Selecting Secure Parameters for Lattice-based Cryptography

Markus Rückert* Michael Schneider

{rueckert,mischnei}@cdc.informatik.tu-darmstadt.de

Cryptography and Computeralgebra
Department of Computer Science
TU Darmstadt

March 12, 2010

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 question of how to systematically choose secure parameters in practice is 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 for estimating secure parameter sets by relating the hardness of practical lattice basis reduction to symmetric "bit security". 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.

While we restrict the discussion to encryption and signature schemes, our result is applicable to almost all lattice-based cryptosystems. More precisely, on those that are based on the learning with errors problem (LWE) or the small integer solution problem (SIS).

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

*This work was supported by CASED (www.cased.de).

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 a lattice.

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 desireable 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 $\mathcal{O}(n^2 \log(n))$ with rather large constants, where n is the natural system parameter. Fortunately, we can use ideal lattices, introduced by Micciancio [Mic07] and 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 for lattice-based cryptography. Indeed, the task is more involved than in the case of RSA or symmetric ciphers. Lattice cryptosystems have more than one parameter that affects security and dealing with n alone is not sufficient.

So far, only Micciancio and Regev [MR08] and Lyubashevky [Lyu09] have proposed secure parameters for their schemes based on an interesting observation by Gama and Nguyen [GN08b]. They consider the Hermite Short Vector Problem HSVP with parameter $\delta \geq 1$ in lattices L of dimension m. 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 and Mayer distribution [GM03] that are considered to provide hard instances of HSVP. Their observation is that δ is the dominating parameter and n only plays a minor role. They conjecture, that HSVP is infeasible for $\delta < 1.01$ 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, δ can 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. Micciancio and Regev describe this sublattice attack in [MR08]. The bottom line is that solving ζ-HSVP_q in a q-ary lattice of dimension m is only as hard as solving δ-HSVP in dimension d, where $d < m$ and $\delta > \zeta$. Thus, HSVP becomes strictly easier in q-ary lattices because there is some "slack" in the required attack dimension. Moreover, the numbers involved are bounded by $q \leq \mathsf{poly}(n)$, whereas random Goldstein-Mayer lattices require that q is exponential in n.

With this knowledge, two 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 $\mathcal{O}(\sqrt{n \log(n)})$. 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 computations units.

With our work, we focus on lattice-based encryption [Reg09, GPV08, Pei09, SSTX09] and signature schemes [GPV08, SSTX09, Lyu08, LM08, CHKP10] because they are the main building blocks of public-key cryptography. 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 schemes like NTRU [HPS98] that come without a security proof because secure parameters for this efficient scheme are already known and standardized. With our work, we rather demonstrate how practical (or impractical) certain provably secure schemes currently are.

Our Contribution. Inspired by the works of A.K. Lenstra and Verheul [LV01] and the subsequent update by Lenstra [Len05], we propose a methodology for selecting secure parameters for lattice-based cryptography. To this end, we adopt the handy notion of dollar-days, i.e., equipment cost in dollar times running 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 conduct experiments on a wide range of cryptographic lattices. Like Gama and Nguyen, 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. Moreover, we assume that both, technological and algorithmic progress, follow the "double Moore Law", i.e., the required attack effort decreases by a factor 2 every 9 months. Putting this assumption and $T(\delta)$ together, we suggest secure parameter sets for the above encryption and signature schemes in Section 4. Here, we also provide a comprehensive comparison of the state-of-the-art in lattice cryptography. For each scheme, we deal with all attacker types and present parameters that are secure for the next $10, 20, \ldots, 100$ years. Our extrapolation is quite conservative and overestimates the attacker to account for algorithmic improvements beyond our "double Moore law".

Interestingly, our estimation shows that, today, $\delta = 1.01$ seems 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.

2 Preliminaries

We denote $\log x$ the logarithm to base e, all other logarithms are marked, e.g., $\log_2 x$. Vectors and matrices are written in bold, e.g., $\mathbf{v}$ and $\mathbf{M}$.

2.1 Lattices

A lattice in $\mathbb{R}^n$ is a discrete subgroup $\Lambda = \{\sum_{i=1}^{d} x_i \, \mathbf{b}_i \,|\, x_i \in \mathbb{Z}\}$, generated by a matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_d] \in \mathbb{Z}^{n \times d}$ of $\mathbb{R}$-linearly independent vectors ($d \leq n$). The matrix $\mathbf{B}$ is a basis of the lattice Λ and we write $\Lambda = \Lambda(\mathbf{B})$. For $d \geq 2$, there are infinitely many bases for the same lattice. 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 invariant is the first successive minimum $\lambda_1(\Lambda)$, which is the Euclidean length of the shortest, non-zero vector in Λ. For a lattice $\Lambda(\mathbf{B})$ with $\mathbf{B} \in \mathbb{R}^{n\times n}$ define the 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 $\mathbf{y} \in \Lambda(\mathbf{B})$. We know that $(\Lambda(\mathbf{B}))^* = \Lambda((\mathbf{B}^{-1})^T)$.

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 $\|\mathbf{v}\|_2 \leq \gamma\lambda_1(\Lambda)$. 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 algorithm is exponential (space and time) [AKS01]. The algorithm mostly used in practice is the BKZ algorithm [SE94].

In cryptography, we use lattices of special structure, 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., it is $1/q \cdot \Lambda_q^\perp(\mathbf{A}) = (\Lambda_q(\mathbf{A}))^*$. The determinant of a q-ary lattice is q^n.

The main computational problem in $\Lambda_q^\perp(\mathbf{A})$ is the "small integer solution" problem (SIS): given $n, m, q, \mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a norm bound ν, find $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A})$ with $\|\mathbf{v}\|_2 \leq \nu$. 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, rounded 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. Finally, the Shortest Independent Vectors Problem SIVP asks to find n linearly independent vectors in a lattice, that minimize the quantity $\|\mathbf{V}\| = \max_i \|\mathbf{v}_i\|_2$.

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 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 $\delta \geq 1$.

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 solving the problem for $\delta \geq 1.01$ may be possible even in high dimensions. For smaller δ, the problem is intractable. For every ϵ, δ-HSVP is solvable for all $\delta = 1 + \epsilon$ in time polynomial in the lattice dimension and in $1/\epsilon$ [Sch87, GHGKN06, GN08a]. 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 $\mathcal{O}(2^{\dim(\Lambda)})$. We rather have bases with entries of bit length $\log_2(q) = \mathcal{O}(\log_2(n))$. 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 4.2, we discuss how LWE can be interpreted 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 $\zeta = \sqrt[m]{\nu/q^{n/m}}$ and ask the Hermite-SVP solver to find $\mathbf{v}$ with $0 < \|\mathbf{v}\|_2 \leq \zeta^m q^{n/m}$. We write $\mathsf{HSVP_q}$ for this direct translation from SIS to HSVP.

Micciancio and Regev demonstrate that ζ-$\mathsf{HSVP_q}$ is typically easier that ζ-HSVP in arbitrary lattices because it is possible to solve $\mathsf{HSVP_q}$ in a sublattice [MR08]. Their approach involves removing random columns from $\mathbf{A}$ such that the resulting lattice has a lower dimension and still contains short vectors. For a given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ they find that the optimum dimension for solving ζ-$\mathsf{HSVP_q}$ is $d = \min\{\sqrt{n \log(q)/\log(\zeta)}, m\}$. 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 smaller than ν in $\Lambda_q^\perp(\mathbf{A})$.

In consequence, this allows us to *normalize* ζ-$\mathsf{HSVP_q}$ by removing the "slack" in the dimension parameter. We end up with an instance of δ-HSVP with $\delta = \sqrt[d]{\nu/q^{n/d}} > \zeta$. The resulting distribution of lattices is what we will analyze by directly applying lattice basis reduction.

