Towards Efficient Lattice-Based Cryptography

Vom Fachbereich Informatik der
Technischen Universität Darmstadt genehmigte

Dissertation

zur Erlangung des Grades
Doctor rerum naturalium (Dr. rer. nat.)
von

Dipl.-Math. Richard Lindner

geboren in Dresden.

Referenten:	Prof. Dr. Johannes Buchmann Prof. Dr. Christopher Peikert
Tag der Einreichung:	8. November 2010
Tag der mündlichen Prüfung:	20. Dezember 2010
Hochschulkennziffer:	D 17

Darmstadt 2011

Wissenschaftlicher Werdegang

September 2006 – heute

Wissenschaftlicher Mitarbeiter und Promotionsstudent in der Arbeitsgruppe von Prof. Johannes Buchmann, Fachgebiet Informatik, an der Technischen Universität Darmstadt

Oktober 2001 – July 2006

Studium von „Mathematics with Computer Science“ an der Technischen Universität Darmstadt

September 2003 – Juni 2004

Auslandsstudium in England an der University of Birmingham

List of Publications

[BDL08] J. Buchmann, M. Döring, and R. Lindner. Efficiency improvement for NTRU. In A. Alkassar and J. H. Siekmann, editors, *Sicherheit 2008: Sicherheit, Schutz und Zuverlässigkeit. Konferenzband der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 2.-4. April 2008 im Saarbrücker Schloss*, volume 128 of *LNI*, pages 163–178. GI, 2008. Cited on page 23.

[BL09a] J. Buchmann and R. Lindner. Density of ideal lattices. In *WEWoRC, to appear*. Springer, 2009. Cited on page 11.

[BL09b] J. Buchmann and R. Lindner. Secure parameters for SWIFFT. In B. Roy and S. Nicolas, editors, *Progress in Cryptology - INDOCRYPT 2009, 10th International Conference on Cryptology in India, New Delhi, India, December 13-16, 2009. Proceedings*, volume 5922 of *LNCS*, pages 1–17. Springer, 2009. Cited on page 17.

[BLR08] J. Buchmann, R. Lindner, and M. Rückert. Explicit hard instances of the shortest vector problem. In J. Buchmann and J. Ding, editors, *Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings*, volume 5299 of *LNCS*, pages 79–94. Springer, 2008. Cited on page 8.

[BLRS09] J. Buchmann, R. Lindner, M. Rückert, and M. Schneider. Post-quantum cryptography: Lattice signatures. *Computing*, 85:105–125, 2009. Cited on page 5.

[BLS09] J. Buchmann, R. Lindner, and M. Schneider. Probabilistic analysis of LLL reduced bases. In *WEWoRC, to appear*. Springer, 2009.

[CLRS10a] P.-L. Cayrel, R. Lindner, M. Rückert, and R. Silva. Improved zero-knowledge identification with lattices. In S.-H. Heng and K. Kurosawa, editors, *Provable Security, 4th International Conference, ProvSec 2010, Malacca, Malaysia, October 2010. Proceedings*, volume 6402 of *LNCS*, pages 1–17. Springer, 2010. Cited on page 31.

[CLRS10b] P.-L. Cayrel, R. Lindner, M. Rückert, and R. Silva. A lattice-based threshold ring signature scheme. In M. Abdalla and P. S. L. M. Barreto, editors, *LATINCRYPT*, volume 6212 of *Lecture Notes in Computer Science*, pages 255–272. Springer, 2010. Cited on page 39.

[DL07] J. Ding and R. Lindner. Identifying ideal lattices. Technical Report 2007/322, Cryptology ePrint Archive, 2007. http://eprint.iacr.org/. Cited on page 12.

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA 2011, to appear*. Springer, 2011. Cited on page 45.

Acknowledgments

I would like to thank many people for their help, support, and encouragement on my long journey of writing this thesis. First and foremost, I thank Johannes Buchmann for his ideas and supervision, and for giving me the opportunity to travel the world to present my results. I am also extraordinary thankful to Chris Peikert for wanting to work with me, which was so much fun and productive, and for agreeing to be my co-referee.

For interesting and beneficial collaborations, I thank Paulo Barreto, Pierre-Louis Cayrel, Jintai Ding, Martin Döring, Rafael Misoczki, Markus Rückert, Michael Schneider, and Rosemberg André da Silva.

For supporting my research with remarks, questions, and answers, I thank all of my colleagues, as well as Indranil Banerjee, Marc Fischlin, Vadim Lyubashevsky, Oded Regev, and Alon Rosen. For advice in the fascinating field of number theory, I thank Jan Hendrik Bruinier, Michael Pohst, and Jonathan Sands. For advice in the field of lattice reduction that is essential for my work, I thank Bartol Filipović, Henrik Koy, and Christoph Ludwig. Furthermore, I thank Paul Baecher, Erik Dahmen, Roswitha Jäger-Beck, Lucie Langer, Anja Lehmann, Axel Schmidt, Michael Schneider, and Marita Skrobić for creating a pleasant office atmosphere.

Next, I thank my parents Bernd and Eva Lindner for their mixture of unfaltering encouragement and nagging questions, which instilled in me the drive necessary to finish this undertaking. I thank all of my friends both old and new for the net of support they wove for me. My special thanks go to Pierre-Louis Cayrel for our restaurant tours and the pool holiday, Ben Cohen for being my editor in chief, Rafaël Dahmen for the ever fruitful tomato wheat sessions, Frank Karinda for showing that it can be done, Anja Lehmann for engaging discussions in the park, and Michael Schneider for being funny sometimes.

Richard Lindner
Darmstadt, 2010

Abstract

One essential quest in cryptography is the search for hard instances of a given computational problem that is known to be hard in the worst-case. In lattice cryptography we are in the unique situation that we have found a way of picking random instances which are at least as hard as well-studied lattice problems in the worst-case. At the same time, no attack running in subexponential time is known to break these problems, even for an adversary using quantum computers. Virtually all public-key schemes in use today are subject to such attacks, and the development of quantum computers is actively pursued, so it is prudent to investigate lattice-based alternatives.

There are two fundamental open problems in lattice cryptography today and this thesis contributes to solving them. First, there exists a widely used efficiency improvement that allows for trapdoors whose asymptotic keysize and evaluation time are both quasilinear in the dimension of an associated lattice. This is accomplished by restricting oneself to lattices with special structure, so-called ideal lattices. This entails the use of newer security assumptions, but these have not been analyzed thoroughly so far. We start this work by comparing the class of ideal lattice problems with its general counterpart in terms of size. Affirming folklore, we find the number of restricted instances among all instances to be asymptotically negligible for those classes of lattices suggested for practical use.

The second open problem is that while the connection to worst-case problems is well understood, the practical hardness of the related average-case problems is not. Specifically, there have been parameters suggested for practical usage, where current lattice basis reduction algorithms can solve these worst-case problems, but the related average-case problems, which these are reduced to and which represent the basis of practical security of the cryptosystems, are completely infeasible. In most cases, this lack of understanding has lead either to a very conservative choice of parameters, or none at all. This in turn makes it impossible for the resulting lattice schemes to compete with their counterparts based on other paradigms. We further the understanding of this practical security and at the same time improve the efficiency of several common related cryptographic schemes. Among other things, we find that for the SWIFFT compression function, solving certain problems closely related to finding collisions is easier than previously thought and we suggest efficient replacement parameters. We propose a novel zero-knowledge identification scheme that, to our knowledge, beats all competing post-quantum schemes, even those based on other paradigms. Possibly most important, we help to tighten the efficiency gap between lattice encryption schemes that are provably secure and the acclaimed ad-hoc encryption scheme NTRU. This is done by unifying many recent developments into a new provably secure design and providing a comprehensive analysis of practical security, which together results in a great leap of efficiency.

Zusammenfassung

In der Kryptographie ist eine der wesentlichen Aufgaben die Suche nach schweren Instanzen eines Berechnungsproblems, von dem man weiß, dass es im Worst-Case schwer ist. In der Gitterkryptographie sind wir in der einzigartigen Situation, einen Weg gefunden zu haben wie man zufällige Instanzen auswählt, die mindestens genauso schwer wie wohluntersuchte Gitterprobleme im Worst-Case sind. Gleichzeitig ist kein Angriff bekannt, der diese Probleme in subexponentieller Zeit löst, selbst wenn der Angreifer einen Quantencomputer verwendet. Praktisch alle asymmetrischen Kryptographieverfahren, die heute im Einsatz sind, unterliegen solchen Angriffen, und die Entwicklung von Quantencomputern, mit denen man die Angriffe praktisch realisieren kann, wird aktiv verfolgt. Es ist daher langfristig gesehen ratsam, gitterbasierte Alternativen zu untersuchen.

