need to exploit the duality of SIS and LWE before making such a statement. Once we have the hardness estimate $\delta = \delta(n)$, we can easily determine the least n, such that it provides sufficient hardness against various attacker types and for the desired period of time via Conjecture 2. All parameter sets correspond to security against the attacker type "Lenstra" but the analysis easily extends to any other type.

We conclude this section with a set of remarks about our findings.

For some of the schemes, we require ideal lattices and some additional notation. We define ideal lattices over the ring $\mathbf{R} = \mathbb{Z}_q[x]/\langle f \rangle$ for an irreducible polynomial f of degree n. The description $\mathbf{A}$ in q-ary lattices is replaced by a small number of degree-n polynomials, denoted with $\hat{\mathbf{a}} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathbf{R}^m$. Since $\mathbf{R}^m \cong \mathbb{Z}_q^{mn}$, the parameter m is in $\Omega(\log(n))$ for ideal lattices. The resulting lattices dimension, however, is $mn = \Omega(n \log(n))$. In addition, in ideal lattices, the matrix-vector product $\mathbf{Av}$ is replaced with the product $\hat{\mathbf{a}} \circledast \hat{\mathbf{v}} := \sum_{i=1}^{m} \mathbf{a}_i \mathbf{v}_i$ (modulo f and q).

A.1. Signature Schemes

All modern lattice-based signature schemes are based on the hardness of the SIS problem. In other words, for each scheme, we can easily describe an equivalent instance of SIS in terms of the parameters n, m, q, ν that also fully determine the hardness estimate δ for HSVP. For our choices of $n, m,$ and q, by the following worst-case to average-case reduction, the SIS instances in dimension m are provably at least as hard as all instances of the shortest vector problem in dimension n.

Proposition A.1 (Worst-case to Average-case [GPV08]) *For any $m \leq \mathsf{poly}(n)$, $\nu \leq \mathsf{poly}(n)$, and for any prime $q \geq \nu\omega(\sqrt{n \log(n)})$, the average-case problem $\mathsf{SIS}(n, m, q, \nu)$ is as hard as approximating the* SIVP *in the worst case within certain $\gamma = \nu\widetilde{\mathcal{O}}(\sqrt{n})$ factors.*

Using the attacker dimension d of Proposition 3.1, we can compute $\delta = \sqrt[d]{\nu/q^{n/d}}$. We let q be governed by a constraint in the worst-case to average-case reduction. As this constraint introduces a circular dependency, we typically choose a fixed relation $q \geq n^t$, for $t \in \mathbb{N}$, before the other parameters to resolve this issue. Having these relations at hand, we can also fix a δ and find suitable n, m, q, ν such that they are valid parameters that guarantee security until the desired year. Combined with the infeasible values for δ for each year and attacker type (Table 2) we generate tables that present suitable parameters for each signature scheme. We propose exemplary parameters for GPV [GPV08], Lyubashevsky's treeless signature scheme [Lyu09], the ideal lattice variant of GPV, the Bonsai tree scheme [CHKP10], its ideal lattice variant, and the Lyubashevsky-Micciancio one-time signature scheme [LM08].

GPV Signatures. The GPV signature scheme [GPV08] is due to Gentry, Peikert, and Vaikuntanathan. It benefits from the improved trapdoor generation algorithm in [AP09], which demands $m_1 \geq (1 + \varphi) n \log_2(q)$, $m_2 \geq (4 + 2\varphi) n \log_2(q)$, $m = m_1 + m_2$, and odd prime $q \geq 3$ (q has to satisfy $q \geq \nu\omega(\sqrt{n \log n})$, for the worst-case to average-case reduction). For our choices of n ($n \geq 100$), m ($m \geq 1000$), and q ($q \geq n^3$), $\varphi = 0.1$ is a suitable choice. For $\varphi = 0.1$, the statistical distance from uniformity, $m_2 \cdot q^{-\varphi n/2}$ in [AP09], is smaller than 2^{-80}.

The most recent sampling algorithm [Pei10] improves the efficiency of the signature generation process in GPV and in all derived schemes. However, it does not change the parameters.

The GPV scheme is strongly unforgeable in the random oracle model as long as the respective instance of SIS with norm bound $\nu = 2s\sqrt{m}$ is hard, for a Gaussian parameter $s \geq (1 + 20\sqrt{m_1}) \cdot \omega\left(\sqrt{\log(n)}\right)$. Choosing $\log(n)$ for $\omega\left(\sqrt{\log(n)}\right)$ we get $\nu = 2(1 + 20\sqrt{m_1})\log(n)\sqrt{m}$.[5]

We choose $m_1 = \lceil(1 + 0.1)n\log_2(q)\rceil$ and $m_2 = \lceil(4 + 0.2)n\log_2(q)\rceil$. For q we choose the smallest prime bigger than n^t for the smallest t such that $q \geq 2\nu\sqrt{n}\log_2(n)$ (worst-case to average-case reduction). In our case, we could choose a prime $q \geq n^4$. Messages are mapped to $\mathbb{Z}_q^n$ via a full-domain hash. This set is always bigger than 2^λ.

Here we describe the structure of the scheme, in order to compute the key and signature sizes. The parameters for GPV are presented in Table 4.

Secret Key: $\mathbf{S} \in \mathbb{Z}^{m \times m}$ with $\|\mathbf{S}\| \leq 20n\log(q)$. A close look at the trapdoor construction allows to store the key in $2m_1m_2 + m_1\log_2(q))$ bits, without storing the orthogonalized basis. This implies that generating signatures gets a bit more expensive, as it requires computation of the QR decomposition of the trapdoor basis.

Public Key: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, i.e, $nm\log_2(q)$ bits.

Signature: $\sigma \in \mathbb{Z}^m$ with $\|\sigma\|_2 \leq s\sqrt{m}$, i.e., $m\log_2(s\sqrt{m})$ bits.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	λ	160	225	246	264	285	306	324	345	366	384
Lenstra	n	330	289	338	391	440	489	542	592	641	695
	q	1.19e+10	6.98e+09	1.31e+10	2.34e+10	3.75e+10	5.72e+10	8.63e+10	1.23e+11	1.69e+11	2.33e+11
	m_1	12148	10396	12494	14815	17001	19222	21660	23989	26298	28871
	m	58531	50087	60198	71380	81913	92615	104359	115583	126709	139103
	$\lvert\mathsf{sk}\rvert$	137613	100780	145562	204654	269498	344507	437415	536545	644799	777112
	$\lvert\mathsf{pk}\rvert$	78904	57779	83462	117348	154538	197556	250834	307694	369782	445659
	$\lvert\sigma\rvert$	154	130	158	190	221	252	286	320	353	390

Table 4: Recommended parameters for GPV signatures. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Ideal GPV. In [SSTX09], the authors explain how to create an ideal-lattice variant of the GPV signature, in order to reduce the key sizes of the secret and public key. This variant comes with $\tilde{\mathcal{O}}(n)$ verification time and signature length. Here we apply their idea and instantiate the GPV scheme with ideal lattices.

Choose $k > 0$ and $n = 2^k$ for the smallest possible k, $\sigma = 1$ and $\rho = \lceil 1 + \log_3(q)\rceil$. The ring $\mathbf{R}$ is $\mathbf{R} = \mathbb{Z}_q[x]/(x^n + 1)$. Choose the norm bound $d = s\sqrt{mn}$. No bound on $\tilde{L}$ is known, but it is always possible to assume $\tilde{L} \leq L = \sqrt{2n(9\rho + \sigma)}$. The dimension has to satisfy $m \geq (\lceil\log_2(q)\rceil + 1)(\sigma + \rho)$, we choose m equal to that bound. Choose the Gaussian parameter as $s = \tilde{L}\log(n) = \sqrt{2n(9\rho + \sigma)} \cdot \log(n)$. The modulus q is chosen to be the smallest prime bigger than or equal to n^7 satisfying $q \equiv 3 \pmod 8$, as in that case $m > \log_2(q)/\log_2(2d)$ and $q > 4dmn\sqrt{n}\log_2(n)$ hold. With $\|\sigma\|_2 \leq d$ we have $\nu^{(2)} = 2d$ (in the Euclidean norm). We can use the same bound $2d$ in the maximum norm, i.e., $\nu = 2d$.