Notice that the bases of ideal lattices have essentially the same structure. However, there is no lattice basis reduction algorithm that can take significant advantage of the ideal structure. Therefore we can adapt our analysis to the ideal lattice case.

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 ζ. 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. Therefore we perform new experiments using q-ary lattices $\Lambda_q^\perp$ to get a better estimate of the worst-case hardness of those lattices that are really used in cryptography.

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 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.

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.

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 our experiments were conducted 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 by [GN08a] and [GHGKN06] 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, the time of conducting the experiments. For year y, we need to satisfy the inequality $T(k) \geq T_{2009} \cdot 2^{(y-2009)\cdot 12/9}$ to obtain a secure k 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 be easily replaced if there is a new attack.

3 Analysis

Before we can propose actual parameters, we need to assess the practical hardness of the underlying problem. As we will see in Section 4, the best known attacks against the most recent signature and encryption schemes involve a q-ary lattice $\Lambda = \Lambda_q^{\perp}(\mathbf{A})$ of dimension $m = \Omega(n \log(n))$ and the SIS problem with a scheme-specific norm bound ν. The required norm bound can be obtained by studying the security reductions. Thus, the main goal of this section is to determine the effort T_{2009} (in dollar-days) that is required today for mounting these attacks. 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 is adaptive to focus on the interesting invervals. These parameters also determine n if we demand that $m > n \log_2(q)$. In this setting, we know that $(1+\epsilon)$-HSVP_q has a solution for any $\epsilon > 0$. This is because the function $f_{\mathbf{A}}(\mathbf{v}) = \mathbf{A}\mathbf{v} \pmod q$ admits a collision $(\mathbf{v}, \mathbf{v}') \in \{0,1\}^m \times \{0,1\}^m$

and therefore $\mathbf{v} - \mathbf{v}' \in \Lambda_q^\perp(\mathbf{A})$ with $\|\mathbf{v} - \mathbf{v}'\|_2 \leq \sqrt{m}$. The corresponding ζ quickly tends to 1 with increasing n. We use $\zeta \in [1.01, 1.04]$ to bootstrap and then normalize the resulting problem instances by reducing the dimension. We assume the following conjecture.

Conjecture 1 *For every $n \in \mathbb{N}_{>0}$, constant $c \geq 2$, prime $q \geq n^c$, and $m > n\log_2(q)$, the best known approach to solve* SIS *with parameters (n, q, m, ν) involves solving δ-*HSVP *in dimension $d = \sqrt{n\log(q)/\log(\zeta)}$ for $\zeta = \sqrt[m]{\nu/q^{n/m}} \leq \delta = \sqrt[d]{\nu/q^{n/d}}$.*

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.[1] We apply BKZ with an increasing block size parameter until a vector of the desired length is found. Our first observation is that q plays a minor role if $\delta \in (1, 1.02]$. To see this, compare Figures 2(a) ($q \approx n^2$) and 2(c) ($q \approx n^8$) in Appendix A. 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. Although the dimension plays a noticeable role, the hardness of HSVP is mainly governed by δ and different dimensions result in slightly shifted cost functions. To arrive at very conservative estimates, we use SIS instances with a fix $m = 175$ and n, q accordingly as our reference.[2] For similar reasons, we choose a fix relation $q \approx n^3$ because all cryptosystems in Section 4 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

$$T(\delta) = a2^{-(\log_2(\delta)^b)} + c \tag{1}$$

for real constants a, b, c. We use the (averaged) data samples in Figure 2(d) to find parameters a, b, c for (1) by a least-squares approximation. Now, we can draw our main conjecture, where $n \geq 100$ rules out unnaturally easy cases in small lattice dimensions.

Conjecture 2 *Let all other parameters and relations as in Conjecture 1. For $n \geq 100$ and any $\delta \in (1, 1.02]$, solving δ-*HSVP *(in normalized q-ary lattices) of dimension d involves an effort of at least $T(\delta) = 10^{-15}2^{-(\log_2(\delta)^{1.001})} + 0.005$ dollar-days.*

Extrapolating T for smaller δ yields Figure 1. 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

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 i 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

[1] An AMD Opteron, running at 2.4 GHz.

[2] Choosing a rather small problem dimension m, and therefore a small attack dimension d, is very conservative but it also guarantees that we can average over many data samples for small δ. Our choice was also influenced by the fact that the BKZ algorithm tends to behave badly in large dimensions for block size parameters bigger than 30. With our experiments we avoid this potential bias.

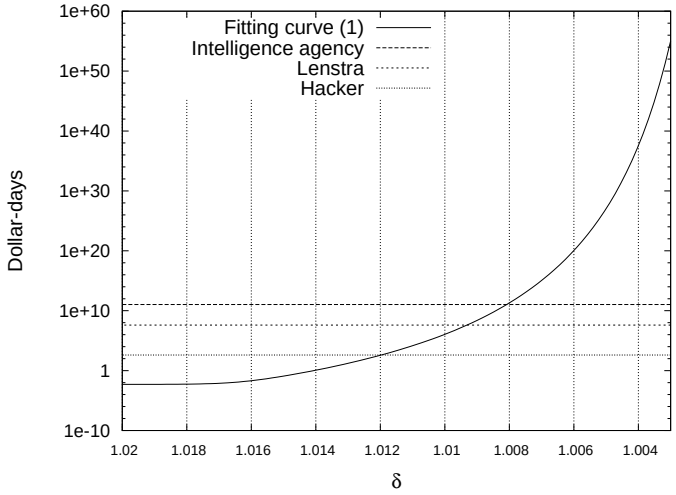


Figure 1: Estimated time complexity of δ-HSVP for $\delta \in [1.003, 1.02]$. The plots include horizontal lines, illustrating today's power of different attacker types.

year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
bit security	75	82	88	95	102	108	115	122	128	135
λ	225	246	264	285	306	324	345	366	384	405
κ	150	164	176	190	204	216	230	244	256	270
Hacker	1.01177	1.00965	1.00808	1.00702	1.00621	1.00552	1.00501	1.00458	1.00419	1.00389
Lenstra	1.00919	1.00785	1.00678	1.00602	1.00541	1.00488	1.00447	1.00413	1.00381	1.00356
Int. agency	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).

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.

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 secure parameters for all SIS and LWE-based cryptosystems in the next section. It also shows the equivalent level of symmetric security, sometimes referred to as "bit security".

4 Estimating Secure Parameters

In this section we analyze the individual signature and encryption schemes.

4.1 Signature Schemes

All 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 (sometimes we need more parameters, but those are the most important) that also fully determines the hardness estimate δ for HSVP. For our choices of n, m, and q, by 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.

Using the attacker dimension $d = \sqrt{n\log(q)/\log(\zeta)}$ we can compute $\delta = \sqrt[d]{\nu/q^{n/d}}$. 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. In this chapter, for each signature scheme we present an extract of the complete parameter tables, which are given in Appendix C. More precise, we present the signature scheme of GPV [GPV08], the Bonsai tree scheme [CHKP10], the one-time signature scheme of [LM08], and Lyubashevsky's treeless signature scheme [Lyu09].

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 $q \geq 3$ (q has to satisfy $q \geq \omega(\sqrt{n\log n}) \cdot poly(n)$, for that the hardness of breaking the scheme can be reduced to a hard worst-case problem). 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 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 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 $\nu = 2(1 + 20\sqrt{m_1})\log(n)\sqrt{m}$. 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].

We choose $m_1 = \lceil (1 + 0.1)n\log_2(q) \rceil$ and $m_2 = \lceil (4 + 0.2)n\log_2(q) \rceil$. An attacker would only use the first m_1 columns of $\mathbf{A}$ for an attack, therefore when calculating ζ we only consider those first columns: $\zeta = (\nu/q^{n/m_1})^{1/m_1}$. For q we choose $q = n^t$ for the smallest t such that $q \geq 2\nu\sqrt{n}\log_2(n)$ (worst-case to average-case reduction). 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 3.