Es gibt aktuell zwei grundlegende offene Probleme in der Gitterkryptographie, und diese Arbeit trägt zu ihrer Lösung bei. Das erste Problem betrifft eine weit verbreitete Effizienzsteigerung. Diese ermöglicht die Konstruktion von Trapdoor-Einwegfunktionen, deren asymptotische Schlüssellänge und Laufzeit beide quasilinear in der Dimension eines assoziierten Gitters sind. Dies wird durch die Einschränkung des Verfahrens auf Gitter mit spezieller Struktur erreicht, so genannte Idealgitter. Dies ist jedoch nur möglich, wenn man von neuen Sicherheitsannahmen ausgeht, deren gründliche Untersuchung noch aussteht. Wir beginnen diese Untersuchung, indem wir die Klasse von Berechnungsproblemen in Idealgittern mit ihrem allgemeineren Gegenstück in Bezug auf deren Größe vergleichen. Wir finden in den praxisrelevanten Gitterkategorien, in Übereinstimmung mit allgemeinen Vermutungen, heraus, dass die Anzahl der beschränkten Instanzen unter allen Instanzen asymptotisch vernachlässigbar ist.

Das zweite offene Problem betrifft die Berechnungsprobleme im Average-Case, die als direkte Sicherheitsgrundlage vieler praktischer Verfahren dienen. Es ist bekannt, dass diese Probleme asymptotisch mindestens so schwer wie verwandte Worst-Case-Probleme sind, aber darüber hinaus ist wenig über ihre Robustheit gegenüber praktischen Angriffen bekannt. Insbesondere gibt es praxisrelevante Parameter, für die alle bekannten Algorithmen die entsprechenden Average-Case-Probleme nicht lösen können, die zugehörigen Worst-Case-Probleme aber teilweise schon. In den meisten Fällen hat diese Wissenslücke dazu geführt, dass beim Vorstellen neuer Gitterverfahren die Parameter entweder sehr konservativ ausgewählt oder überhaupt nicht angegeben wurden. Dies wiederum macht es unmöglich, die entsprechenden Verfahren mit Konkurrenten zu vergleichen, die auf anderen Paradigmen basieren. Wir erweitern das Verständnis der praktischen Sicherheit mehrerer verbreiteter kryptographischer Verfahren und verbessern gleichzeitig deren Effizienz. Unter anderem finden wir heraus, dass für die Kompressionsfunktion SWIFFT die Lösung bestimmter sicherheitsrelevanter Probleme, die eng mit der Suche von Kollisionen zusammenhängen, leichter ist als bisher angenommen und empfehlen hinreichend effiziente Ersatzparameter mit denen das Problem den ursprünglichen Annahmen gerecht wird. Wir schlagen ein neues Zero-Knowledge-Identifikationsverfahren vor, dass unseres Wissens alle konkurrierenden Post-Quantum-Verfahren übertrifft, insbesondere auch solche Verfahren, die auf anderen Paradigmen basieren. Als vielleicht bedeutendstes Ergebnis tragen wir dazu dabei, die Effizienzlücke zwischen beweisbar sicheren Gitterverschlüsselungsverfahren und dem vielgepriesenen Ad-hoc-Verfahren NTRU zu verkleinern. Dies erreichen wir, indem wir viele der jüngeren Entwicklungen zu einem neuen beweisbar sicheren Design zusammenfassen und eine umfassende Analyse zu dessen praktischer Sicherheit durchführen, was zusammen einen bedeutenden Effizienzsprung bei gleichbleibender Sicherheit in der Praxis zur Folge hat.

Contents

1 Introduction

The usage of public-key cryptography has become commonplace in our lives. The security of virtually all such cryptographic schemes in use today rests on either of two problems, namely integer factorization and discrete logarithms. Due to an algorithm by Shor, the computational difficulty of both these problems can be overcome by large enough quantum computers [Sho94]. Physicists have successfully been working towards the goal of building such a quantum computer. Looking ahead, we will at some point be required to diversify our computational assumptions.

Several mathematical fields offer computational problems suitable for cryptography. The prominent ones today are *multivariate polynomial systems*, *coding theory*, and *lattice theory*. We will concern ourselves only with the latter, because it carries the additional and unique benefit that in order to *guarantee* security of associated cryptographic primitives, one only requires the computational hardness of lattice problems in the *worst-case*. Primitives based on computational problems from other fields always require the hardness of *average-case* problems following some distribution usually chosen heuristically by the designers and attackers of the primitive.

For the case of lattice-based cryptography, Ajtai proposed a method of randomly choosing instances of computational problems suitable for cryptography, which are secure assuming only the hardness of associated worst-case problems [Ajt96a]. These instances are now known as SIS problems and their unusual connection to *worst-case* problems has led to their usage for many basic cryptographic primitives such as hash functions, signature schemes, identification schemes, and more. Later, Regev presented a similar connection for a related method of choosing random instances, but on the dual lattices. These instances became known as LWE problems [Reg05a]. These are suitable for constructing more advanced primitives such as public-key encryption, hierarchical identity-based encryption, and many more.

There are two fundamental problems in lattice-based cryptography today. First, there exists a widely used efficiency improvement which entails the use of newer security assumptions. However, analyzing these throughly is still an open problem. The efficiency gain is based on the fact that some additive groups can be extended to *rings*. Restricting the aforementioned SIS and LWE problems to such groups, i.e., those which are themselves ideals, greatly improves both efficiency and key-sizes for the resulting schemes [LM06, LPR10]. It was also shown that a connection to appropriately restricted *worst-case* problems still exists, however, these restricted variants are new and much less studied, so for the moment their benefits come at a cost of faith.

The second problem is that while the connection to *worst-case* problems is well understood, the *practical hardness* of the related *average-case* SIS and LWE problems is not. There is a wide range of parameters for which current lattice basis reduction algorithms can solve the *worst-case* problems we discussed, but the related *average-case* problems, which these are reduced to and which represent the actual basis of security of the cryptosystems, are completely infeasible. In most cases, this lack of understanding has lead either to a very conservative choice of parameters, or none at all. This in turn makes it impossible for the resulting lattice schemes to compete with their counterparts based on other paradigms.

This thesis contributes to solving these problems. After explaining the basic notions of lattice-based cryptography, we present the following four results.

Density of Ideal Lattices. We start in Chapter 3 with the first thorough analysis of the only known drawback when restricting SIS and LWE to ideals, which is common practice for current schemes. In our analysis, we compare the related classes of worst-case problems in terms of size as the parameters defining both classes tend to infinity. The size of the restricted variants is always found to be negligibly small in this comparison. However, we also find that no practical algorithm is known to take noticeable advantage of this gap.

Several papers relate to our work, since they make use of worst-case problems in ideal lattices as basis of security for new cryptographic schemes. A comprehensive survey of these schemes is given in Lyubashevsky's PhD thesis [Lyu08b]. Though many authors suggest that the problems in ideal lattices may be easier, none of them give a quantitative comparison of worst-case problems in general lattices and ideal ones as we do.

Mathematically, our results are a new interpretation of the work by Murty and Van Order [MVO07], who analyzed the same objects we deal with but for purely mathematical reasons, namely giving explicit bounds for the Riemann zeta function at $s = 1$. We, on the other hand, provide a cryptographic context where their results give new insights.

Lattice-Based Compression Functions. In Chapter 4, we introduce the lattice-based compression function SWIFFT proposed by Lyubashevsky, Micciancio, Peikert, and Rosen [LMPR08]. It is the major internal part of the hash function SWIFFTX, which in turn was a long running candidate for the hash design competition SHA-3. We propose an efficiency improvement for SWIFFT, which is universally applicable to *all* schemes based on SIS and comes at no cost. We also propose a parameter generator for SWIFFT and present an analysis which shows that finding ℓ_2-pseudo-collisions for SWIFFT is not as hard as previously thought. We finish by proposing appropriately efficient replacement parameters for which pseudo-collisions are hard to find.