The parameters for Ideal-GPV are presented in Table 5.

Here we describe the structure of the scheme, in order to compute the key and signature sizes. Instead of storing the trapdoor basis, which implies the necessity to calculate orthogonalizations on the fly, it would also we possible to store the Gram-Schmidt orthogonalized basis.

[5] This choice is suitable for all dimensions $m \geq 83$; for those m, the smoothing parameter index ϵ (see [MR07, Pei07, GPV08] for more details) is smaller than 2^{-79}. This renders the statistical distance between a uniform distribution and the "blurred" lattice negligible (i.e., 2^{-80}). This is due to the fact that $\log(m) \geq \sqrt{\log(2m(1 + 1/\epsilon))/\pi}$ for $m \geq 83$ and $\lambda_1^\infty(\mathbb{Z}^*) = 1$ (a lattice constant) in [GPV08, Lemma 4.3], using [Pei07, Lemma 3.5].

Secret Key: Trapdoor $\mathbf{S} \in \mathbb{Z}^{mn \times mn}$ such that $\hat{\mathbf{a}}\hat{\mathbf{s_i}} \equiv 0 \bmod q$ for every column $\hat{\mathbf{s}}$ in $\mathbf{S}$ (interpreted as an element of $\mathbf{R}^m$). The basis length is $\|\mathbf{S}\| \leq \sqrt{2n(9\rho + \sigma)}$. When looking closely at the construction, we find that the trapdoor can be reconstructed from $\sigma(m - \sigma)n\sqrt{\sigma n} + \rho(m - \rho)n \log_2(3)$ bits.

Public Key: $\hat{\mathbf{a}} \in \mathbf{R}^m$ determining the ideal lattice, i.e., $mn \log_2(q)$ bits.

Signature: $\sigma \in \mathbf{R}^m$ with $\|\sigma\|_2 \leq d$, i.e., $mn \log_2(d)$ bits.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	λ	160	225	246	264	285	306	324	345	366	384
Lenstra	n	256	256	256	256	256	512	512	512	512	512
	q	7.21e+16	7.21e+16	7.21e+16	7.21e+16	7.21e+16	9.22e+18	9.22e+18	9.22e+18	9.22e+18	9.22e+18
	m	2204	2204	2204	2204	2204	2688	2688	2688	2688	2688
	$\|\mathsf{sk}\|$	5072	5072	5072	5072	5072	14550	14550	14550	14550	14550
	$\|\mathsf{pk}\|$	3857	3857	3857	3857	3857	10584	10584	10584	10584	10584
	$\|\sigma\|$	1151	1151	1151	1151	1151	2957	2957	2957	2957	2957

Table 5: Recommended parameters for Ideal-GPV signatures. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Bonsai Trees. Here we describe the original Bonsai tree scheme by Cash, Hofheinz, Kiltz, and Peikert [CHKP10]. It does not require random oracles for the security proof of existential unforgeability. A modified version by Rückert [Rüc10] with essentially the same efficiency supports strong unforgeability. The Bonsai tree scheme makes use of the [AP09] trapdoor, which was used in the GPV case as well.

The parameters are: $m_1 = \lceil (1 + \varphi)n \log_2(q) \rceil$, $m_2 = \lceil (4 + 2\varphi)n \log_2(q) \rceil$, hashed message length λ, total dimension $m = m_1 + (\lambda + 1)m_2$.[6] Again, we can use $\varphi = 0.1$. We choose the Gaussian parameter $s = (1 + 20\sqrt{m_1}) \log(n)$ and let $q \geq n^5$. If there exists a PPT attack against unforgeability on the signature scheme, then there is a PPT algorithm attacking SIS for $\nu = 2s\sqrt{m}$. For the overview of the parameters, refer to Table 6.

Here we describe the keys and the signature of the scheme, in order to derive the key and signature sizes.

Secret Key: $\mathbf{S} \in \mathbb{Z}^{(m_1+m_2)\times(m_1+m_2)}$ with $\|\mathbf{S}\| \leq 20n \log(q)$. A close look at the trapdoor construction allows to store the key in $2m_1m_2 + m_1 \log_2(q))$ bits, without storing the orthogonalized basis. This implies that generating signatures gets a bit more expensive, as it requires computation of the QR decomposition of the trapdoor basis.

Public Key: $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times(m_1+m_2)}, \mathbf{A}_j^{(k)} \in \mathbb{Z}_q^{n\times m_2}$, 2λ many, i.e., $n(m_1 + m_2) \log_2(q) + 2\lambda \cdot nm_2 \log_2(q)$ bits.

Signature: $\sigma \in \mathbb{Z}^m$ with $\|\sigma\|_2 \leq s\sqrt{m}$, i.e., $m \log_2(s\sqrt{m})$ bits.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	λ	160	225	246	264	285	306	324	345	366	384
Lenstra	n	360	322	377	436	491	547	607	663	718	779
	q	6.05e+12	3.46e+12	7.62e+12	1.58e+13	2.85e+13	4.90e+13	8.24e+13	1.28e+14	1.91e+14	2.87e+14
	m_1	16814	14755	17746	21027	24142	27364	30867	34179	37468	41155
	m_2	64199	56334	67758	80282	92177	104479	117854	130499	143058	157137
	m	10352853	12746239	16753972	21295757	26386764	32102417	38333417	45186833	52539754	60538900
	$\|\mathsf{sk}\|$	263622	203006	293655	412243	543426	698141	888308	1089142	1308834	1579092
	$\|\mathsf{pk}\|$	38483319	41622485	65819290	99143682	141070584	194564698	262099312	342149385	436157286	552063776
	$\|\sigma\|$	32290	39799	53067	68312	85551	105088	126602	150405	176114	204315

Table 6: Recommended parameters for Bonsai signature scheme. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

[6] We apply the original construction due to Peikert, as mentioned in a footnote in [CHKP10].

Note that there is an improvement due to Boyen [Boy10] that reduces the dimension of the generated signatures at the expense of a stronger assumption. Unfortunately, this improvement does not seem to yield practical parameters either.

Ideal Bonsai. Here we describe how to instantiate the Bonsai tree scheme of [CHKP10] with ideal lattices. As the security reduction to a worst case problem is stated in the infinity norm (and this norm is more natural for ideal lattices and ring elements) we describe the scheme using the infinity norm. Following [SSTX09], the parameters are: n which is a power of 2, $f = x^n + 1$, prime $q \equiv 3 \pmod 8$, $\sigma = 1$, $\rho = \lceil \log_3(q) + 1 \rceil$. The output length of a secure hash function is denoted by λ. We choose $\tilde{L} = \sqrt{2n(9\rho + \sigma)}$ as bound for the length of the trapdoor. $\mathbf{R}$ is again the ring $\mathbb{Z}_q[x]/\langle f \rangle$. We use a Gaussian parameter $s = \tilde{L} \log(n)$ and $d = s\sqrt{mn}$.

It is required that $m_1 + m_2 \geq (\lceil \log(q) \rceil + 1)(\sigma + r)$. We can choose $m_1 = \sigma = 1$ and $m_2 = \lceil \log(q) + 1 \rceil (\sigma + \rho) - 1$. Let $m = m_1 + (\lambda + 1)m_2$. Following the worst-case to average-case reduction for ideal lattices, we choose a prime $q \geq n^8$ such that $m > \log_2(q)/\log_2(2d)$ and $q > 4dmn\sqrt{n}\log_2(n)$. The corresponding approximation factor for SIS is $\nu = 2d$. The overview of the parameters for the Ideal Bonsai scheme are presented in Table 7.