Secret Key: $\mathbf{S} \in \mathbb{Z}^{m \times m}$ with $\|\mathbf{S}\| \leq 20n\log(q)$, i.e, $m^2\log_2(20n\log(q))$ bits.

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	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Lenstra	n	186	216	249	279	310	342	373	402	435	464
	q	1.20e+09	2.18e+09	3.84e+09	6.06e+09	9.24e+09	1.37e+10	1.94e+10	2.61e+10	3.58e+10	4.64e+10
	m_1	6171	7371	8721	9974	11289	12668	14021	15302	16776	18085
	m	29730	35512	42020	48054	54392	61034	67556	73728	80830	87135
	$\|\mathsf{sk}\|$	1.75e+06	2.54e+06	3.61e+06	4.77e+06	6.18e+06	7.86e+06	9.71e+06	1.17e+07	1.41e+07	1.65e+07
	$\|\mathsf{pk}\|$	20356.39	29045.2	40666.67	53183.9	68138.69	85796.45	105112.61	125198.02	150479.54	174870.04
	$\|\sigma\|$	74.16	89.87	107.77	124.56	142.35	161.16	179.76	197.48	218.0	236.32

Table 3: 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).

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. We do not examine this variant here, but we expect that we can apply the same parameters as in the original GPV case.

Bonsai Trees. In [CHKP10], Cash, Hofheinz, Kiltz, and Peikert introduce a tree based signature scheme. It does not require random oracles for the security proof of existential unforgeability. A modified version by Rückert [Rüc10] with 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 $m = m_1 + (\lambda + 1)m_2$.[3] Again, we can use $\varphi = 0.1$. We choose the Gaussian parameter $s = (1 + 20\sqrt{m_1})\log(n)$. q is chosen the same as in the GPV case. If

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

there exists an attack against unforgeability on the signature scheme, then there is a PPT oracle algorithm attacking SIS for $\nu = 2s\sqrt{m}$. The same as in the GPV case, an attacker would only use the first m_1 columns of $\mathbf{A}$, therefore we take this prefix-attack into account when calculating ζ. For the overview of the parameters, refer to Table 4.

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

Secret Key: $\mathbf{S} \in \mathbb{Z}^{m \times m}$ with $\|\mathbf{S}\| \leq 20n\log(q)$, i.e, $m^2 \log_2(20n \log q)$ bits.

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

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

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Lenstra	n	197	229	264	296	328	363	395	427	462	494
	q	2.97e+11	6.30e+11	1.28e+12	2.27e+12	3.80e+12	6.30e+12	9.62e+12	1.42e+13	2.10e+13	2.94e+13
	m_1	8259	9874	11681	13365	15078	16978	18740	20522	22493	24313
	m_2	31533	37699	44599	51030	57567	64825	71551	78355	85880	92831
	m	7134717	9321527	11830416	14607945	17688147	21085103	24775386	28776807	33086293	37713699
	$\|\mathsf{sk}\|$	15949.16	22507.89	31136.82	40397.03	51005.06	64184.71	77709.88	92669.11	110702.84	128743.86
	$\|\mathsf{pk}\|$	1.30e+07	2.04e+07	3.06e+07	4.32e+07	5.91e+07	7.93e+07	1.03e+08	1.31e+08	1.65e+08	2.03e+08
	$\|\sigma\|$	21436.57	28419.02	36545.11	45621.8	55782.97	67106.78	79480.99	92992.88	107668.03	123491.47

Table 4: 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).

LM-OTS. The one-time signature scheme of [LM08] does not require random oracles, and it is asymptotically optimal (almost linear in the lattice dimension) 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]: $\mathsf{H} \in \mathcal{H}_{R,m} = \{\mathsf{H}_a : a \in R^m\}$ that maps elements from R^m to 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 $R = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$. We choose $q = n^3$ and $m = \lceil \log(n) \rceil$, as proposed in the original work [LM08]. 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 $\nu = 20q^{1/m} n \log^2(n)\sqrt{m}$ for SIS. As we operate in ideal lattices, we have $\zeta = (\nu/q^{(nm)/m})^{1/m}$. See Table 5 for the proposed LM-OTS parameters.

Secret Key: $\mathbf{k} \in R^m, \mathbf{l} \in R^m$ with $\|\mathbf{k}\|_\infty \leq 5 \lfloor \log_2(n) \rfloor q^{1/m}, \|\mathbf{l}\|_\infty \leq 5n \lfloor \log_2(n) \rfloor q^{1/m}$, 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: $\mathsf{H} \in \mathcal{H}_{R,m}, \mathsf{H}(\mathbf{k}), \mathsf{H}(\mathbf{l})$, 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: $\sigma \in R^m$ with $\|\sigma\|_\infty \leq 10q^{1/m}n\log^2(n)$, i.e., $m\log_2(10q^{1/m}n\log^2(n))$ bits.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Lenstra	n	512	512	512	1024	1024	1024	1024	1024	1024	1024
	q	1.34e+08	1.34e+08	1.34e+08	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09
	m	7	7	7	7	7	7	7	7	7	7
	$\|\mathsf{sk}\|$	4.11	4.11	4.11	8.71	8.71	8.71	8.71	8.71	8.71	8.71
	$\|\mathsf{pk}\|$	15.19	15.19	15.19	33.75	33.75	33.75	33.75	33.75	33.75	33.75
	$\|\sigma\|$	9.39	9.39	9.39	20.29	20.29	20.29	20.29	20.29	20.29	20.29

Table 5: 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 $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$.

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 an integer $q \approx (2d_s + 1)^m \cdot 2^{-128/n}$.

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

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 $q \geq 4m^2n^{2.5}d_sd_c\log(n)$ and $m > \log(q)/\log(2mnd_sd_c)$ 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. As the determinant in the ideal lattice case is q^{nm}, we calculate $\zeta = (\nu/q^{(nm)/m})^{1/m}$. 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 6.

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

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

Signature: $\sigma \in R^m$ with $\|\sigma\|_\infty \leq nmd_sd_c$, i.e., $m\log_2(nmd_sd_c)$ bits.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Lenstra	n	256	512	512	512	512	512	1024	1024	1024	1024
	q	1.76e+12	7.25e+12	1.32e+13	1.32e+13	1.32e+13	1.32e+13	1.03e+14	1.03e+14	1.03e+14	1.03e+14
	m	8	9	9	9	9	9	10	10	10	10
	κ	48	41	45	50	55	60	52	56	59	60
	σ	18	14	15	15	15	15	13	13	13	13
	$\lvert\mathsf{sk}\rvert$	1.3	2.73	2.79	2.79	2.79	2.79	5.94	5.94	5.94	5.94
	$\lvert\mathsf{pk}\rvert$	1.27	2.67	2.72	2.72	2.72	2.72	5.82	5.82	5.82	5.82
	$\lvert\sigma\rvert$	5.44	12.56	12.69	12.78	12.86	12.93	29.65	29.79	29.88	29.91

Table 6: 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).

4.2 Encryption Schemes

In contrast to lattice signatures that rely on (search) SIS, lattice-based encryption schemes are usually based on the decision LWE problem. After pointing out the relation of these two problems, we have a close look at LWE, its parameters, and properties. Then, we discuss the parameter choices 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]. As an aside, we also deal with chosen ciphertext secure encryption [RS09, Pei09] and possible improvements with ideal lattices [SSTX09] in Appendix B. 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).

In the following, we only show selected parameter sets. The full tables are in Appendix D.

The LWE Assumption. Let $n \in \mathbb{N}$, $m \leq \mathsf{poly}(n)$, $q \leq \mathsf{poly}(n)$, and $\alpha > 0$. Furthermore, let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e} \xleftarrow{\$} \chi_\alpha^m$ with χ_α being a 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.

The assumption that $(\mathbf{A}, \mathbf{v})$ is close to uniform helps in proving CPA security of all subsequent constructions, except in the ideal lattice case. 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].