Numerous recent results relate to this. For example, in [MR09], Micciancio and Regev suggest an improvement for schemes based on the LWE problem. It comes with the same benefits that our proposition has for SIS-based schemes, i.e., in both cases the problems require less description bits. However, they do not provide a proof that for all relevant parameters the improved and original problem are asymptotically equivalent. They also do not apply their improvement to the case of ideal lattices, since no reduction from worst-case problems for LWE in ideals was known at the time.

There is also an eminent study on the practical hardness of several lattice problems by Gama and Nguyen [GN08] related to our work. Like us, they have performed extensive experiments in order to predict the behavior of lattice basis reduction algorithms. They do not, however, cover the SWIFFT lattices in their analysis. Furthermore, they only provide relative hardness statements in terms of the Hermite factor of a lattice problem, and make no link to, say, an absolute runtime required to achieve such a factor on a given computer.

Lattice-Based Zero-Knowledge Identification. Having thoroughly analyzed SWIFFT, we continue in Chapter 5 by using it in conjunction with a recent zero-knowledge identification scheme from coding theory by Cayrel and Véron [CV10] to construct a lattice counterpart. Due to the efficiency inherent in SWIFFT, our streamlined adaptation has smaller communication costs than its coding partner, while keeping the same soundness error of 1/2 and perfect completeness. At the time of writing, we are not aware of any similar post-quantum primitive that has less communication cost. The adaptation is also of independent interest, because it can be used as a roadmap for translating simple schemes based on the syndrome decoding problem into lattice ones.

There exist many propositions for post-quantum, zero-knowledge identification schemes. In lattice cryptography, one of the most prominent ones has been proposed by Kawachi, Tanaka, and Xagawa [KTX08]. Like our scheme, it is an adaptation of a code-based primitive, in their case one by Stern [Ste93]. Starting from Stern's scheme, which requires more communication than Cayrel and Véron's, they naturally end up with a proposal that is less efficient than ours. Also, their focus is on showing the theoretical possibility of deriving a scheme based on worst-case hardness assumptions, rather than on analyzing the practical costs of their scheme in comparison to other proposals.

Another well-known example of lattice-based identification is a scheme by Lyubashevsky [Lyu09]. His scheme is different in the sense that its proofs of authenticity are only witness-indistinguishable and not zero-knowledge, which is somewhat weaker. Lyubashevsky's scheme has no soundness error, but a small completeness error of $1 - 1/e$, creating the undesirable possibility that an honest prover is rejected. Lyubashevsky's proposal is well tailored towards the Fiat-Shamir heuristic for transforming an identification scheme into a signature. So, it does have a more efficient signature counterpart than ours. As an identification scheme, however, it is slightly less efficient.

Lattice-Based Encryption. Finally, in Chapter 6, we wrap things up by analyzing the more sophisticated LWE problem, as proposed by Regev [Reg05a], in the same way we analyzed SIS in the context of SWIFFT. The LWE problem is much more versatile in terms of schemes that are built upon it, so for clarity we focus on basic public-key encryption. We unify several recent developments concerning LWE into a new variant of the encryption scheme which features public keys that are 10 times smaller than previous propositions. At the same time, we propose a new two-phase hybrid attack on LWE which combines lattice basis reduction and an extended enumeration variant of Babai's nearest plane algorithm to achieve better results.

Several papers contain studies of the concrete hardness of lattice problems. For instance, the one we mentioned before by Gama and Nguyen [GN08]. They performed a comprehensive study of the behavior of basis reduction algorithms for various families of lattices. However, aside from the Goldstein-Mayer distribution for very large moduli (which are not typically used in cryptographic constructions), their work did not attempt to document the behavior of basis reduction on q-ary lattices, the class of lattices related to SIS and LWE. In addition, they were not concerned with the *use* of a reduced basis to solve bounded-distance decoding problems (of which LWE is a special case), where additional algorithmic ideas and trade-offs are possible.

Another related work is a recent survey by Micciancio and Regev [MR09], who proposed parameters for various lattice-based schemes from the contemporary literature. Their parameters were derived using Gama and Nguyen's general estimates for the best obtainable Hermite factor, and as such do not include concrete security estimates (e.g., of symmetric bit security), nor do they incorporate specific information about q-ary lattices or post-reduction decoding. Their parameters also apply to a less efficient LWE-based encryption scheme that has larger keys than the one we describe and analyze.

2 Lattices in Cryptography

In this chapter we will introduce some basic mathematical definitions and properties related to lattices. Our explanations aim to be sound and sufficient to make the rest of this work accessible. For a complete and thorough introduction we recommend the surveys [Ngu09, MR09, BLRS09, MG02].

2.1 Basics

Before we to start with lattices, let us first make some remarks about notation. We will use boldface letters for matrices and vectors and normal other typefaces for their entries.

Geometrically, a lattice is a set of regularly recurring points in real vector space. This geometric perspective is especially useful for visualizing computational problems in lattices. On the other hand, algebraically, lattices are free $\mathbb{Z}$-modules in real vector space. This second perspective is also useful, since most algorithms and results are not tied to the real vector space, but apply to free $\mathbb{Z}$-modules in general. We use the following formal definition.

Definition 2.1.1 (Lattice)**.** Let $\mathcal{L}$ be a discrete, additive subgroup of $\mathbb{R}^m$, then it is a *lattice* and the maximum number of linearly independent vectors in $\mathcal{L}$ is the lattice *dimension* or *rank*.

So, except for $\mathcal{L} = \{\mathbf{0}\}$, lattices are (countably) infinite point sets and as such hard to work with. Fortunately, lattices can be described compactly.

Theorem 2.1.2. *Let $\mathcal{L}$ be a subset of $\mathbb{R}^m$, then these are equivalent:*

- *$\mathcal{L}$ is an n-dimensional lattice.*
- *There exist linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, such that*

$$\mathcal{L} = \mathbf{b}_1\mathbb{Z} + \cdots + \mathbf{b}_n\mathbb{Z}.$$

For the interested reader, the proof is given in [Ngu09]. This theorem motivates the following definition.

Definition 2.1.3 (Lattice basis)**.** A matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ with linearly independent columnvectors is a *basis* of the lattice $\mathcal{L}(\mathbf{B}) = \mathbf{b}_1\mathbb{Z} + \cdots + \mathbf{b}_n\mathbb{Z}$.

We use the indefinite article on purpose here, since lattice bases are not necessarily unique. In fact, the 0-dimensional lattice has exactly one basis, any 1-dimensional lattice has exactly 2, and n-dimensional lattices with $n > 1$ have a (countably) infinite number of bases. More precisely, given any particular basis $\mathbf{B}$ of an n-dimensional lattice $\mathcal{L}$, *all* bases of $\mathcal{L}$ are in the orbit of $\mathbf{B}$ under the general linear group $\mathrm{GL}(n, \mathbb{Z})$, i.e., the group of $n \times n$ matrices that are invertible over the integers.

So, for any two bases $\mathbf{B}, \mathbf{B}'$ of the same lattice $\mathcal{L}$ there exists a unique transformation matrix $\mathbf{T}$, which is invertible over the integers, such that $\mathbf{BT} = \mathbf{B}'$, and conversely, multiplying $\mathbf{B}$ with any transformation matrix $\mathbf{T}$, which is invertible over the integers, will yield another basis of the same lattice. This process is called *basis transformation.*

Definition 2.1.4 (Determinant)**.** Given a basis $\mathbf{B}$ of the lattice $\mathcal{L}$, we define the lattice *determinant* to be

$$\det(\mathcal{L}) = \sqrt{\det(\mathbf{B}^T\mathbf{B})}.$$

Note that this definition is sound, since the lattice determinant, as defined here, remains invariant under basis transformations.

In cryptography, we will often work with integral lattices of full rank, i.e., m-dimensional sublattices of $\mathbb{Z}^m$. For these, there is one particular basis of special interest. It can be computed efficiently from any given lattice basis using an integer variant of Gaussian elimination.

Definition 2.1.5 (HNF)**.** An integer matrix $\mathbf{B} \in \mathbb{Z}^{m \times m}$ is in *Hermite normal form* if

(i) $\mathbf{B}$ is upper triangular,

(ii) $b_{i,i} > 0$ for all $1 \leq i \leq m$, and

(iii) $0 \leq b_{i,j} < b_{i,i}$, for all $1 \leq i < j \leq m$.