Here we describe the keys and the signature of the scheme, in order to derive the key and signature sizes. Instead of storing the trapdoor basis, which implies the necessity to calculate orthogonalizations on the fly, it would also we possible to store the Gram-Schmidt orthogonalized basis.

Secret Key: Trapdoor $\mathbf{S} \in \mathbb{Z}^{mn \times mn}$ such that $\hat{\mathbf{a}}\hat{\mathbf{s_i}} \equiv 0 \bmod q$ for every column $\hat{\mathbf{s}}$ in $\mathbf{S}$ (interpreted as an element of $\mathbf{R}^m$). The basis length is $\|\mathbf{S}\| \leq \sqrt{2n(9\rho + \sigma)}$. When looking closely at the construction, we find that the trapdoor can be reconstructed from $\sigma(m - \sigma)n\sqrt{\sigma n} + \rho(m - \rho)n\log_2(3)$ bits.

Public Key: $\hat{\mathbf{a}}_0 \in \mathbf{R}^{m_1+m_2}$, $\hat{\mathbf{b}}_i^{(k)}$ for $k \in \{0, 1\}$ and $i \in \{1, \ldots, \lambda\}$, random elements in $\mathbf{R}^{m_2}$, i.e., $n\log_2(q) \cdot (m_1 + m_2 + 2\lambda m_2)$ Bits

Signature: $\sigma \in \mathbf{R}^m$ with $\|\sigma\|_2 \leq s\sqrt{mn}$, i.e., $mn\log_2(s\sqrt{mn})$ bits.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	λ	160	225	246	264	285	306	324	345	366	384
Lenstra	n	512	512	1024	1024	1024	1024	1024	1024	1024	2048
	q	4.72e+21	4.72e+21	1.21e+24	1.21e+24	1.21e+24	1.21e+24	1.21e+24	1.21e+24	1.21e+24	3.09e+26
	m_1	1	1	1	1	1	1	1	1	1	1
	m_2	2447	2447	3020	3020	3020	3020	3020	3020	3020	3595
	m	393968	553023	745941	800301	863721	927141	981501	1044921	1108341	1384076
	$\lvert\mathsf{sk}\rvert$	14639	14639	42667	42667	42667	42667	42667	42667	42667	120603
	$\lvert\mathsf{pk}\rvert$	1772856	2488603	7459410	8003010	8637210	9271410	9815010	10449210	11083410	30449672
	$\lvert\sigma\rvert$	635248	900169	2562702	2754533	2978756	3203399	3396263	3621613	3847310	10080821

Table 7: Recommended parameters for Ideal Bonsai signature scheme. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

It is noticeable that for the ideal Bonsai signature scheme, we need to choose a bigger modulus q than for the original Bonsai tree scheme.

LM-OTS. The one-time signature scheme of [LM08] does not require random oracles, and it is asymptotically optimal (almost linear in the security parameter n) in concerns of key size and signature/verification time. It is equipped with a security proof of worst-case complexity assumptions. Using a tree construction it can be transformed into a regular signature scheme, with logarithmic overhead [Mer89]. The LM-OTS scheme is based on the collision resistant hash function of [LM06, Mic07, PR06]: $\mathsf{H} \in \mathcal{H}_{\mathbf{R},m} = \{\mathsf{H}_{\hat{\mathbf{a}}} : \hat{\mathbf{a}} \in \mathbf{R}^m\}$ that maps elements from $\mathbf{R}^m$ to $\mathbf{R}$. For a λ-bit message signing and verification take time $\tilde{\mathcal{O}}(\lambda) + \tilde{\mathcal{O}}(n)$, signature size is $\tilde{\mathcal{O}}(n)$.

We fix the ring defining polynomial and operate in $\mathbf{R} = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$. We choose a prime $q \geq n^3$ and $m = \lceil \log(n) \rceil$, as proposed in the original work [LM08]. The main parameter n is chosen to be a power of 2. Messages are encoded in $\{-1, 0, 1\}^n$, but $|\{-1, 0, 1\}^n| \geq 2^\lambda$ does not introduce an additional constraint here.

An attacker that, after seeing a signature/message pair, can output a valid signature of another message, can use a polynomial-time algorithm to find a collision in the underlying hash function and from this we derive $\nu = 20q^{1/m} n \log^2(n)\sqrt{mn}$ for SIS. See Table 8 for the proposed LM-OTS parameters.

Secret Key: $\hat{\mathbf{k}} \in \mathbf{R}^m, \hat{\mathbf{l}} \in \mathbf{R}^m$ with $\left\|\hat{\mathbf{k}}\right\|_\infty \leq 5 \lfloor \log_2(n) \rfloor q^{1/m}$, $\left\|\hat{\mathbf{l}}\right\|_\infty \leq 5n \lfloor \log_2(n) \rfloor q^{1/m}$, i.e, $mn \log_2(5 \lfloor \log_2(n) \rfloor q^{1/m}) + mn \log_2(5n \lfloor \log_2(n) \rfloor q^{1/m})$ bits.

Public Key: $\mathsf{H} \in \mathcal{H}_{\mathbf{R},m}, \mathsf{H}(\hat{\mathbf{k}}), \mathsf{H}(\hat{\mathbf{l}})$, i.e., $mn \log_2(q) + 2 \cdot n \log_2(q)$ bits. H is shared among all users and generated from a trusted source of random bits, e.g., from the random bits of π.

Signature: $\sigma \in \mathbf{R}^m$ with $\|\sigma\|_\infty \leq 10q^{1/m} n \log^2(n)$, i.e., $mn \log_2(10q^{1/m} n \log^2(n))$ bits.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	λ	160	225	246	264	285	306	324	345	366	384
Lenstra	n	1024	1024	1024	1024	1024	1024	2048	2048	2048	2048
	q	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	8.59e+09	8.59e+09	8.59e+09	8.59e+09
	m	7	7	7	7	7	7	8	8	8	8
	$\lvert\mathsf{sk}\rvert$	8.71	8.71	8.71	8.71	8.71	8.71	19.83	19.83	19.83	19.83
	$\lvert\mathsf{pk}\rvert$	33.75	33.75	33.75	33.75	33.75	33.75	82.5	82.5	82.5	82.5
	$\lvert\sigma\rvert$	20.29	20.29	20.29	20.29	20.29	20.29	48.62	48.62	48.62	48.62

Table 8: Recommended parameters for LM-OTS signature scheme. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Lyubashevsky Treeless Signatures. In [Lyu09] Lyubashevsky presents a signature scheme secure in the random oracle model with key generation, signing, and verification time $\tilde{\mathcal{O}}(n)$. Its security is based on the hardness of approximating the shortest, non-zero vector to within a factor of $\tilde{\mathcal{O}}\left(n^2\right)$ in lattices corresponding to ideals in $\mathbf{R} = \mathbb{Z}[x]/\langle x^n + 1\rangle$.

The parameters involved are: n, a power of 2, an integer m, an integer d_c such that $2^{d_c}\binom{n}{d_c} \geq 2^\lambda$ (for encoding messages), and a prime integer $q \geq (2d_s + 1)^m \cdot 2^{-128/n}$.

If the scheme is not strongly unforgeable, then there exists a polynomial time algorithm that solves SIS in every lattice corresponding to ideals in $\mathbf{R}$ for $\nu = 2\sqrt{mn} \cdot nmd_s d_c$.

We choose $m = \lceil \log_2(n) \rceil$ and compute the smallest d_c such that $2^{d_c}\binom{n}{d_c} \geq 2^\lambda$ holds. Further, for d_s we choose the smallest value such that $q \geq 4m^2 n^{2.5} d_s d_c \log(n)$ and $m > \log(q)/\log(2mnd_s d_c)$ hold because of the worst-case to average-case reduction. This choice of parameters implies that finding collisions in the underlying hash function is hard. Notice that the scheme allows various trade-offs. For example, a larger d_s increases the key size but allows for smaller m, as demonstrated in [Lyu09]. The scheme has the following structure. See [Lyu09] for a full description of the numerous parameters. Our proposed parameter sets are in Table 9.