Attacking LWE. As pointed out by Micciancio and Regev in [MR08], the most natural approach to distinguish $(\mathbf{A}, \mathbf{v})$ from uniform is solving an instance of the SIS problem. An even more compelling reason for this approach is the quantum reduction from SIS to search-LWE in [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 $\mathbf{w} \in 1/q\Lambda_q^\perp(\mathbf{A})$ and checking whether $\langle \mathbf{v}, \mathbf{w} \rangle$ is close to $\mathbb{Z}$ solves the decision problem. 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.

For concreteness, we will replace $\|\mathbf{w}\|$ with $\nu = \zeta^m q^{n/m}$ and require that *no* adversary can solve SIS in $1/q\Lambda_q^\perp(\mathbf{A})$ with ν in dimension $d = \min\{\sqrt{n\log(q)/\log(\zeta)}, m\}$. Recall that now, the adversary has to attack SIS with $\nu = \delta^d q^{n/d}$ in dimension d. Again, we typically have $\delta > \zeta$, i.e., the lattice problem becomes easier when reducing the dimension. Thus, for concrete parameter choices, we require that δ falls below feasible levels according to our hardness estimation in Section 3.

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 that can be bounded with certainty by choosing a q that is sufficiently large. 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 $\text{Prob}[\,|x| \geq ts]$, $t \geq 1$. It states that $e^{-\pi t^2}$ is a very good approximation (see, e.g., [Pei07]). We want the 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 ℓ ($\ell \leq 270$), setting $t = 5$ is sufficient. Moreover, we typically have $ts < 1/c$ for some constant $c \geq 2$. Consequently, we need to ensure that the error is distributed with $s = 1/(5c)$.

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

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

Public Key: $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{P} = \mathbf{A}^t\mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m\times\ell}$ for $\mathbf{E} \leftarrow \chi_\alpha^{m\times\ell}$. 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 $(m-n)\ell\log_2(q)$ bits.

Plaintext: $\mathbf{k} \in \mathbb{Z}_t^\ell$, i.e., $\kappa \leq \ell\log_2(t)$.

Ciphertext: $\mathbf{u} = \mathbf{A}\mathbf{a} \in \mathbb{Z}_q^n$, $\mathbf{c} = \mathbf{P}^t\mathbf{a} + f(\mathbf{k})$, where f encodes $\mathbf{k}$ into $\mathbb{Z}_q^\ell$ and $\mathbf{a} \xleftarrow{\$} \{-r, \ldots, r\}_1^m$, $r \geq 1$. The ciphertext has $\ell\log_2(q)$ bits.

We want t to be a power of two, which makes message encoding easy. A large t reduces the public-key size but introduces decryption errors. We fix $t = 4$ and encrypt $\ell = \kappa/2$ letters.

Having fixed t, we need to set set $\alpha = 1/(40\sqrt{m+1})$ to eliminate decryption errors because then the error is distributed as a Gaussian with parameter $s = 1/40$ and correctable error is $1/8$ per component. Alternatively, this can be verified by applying the error estimation formula in [MR08] to ℓ letters. Another simplification is that we set $r = 1$, which speeds up encryption. We let $q = q(n)$ be the smallest prime between $2n^2$ and $4n^2$ to resolve a circular dependency. Then, we set $m = m(n) = \lceil ((n+\ell)\log_2(q) + 2\kappa)/\log_2(3) \rceil$ to tie the probability of being able to distinguish ciphertexts from uniform to the symmetric security level, i.e., the probability is at most $\sqrt{q^{n+\ell}/(2r+1)^m} = \sqrt{q^{n+\ell}/3^m} < \sqrt{q^{n+\ell}/(q^{n+\ell}2^{2\kappa})} = 2^{-\kappa}$. After taking all this into account, we propose various parameter sets in Table 7. 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	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	κ	150	164	176	190	204	216	230	244	256	270
	(ℓ, t)	$(75, 4)$	$(82, 4)$	$(88, 4)$	$(95, 4)$	$(102, 4)$	$(108, 4)$	$(115, 4)$	$(122, 4)$	$(128, 4)$	$(135, 4)$
Lenstra	n	158	183	211	237	262	290	315	340	368	392
	q	49937	67003	89051	112339	137303	168211	198461	231223	270859	307337
	α	5.02e-04	4.65e-04	4.34e-04	4.08e-04	3.87e-04	3.67e-04	3.51e-04	3.37e-04	3.24e-04	3.12e-04
	m	2484	2888	3324	3755	4178	4632	5065	5502	5971	6402
	$\lvert\mathsf{sk}\rvert$	22.6	29.4	37.3	46.1	55.7	66.4	77.8	90.2	103.8	117.8
	$\lvert\mathsf{pk}\rvert$	332.4	434.1	549.8	684.5	832.2	993.7	1173.5	1369.8	1580	1805.5
	EPF	24.24	25.91	27.93	29.32	30.45	31.99	32.9	33.74	34.97	35.58

Table 7: Recommended parameters for multi-bit LWE. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB) and EPF is the ciphertext expansion factor.

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: $\mathbf{X} \xleftarrow{\$} \mathbb{Z}_2^{m\times\kappa}$, i.e, $m\kappa$ bits.

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

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

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

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. To ensure that the public key is within distance $2^{-\kappa}$ from uniform, we set $m = \lceil n\log_2(q) + 2 \rceil$. Then, the statistical distance is at most

$\sqrt{q^{n\kappa}/2^{m\kappa}} < \sqrt{q^{n\kappa}/(q^{n\kappa}2^{2\kappa})} = 2^{-\kappa}$. As for α, we need to ensure that the induced errors, distributed according to a Gaussian with parameter at most $\alpha\sqrt{m+1}$, are less than $1/8$. Thus, setting $\alpha = 1/(40\sqrt{m+1})$ is sufficient for all relevant n and ℓ. Given these relations among the parameters, we propose secure parameter sets in Table 8.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	κ	150	164	176	190	204	216	230	244	256	270
Lenstra	n	158	183	210	234	259	285	310	334	360	384
	q	49937	67003	88211	109517	134171	162451	192229	223129	259201	294919
	α	5.03e-04	4.61e-04	4.25e-04	3.99e-04	3.76e-04	3.56e-04	3.39e-04	3.24e-04	3.11e-04	2.99e-04
	m	2469	2936	3453	3920	4414	4936	5444	5937	6477	6980
	$\|\mathsf{sk}\|$	2.9	3.7	4.5	5.4	6.4	7.5	8.7	9.9	11.3	12.7
	$\|\mathsf{pk}\|$	45.2	58.7	74.1	90.9	109.9	130.1	152.8	176.8	202.3	230
	EPF	272.51	303.04	338.75	362.13	385.6	412.87	433.01	450.09	472.99	487.9

Table 8: Recommended parameters for dual-LWE. The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB) and EPF is the ciphertext expansion factor.

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.

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 $\tilde{L}$.

Secret Key: $\mathbf{T} \in \mathbb{Z}^{m\times m}$ such that $\mathbf{AT} \equiv 0 \bmod q$. It has at most $m^2 \log_2(L)$ bits.

Public Key: $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n\times\kappa}$. 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.

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

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

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 9 shows the resulting parameter sets for "nearest plane". See Appendix D for "rounding-off".

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	κ	150	164	176	190	204	216	230	244	256	270
Lenstra	n	165	191	218	242	267	293	318	341	368	391
	q	4492157	6967897	10360241	14172493	19034173	25153763	32157451	39651823	49836043	59776477
	α	3.94e-05	3.61e-05	3.34e-05	3.14e-05	2.96e-05	2.81e-05	2.68e-05	2.57e-05	2.46e-05	2.37e-05
	m	19326	23013	26927	30472	34221	38178	42033	45618	49875	53535
	$\|\mathsf{sk}\|$	736509	1060620	1472135	1905490	2427138	3048981	3725735	4418381	5320559	6165845
	$\|\mathsf{pk}\|$	8602	12197	16699	21385	26972	33570	40691	47930	57291	66009
	EPF	46.41	49.21	52.17	54.01	55.83	57.93	59.42	60.52	62.33	63.24

Table 9: Recommended parameters for trapdoor-LWE with "nearest-plane". The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kB and EPF is the ciphertext expansion factor.