Each integral lattice of full rank has exactly one basis in HNF. Using this particular basis to describe the lattice is often convenient in practice. Since an HNF basis is upper triangular, it will potentially require fewer description bits than other bases.

Every lattice has an associated dual lattice defined as follows.

Definition 2.1.6 (Dual)**.** Let $\mathcal{L}$ be a lattice in $\mathbb{R}^m$, then its *dual lattice* is

$$\mathcal{L}^* = \left\{\mathbf{v} \in \mathbb{R}^m \mid \forall \mathbf{w} \in \mathcal{L} : \mathbf{v}^T\mathbf{w} \in \mathbb{Z}\right\}.$$

Note that the dual lattice of an integral lattice is not necessarily integral, for example $(2\mathbb{Z}^n)^* = (1/2)\mathbb{Z}^n$. If a lattice with basis $\mathbf{B}$ has full rank, then $(\mathbf{B}^{-1})^T$ is a basis of its dual lattice.

2.2 Computational Problems

Having learned the basic properties of lattices, we will continue by listing the most relevant computationally hard lattice problems. For a more comprehensive list, we refer to [MG02]. Computational problems for lattices are closely intertwined with the notion of “length” we choose on the real vectorspace.

Definition 2.2.1 (ℓ_p-Norm)**.** Let $p \in \mathbb{N} \cup \{\infty\}$, then the *$\ell_p$-norm* is given by

$$\|\cdot\|_p \colon \mathbb{R}^m \longrightarrow \mathbb{R} : (v_1, \ldots, v_m)^T \longmapsto \begin{cases} (|v_1|^p + \cdots + |v_m|^p)^{1/p} & \text{for } p < \infty, \\ \max\{|v_1|, \ldots, |v_m|\} & \text{otherwise.} \end{cases}$$

Current research is limited to the named cases, i.e., $p = 1$ (Manhatten norm), $p = 2$ (Euclidean norm), or $p = \infty$ (Max norm). Having chosen an appropriate notion of length, we can define another lattice invariante, namely the successive minima.

Definition 2.2.2 (Successive Minima)**.** Let $p \in \mathbb{N} \cup \{\infty\}$ and $\mathcal{L}$ be an n-dimensional lattice. For $i = 1, \ldots, n$, the i-th *successive minimum* with respect to this norm is

$$\lambda_i^p = \min\left\{r > 0 \mid \exists \text{ linearly independent } \mathbf{b}_1, \ldots, \mathbf{b}_i \in \mathcal{L} : \|\mathbf{b}_j\|_p \le r \text{ for } j = 1, \ldots, i\ \right\}.$$

So, the i-th successive minimum is the smallest radius of a ball containing i linearly independent lattice vectors. Specifically, $\lambda_1^p(\mathcal{L})$ is the length of the shortest nonzero lattice vector. The superscript p is usually omitted when the Euclidean norm ($p = 2$) is used.

We begin with the two oldest and most famous lattice problems.

Definition 2.2.3 (SVP)**.** Given a lattice $\mathcal{L}$ and a real $\gamma > 0$, the *shortest vector problem* SVP_γ^p is to find a nonzero lattice vector $\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}$ such that $\|\mathbf{v}\|_p \le \gamma\lambda_1(\mathcal{L})$.

Definition 2.2.4 (CVP)**.** Given a lattice $\mathcal{L}$, a target vector $\mathbf{t} \in \mathbb{R}^m$, and a real $\gamma > 0$, the *closest vector problem* CVP_γ^p is to find a lattice vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{t} - \mathbf{v}\|_p \le \gamma\lambda_1(\mathcal{L})$.

By convention, if $\gamma = 1$ then the subscript is omitted and if $p = 2$ the superscript is omitted. So for example, solving SVP means finding a shortest nonzero lattice vector in the Euclidean norm.

As stated, these problems are known as search problems. Their solver is required to present a mathematical object, in this case a vector, as opposed to a YES/NO decision. We continue by stating related *promise problems* that have the latter form.

Definition 2.2.5 (GapSVP)**.** Given a tuple $(\mathbf{B}, d)$, where $\mathbf{B} \in \mathbb{Q}^{m \times n}$ is a basis and $d \in \mathbb{Q}$ is some number, the *gap shortest vector problem* GapSVP_γ^p is to answer

$$\text{YES if } \lambda_1^p(\mathcal{L}(\mathbf{B})) \le d, \qquad \text{and} \qquad \text{NO if } \lambda_1^p(\mathcal{L}(\mathbf{B})) > \gamma d.$$

Definition 2.2.6 (GapCVP)**.** Given a tuple $(\mathbf{B}, \mathbf{t}, d)$, where $\mathbf{B} \in \mathbb{Q}^{m \times n}$ is a basis, $\mathbf{t} \in \mathbb{Q}^m$ a target, and $d \in \mathbb{Q}$ is some number, the *gap closest vector problem* GapCVP_γ^p is to answer

$$\text{YES if } \|\mathbf{t} - \mathcal{L}\|_p \le d, \qquad \text{and} \qquad \text{NO if } \|\mathbf{t} - \mathcal{L}\|_p > \gamma d,$$

where we use the shorthand $\|\mathbf{t} - \mathcal{L}\|_p = \min\{\|\mathbf{t} - \mathbf{v}\|_p \mid \mathbf{v} \in \mathcal{L}\}$.

Note that in either case we can set d to some fixed value, say $d = 1$, and scale $\mathbf{B}$ and $\mathbf{t}$ accordingly. Also, if a solver for GapSVP_γ^p is run on a tuple $(\mathbf{B}, d)$ where $d < \lambda_1^p(\mathcal{L}(\mathbf{B})) \le \gamma d$, i.e., the promise inherent in the question is broken, then the solver's behavior is impossible to predict (it may not ever terminate). The same holds for GapCVP.

Since these problems ask for a decision, they fall into the realm of classic complexity classes, such as $\mathsf{P}, \mathsf{NP}, \mathsf{AM}, \ldots$, adapted to the setting of promise problems. We state two main results in this area and refer to a recent survey by Regev for more details [Reg09].

Theorem 2.2.7 ([MG02]). *For $\gamma < \sqrt[p]{2}$, GapSVP^p_γ is NP-hard under randomized reductions.*

Theorem 2.2.8 ([AR04]). *There exists $c > 0$ such that $\mathsf{GapSVP}_{c\sqrt{n}}$ is in $\mathsf{NP} \cap \mathsf{coNP}$.*

So for small approximation factors $\gamma < \sqrt{2}$, the problem is NP-hard and for bigger factors, starting around $\gamma = \sqrt{n}$, the problem is unlikely to be NP-hard. Similar results hold for GapCVP.

2.3 Hard Instances

Having established that SVP_γ is a hard problem, at least for small approximation factors γ, we come to the very heart of lattice-based cryptography, namely the seminal results by Ajtai [Ajt96a] and Regev [Reg05a] relating GapSVP and the average-case problems SIS and LWE. We will see that hard instances of SVP and CVP may be found in the following class of lattices.

Definition 2.3.1 (q-ary Lattices). A lattice $\mathcal{L}$ satisfying $q\mathbb{Z}^m \subseteq \mathcal{L} \subseteq \mathbb{Z}^m$ is *q-ary.*

Given positive integers n, q, m and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we can define two q-ary lattices:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{x} = \mathbf{A}^t\mathbf{y} \bmod q \text{ for some } \mathbf{y} \in \mathbb{Z}^n\ \},$$
$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}.$$

Note that for a fixed matrix $\mathbf{A}$, one is the scaled dual of the other, i.e., $q(\Lambda_q(\mathbf{A}))^* = \Lambda_q^\perp(\mathbf{A})$. At the same time either form exists for all q-ary lattices.

Lemma 2.3.2 ([Mic10]). *For any q-ary lattice $\mathcal{L}$, there exist matrices $\mathbf{A}, \mathbf{A}'$ such that*

$$\mathcal{L} = \Lambda_q(\mathbf{A}) = \Lambda_q^\perp(\mathbf{A}').$$

These lattices play a special role in lattice-based cryptography, because they appear implicitly in the average-case problems SIS and LWE.