Secret Key: $\hat{\mathbf{s}} \in \mathbf{R}^m$ with $\|\hat{\mathbf{s}}\|_\infty \leq d_s$, i.e, $mn \log_2(2d_s + 1)$ bits for a typically small d_s.

Public Key: $\mathsf{H} \in \mathcal{H}_{R,m}, \mathsf{H}(\hat{\mathbf{s}}) \in \mathbf{R}$, i.e., $n \log_2(q)$ bits. H is again global.

Signature: $\sigma \in \mathbf{R}^m$ with $\|\sigma\|_\infty \leq mnd_s d_c$, i.e., $mn \log_2(2mnd_s d_c + 1)$ bits.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	λ	160	225	246	264	285	306	324	345	366	384
Lenstra	n	512	512	512	512	1024	1024	1024	1024	1024	1024
	q	3.81e+12	7.25e+12	7.25e+12	1.32e+13	4.77e+13	1.03e+14	1.03e+14	1.03e+14	1.03e+14	1.03e+14
	m	9	9	9	9	10	10	10	10	10	10
	d_c	23	37	41	45	40	44	48	52	56	59
	d_s	13	14	14	15	12	13	13	13	13	13
	$\lvert\mathsf{sk}\rvert$	2.67	2.73	2.73	2.79	5.8	5.94	5.94	5.94	5.94	5.94
	$\lvert\mathsf{pk}\rvert$	2.61	2.67	2.67	2.72	5.68	5.82	5.82	5.82	5.82	5.82
	$\lvert\sigma\rvert$	12.03	12.48	12.56	12.69	29.04	29.35	29.51	29.65	29.79	29.88

Table 9: Recommended parameters for treeless signatures. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Our parameters for the year 2020 lead to comparable sizes for keys and signatures as the parameters in the weakest sample instantiation of [Lyu09].

A.2. Encryption Schemes

We discuss the parameter choice for the multi-bit variant of Regev's cryptosystem [Reg09, KTX07, PVW08, MR08], the dual-LWE cryptosystem [GPV08, Pei09], and the trapdoor-LWE scheme [RS09, Pei09]. For each scheme, we also present a "ring" version that uses an ideal lattice version of LWE [LPR10]. After briefly recalling the LWE assumption, we describe its modification for rings and deal with decryption errors.

The LWE Assumption. Let $n \in \mathbb{N}$, $m \leq \mathsf{poly}(n)$, $q \leq \mathsf{poly}(n)$, and $\alpha > 0$. Furthermore, let $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$, and $\mathbf{e} \overset{\$}{\leftarrow} \chi_\alpha^m$ with χ_α being a discretized Gaussian distribution with standard deviation $\alpha q/\sqrt{2\pi}$ and mean zero. A theorem in [Reg09] states that $\mathbf{v} \leftarrow \mathbf{A}^t\mathbf{s} + \mathbf{e}$ is indistinguishable from uniform if $\alpha > \sqrt{n}/q$ by a worst-case to average-case reduction, i.e., solving decision LWE implies solving several worst-case lattice problems in dimension n with approximation factors in $\widetilde{\mathcal{O}}(n/\alpha)$. Thus, choosing a large α ensures worst-case hardness but it increases the probability of a decryption error. We let this reduction govern the choice of α but there are further restrictions, coming from the individual cryptosystems. Regev's reduction relies on quantum computation but it was "dequantized" by Peikert in [Pei09]. Although Peikert requires $q = 2^{\mathcal{O}(n)}$ for the dequantization to work, we stick to $q = \mathsf{poly}(n)$. It is more practical and, similar to SIS, the worst-case to average-case reduction should not be more than a guideline for choosing actual parameters. Since there is a circular dependency in the parameters, we will make a sensible choice for q before choosing the remaining parameters. Having chosen a complete set of parameters, we verify that all constraints are satisfied.

The assumption that $(\mathbf{A}, \mathbf{v})$ is close to uniform helps in proving CPA security of all subsequent constructions. In Regev's LWE construction it is used to show indistinguishability of the public key from uniform, while dual-LWE and trapdoor-LWE rely on this assumption for proving the same for the ciphertexts. The uniform distribution of ciphertexts (Regev) and keys (dual, trapdoor) is ensured by the particular choice of m by the leftover-hash lemma [HILL99]. To get $2^{-\kappa}$-uniformity, we essentially require that $\sqrt{q^n/|D|^m} \leq 2^{-\kappa}$, where $D \subset \mathbb{Z}$ is the set from which we choose our randomness.

Ring-LWE. Although the ring (or ideal) analogue of LWE in [LPR10] extends to arbitrary cyclotomic number fields, we will work over a special ring for efficiency reasons and for ease of exposition. Our particular ring $\mathbf{R} = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ requires that n is a power of two and that $q \equiv 1 \pmod{2n}$. Hence, instead of working over matrices, we now work over the ring $\mathbf{R}$, over the subsets $\mathbf{D}_r = (\mathbb{Z} \cap \{-\lfloor r/2 \rfloor, \ldots, \lceil r/2 \rceil\})[x]/\langle x^n + 1\rangle$ for $r \geq 1$, as well as over the $\mathbf{R}$-module $\mathbf{R}^m$. Notice that $\mathbf{D}_1 = (\mathbb{Z} \cap \{0, 1\})[x]/\langle x^n + 1\rangle$. Elements from the $\mathbf{R}$-module $\mathbf{R}^m$ are denoted with a hat, $\hat{\mathbf{x}}$. There are two multiplications in $\mathbf{R}^m$. The first is the usual component-wise $\hat{\mathbf{x}}\mathbf{y} = (\mathbf{x}_1\mathbf{y}, \ldots, \mathbf{x}_m\mathbf{y}) \in \mathbf{R}^m$ and the second is a convolution $\circledast : \mathbf{R}^m \times \mathbf{R}^m \to \mathbf{R}$, $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \mapsto \sum_{i=1}^{m} \mathbf{x}_i \mathbf{y}_i$. Notice that, here, m is not $\Omega(n\log(n))$ but only $\Omega(\log(n))$. The total "dimension", however, is again $\Omega(n\log(n))$ because $\mathbf{R} \cong \mathbb{Z}_q^n$.

Also, the error distribution is different for ring-LWE. The proofs in [LPR10] require an axis-aligned ellipsoidal Gaussian distribution over $\mathbf{R}$, which we will denote with $\chi_{\mathbf{R},\alpha}$. The per-axis Gaussian parameters are bounded by α and the exact shape is inconsequential for our analysis. Hence, we omit the details.

The corresponding decision problem becomes: Given $\hat{\mathbf{a}} \xleftarrow{\$} \mathbf{R}^m$ and either $\hat{\mathbf{r}} \xleftarrow{\$} \mathbf{R}^m$ *or* $\hat{\mathbf{a}}\mathbf{s} + \hat{\mathbf{e}} \in \mathbf{R}^m$ for $s \xleftarrow{\$} \mathbf{R}$ and $\hat{\mathbf{e}} \leftarrow \chi_{\mathbf{R},\alpha}^m$ with certain per-axis parameters, the task is to distinguish the two cases. As with LWE, ring-LWE offers a search-decision equivalence.

The worst-case to average-case reduction for ring-LWE is slightly more demanding than in (ordinary) LWE. Roughly speaking, it states that distinguishing the ring-LWE distribution from uniform for $\alpha > \sqrt{n}\log(n)/q$ is equivalent to solving several ideal lattice problems with approximation factors in $\widetilde{\mathcal{O}}(n\sqrt{n}/\alpha)$.

Again, the decision ring-LWE assumption is used to establish indistinguishability of keys (Regev) and ciphertexts (dual, trapdoor) and the uniform distribution of ciphertexts (Regev) and keys (dual, trapdoor) is now guaranteed by a ring-version of the leftover-hash lemma. The first ring-version due to Micciancio [Mic07] essentially requires $m = \widetilde{\Omega}(n)$, whereas $m/n = \widetilde{\mathcal{O}}(1)$ is sufficient for regular LWE for a negligible statistical distance from uniform. Otherwise, the statistical distance would not be small enough for small, practical values of n. This is because of the complete splitting of $\mathbf{x^n} + \mathbf{1}$ is within the worst case for regularity.