Acknowledgments

The authors thank Chris Peikert for his helpful remarks and for pointing out an efficiency improvement for the trapdoor constructions.

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 Seminar 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.

[BR93] Mihir Bellare and Pil Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*. ACM, 1993.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[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.

[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.

[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 *STOC*, pages 207–216. ACM, 2008.

[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.

[Gol04] Oded Goldreich. *The Foundations of Cryptography*, volume 1. Cambridge University Press, 2004.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Ladner and Dwork [LD08], 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.

[LD08] Richard E. Ladner and Cynthia Dwork, editors. *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. ACM, 2008.

[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.

[LN09] Gaëtan Leurent and Phong Q. Nguyen. How risky is the random-oracle model? In Shai Halevi, editor, *CRYPTO*, volume 5677 of *LNCS*, pages 445–464. Springer, 2009.

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

[Lyu08] Vadim Lyubashevsky. *Towards Practical Lattice-Based Cryptography*. PhD thesis, 2008.

[Lyu09] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures, 2009. Asiacrypt 2009, to appear.

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

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[Mic01] Daniele Micciancio. Improving lattice based cryptosystems using the hermite normal form. In Joseph H. Silverman, editor, *CaLC*, volume 2146 of *Lecture Notes in Computer Science*, 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.

[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.

[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 Ladner and Dwork [LD08], pages 187–196.

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

[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, 2010. To appear in PQC 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.

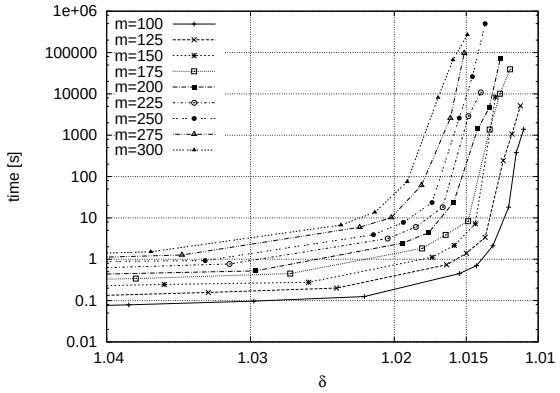
[Sho] Victor Shoup. Number theory library (NTL) for C++. `http://www.shoup.net/ntl/`.

[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.

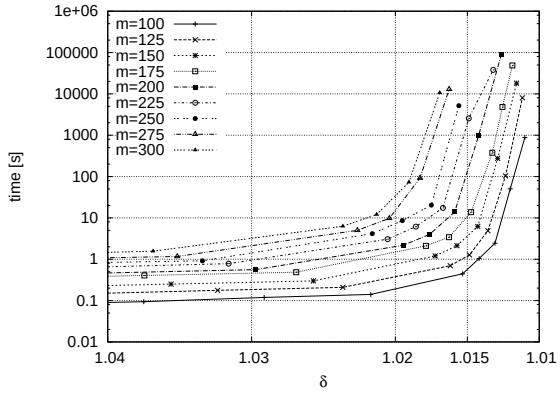
[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, 2009.

A Experimental Data

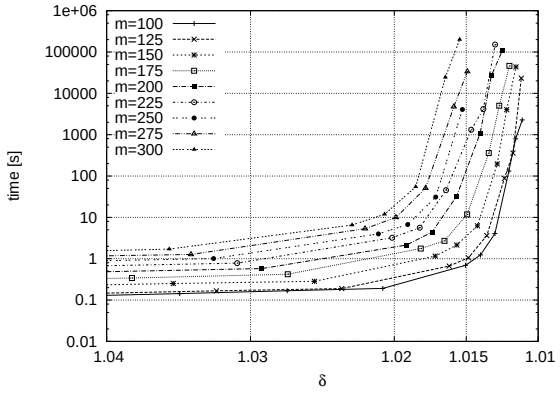
The Figures 2(a) and 2(c) show that the running time of BKZ behaves quite similar for various q and small δ. Here, we compare $q \approx n^2$ with $q \approx n^8$, whereas Figure 2(b) shows the averaged samples for $q \approx n^3$ that were used for the interpolation in Section 3. It appears that the impact of q is negligible, the graphs in the three figures are comparable. The impact of the dimension m is noticeable, but the slope of all graphs seems to be the same. The impact of the Hermite factor δ is compelling. Thus, we can consider δ to be the main security parameter. The fitting curve of Figure 2(d) was used to determine the key sizes in this paper.



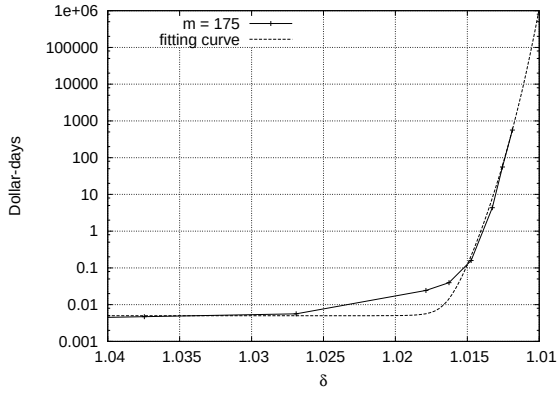
(a) Logarithmic running time in seconds for prime $q \approx n^2$ and selected $100 \leq m \leq 300$ and $1.01 < \delta \leq 1.04$.



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

B Extensions for Encryption Schemes

There are two particularly interesting extensions: using ideal lattices to reduce the key size and making the cryptosystem secure against chosen ciphertext attacks (CCA2).

Ideal Lattices So far, the only work on encryption with ideal lattices is that of Stehlé, Steinfeld, Tanaka, and Xagawa [SSTX09]. They adapt the trapdoor generation algorithm of Alwen and Peikert to work with ideal lattices and propose an ideal version of trapdoor-LWE. The resulting trapdoor $\mathbf{T}$ is a full-rank set of short lattice vectors. Although, it can be converted into a lattice basis via a generic method [MG02, Lemma 7.1], the basis length increases by a factor of $\mathcal{O}(\sqrt{mn})$. Therefore, we recommend using the full-rank set in conjunction with the "nearest plane" approach. The parameter sets remain almost the same as in Table 9. However, they need to assume subexponential hardness of *search* LWE. If this assumption holds, their scheme can encrypt $\lfloor n/\log_2(n) \rfloor$ bits with the Goldreich-Levin hardcore function [Gol04, Section 2.5]. To encrypt $\kappa = 150$ ($\kappa = 270$), this would imply $n \geq 1597$ ($n \geq 3136$). This would result in a large q and big keys, which is exactly what we wanted to avoid by using ideal lattices in the first place. The better approach is using parameters close to the ones in Table 9 and encrypt the symmetric key of length κ in $\lceil n/\lfloor n/\log_2(n) \rfloor \rceil$ chunks. This increases ciphertext expansion by a small factor.

The significant advantage of using ideal lattices is that the bit lengths of public and secret keys are essentially reduced to a $1/n$ fraction. We leave the detailed discussion of schemes based on ideal lattices to a later report.

CCA Security Achieving CCA2 security involves almost the same parameters as in Tables 16 or 9. Let k be the bit length of a verification key in a one-time signature scheme. Then, the ciphertext comprises a "correlated product", i.e, k independent trapdoor functions that are evaluated on the same $\mathbf{s}$. As in trapdoor-LWE, one of the trapdoor evaluations hides the symmetric key $\mathbf{k}$, the others and a one-time signature are merely used as a proof of "well-formedness" to make the scheme CCA2 secure. For the details, refer to [PW08, RS09, Pei09].