Definition 2.3.3 (SIS). In the *small integer solution* problem $\mathsf{SIS}(n, q, m, \beta)$, we are given a matrix $\mathbf{A} \sim U(\mathbb{Z}_q^{n\times m})$ chosen uniformly at random and a real $\beta > 0$. The task is to find a nonzero integer vector $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $\mathbf{A}\mathbf{x} \equiv \mathbf{0} \pmod q$ and $\|\mathbf{x}\| \leq \beta$.

For very small β the problem is insolvable, since there need not be a solution vector that is small enough. However, if $\beta \geq \sqrt{m}\, q^{n/m}$ we know that a valid solution vector exists by the pigeonhole principle.

Note that solving SIS with a random matrix $\mathbf{A}$ amounts to solving SVP in the q-ary lattice $\Lambda_q^\perp(\mathbf{A})$. In [BLR08], the connection of SIS and concrete lattice problems is further explored and the progress of practical algorithms solving SIS is measured in the from of a public contest, the lattice challenge[1], which is open for all enthusiasts.

For the connection to worst-case problems, instead of restating Ajtai's original result, we give a recent result by Micciancio and Regev that offers the tightest connection known today.

[1] See `http://www.latticechallenge.org`.

Theorem 2.3.4 ([MR04]). *For any $m = \Theta(n \log(n))$, there exist $q = O(n^{2.5} \log(n))$, and $\gamma = O(n\sqrt{\log(n)})$, such that solving $\mathsf{SIS}(n, q, m, \beta)$ with $\beta = \sqrt{m}\, q^{n/m}$ is at least as hard as solving GapSVP_γ for all lattices of dimension n.*

So, for large enough n with q, m, β chosen as suggested in the theorem, we expect SIS to be a computationally hard problem.

Definition 2.3.5 (LWE). Let n, q, m be positive integers and χ be some distribution on $\mathbb{Z}_q$. In the *learning with errors* problem $\mathsf{LWE}(n, q, m, \chi)$, we are given a matrix $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$ chosen uniformly at random and a vector $\mathbf{t} \in \mathbb{Z}_q^m$, such that either
(i) $\mathbf{t} \sim U(\mathbb{Z}_q^m)$, or
(ii) $\mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e}$ for some $s \sim U(\mathbb{Z}_q^n)$, and $e \sim \chi^m$.
In the *decision* variant, the task is to distinguish between these distributions for $\mathbf{t}$. In the *search* variant, we are always in the second case and the goal is to recover $\mathbf{s}$ or $\mathbf{e}$.

This problem is sometimes stated without the parameter m. In this case we are given an *arbitrary* number of coordinates of $\mathbf{t}$.

Similar to SIS, solving search-LWE with a random matrix $\mathbf{A}$ amounts to solving CVP in the q-ary lattice $\Lambda_q(\mathbf{A})$.

For the best worst-case connection, we will need to specify a special distribution taking the place of χ. For any positive real α, we let $\bar{\Psi}_\alpha$ denote the distribution on $\mathbb{Z}_q$ obtained by sampling a random variable with mean 0 and standard deviation $\alpha q/\sqrt{2\pi}$, rounding the result to the nearest integer and reducing it modulo q. We are usually interested in the case where $\chi = \bar{\Psi}_\alpha$. For this we will write $\mathsf{LWE}(n, q, m, \alpha)$.

Theorem 2.3.6 ([Reg05a]). *Assume there is an oracle which solves the decision variant of $\mathsf{LWE}(n, q, m, \alpha)$, where $\alpha q > \sqrt{n}$, furthermore $q \leq \operatorname{poly}(n)$ is prime, and $m \leq \operatorname{poly}(n)$. Then there exists a quantum algorithm running in time $\operatorname{poly}(n)$ for solving $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ in all lattices of dimension n.*

The worst-case to average-case reductions for SIS and LWE also hold within a subclass of lattices, namely lattices corresponding to ideals in the ring $\mathbb{Z}[\zeta]$, where ζ is some algebraic integer that is fixed for the reduction [LM06, LPR10]. These reductions use different methodologies and are more recent than the general ones we have described up to now. All involved problems usually get the prefix "Ideal" or just 'I' to clarify the restriction to such lattices.

The additional ring structure of ideal lattices allows all cryptographic schemes which use them to sport smaller keys and, for a slightly more restricted class, be much more efficient as well. In each case, the change in keysizes and trapdoor evaluation time is from $\tilde{O}(n^2)$ for general lattices to $\tilde{O}(n)$ for ideal lattices. For details on such constructions, we refer to Lyubashevsky's PhD thesis [Lyu08b].

3 Density of Ideal Lattices

In this chapter, we will show that for a fixed maximal order $\mathcal{O}_K$, the set of ideal lattices with respect to ideals in $\mathcal{O}_K$ that have bounded determinant, due to its inherent structure, forms a comparatively small subset of all lattices under the same bound.

The security of many *efficient* cryptographic constructions, e.g., collision-resistant hash functions, digital signatures, identification schemes, and more recently public-key encryption, has been proven assuming the hardness of *worst-case* computational problems in ideal lattices. These lattices correspond to ideals in the ring $\mathbb{Z}[\zeta]$, where ζ is some fixed algebraic integer.

Under the assumption that this ring $\mathbb{Z}[\zeta]$ is the maximal order of the number field $\mathbb{Q}(\zeta)$, we show that the density of n-dimensional ideal lattices with determinant $\leq b$ among all lattices under the same bound is in $O(b^{1-n})$ as b grows. So, for lattices of dimension > 1 with bounded determinant, the subclass of ideal lattices is always vanishingly small. Our assumption, though not valid for all algebraic integers ζ, is certainly valid for all ζ that have been suggested for practical use.

Though interesting from a theoretical point of view, we advise the reader that there is no known algorithm solving standard computational lattice problems, such as SVP or CVP, that can use this special structure to noticeably decrease the time required to solve the problem.

This chapter is based on [BL09a]. It was presented both at the *Algorithms and Number Theory* Seminar 2009 in Dagstuhl, Germany and the *Western European Workshop on Research in Cryptology* (WEWoRC) 2009 in Graz, Austria.

3.1 Introduction

Following the seminal result of Ajtai from 1996, which gives a worst-case to average-case reduction for computational problems in lattices [Ajt96a], the security of many lattice-based cryptographic schemes was proven assuming the hardness of these worst-case problems, e.g., [GGH96, LM08, GPV08, Pei09].

Using similar methods, Lyubashevsky and Micciancio found in 2006 that the same worst-case to average-case reduction holds for a different class of lattices, namely lattices corresponding to ideals in the ring $\mathbb{Z}[\zeta]$, where ζ is some algebraic integer that is fixed for the reduction. The additional structure of these lattices allows the cryptographic schemes which use them to be much more efficient and require smaller keys. In each case, the change for keysizes and trapdoor evaluation time is from $\tilde{O}(n^2)$ for general lattices to $\tilde{O}(n)$ for ideal lattices. Again, many cryptographic schemes were proven secure assuming the hardness of worst-case problems in ideal lattices, see [LM06, LM08, LMPR08, SSTX09].

Until today, there has been no in-depth analysis of the hardness relationship of these two worst-case problems which have become the basis of security for so many schemes, although it is folklore that problems in ideal lattices are easier and it is known that the property of being an ideal can be efficiently recognized in any lattice basis [DL07]. We give a first solid indication that worst-case computational problems in ideal lattices are potentially much simpler. We show that the number of n-dimensional lattices with bounded determinant $\leq b$ is $\Omega(b^n)$ as b goes to infinity. The number of ideal lattices under the same constraints is only $O(b)$, a vanishingly small quantity in comparison.

3.2 Algebraic Number Theory

We start with a special subring of the complex numbers. The ring of algebraic integers is

$$\mathcal{A} = \{\zeta \in \mathbb{C} \mid \exists f \in \mathbb{Z}[x] : f \text{ monic and } f(\zeta) = 0\}.$$

Throughout this chapter, let $\zeta \in \mathcal{A}$ be an algebraic integer, and $K = \mathbb{Q}(\zeta)$ be a number field of *degree* $\deg(K) = [K : \mathbb{Q}] = n$.

Definition 3.2.1. An *order* $\mathcal{O}$ is a subring of $\mathcal{A}$ that is a free $\mathbb{Z}$-module.