There is a second ring-version of the leftover-hash lemma that has been communicated to us by Regev [Reg10]. It studies regularity of the convolution $\hat{\mathbf{a}} \circledast \hat{\mathbf{x}}$, where the $\mathbf{a}_i$ are invertible in $\mathbf{R}$, i.e., all coefficients of $\mathbf{a}_i$ are non-zero. We defer the details and work with the "normal" leftover hash lemma by replacing m with nm for now.

As will become obvious below, ring-LWE helps reduce the public key size at the expense of having a larger ciphertext and modulus. In addition, ring-LWE can improve the computational efficiency due to fast FFT-multiplications in the employed polynomial rings.

Decryption Errors. For the decryption process to work, we need to bound the errors that are induced during encryption. In each cryptosystem, the error comes from two sources. Firstly, a rounding error of magnitude $1/(2q)$ that can be bounded with certainty by choosing a q that is sufficiently large. We will assume $q > 6$, i.e., a rounding error of $< 1/12$. Secondly, there is an error x that follows a normal distribution with parameter s. Thus, in principle, the error can be arbitrarily large. However, there is a tail bound for $\mathrm{Prob}[\,|x| \geq ts]$, $t \geq 1$. It states that $e^{-\pi t^2}$ is a very good approximation (see, e.g., [Pei07]). We want the decryption-error probability to be less than 2^{-80} in all ℓ components of the ciphertext. Thus, we need $1 - (1 - e^{-\pi t^2})^\ell < 2^{-80}$.

For all relevant parameters, setting $t = 5$ is sufficient. In order for the relative total error to be less than $1/4$ (to be able to decrypt), we require that $ts < 1/6$. Consequently, we need to ensure that the error is distributed with $s = 1/30$.

Hybrid Encryption. We assume that one uses hybrid encryption in practice. The employed block cipher has key length κ and we want it to remain secure in the presence of quantum computers (see Table 2).

Multi-bit LWE. The multi-bit version of Regev's LWE cryptosystem [Reg09] looks as follows.

Secret Key: $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n \times \kappa}$, i.e, $n\kappa\log_2(q)$ bits.

Public Key: $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{P} = \mathbf{A}^t\mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m \times \kappa}$ for $\mathbf{E} \leftarrow \chi_\alpha^{m \times \kappa}$. The matrix $\mathbf{A}$ can be the same for all users, e.g., generated from the random bits of π. Using the HNF technique of [Mic01], the key is reduced to $(m-n)\kappa \log_2(q)$ bits.

Plaintext: $\mathbf{k} \in \mathbb{Z}_2^{\kappa}$.

Ciphertext: $\mathbf{u} = \mathbf{A}\mathbf{a} \in \mathbb{Z}_q^n$, $\mathbf{c} = \mathbf{P}^t\mathbf{a} + \mathbf{k}\frac{q-1}{2}$, where $\mathbf{a} \xleftarrow{\$} \{-\lfloor r/2 \rfloor, \ldots, \lceil r/2 \rceil\}^m$, $r \geq 1$. The ciphertext has $(n+\kappa)\log_2(q)$ bits.

Decryption: $\mathbf{c} - \mathbf{S}^t\mathbf{u} \approx \mathbf{k}\frac{q-1}{2}$.

We need to set $\alpha = 1/(30\sqrt{m}\lceil r/2 \rceil)$ to eliminate decryption errors because then the accumulated error in $\mathbf{c}$ is distributed as a Gaussian with parameter $s = 1/30$, which limits it to at most $1/6$ per component with high probability. For simplicity, we choose $r = 2$. Notice that other trade offs, e.g., choosing a different (non-binary) alphabet or choosing a larger r, are possible and easy to implement.

We let $q = q(n)$ be the smallest prime between $2n^2$ and $4n^2$ to resolve a circular dependency. Then, we set $m = m(n) = \lceil ((n+\kappa)\log_2(q) + 2\kappa)/\log_2(r+1) \rceil$ to tie the probability of being able to distinguish ciphertexts from uniform to the symmetric security level, i.e., the probability is at most $\sqrt{q^{n+\kappa}/(r+1)^m} \leq \sqrt{q^{n+\kappa}/(q^{n+\kappa}2^{2\kappa})} = 2^{-\kappa}$. After taking all this into account, we propose various parameter sets in Table 10. Our parameters differ from the proposed sets of parameters in [MR08] as they are chosen via a completely different methodology. In addition, our parameters do not yield decryption errors but with negligible probability, whereas in [MR08] the error probability is only guaranteed to be $\leq 1/100$ without an additional error correcting code.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	κ	128	150	164	176	190	204	216	230	244	256
Lenstra	n	214	191	221	253	283	314	346	376	405	438
	q	91621	72973	97687	128021	160183	197203	239441	282767	328051	383693
	α	5.47e-04	5.51e-04	5.12e-04	4.80e-04	4.54e-04	4.30e-04	4.10e-04	3.92e-04	3.77e-04	3.63e-04
	m	3719	3665	4234	4815	5400	6006	6609	7215	7811	8446
	|sk|	55.1	56.5	73.3	92.2	113.5	137.5	163	191.2	221	253.9
	|pk|	54.8	63.6	80.3	98	118.7	141.7	165.1	192	220.6	250.3
	$\|C\|$	0.7	0.7	0.8	0.9	1	1.1	1.2	1.3	1.5	1.6

Table 10: Recommended parameters for multi-bit LWE. The rows correspond to attacker types and the columns correspond to security until a given year. C is the ciphertext sizes and all sizes are in kilobytes (kB).

Dual Ring-LWE. Gentry, Peikert, and Vaikuntanathan proposed a dual version of Regev's cryptosystem in [GPV08]. It is "dual" in the sense that public keys and ciphertexts are essentially exchanged. Therefore, the LWE assumption ensures that ciphertexts are indistinguishable from random. The keys are unconditionally random for the proposed parameters. When adapted to the ring setting, the dual cryptosystem looks as follows.

Secret Key: $\hat{\mathbf{r}} \xleftarrow{\$} \mathbf{D}_r^m$, i.e, $mn\log_2(r+1)$ bits.

Public Key: $\hat{\mathbf{a}} \xleftarrow{\$} \mathbf{R}^m$, $\mathbf{u} = \hat{\mathbf{a}} \circledast \hat{\mathbf{r}} \in \mathbf{R}$. Again, $\hat{\mathbf{a}}$ is global and the key requires $n\log_2(q)$ bits.

Plaintext: $\mathbf{k} \in \mathbf{D}_1$, i.e., $\kappa \leq n$.

Ciphertext: $\hat{\mathbf{c}}_1 = \hat{\mathbf{a}}\mathbf{s} + \hat{\mathbf{x}}_1 \in \mathbf{R}^m$, $\mathbf{c}_2 = \mathbf{u}\mathbf{s} + \mathbf{x}_2 + \mathbf{k}\frac{q-1}{2} \in \mathbf{R}$, where $\hat{\mathbf{x}}_1 \leftarrow \chi_{\mathbf{R},\alpha}^m$, $\mathbf{x}_2 \leftarrow \chi_{\mathbf{R},\alpha}$ and $\mathbf{s} \xleftarrow{\$} \mathbf{R}$. The ciphertext has $(m+1)n\log_2(q)$ bits.

Decryption: $\mathbf{c}_2 - \hat{\mathbf{r}} \circledast \hat{\mathbf{c_1}} \approx \mathbf{k}\frac{q-1}{2}$.