For the construction in [Pei09] to work, each user needs two matrix-trapdoor pairs $(\mathbf{A}_1^{(0)}, \mathbf{T}_1^{(0)})$ and $(\mathbf{A}_1^{(1)}, \mathbf{T}_1^{(1)})$. The remaining (public) trapdoor descriptions $\mathbf{A}_2, \ldots, \mathbf{A}_k$ and $\mathbf{U}$ can be shared among all users. Thus, the key sizes are doubled. The ciphertext requires $(kn + \kappa)\log_2(q) + k$ bits, compared to $(n + \kappa)\log_2(q)$ bits in the CPA secure version.

In practice, it is widely accepted to use random oracles and generically transform a CPA secure scheme into a CCA2 secure one [BR93]. The schemes remain largely unchanged, except for the ciphertext. The random oracle is used to append a hash proof of "well-formedness" of the ciphertext. This is very efficient compared to the standard model approach. However, it may be dangerous as all proofs in the random oracle model are only heuristic because random oracles do not exist. Canetti, Goldreich, and Halevi show that proofs in the random oracle model can actually be wrong [CGH04] and Leurent and Nguyen [LN09] demonstrate some practical problems.

C Secure Parameters for Lattice-based Signature Schemes

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Hacker	n	122	178	211	241	271	304	334	364	397	426
	q	2.70e+10	1.00e+09	1.98e+09	3.37e+09	5.39e+09	8.54e+09	1.24e+10	1.76e+10	2.48e+10	3.29e+10
	m_1	4651	5855	7169	8391	9638	11033	12321	13627	15081	16373
	m	22408	28211	34539	40429	46435	53157	59364	65654	72660	78886
	$\lvert\mathsf{sk}\rvert$	9.71e+05	1.57e+06	2.40e+06	3.33e+06	4.45e+06	5.89e+06	7.42e+06	9.15e+06	1.13e+07	1.34e+07
	$\lvert\mathsf{pk}\rvert$	11564.38	18329.99	27475.23	37645.68	49660.56	65080.21	81166.28	99277.22	121595.48	143327.01
	$\lvert\sigma\rvert$	54.45	70.07	87.21	103.38	120.04	138.87	156.41	174.32	194.41	212.37
Lenstra	n	186	216	249	279	310	342	373	402	435	464
	q	1.20e+09	2.18e+09	3.84e+09	6.06e+09	9.24e+09	1.37e+10	1.94e+10	2.61e+10	3.58e+10	4.64e+10
	m_1	6171	7371	8721	9974	11289	12668	14021	15302	16776	18085
	m	29730	35512	42020	48054	54392	61034	67556	73728	80830	87135
	$\lvert\mathsf{sk}\rvert$	1.75e+06	2.54e+06	3.61e+06	4.77e+06	6.18e+06	7.86e+06	9.71e+06	1.17e+07	1.41e+07	1.65e+07
	$\lvert\mathsf{pk}\rvert$	20356.39	29045.2	40666.67	53183.9	68138.69	85796.45	105112.61	125198.02	150479.54	174870.04
	$\lvert\sigma\rvert$	74.16	89.87	107.77	124.56	142.35	161.16	179.76	197.48	218.0	236.32
Int. agency	n	213	243	276	306	336	368	398	428	461	491
	q	2.06e+09	3.49e+09	5.80e+09	8.77e+09	1.27e+10	1.83e+10	2.51e+10	3.36e+10	4.52e+10	5.81e+10
	m_1	7249	8474	9847	11118	12408	13802	15125	16462	17949	19314
	m	34927	40827	47445	53568	59781	66499	72873	79317	86480	93055
	$\lvert\mathsf{sk}\rvert$	2.45e+06	3.40e+06	4.65e+06	5.99e+06	7.53e+06	9.40e+06	1.14e+07	1.36e+07	1.62e+07	1.89e+07
	$\lvert\mathsf{pk}\rvert$	28096.68	38389.53	51845.54	66090.58	82310.34	101848.37	122310.52	144898.68	172251.23	199438.19
	$\lvert\sigma\rvert$	88.27	104.48	122.86	140.03	157.6	176.74	195.02	213.62	234.41	253.6

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Table 10: Recommended parameters for GPV signatures.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Hacker	n	157	189	224	256	288	323	355	387	422	454
	q	9.54e+10	2.41e+11	5.64e+11	1.10e+12	1.98e+12	3.52e+12	5.64e+12	8.68e+12	1.34e+13	1.93e+13
	m_1	6299	7861	9619	11265	12942	14808	16541	18297	20242	22040
	m_2	24051	30015	36726	43009	49412	56539	63157	69862	77287	84153
	m	5441825	7421566	9742009	12311839	15182426	18389983	21868863	25657651	29775737	34188158
	$\lvert\mathsf{sk}\rvert$	9467.4	14502.51	21400.34	29028.26	37957.52	49256.98	61025.02	74194.71	90235.93	106425.08
	$\lvert\mathsf{pk}\rvert$	7.59e+06	1.29e+07	2.07e+07	3.07e+07	4.35e+07	6.03e+07	8.01e+07	1.04e+08	1.34e+08	1.67e+08
	$\lvert\sigma\rvert$	16048.59	22281.62	29709.39	38024.21	47411.64	58021.51	69606.72	82320.89	96264.2	111273.91
Lenstra	n	197	229	264	296	328	363	395	427	462	494
	q	2.97e+11	6.30e+11	1.28e+12	2.27e+12	3.80e+12	6.30e+12	9.62e+12	1.42e+13	2.10e+13	2.94e+13
	m_1	8259	9874	11681	13365	15078	16978	18740	20522	22493	24313
	m_2	31533	37699	44599	51030	57567	64825	71551	78355	85880	92831
	m	7134717	9321527	11830416	14607945	17688147	21085103	24775386	28776807	33086293	37713699
	$\lvert\mathsf{sk}\rvert$	15949.16	22507.89	31136.82	40397.03	51005.06	64184.71	77709.88	92669.11	110702.84	128743.86
	$\lvert\mathsf{pk}\rvert$	1.30e+07	2.04e+07	3.06e+07	4.32e+07	5.91e+07	7.93e+07	1.03e+08	1.31e+08	1.65e+08	2.03e+08
	$\lvert\sigma\rvert$	21436.57	28419.02	36545.11	45621.8	55782.97	67106.78	79480.99	92992.88	107668.03	123491.47
Int. agency	n	225	257	292	324	356	390	422	454	489	522
	q	5.77e+11	1.12e+12	2.12e+12	3.57e+12	5.72e+12	9.02e+12	1.34e+13	1.93e+13	2.80e+13	3.88e+13
	m_1	9670	11316	13153	14862	16596	18463	20242	22040	24028	25920
	m_2	36921	43207	50221	56745	63365	70495	77287	84153	91741	98964
	m	8353816	10683445	13321718	16243932	19469651	22929338	26761544	30906191	35344313	40205304
	$\lvert\mathsf{sk}\rvert$	21619.72	29286.71	39167.13	49604.35	61415.37	75502.61	90235.93	106425.08	125827.14	145764.2
	$\lvert\mathsf{pk}\rvert$	1.79e+07	2.68e+07	3.88e+07	5.35e+07	7.16e+07	9.38e+07	1.20e+08	1.51e+08	1.88e+08	2.31e+08
	$\lvert\sigma\rvert$	25367.95	32867.15	41472.33	51079.84	61778.34	73363.75	86267.89	100317.27	115484.2	132165.77