The integral combinations of powers of ζ form an order $\mathbb{Z}[\zeta] = [1, \zeta, \dots, \zeta^{n-1}]\mathbb{Z}^n$. Another order, the *ring of integers* in K, is

$$\mathcal{O}_K = \mathcal{A} \cap K.$$

This order is maximal in the sense that it contains *all* other orders in K. Note that the rank of an order in K, as a $\mathbb{Z}$-module, cannot exceed the extension degree of K. So, there exist $\beta_1, \dots, \beta_n \in \mathcal{O}_K$ such that $\mathcal{O}_K = [\beta_1, \dots, \beta_n]\mathbb{Z}^n$.

We can embed K into $\mathbb{Q}^n$ via the *coefficients*

$$\sigma \colon K \longrightarrow \mathbb{Q}^n : a_0 + a_1\zeta + \cdots + a_{n-1}\zeta^{n-1} \longmapsto (a_0, a_1, \dots, a_{n-1})^T = \mathbf{a}.$$

Definition 3.2.2. Let $\mathcal{O}$ be an order in K. An *$\mathcal{O}$-ideal lattice* is a lattice $L \subseteq \mathbb{Z}^n$ such that $L = \sigma(\mathfrak{i})$ for some ideal $\mathfrak{i} \subseteq \mathcal{O}$.

In the special case $\mathcal{O} = \mathbb{Z}[\zeta]$, this matches the definition of Lyubashevsky and Micciancio in [LM06].

We will often use the embedding σ implicitly and write, for example, $\det(\mathfrak{i})$ instead of $\det(\sigma(\mathfrak{i}))$. The *norm* of an ideal $\mathfrak{i}$ in $\mathcal{O}$ is $N(\mathfrak{i}) = |\mathcal{O} \,/\, \mathfrak{i}|$. This is related to the determinant of the corresponding ideal lattice

$$N(\mathfrak{i}) = \det(\mathfrak{i}) \cdot \det(\mathcal{O}). \tag{3.2.1}$$

For the case $\mathcal{O} = \mathcal{O}_K$, this is the *field norm.*

Conforming with notations in related works, we will use greek letters for elements of K and fraktur for (fractional) ideals of $\mathcal{O}_K$.

3.3 Counting Lattices

General lattices. For integers $n, b > 0$, let all n-dimensional full-rank integral lattices with determinant $\leq b$ be

$$L_n(b) = \{L \subseteq \mathbb{Z}^n : \dim(L) = n, \det(L) \leq b\}$$

and let their number be $l_n(b) = |L_n(b)|$.

In 1968 Schmidt showed in [Sch68] that as b goes to infinity, we have $l_n(b) \in O(b^n)$. We will use a similar methodology to derive a *lower* bound.

Theorem 3.3.1. *For integers $n, b > 0$, we have $l_n(b) \geq b^n/n$.*

Proof. Let $L'_n(d) = \{L \subseteq \mathbb{Z}^n : \det(L) = d\}, l'_n(d) = |L'_n(d)|$. We start by showing

$$l'_n(1) = l'_1(d) = 1, \tag{3.3.1}$$

$$l'_n(d) = \sum_{c|d} c^{n-1}\, l'_{n-1}(d/c). \tag{3.3.2}$$

It suffices to count the number of possible lattice bases in HNF, because this form is unique for each lattice. Equations 3.3.1 are an immediate consequence.

Now, let $L \in L'_n(d)$, $\mathbf{B} = \mathrm{HNF}(L)$, and $c = b_{n,n}$. Consider the last row of $\mathbf{B}$. We know $c \mid d$ and $0 \leq b_{n,i} < c$ for $i = 1, \ldots, n-1$. These are $\sum_{c|d} c^{n-1}$ possible rows. The remaining upper left $(n-1) \times (n-1)$ submatrix of $\mathbf{B}$ could be the HNF of *any* lattice in $L'_{n-1}(d/c)$, which shows Equation 3.3.2.

We can now prove the claim

$$l_n(b) = \sum_{d=1}^{b} l'_n(d) = \sum_{d=1}^{b} \sum_{c|d} c^{n-1} l'_{n-1}(d/c) \geq \sum_{d=1}^{b} d^{n-1} \geq \int_0^b d^{n-1}\,\mathrm{d}d \geq b^n/n.$$

□

Note that, during the proof, we counted lattices whose HNF differs from the identity matrix only in the last row and we found there are at least $\Omega(b^n)$ many of those. Since Schmidt showed in [Sch68] that $O(b^n)$ is also an upper bound on the number of n-dimensional lattices with determinant $\leq b$, it follows that lattices with these special bases are a *dense* subset of all lattices. This was shown less elementarily by Goldstein and Mayer [DG03].

Ideal lattices. For integers $n, b > 0$, let $\mathcal{O}$ be an order of rank n in K. Furthermore, let the set of all $\mathcal{O}$-*ideal lattices* with determinant $\leq b$ be

$$I_n^{\mathcal{O}}(b) = \{L \subseteq \mathbb{Z}^n : L \text{ is } \mathcal{O}\text{-ideal lattice}, \dim(L) = n, \det(L) \leq b\}$$

and let their number be $i_n^{\mathcal{O}}(b) = |I_n^{\mathcal{O}}(b)|$. We adapt an old result of Dedekind and Weber, which was recently made more precise by Murty and Van Order [MVO07].

Theorem 3.3.2. *Let K be a number field of degree n, then for integers $b > 0$ we have*

$$i_n^{\mathcal{O}_K}(b) \leq h_K(2c_K b^{1/n} + 1)^n / (w \det(\mathcal{O}_K)),$$

where h_K is the number of ideal classes, w is the number of roots of unity in K, and c_K is another real constant depending only on K.

Proof. Let $\mathcal{C}$ be some ideal class in $\mathcal{O}_K$,

$$I_n^{\mathcal{C}}(b) = \{\mathfrak{a} \in \mathcal{C} : 0 < N(\mathfrak{a}) \leq b\}, \qquad i_n^{\mathcal{C}}(b) = |I_n^{\mathcal{C}}(b)|.$$

We start by showing for any ideal $\mathfrak{b} \in \mathcal{C}^{-1}$, $i_n^{\mathcal{C}}(b) = |\mathfrak{b} I_n^{\mathcal{C}}(b)|$. Obviously, $\geq$ holds and we also have $|\mathfrak{b} I_n^{\mathcal{C}}(b)| \geq |(\mathfrak{b}^{-1})\mathfrak{b} I_n^{\mathcal{C}}(b)| = |I_n^{\mathcal{C}}(b)|$, which gives us $\leq$. Note that

$$\mathfrak{b} I_n^{\mathcal{C}}(b) = \{\langle\alpha\rangle \subseteq \mathfrak{b} : 0 < N(\alpha) \leq bN(\mathfrak{b})\},$$

so in order to count ideals in $\mathcal{C}$ it suffices to count principal ideals in $\mathfrak{b}$.

The span of two elements is equal if and only if they differ by a ring unit: $\langle\alpha\rangle = \langle\alpha'\rangle \iff$ there exists a unit $\epsilon \in \mathcal{O}_K$ such that $\alpha' = \epsilon\alpha$.

Let (r_1, r_2) be the signature of K and $r = r_1 + r_2 - 1$. Dirichlet proved the following classification (see, e.g., [ME05, p. 99]): There exist fundamental units $\epsilon_1, \ldots, \epsilon_r \in \mathcal{O}_K$ such that ϵ is a unit in $\mathcal{O}_K$ if and only if $\epsilon = \zeta\epsilon_1^{n_1} \cdots \epsilon_r^{n_r}$, where $\zeta \in K$ is a root of unity, and $n_1, \ldots, n_r \in \mathbb{Z}$. Recall that the total number of roots of unity in K is w.

We continue by showing that for each principal ideal $\langle\alpha\rangle \in \mathfrak{b} I_n^{\mathcal{C}}(b)$ there exist w many reals $0 \leq c_1, \ldots, c_r < 1$ such that

$$\sum_{j=1}^{r} c_j \log|\epsilon_j^{(i)}| = \log(|\alpha^{(i)}| N(\alpha)^{-1/n}) \qquad \text{for } 1 \leq i \leq n. \tag{3.3.3}$$

Note that the $r \times r$ matrix $(\log|\epsilon_j^{(i)}|)_{1 \leq i,j \leq r}$ is non-singular, so for each $\alpha \in \mathfrak{b}$ there exist (unrestricted) reals $c_1, \ldots, c_r$ such that Equation 3.3.3 holds for $1 \leq i \leq r$. Let $\alpha' = \epsilon\alpha$ for some unit ϵ, then we have

$$\log\left(|\alpha'^{(i)}| N(\alpha')^{-1/n}\right) = \sum_{j=1}^{r} n_j \log|\epsilon_j^{(i)}| + \log\left(|\alpha^{(i)}| N(\alpha)^{-1/n}\right) = \sum_{j=1}^{r} (n_j + c_j) \log|\epsilon_j^{(i)}|.$$

So, by Dirichlet's classification, restricting the reals to $0 \leq c_1, \ldots, c_r < 1$ leaves only w many for each principal ideal. For the rest, fix any of the w many.