We need to set $m = \lceil (\log_2(q) + 2\kappa/n)/\log_2(r+1) \rceil$ to achieve unconditional ($2^{-\kappa}$) uniformity of $\mathbf{u}$ and we choose $q > n^{2.5}$. We use a binary secret key, which makes the ciphertext somewhat larger. Full "duality" with multi-bit LWE is established with a ternary secret key ($r = 2$). When analyzing the Gaussian error, we need to be more careful as it comes from two sources, $\hat{\mathbf{r}} \circledast \hat{\mathbf{x}}_1$ and $\mathbf{x}_2$ in the dual construction. The errors accumulate in a different way because of the convolution $\circledast$. Here, we have that $\hat{\mathbf{r}} \circledast \hat{\mathbf{x}}_1 + \hat{\mathbf{x}}_2$ is distributed like a Gaussian with parameter $(\sqrt{mn}\,\lceil r/2 \rceil + 1)\alpha$. Hence, setting $\alpha = 1/(30(\sqrt{mn}\,\lceil r/2 \rceil + 1))$ Our proposed parameter sets are in Table 11.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	κ	128	150	164	176	190	204	216	230	244	256
Lenstra	n	256	256	256	256	256	512	512	512	512	512
	q	1049089	1049089	1049089	1049089	1049089	5941249	5941249	5941249	5941249	5941249
	α	4.38e-04	4.38e-04	4.38e-04	4.38e-04	4.38e-04	2.98e-04	2.98e-04	2.98e-04	2.98e-04	2.98e-04
	m	22	22	22	22	22	24	24	24	24	24
	$\lvert\mathsf{sk}\rvert$	0.7	0.7	0.7	0.7	0.7	1.5	1.5	1.5	1.5	1.5
	$\lvert\mathsf{pk}\rvert$	0.6	0.6	0.6	0.6	0.6	1.4	1.4	1.4	1.4	1.4
	$\lvert C\rvert$	14.4	14.4	14.4	14.4	14.4	35.2	35.2	35.2	35.2	35.2

Table 11: Recommended parameters for dual ring-LWE. The rows correspond to attacker types and the columns correspond to security until a given year. C is the ciphertext sizes and all sizes are in kilobytes (kB).

Multi-bit Ring-LWE. The ring version of multi-bit ring-LWE can be defined as follows using the sets $\mathbf{R}, \mathbf{D}_r$ from above.

Secret Key: $\mathbf{s} \xleftarrow{\$} \mathbf{R}$, i.e, $n \log_2(q)$ bits.

Public Key: $\hat{\mathbf{a}} \xleftarrow{\$} \mathbf{R}^m$, $\hat{\mathbf{p}} = \hat{\mathbf{a}}\mathbf{s} + \hat{\mathbf{e}} \in \mathbf{R}^m$ for $\hat{\mathbf{e}} \leftarrow \chi^m_{\mathbf{R},\alpha}$. The element $\hat{\mathbf{a}}$ can be the same for all users. The public-key size is $mn \log_2(q)$ bits.

Plaintext: $\mathbf{k} \in \mathbf{D}_1$, i.e., $\kappa \leq n$.

Ciphertext: $\mathbf{u} = \hat{\mathbf{a}} \circledast \hat{\mathbf{r}} \in \mathbf{R}$, $\mathbf{c} = \hat{\mathbf{p}} \circledast \hat{\mathbf{r}} + \mathbf{k}\frac{q-1}{2} \in \mathbf{R}$, where $\hat{\mathbf{r}} \xleftarrow{\$} \mathbf{D}^m_r$. The ciphertext has $2n \log_2(q)$ bits.

Decryption: $\mathbf{c} - \mathbf{s}\mathbf{u} \approx \mathbf{k}\frac{q-1}{2}$.

Notice that we actually encrypt more than κ bits because it is always less than the plaintext size n. This slack can be used to simultaneously encapsulate more than one key. See above for the general setup for ring-LWE. In order to be able to decrypt, we require that the accumulated error term $\hat{\mathbf{e}} \circledast \hat{\mathbf{r}}$ has a small max-norm of at most $q/4$. The accumulated error is now generated differently, namely as a sum of m products of polynomials, where one polynomial is the error term and the second is always a polynomial in $\mathbf{D}_r$. Thus, the resulting error is a Gaussian with parameter $\leq \sqrt{mn}\,\lceil r/2 \rceil\,\alpha$ and we can set $\alpha = 1/(30\sqrt{mn}\,\lceil r/2 \rceil)$ to eliminate decryption errors because then the error is distributed as a Gaussian with parameter $s = 1/30$ and very likely to be less than $1/6$ per component. For simplicity, we let $r = 2$ as in multi-bit LWE. We let $q = q(n)$ be the least prime $> n^{2.5}$ according to the requirements of our specific ring $\mathbf{R}$ that are discussed above.

Then, we set $m = m(n) = \lceil (2\kappa/n + \log_2(q))/\log_2(r+1) \rceil$ to make $\mathbf{u}$ $2^{-\kappa}$-uniform by Micciancio's ring version of the leftover hash lemma. Again, we only show one option of choosing the parameters. For example, a bigger r allows smaller m and therefore smaller key sizes, but bigger errors. We propose various parameter sets in Table 12.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	κ	128	150	164	176	190	204	216	230	244	256
Lenstra	n	256	256	256	256	256	512	512	512	512	512
	q	1049089	1049089	1049089	1049089	1049089	5941249	5941249	5941249	5941249	5941249
	α	5.57e-04	5.57e-04	5.57e-04	5.57e-04	5.57e-04	3.80e-04	3.80e-04	3.68e-04	3.68e-04	3.68e-04
	m	14	14	14	14	14	15	15	16	16	16
	$\lvert\mathsf{sk}\rvert$	0.6	0.6	0.6	0.6	0.6	1.4	1.4	1.4	1.4	1.4
	$\lvert\mathsf{pk}\rvert$	8.8	8.8	8.8	8.8	8.8	21.1	21.1	22.5	22.5	22.5
	$\lvert C\rvert$	1.3	1.3	1.3	1.3	1.3	2.8	2.8	2.8	2.8	2.8

Table 12: Recommended parameters for multi-bit ring-LWE. The rows correspond to attacker types and the columns correspond to security until a given year. C is the ciphertext sizes and all sizes are in kilobytes (kB).

Dual-LWE. Gentry, Peikert, and Vaikuntanathan proposed a dual version of Regev's cryptosystem in [GPV08]. It is "dual" in the sense that public keys and ciphertexts are essentially exchanged. Therefore, the LWE assumption ensures that ciphertexts are indistinguishable from random. The keys are unconditionally random for the proposed parameters. We use a variant of the scheme in [Pei09].

Secret Key: $\mathbf{X} \overset{\$}{\leftarrow} \{-\lfloor r/2 \rfloor, \ldots, \lceil r/2 \rceil\}_2^{m\times\kappa}$ for $r \geq 1$, i.e, $m\kappa \log_2(r+1)$ bits.

Public Key: $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times m}$, $\mathbf{U} = \mathbf{AX} \in \mathbb{Z}_q^{n\times\kappa}$. Again, $\mathbf{A}$ is global. The key requires $n\kappa \log_2(q)$ bits.

Plaintext: $\mathbf{k} \in \mathbb{Z}_2^{\kappa}$.

Ciphertext: $\mathbf{c}_1 = \mathbf{A}^t\mathbf{s} + \mathbf{x}_1 \in \mathbb{Z}_q^m$, $\mathbf{c}_2 = \mathbf{U}^t\mathbf{s} + \mathbf{x}_2 + \mathbf{k}\frac{q-1}{2} \in \mathbb{Z}_q^{\kappa}$, where $\mathbf{x}_1 \leftarrow \chi_\alpha^m$, $\mathbf{x}_2 \leftarrow \chi_\alpha^\kappa$ and $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$. The ciphertext has $(m+\kappa)\log_2(q)$ bits.