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Table 11: Recommended parameters for Bonsai signature scheme.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Hacker	n	512	512	512	512	1024	1024	1024	1024	1024	1024
	q	1.34e+08	1.34e+08	1.34e+08	1.34e+08	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09
	m	7	7	7	7	7	7	7	7	7	7
	$\lvert\mathsf{sk}\rvert$	4.11	4.11	4.11	4.11	8.71	8.71	8.71	8.71	8.71	8.71
	$\lvert\mathsf{pk}\rvert$	15.19	15.19	15.19	15.19	33.75	33.75	33.75	33.75	33.75	33.75
	$\lvert\sigma\rvert$	9.39	9.39	9.39	9.39	20.29	20.29	20.29	20.29	20.29	20.29
Lenstra	n	512	512	512	1024	1024	1024	1024	1024	1024	1024
	q	1.34e+08	1.34e+08	1.34e+08	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09
	m	7	7	7	7	7	7	7	7	7	7
	$\lvert\mathsf{sk}\rvert$	4.11	4.11	4.11	8.71	8.71	8.71	8.71	8.71	8.71	8.71
	$\lvert\mathsf{pk}\rvert$	15.19	15.19	15.19	33.75	33.75	33.75	33.75	33.75	33.75	33.75
	$\lvert\sigma\rvert$	9.39	9.39	9.39	20.29	20.29	20.29	20.29	20.29	20.29	20.29
Int. agency	n	512	512	1024	1024	1024	1024	1024	1024	1024	2048
	q	1.34e+08	1.34e+08	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	1.07e+09	8.59e+09
	m	7	7	7	7	7	7	7	7	7	8
	$\lvert\mathsf{sk}\rvert$	4.11	4.11	8.71	8.71	8.71	8.71	8.71	8.71	8.71	19.83
	$\lvert\mathsf{pk}\rvert$	15.19	15.19	33.75	33.75	33.75	33.75	33.75	33.75	33.75	82.5
	$\lvert\sigma\rvert$	9.39	9.39	20.29	20.29	20.29	20.29	20.29	20.29	20.29	48.62

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Table 12: Recommended parameters for LM-OTS signature scheme.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	λ	225	246	264	285	306	324	345	366	384	405
Hacker	n	256	256	512	512	512	512	512	512	1024	1024
	q	1.76e+12	1.76e+12	1.32e+13	1.32e+13	1.32e+13	1.32e+13	1.32e+13	1.32e+13	1.03e+14	1.03e+14
	m	8	8	9	9	9	9	9	9	10	10
	κ	48	56	45	50	55	60	60	60	59	60
	σ	18	18	15	15	15	15	15	15	13	13
	$\lvert\mathsf{sk}\rvert$	1.3	1.3	2.79	2.79	2.79	2.79	2.79	2.79	5.94	5.94
	$\lvert\mathsf{pk}\rvert$	1.27	1.27	2.72	2.72	2.72	2.72	2.72	2.72	5.82	5.82
	$\lvert\sigma\rvert$	5.44	5.49	12.69	12.78	12.86	12.93	12.93	12.93	29.88	29.91
Lenstra	n	256	512	512	512	512	512	1024	1024	1024	1024
	q	1.76e+12	7.25e+12	1.32e+13	1.32e+13	1.32e+13	1.32e+13	1.03e+14	1.03e+14	1.03e+14	1.03e+14
	m	8	9	9	9	9	9	10	10	10	10
	κ	48	41	45	50	55	60	52	56	59	60
	σ	18	14	15	15	15	15	13	13	13	13
	$\lvert\mathsf{sk}\rvert$	1.3	2.73	2.79	2.79	2.79	2.79	5.94	5.94	5.94	5.94
	$\lvert\mathsf{pk}\rvert$	1.27	2.67	2.72	2.72	2.72	2.72	5.82	5.82	5.82	5.82
	$\lvert\sigma\rvert$	5.44	12.56	12.69	12.78	12.86	12.93	29.65	29.79	29.88	29.91
Int. agency	n	512	512	512	512	512	512	1024	1024	1024	1024
	q	7.25e+12	7.25e+12	1.32e+13	1.32e+13	1.32e+13	1.32e+13	1.03e+14	1.03e+14	1.03e+14	1.03e+14
	m	9	9	9	9	9	9	10	10	10	10
	κ	37	41	45	50	55	60	52	56	59	60
	σ	14	14	15	15	15	15	13	13	13	13
	$\lvert\mathsf{sk}\rvert$	2.73	2.73	2.79	2.79	2.79	2.79	5.94	5.94	5.94	5.94
	$\lvert\mathsf{pk}\rvert$	2.67	2.67	2.72	2.72	2.72	2.72	5.82	5.82	5.82	5.82
	$\lvert\sigma\rvert$	12.48	12.56	12.69	12.78	12.86	12.93	29.65	29.79	29.88	29.91

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB).

Table 13: Recommended parameters for treeless signatures.

D Secure Parameters for Lattice-based Encryption Schemes

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	κ	150	164	176	190	204	216	230	244	256	270
	(ℓ, t)	$(75, 4)$	$(82, 4)$	$(88, 4)$	$(95, 4)$	$(102, 4)$	$(108, 4)$	$(115, 4)$	$(122, 4)$	$(128, 4)$	$(135, 4)$
Hacker	n	123	149	177	203	229	256	281	306	334	359
	q	30259	44417	62659	82421	104891	131101	157931	187273	223129	257783
	α	5.52e-04	5.04e-04	4.65e-04	4.34e-04	4.09e-04	3.87e-04	3.68e-04	3.52e-04	3.37e-04	3.24e-04
	m	2049	2458	2887	3311	3741	4177	4605	5038	5503	5944
	$\lvert\mathsf{sk}\rvert$	16.8	23	30.3	38.4	47.6	57.4	68.1	79.8	92.7	106.3
	$\lvert\mathsf{pk}\rvert$	262.5	356.8	463.9	588.6	729.3	878.8	1048.2	1234.3	1435	1654.5
	EPF	19.65	21.75	23.99	25.61	27.06	28.65	29.73	30.72	32.06	32.89
Lenstra	n	158	183	211	237	262	290	315	340	368	392
	q	49937	67003	89051	112339	137303	168211	198461	231223	270859	307337
	α	5.02e-04	4.65e-04	4.34e-04	4.08e-04	3.87e-04	3.67e-04	3.51e-04	3.37e-04	3.24e-04	3.12e-04
	m	2484	2888	3324	3755	4178	4632	5065	5502	5971	6402
	$\lvert\mathsf{sk}\rvert$	22.6	29.4	37.3	46.1	55.7	66.4	77.8	90.2	103.8	117.8
	$\lvert\mathsf{pk}\rvert$	332.4	434.1	549.8	684.5	832.2	993.7	1173.5	1369.8	1580	1805.5
	EPF	24.24	25.91	27.93	29.32	30.45	31.99	32.9	33.74	34.97	35.58
Int. agency	n	181	207	235	260	286	313	338	363	390	416
	q	65537	85703	110459	135209	163601	195967	228509	263561	304211	346117
	α	4.75e-04	4.42e-04	4.14e-04	3.92e-04	3.73e-04	3.56e-04	3.41e-04	3.28e-04	3.16e-04	3.05e-04
	m	2774	3195	3637	4058	4498	4943	5379	5819	6276	6738
	$\lvert\mathsf{sk}\rvert$	26.5	34	42.3	51.4	61.7	72.5	84.5	97.4	111	126.1
	$\lvert\mathsf{pk}\rvert$	379.8	490.1	612.2	750.7	908.3	1073.1	1259.8	1463.2	1675.2	1917.1
	EPF	27.31	28.88	30.75	31.85	32.94	34.27	35.06	35.79	36.86	37.55

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB) and EPF is the ciphertext expansion factor.