For $r + 1 < i \leq n$, we have $|(\cdot)^{(i)}| = |\overline{(\cdot)}^{(i-r_2)}| = |(\cdot)^{(i-r_2)}|$, so Equation 3.3.3 holds for these.

Since $N(\alpha) = \prod_{i=1}^n |\alpha^{(i)}|$ and $1 = N(\epsilon_j) = \prod_{i=1}^n |\epsilon_j^{(i)}|$ for $1 \leq j \leq r$, we get

$$\sum_{i=1}^n \sum_{j=1}^r c_j \log |\epsilon_j^{(i)}| = \sum_{j=1}^r c_j \left(\sum_{i=1}^n \log |\epsilon_j^{(i)}| \right) = 0 = \sum_{i=1}^n \log \left(|\alpha^{(i)}| N(\alpha)^{-1/n} \right).$$

We already knew that the summands of the left- and rightmost sum are equal for $i \neq r+1$, so this equality gives us the final case $i = r+1$ for Equation 3.3.3.

Finally, we prove the theorem. Let h_K be the number of ideal classes,

$$i_n^{\mathcal{O}_K}(b) \leq h_K \max_{\mathcal{C}} \{ i_n^{\mathcal{C}}(b) \} / \det(\mathcal{O}_K).$$

Let $\beta_1, \dots, \beta_n$ be an integral basis of $\mathcal{O}_K$. For each principal ideal in $\mathfrak{b} I_n^{\mathcal{C}}(b)$, there are w many α subject to Equation 3.3.3. For each of these α, there exist unique integers $x_1, \dots, x_n$ such that $\alpha = x_1\beta_1 + \cdots x_n\beta_n$. We will show that the total number of these integers and thus $i_n^{\mathcal{C}}(b)$ is bounded.

The $\beta_1, \dots, \beta_n$ form a basis, so the matrix $\mathbf{B} = (\beta_j^{(i)})_{1 \leq i,j \leq n}$ is invertible and

$$\|(x_i)_{1 \leq i \leq n}\|_\infty \leq \|\mathbf{B}^{-1}\|_\infty \, \|(\alpha^{(i)})_{1 \leq i \leq n}\|_\infty.$$

Let $m_\epsilon = \max\{\log |\epsilon_j^{(i)}| : 1 \leq i, j \leq r\}$. By Equation 3.3.3 we know

$$\|(\alpha^{(i)})_{1 \leq i \leq n}\|_\infty \leq \exp(r m_\epsilon) |N(\alpha)^{1/n}| \leq \exp(r m_\epsilon)(b N(\mathfrak{b}))^{1/n}.$$

Minkowski showed that an ideal $\mathfrak{b}$ in class $\mathcal{C}^{-1}$ can always be chosen such that

$$N(\mathfrak{b}) \leq (4/\pi)^{r_2} n! \sqrt{|d_K|}/n^n,$$

where d_K is the discriminant of K (see, e.g., [Ash10, Ch. 5]). Altogether, we have

$$\|(x_i)_{1 \leq i \leq n}\|_\infty \leq \underbrace{(4/\pi)^{r_2/n} \|\mathbf{B}^{-1}\|_\infty \exp(r m_\epsilon)(n! \sqrt{|d_K|}/n^n)^{1/n}}_{=c_K} \cdot b^{1/n}.$$

Since all possible $x_1, \dots, x_n$ are bounded in this way, the total number of α subject to Equation 3.3.3 is $(2c_K + 1)^n$. As we know there exist at most w many of these α for every principal ideal in $\mathfrak{b} I_n^{\mathcal{C}}(b)$, we get

$$i_n^{\mathcal{C}}(b) \leq (2c_K b^{1/n} + 1)^n / w,$$

which completes the proof. □

Now, we can derive the claimed density statements from this theorem and theorem 3.3.1.

Corollary 3.3.3. *For integers $n, b > 0$, as $b \to \infty$ we have*

$$i_n^{\mathcal{O}_K}(b)/l_n(b) \in O(b^{1-n}).$$

For all non-trivial dimensions the density ratio goes to zero. Though the density statement we derived is asymptotic, the theorems we have presented contain all the necessary constants to make it explicit. In practical experiments, we observed that for dimensions ≥ 100 the amount of bounded $\mathcal{O}_K$-ideal lattices is vanishingly small compared to all lattices under the same bound.

This tells us that whenever the maximal order $\mathcal{O}_K$ of some number field $K = \mathbb{Q}(\zeta)$ coincides with the order $\mathbb{Z}[\zeta]$, the number of ideal lattices with respect to this ring, which are exactly the ones used in worst-case hardness assumptions, is vanishingly small among all lattices (given the same bound on the determinant).

In practice, the only rings that have been used and suggested for constructing ideal lattices are maximal orders of cyclotomic number fields. This is because these fields allow an especially efficient multiplication due to the Fast Fourier Transform.

3.4 General Orders

In the previous section we have shown that the number of lattices with bounded determinant corresponding to ideals in the maximal order of a number field is small compared to the total number of lattices with the same bound on the determinant.

For all cyclotomic fields (and a lot more), it is certainly true that $\mathbb{Z}[\zeta]$ is the maximal order and our result stands. However, we may ask ourselves what happens to our counting argument if we use other (non-maximal) orders to construct our ideal lattices.

We will give a partial answer to this question here. We will use an old result of Dedekind and the result of last section to show that the number of bounded ideals coprime to the conductor is small. This leaves open the problem of counting the number of bounded ideals not coprime to the conductor.

Definition 3.4.1 (Conductor)**.** Given any order $\mathcal{O} \subseteq K$, we define its *conductor* to be $\mathfrak{c} = \{\, \alpha \in K : \alpha\mathcal{O}_K \subseteq \mathcal{O} \,\}$.

Note that $\mathfrak{c}$ is the largest ideal of $\mathcal{O}$ that is also an ideal of $\mathcal{O}_K$.

Definition 3.4.2. Let $\mathcal{O} \subseteq K$ be an order. We say that two ideals $\mathfrak{a}, \mathfrak{b} \subseteq \mathcal{O}$ are *coprime* if and only if $\mathfrak{a} + \mathfrak{b} = \mathcal{O}$.

With all the definitions in place, we can formulate Dedekind's result (e.g., [Lan73, pp. 92,94]).

Theorem 3.4.3. *Let $\mathcal{O} \subseteq K$ be any order with conductor $\mathfrak{c}$. Then the two monoids*

$$A = \{\, \mathfrak{a} \subseteq \mathcal{O} \text{ ideal} : \mathfrak{a} + \mathfrak{c} = \mathcal{O} \,\}, \qquad B = \{\, \mathfrak{b} \subseteq \mathcal{O}_K \text{ ideal} : \mathfrak{b} + \mathfrak{c} = \mathcal{O}_K \,\}$$

are isomorphic via the mapping

$$\phi\colon A \longrightarrow B : \mathfrak{a} \longmapsto \mathfrak{a}\mathcal{O}_K, \qquad \phi^{-1}\colon B \longrightarrow A : \mathfrak{b} \longmapsto \mathfrak{b} \cap \mathcal{O}.$$

This theorem implies the claims we made earlier. It shows that ideals in $\mathcal{O}$ coprime to the conductor correspond 1-to-1 to ideals in $\mathcal{O}_K$. So there can never be more ideals in $\mathcal{O}$ coprime to the conductor than there are ideals $\mathcal{O}_K$ in total. By the argument given in Corollary 3.3.3, the number of the related $\mathcal{O}$-ideal lattices is vanishingly small.

4 Lattice-Based Compression Functions

In this chapter, we will discuss the SWIFFT compression functions, proposed by Lyubashevsky *et al.* at FSE 2008. They are very efficient instantiations of generalized compact knapsacks for a specific set of parameters. In particular, they have the property that, *asymptotically*, finding collisions for a randomly chosen compression function implies being able to solve computationally hard ideal lattice problems in the *worst-case*.