Decryption: $\mathbf{c}_2 - \mathbf{X}^t\mathbf{c}_1 \approx \mathbf{k}\frac{q-1}{2}$

We do not explicitly consider the dequantization of LWE in [Pei09] as it requires $q = 2^{\mathcal{O}(n)}$, which dramatically increases the public-key size. Moreover, by choosing $q \leq \mathsf{poly}(n)$, the encryption process is slightly simpler. Here, we let $q = q(n)$ be the smallest prime between $2n^2$ and $4n^2$ to resolve a circular dependency. As for the secret key, we choose $r = 1$ to demonstrate how small the secret key can be, but choosing $\mathbf{X}$ from a larger set has the advantage of a smaller ciphertext (but bigger accumulted errors). The desired trade off depends on the target application. To ensure that the public key is within distance $2^{-\kappa}$ from uniform, we set $m = \lceil (n\log_2(q) + 2\kappa)/\log_2(r+1)\rceil$. Then, the statistical distance is at most $\sqrt{q^{n\kappa}/(r+1)^{m\kappa}} \leq \sqrt{q^{n\kappa}/(q^{n\kappa}2^{2\kappa})} = 2^{-\kappa}$. As for α, we need to ensure that the induced errors, distributed according to a Gaussian with parameter at most $\alpha(\sqrt{m}\lceil r/2\rceil + 1)$, are less than $1/6$. Thus, setting $\alpha = 1/(30(\sqrt{m}\lceil r/2\rceil + 1))$ is sufficient. Given these relations among the parameters, we propose secure parameter sets in Table 13.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	κ	128	150	164	176	190	204	216	230	244	256
Lenstra	n	215	190	220	253	284	314	347	377	407	440
	q	92459	72211	96821	128021	161323	197203	240829	284261	331301	387203
	α	5.32e-04	5.65e-04	5.21e-04	4.82e-04	4.52e-04	4.27e-04	4.04e-04	3.86e-04	3.70e-04	3.54e-04
	m	3803	3367	3972	4645	5294	5932	6636	7291	7952	8680
	$\lvert\mathsf{sk}\rvert$	59.4	61.7	79.5	99.8	122.8	147.7	175	204.7	236.9	271.3
	$\lvert\mathsf{pk}\rvert$	55.4	56.2	72.9	92.2	114	137.5	163.6	191.8	222.3	255.2
	$\lvert C\rvert$	7.9	6.9	8.4	10	11.6	13.2	15	16.6	18.3	20.2

Table 13: Recommended parameters for dual-LWE. The rows correspond to attacker types and the columns correspond to security until a given year. C is the ciphertext sizes and all sizes are in kilobytes (kB).

Trapdoor Ring-LWE. In this section, we show how to combine the result in [LPR10] with an earlier work on an ideal version of LWE [SSTX09]. There, the authors show how to generate a trapdoor for LWE as in trapdoor-LWE (similar to the construction in [AP09]). However, their result does not guarantee the hardness of the LWE decision problem, which is why they rely on generic hardcore bits and a subexponential-time reduction. To eliminate this need, we demonstrate that their trapdoor generation algorithm also works in the setting of [LPR10]. We focus on the "rounding-off" version of trapdoor ring-LWE because the construction in [SSTX09] does not bound the length $\tilde{L}$ of the orthogonalized trapdoor. It only guarantees that the basis itself has length at most L. Nevertheless, our approach generalizes to the "nearest-plane" version (see trapdoor-LWE for the details). The scheme works as follows.

Public Key: $\hat{\mathbf{a}} \in \mathbf{R}^m$, $\mathbf{u} \overset{\$}{\leftarrow} \mathbf{R}$. Notice that $\hat{\mathbf{a}}$ cannot be global here as it contains a trapdoor. Fortunately, $\mathbf{u}$ can be the same for all users. Thus, $|\mathsf{pk}| = mn \log_2(q)$ bits.

Secret Key: $\mathbf{T} \in \mathbb{Z}^{mn \times mn}$ such that $\hat{\mathbf{a}}\hat{\mathbf{t_i}} \equiv 0 \bmod q$ for every column $\hat{\mathbf{t}}$ in $\mathbf{T}$ (interpreted as an element of $\mathbf{R}^m$). The basis length is $\|\mathbf{T}\| \leq L = \sqrt{2n(9\rho + \sigma)}$. When looking closely at the construction, we find that the trapdoor can be reconstructed from $\sigma(m-\sigma)n\sqrt{\sigma n} + \rho(m-\rho)n\log_2(3)$ bits.

Plaintext: $\mathbf{k} \in \mathbf{D}_1$, i.e., $\kappa \leq n$.

Ciphertext: $\hat{\mathbf{c}}_1 = \hat{\mathbf{a}}\mathbf{s} + \hat{\mathbf{x}}_1 \in \mathbf{R}^m$, $\mathbf{c}_2 = \mathbf{u}\mathbf{s} + \mathbf{x}_2 + \mathbf{k}\lfloor q/2 \rfloor \in \mathbf{R}$, where $\hat{\mathbf{x}}_1 \leftarrow \chi^m_{\mathbf{R},\alpha}$, $\mathbf{x}_2 \leftarrow \chi_{\mathbf{R},\alpha}$ and $\mathbf{s} \overset{\$}{\leftarrow} \mathbf{R}$. The ciphertext has $(mn+n)\log_2(q)$ bits.

The parameters σ and ρ control the success probability of the trapdoor generator and the uniformity of $\hat{\mathbf{a}}$, respectively. Furthermore, the influence the total lattice dimension mn, namely, $m = (\lceil \log_2(q) + \sigma \rceil)(\sigma + \rho)$. Unfortunately, the setting required in [LPR10] is within the worst-case for the trapdoor generation algorithm in [SSTX09]. Particularly, the fact that $x^n + 1$ splits completely into n degree-1 polynomials over $\mathbb{Z}_q$ makes it necessary to increase the overall lattice dimension. In particular, we require $\rho = \Omega(\kappa + \log(q))$ instead of just $\rho = \mathcal{O}(\log(q))$ (as in ideal GPV) to ensure a well-distributed $\hat{\mathbf{a}}$.

We fix $\sigma = 1$, resulting in a slightly skewed ($\leq 1-(1-1/q)^n$ distance) distribution, where $\mathbf{a}_1$ is always invertible in $\mathbf{R}$ and a success probability $\geq (1-1/q)^n$ that converges to 1 as n increases. This does not harm security. However, we require that the remaining $\mathbf{a}_i$, $i > 1$, are within $2^{-\kappa}$ distance from uniform. To this end, it is sufficient to set $\rho = (y + \log_2(q))/\log_2(3)$ for $y = 1/2\sqrt{8\kappa + 16\log\log_2(q) + 1} + 1 + 2\kappa + 4\log\log_2(q)$. Alternatively, we can re-run the algorithm until we obtain $\hat{\mathbf{a}}$ with only non-zero coefficients. Then, the modified regularity lemma holds and we can use $\rho = \rho(n) \geq \lceil (2\kappa/n + \log_2(q))/\log_2(3) \rceil$.

The induced error is a rounding error $\leq 1/4$ if $q \geq 2L\sqrt{m}$ and a Gaussian with parameter $\leq \alpha L$. The Gaussian error needs to be $< 1/4$, i.e., setting $\alpha = 1/(L20)$ is sufficient. An admissible q is the smallest prime $\geq 2n^{2.5}$ with $q \equiv 1 \pmod{2n}$. Table 14 shows the resulting parameter sets.

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	κ	128	150	164	176	190	204	216	230	244	256
Lenstra	n	256	256	256	256	512	512	512	512	512	512
	q	2100737	2100737	2100737	2100737	11867137	11867137	11867137	11867137	11867137	11867137
	α	1.89e-04	1.89e-04	1.89e-04	1.89e-04	1.30e-04	1.30e-04	1.30e-04	1.30e-04	1.30e-04	1.30e-04
	m	368	368	368	368	425	425	425	425	425	425
	$\lvert\mathsf{sk}\rvert$	446	446	446	446	1248	1248	1248	1248	1248	1248
	$\lvert\mathsf{pk}\rvert$	242	242	242	242	624	624	624	624	624	624
	$\lvert C\rvert$	242	242	242	242	626	626	626	626	626	626

Table 14: Recommended parameters for trapdoor ring-LWE with "rounding-off". The rows correspond to attacker types and the columns correspond to security until a given year. C is the ciphertext sizes and all sizes are in kilobytes (kB).