Table 14: Recommended parameters for multi-bit LWE.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	κ	150	164	176	190	204	216	230	244	256	270
Hacker	n	126	151	178	203	228	254	278	303	329	353
	q	31769	45613	63377	82421	103969	129037	154571	183637	216493	249229
	α	5.75e-04	5.17e-04	4.69e-04	4.34e-04	4.05e-04	3.81e-04	3.61e-04	3.43e-04	3.27e-04	3.14e-04
	m	1887	2340	2842	3318	3802	4315	4795	5301	5834	6331
	$\lvert\mathsf{sk}\rvert$	2.3	3	3.8	4.7	5.7	6.7	7.8	9	10.3	11.6
	$\lvert\mathsf{pk}\rvert$	34.5	46.8	61	76.9	94.6	113.7	134.5	157.8	182.2	208.6
	EPF	203.09	236.31	273.53	301.52	327.27	356.13	376.61	397.39	421.64	438.28
Lenstra	n	158	183	210	234	259	285	310	334	360	384
	q	49937	67003	88211	109517	134171	162451	192229	223129	259201	294919
	α	5.03e-04	4.61e-04	4.25e-04	3.99e-04	3.76e-04	3.56e-04	3.39e-04	3.24e-04	3.11e-04	2.99e-04
	m	2469	2936	3453	3920	4414	4936	5444	5937	6477	6980
	$\lvert\mathsf{sk}\rvert$	2.9	3.7	4.5	5.4	6.4	7.5	8.7	9.9	11.3	12.7
	$\lvert\mathsf{pk}\rvert$	45.2	58.7	74.1	90.9	109.9	130.1	152.8	176.8	202.3	230
	EPF	272.51	303.04	338.75	362.13	385.6	412.87	433.01	450.09	472.99	487.9
Int. agency	n	180	205	231	256	280	307	331	355	381	406
	q	64811	84053	106727	131101	156817	188519	219133	252079	290327	329677
	α	4.66e-04	4.31e-04	4.02e-04	3.79e-04	3.59e-04	3.41e-04	3.26e-04	3.13e-04	3.01e-04	2.90e-04
	m	2880	3356	3861	4355	4835	5382	5875	6372	6917	7445
	$\lvert\mathsf{sk}\rvert$	3.3	4.1	5	5.9	7	8.1	9.3	10.6	11.9	13.4
	$\lvert\mathsf{pk}\rvert$	52.7	67.1	82.9	100.9	120.3	141.9	164.9	189.7	216.1	245.3
	EPF	322.88	351.12	383.14	406.67	426.31	454.17	470.92	486.53	508.48	523.78

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB) and EPF is the ciphertext expansion factor.

Table 15: Recommended parameters for dual-LWE.

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	κ	150	164	176	190	204	216	230	244	256	270
Hacker	n	165	198	234	267	300	335	367	399	434	466
	q	7.41e+08	1.54e+09	3.00e+09	5.08e+09	8.10e+09	1.26e+10	1.81e+10	2.53e+10	3.55e+10	4.72e+10
	α	5.14e-07	4.14e-07	3.39e-07	2.90e-07	2.53e-07	2.22e-07	2.00e-07	1.81e-07	1.64e-07	1.51e-07
	m	25768	32026	39044	45628	52337	59572	66287	73087	80615	87572
	$\|\mathsf{sk}\|$	1342990	2113786	3194896	4420390	5882053	7701655	9618429	11784915	14449838	17163278
	$\|\mathsf{pk}\|$	15293	23622	35110	47950	63087	81737	101201	123029	149678	176628
	EPF	61.88	67.36	73.34	77.55	81.32	85.59	88.46	91.08	94.46	96.65
Lenstra	n	208	240	276	308	341	376	408	440	475	507
	q	1.87e+09	3.32e+09	5.80e+09	9.00e+09	1.35e+10	2.00e+10	2.77e+10	3.75e+10	5.09e+10	6.61e+10
	α	3.90e-07	3.29e-07	2.79e-07	2.45e-07	2.18e-07	1.94e-07	1.77e-07	1.62e-07	1.48e-07	1.37e-07
	m	33957	40231	47445	53980	60824	68191	75014	81913	89541	96584
	$\|\mathsf{sk}\|$	2388260	3400645	4794946	6273024	8042334	10202109	12440335	14937797	17975165	21038503
	$\|\mathsf{pk}\|$	26557	37278	51846	67110	85208	107099	129602	154539	184661	214853
	EPF	73.51	77.91	83.3	86.67	89.91	93.78	96.23	98.47	101.56	103.44
Int. agency	n	236	269	305	337	369	404	436	468	503	535
	q	3.10e+09	5.24e+09	8.65e+09	1.29e+10	1.85e+10	2.66e+10	3.61e+10	4.80e+10	6.40e+10	8.19e+10
	α	3.36e-07	2.88e-07	2.48e-07	2.21e-07	1.99e-07	1.79e-07	1.63e-07	1.51e-07	1.38e-07	1.29e-07
	m	39440	46031	53363	59990	66710	74157	81047	88009	95701	102798
	$\|\mathsf{sk}\|$	3262791	4502102	6124674	7814536	9746558	12146578	14611315	17341794	20640753	23948764
	$\|\mathsf{pk}\|$	35825	48801	65585	82886	102496	126658	151288	178396	210941	243388
	EPF	81.14	85.24	90.22	93.16	95.81	99.41	101.56	103.54	106.43	108.09

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB) and EPF is the ciphertext expansion factor.

Table 16: Recommended parameters for trapdoor-LWE with "rounding-off".

	year	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
	κ	150	164	176	190	204	216	230	244	256	270
Hacker	n	133	158	186	211	236	262	286	311	337	361
	q	2352641	3944329	6434861	9393949	13144259	17984749	23393659	30080233	38272757	47045897
	α	4.49e-05	4.05e-05	3.67e-05	3.41e-05	3.19e-05	3.00e-05	2.85e-05	2.71e-05	2.59e-05	2.48e-05
	m	14921	18350	22297	25905	29580	33466	37107	40949	44992	48767
	$\|\mathsf{sk}\|$	428881	660921	992883	1357931	1790990	2316825	2873417	3528329	4293021	5077380
	$\|\mathsf{pk}\|$	5127	7755	11450	15455	20152	25795	31713	38619	46623	54774
	EPF	39.93	43.02	46.52	48.89	51.01	53.33	54.92	56.51	58.35	59.57
Lenstra	n	165	191	218	242	267	293	318	341	368	391
	q	4492157	6967897	10360241	14172493	19034173	25153763	32157451	39651823	49836043	59776477
	α	3.94e-05	3.61e-05	3.34e-05	3.14e-05	2.96e-05	2.81e-05	2.68e-05	2.57e-05	2.46e-05	2.37e-05
	m	19326	23013	26927	30472	34221	38178	42033	45618	49875	53535
	$\|\mathsf{sk}\|$	736509	1060620	1472135	1905490	2427138	3048981	3725735	4418381	5320559	6165845
	$\|\mathsf{pk}\|$	8602	12197	16699	21385	26972	33570	40691	47930	57291	66009
	EPF	46.41	49.21	52.17	54.01	55.83	57.93	59.42	60.52	62.33	63.24
Int. agency	n	188	213	239	264	288	314	339	362	388	413
	q	6644681	9663629	13651943	18399749	23887933	30959167	38958229	47437961	58411097	70445017
	α	3.65e-05	3.39e-05	3.16e-05	2.98e-05	2.84e-05	2.70e-05	2.58e-05	2.48e-05	2.38e-05	2.30e-05
	m	22583	26196	30025	33768	37413	41412	45305	48924	53056	57066
	$\|\mathsf{sk}\|$	1019662	1389964	1847655	2360630	2923025	3611972	4355466	5111489	6051544	7042658
	$\|\mathsf{pk}\|$	11746	15805	20763	26262	32238	39499	47274	55128	64832	75003
	EPF	51.07	53.34	55.89	57.67	59.11	61.06	62.38	63.33	64.9	65.95

The rows correspond to attacker types and the columns correspond to security until a given year. Sizes are in kilobytes (kB) and EPF is the ciphertext expansion factor.

Table 17: Recommended parameters for trapdoor-LWE with "nearest-plane".