Trapdoor-LWE. The trapdoor-LWE cryptosystem [GPV08, Pei09] is similar to dual-LWE. The main difference is that the secret key is a trapdoor $\mathbf{T}$ for the lattice $\Lambda_q^\perp(\mathbf{A})$, i.e., a short basis thereof. It is generated via [AP09]. The secret key $\mathbf{X}$ in dual-LWE disappears and we cannot share the matrix $\mathbf{A}$ among all users. The scheme comes in two flavours. The first uses what is called "rounding-off" for decryption and the second involves Babai's nearest plane algorithm [Bab86]. The advantage of Babai's algorithm is that we can correct bigger errors compared to rounding-off. However, rounding-off is more efficient. We describe both in the following.

Obviously, trapdoor-LWE has numerous caveats when compared to its "trapdoor-less" counterparts. It should not be used for plain CPA encryption but it is, e.g., necessary for constructing chosen-ciphertext (CCA) secure encryption [PW08, RS09, Pei09] based on LWE by essentially applying $\Theta(n)$ independent trapdoors to the same input.

Let $L = \|\mathbf{T}\| = \max_i(\|\mathbf{t}_i\|_2)$ be the basis length, where the $\mathbf{t}_i$ are the columns of $\mathbf{T}$. Similarly, we denote the basis length of the Gram-Schmidt orthogonalization $\tilde{\mathbf{T}}$ of $\mathbf{T}$ with $\tilde{L}$.

Public Key: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{U} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times \kappa}$. Notice that $\mathbf{A}$ cannot be global here as it contains a trapdoor. Fortunately, $\mathbf{U}$ can be the same for all users. Thus, $|\mathsf{pk}| = nm\log_2(q)$ bits.

Secret Key: $\mathbf{T} \in \mathbb{Z}^{m \times m}$ such that $\mathbf{AT} \equiv 0 \bmod q$. By looking closely at the construction in [AP09], we find that it can be restored from just $2m_1m_2 + m_1\log_2(q)$ bits for "rounding-off" and $2m_1m_2 + m_1\log_2(q) + 64 * (m_1 + m_2)m_1$ for "nearest-plane" because one needs the Gram-Schmidt orthogonalization. Here, we assume a IEEE 754 double precision data type is sufficient. The length is $\tilde{L} \leq 1 + 20\sqrt{m_1}$ for "rounding-off" and $L \leq 20n\log(q)$ for "nearest-plane".

Plaintext: $\mathbf{k} \in \mathbb{Z}_t^{\kappa}$.

Ciphertext: $\mathbf{c}_1 = \mathbf{A}^t\mathbf{s} + \mathbf{x}_1 \in \mathbb{Z}_q^m$, $\mathbf{c}_2 = \mathbf{U}^t\mathbf{s} + \mathbf{x}_2 + \mathbf{k}\frac{q-1}{2} \in \mathbb{Z}_q^{\kappa}$, where $\mathbf{x}_1 \leftarrow \chi_\alpha^m$, $\mathbf{x}_2 \leftarrow \chi_\alpha^\kappa$ and $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$. The ciphertext has $(m + \kappa)\log_2(q)$ bits.

Decryption: Recover $\mathbf{s}$ from $\mathbf{c}_1$, using the trapdoor. Then, $\mathbf{c}_2 - \mathbf{U}^t\mathbf{s} \approx \mathbf{k}\frac{q-1}{2}$.

The parameters $m = m_1 + m_2$ is determined by the trapdoor algorithm in [AP09]. The algorithm requires $m_1 = \lceil(1+\varphi)n\log_2(q)\rceil$ and $m_2 = \lceil(4+2\varphi)n\log_2(q)\rceil$, where q depends on the decryption method as we will see below and φ is chosen 0.1 as explained in the GPV signature case.

In both variants, decryption recovers $\mathbf{s}$ from $\mathbf{c}_1$ and then $\mathbf{k}$ from $\mathbf{c}_2$. The induced error is a rounding error $\leq 1/4$ if $q \geq 2L\sqrt{m}$ ($q \geq 2\tilde{L}\sqrt{m}$) and a Gaussian with parameter $\leq \alpha L$ (rounding-off) or $\leq \alpha\tilde{L}$ (Nearest plane). The Gaussian error needs to be $< 1/4$, i.e., setting $\alpha = 1/(L20)$ or $\alpha = 1/(\tilde{L}20)$ is sufficient. The advantage of the "nearest plane" approach becomes obvious as we can have a bigger α and with that a harder worst-case problem. This also affects q because we require $q > \sqrt{n}/\alpha$ in the worst-case to average-case reduction. An admissible q is the smallest prime between n^4 and $2n^4$ (rounding-off), or between n^3 and $2n^3$ (nearest plane). Table 15 shows the resulting parameter sets for "nearest plane".

	year	2018	2010	2020	2030	2040	2050	2060	2070	2080	2090
	κ	128	150	164	176	190	204	216	230	244	256
Lenstra	n	259	229	264	302	338	373	410	445	480	517
	q	17373989	12008999	18399749	27543611	38614483	51895141	68921003	88121141	1.11e+08	1.38e+08
	α	3.02e-05	3.25e-05	2.98e-05	2.76e-05	2.58e-05	2.44e-05	2.31e-05	2.20e-05	2.10e-05	2.02e-05
	m	33015	28545	33768	39560	45149	50667	56583	62249	67978	74099
	$\|\mathsf{sk}\|$	1811121	1354059	1894872	2600580	3387284	4265727	5320089	6438898	7678583	9123402
	$\|\mathsf{pk}\|$	25104	18766	26262	36044	46948	59126	73739	89246	106431	126460
	$\|C\|$	97	82	100	120	139	159	181	201	223	245

Table 15: Recommended parameters for trapdoor-LWE with "nearest-plane". The rows correspond to attacker types and the columns correspond to security until a given year. C is the ciphertext sizes and all sizes are in kilobytes (kB).

A.3. Remarks

When looking at how modestly the parameters need to grow with increasing security demands, we clearly see one of the advantages for lattice-based cryptography. The downside is that all schemes that require an actual trapdoor are quite impractical. Here, our secret key sizes reflect the least number of bits that are necessary to reconstruct the trapdoor. This introduces a significant computational overhead as the Gram-Schmidt orthogonalization of the trapdoor is often required. Storing the orthogonalization of the matrix, however, results in a secret key that is bigger by magnitudes.

A general observation regarding ideal lattices over the ring $\mathbb{Z}_q[x]/(x^n + 1)$ is that it is desirable for efficient implementations but it does not allow a fine-grained parameter selection because n needs to be a power of 2. In consequence, some of the proposed parameter sets provide more security than required.

Signatures. All signature schemes using a trapdoor come with large key, in the order of megabytes or even gigabytes, and signature sizes. The most practical scheme is the Treeless signature scheme (requiring random oracles). The LM-OTS scheme has small keys and signatures, but it is only "one-time". The GPV and Bonsai schemes, even when instantiated with ideal lattices, are far from being practical.

Encryption. Regarding lattice-based encryption schemes, there is no perfect choice. The most suitable scheme depends on the exact application scenario. However, there is a simple classification: multi-bit (ring-)LWE offers the smallest ciphertexts, dual (ring-)LWE has the smallest public keys, and trapdoor (ring-)LWE gives rise to CCA secure encryption. For plain CPA encryption, using trapdoor-LWE is discouraged because it is rather impractical due to its huge secret key. The effect of using the respective "ring" variants is a significant improvement of the public-key size and of the computational efficiency. Furthermore, it improves the secret-key size. The caveat is that the modulus q increases, and with it the ciphertext size. Regarding the ring-version trapdoor-LWE, we conclude that it helps reduce both, the secret- and public-key sizes at the expense of a rather large ciphertext.