We will present three results. First, we present new average-case problems, which may be used for all lattice schemes whose security is based on SIS either with general or ideal lattices. The new average-case problems require less description bits, resulting in improved keysize and speed for these schemes. Second, we propose a parameter generation algorithm for SWIFFT, where the main parameter n can be any integer in the image of Euler's totient function, and not necessarily a power of 2 as before. Third, we give experimental evidence that finding ℓ_2-*pseudo-collisions*[1] for SWIFFT is as hard as breaking a 68-bit symmetric cipher according to the well-known heuristic by Lenstra and Verheul. We also recommend conservative parameters corresponding to a 127-bit symmetric cipher.

This chapter is based on [BL09b]. It was presented at the *10th International Conference on Cryptology in India* (Indocrypt) 2009 in New Delhi, India.

4.1 Introduction

Collision-resistant hash functions play a key role in the IT world. They are an important part of digital signatures as well as authentication protocols.

Despite their fundamental importance, several established hash function designs have turned out to be insecure, for example MD5 and SHA-1 [SLdW07, CMR07]. To avoid this lack of security in a central place for the future, we need efficient hash functions with strong security guarantees.

One such hash function with an intriguing design is SWIFFTX [ADL$^+$08]. In contrast to

[1] These ℓ_2-pseudo-collisions are not related to the usual pseudo-collisions as defined in, e.g., the Handbook of Applied Cryptography [MvOV01].

all other practical hash functions, including all SHA-3 candidates, it remains the only hash function where the most prominent security property, namely collision-resistance, relies solely on the hardness of a well studied mathematical problem. This guarantee on the collision-resistance of SWIFFTX is a feature derived directly from SWIFFT [LMPR08], the internal compression function, which has the same guarantee.

SWIFFTX was part of a hash design competition by the National Institute for Standards and Technology (NIST). It did not survive the competition, and we suspect this is due to inefficiency, with the main bottleneck being SWIFFT.

This chapter has three contributions. First, we present new compact average-case problems that require less description bits than the ones previously used. These are shown to be at least as hard as previous ones. To demonstrate the improvement, we then show how the new problems can be used to create more efficient SWIFFT instances. This improvement to space and time requirements applies not only to SWIFFT, but universally to all lattice-schemes based on worst-case problems via Ajtai's reduction (e.g., [GGH96, GPV08, LMPR08, LM08, XT08]).

Second, we present a parameter generation algorithm that produces parameters sets, where the main parameter may be any number in the image of Euler's totient function and not just a power of 2 as before. We show that all efficiency improvements that were proposed for the standard SWIFFT parameters also apply to an exemplary parameter set from our generator. These parameters are later shown to be far more secure than the standard SWIFFT parameters.

Third, we show that SWIFFT is subject to lattice basis reduction attacks in dimensions that are far smaller than the ones the authors claim are necessary. We give evidence that finding ℓ_2-pseudo-collisions takes an effort comparable to breaking a 68-bit symmetric cipher, i.e., it is considered feasible today. Such pseudo-collisions are closely related to actual collisions and being able to recover them should be infeasible for high security applications. We give alternative parameters which realize.

This chapter is organized as follows. Section 4.2 introduces the new average-case problems and reductions from SIS respective IdealSIS. Section 4.3 describes the SWIFFT compression function family and the SWIFFT lattice. Section 4.4 presents the parameter generation algorithm and Section 4.5 discusses SWIFFT's security.

4.2 Compact (Ideal)SIS

In this section we present a new average-case problem SIS'. We show that the average-case *small integer solution* problem (SIS) reduces to SIS'. Hence, SIS' can be used, for example, to solve worst-case problems that reduce to SIS without any loss in the parameters. The advantage is that SIS' requires $n^2 \log(q)$ less random bits. A similar construction is possible for the average-case problem LWE and has indeed been suggested (without naming it or proving reductions) by Regev and Micciancio in [MR09].

All cryptographic schemes whose security relies on SIS can switch to SIS', resulting in a scheme with smaller keys which is also slightly faster (due to the structure of SIS'). This includes *all* systems based on worst-case lattice problems via Ajtai's reduction [Ajt96a] or the adaptions thereof (e.g., [GGH96, GPV08]).

We will also show that the same idea can be adapted to the IdealSIS problem, which is SIS restricted to the class of ideal lattices. The number of description bits we save in this case is $n\log(q)$. Hence, *all* schemes based on worst-case problems in ideal lattices via the reduction of Lyubashevsky and Micciancio [LM08] can benefit from using IdealSIS′ (e.g., [LMPR08, LM08, XT08, SSTX09]). How these improvements apply to SWIFFT may be seen in Section 4.3.

The technical difference is that SIS chooses a somewhat random basis for a random lattice, whereas SIS′ chooses only a random lattice and takes the basis in Hermite normal form. This is analogous to using the standard (or systematic) form for linear codes in coding theory.

To illustrate the difference between the two problems we restate the definition of SIS as seen in Chapter 2 and give the definition of SIS′.

Definition 4.2.1 (SIS)**.** In the *small integer solution* problem $\mathsf{SIS}(n,q,m,\beta)$, we are given a matrix $\mathbf{A} \sim U(\mathbb{Z}_q^{n\times m})$ chosen uniformly at random and a real $\beta > 0$. The task is to find a nonzero integer vector $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $\mathbf{A}\mathbf{x} \equiv \mathbf{0} \pmod q$ and $\|\mathbf{x}\| \leq \beta$.

In contrast, the distribution for $\mathbf{A}$ used in SIS′ is different.

Definition 4.2.2 (SIS′)**.** In the *compact small integer solution* problem $\mathsf{SIS}'(n,q,m,\beta)$, we are given a matrix $\mathbf{A}' \sim U(\mathbb{Z}_q^{n\times(m-n)})$ chosen uniformly at random and a real $\beta > 0$. The task is to find a nonzero integer vector $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $[\mathbf{I}_n, \mathbf{A}']\mathbf{x} \equiv \mathbf{0} \pmod q$ and $\|\mathbf{x}\| \leq \beta$. Here, $\mathbf{I}_n$ is the n-dimensional identity matrix.

This problem corresponds to the previous one, but with $\mathbf{A} = [\mathbf{I}_n, \mathbf{A}']$, so it requires $n^2\log(q)$ less descriptions bits. We show that new problem is as hard as the old one.

Theorem 4.2.3. *Let n, $q(n) \geq 2$, $m(n) \geq (1+\epsilon)n$ be positive integers, and $\beta(n) > 0$ be a positive real, then $\mathsf{SIS}(n,q(n),m(n),\beta(n))$ reduces to $\mathsf{SIS}'(n,q(n),m(n),\beta(n))$. Here, $\epsilon > 0$ is some real number independent of n.*

Proof. Given $\mathbf{A}$, an instance of $\mathsf{SIS}(n,q(n),m(n),\beta(n))$, let E be the event that there are n column vectors in $\mathbf{A}$ which are linearly independent mod $q(n)$. For brevity, we will write q, m, β instead of $q(n)$, $m(n)$, $\beta(n)$.

Assuming E holds, there is a permutation matrix $\mathbf{P} \in \{0,1\}^{m\times m}$ such that $\mathbf{AP} = [\mathbf{A}', \mathbf{A}'']$ and $\mathbf{A}'$ is invertible mod q. We use the SIS′ oracle to solve $(q, [\mathbf{I}_n, \mathbf{A}'^{-1}\mathbf{A}''], \beta)$. This instance is distributed according to SIS′, when the matrix $\mathbf{A}'^{-1}\mathbf{A}''$ is distributed according to $U(\mathbb{Z}_q^{n\times(m-n)})$. This is the case, since $\mathbf{A}''$ was distributed this way and $\mathbf{A}'^{-1}$ is invertible mod q, so it is a permutation on the vectors $\mathbb{Z}_q^n$ which does not affect the uniform distribution. From the SIS′ oracle, we obtain a solution $\mathbf{z}$. The vector $\mathbf{Pz}$ solves our SIS instance because

$$\mathbf{0} = [\mathbf{I}_n, \mathbf{A}'^{-1}\mathbf{A}'']\mathbf{z} = [\mathbf{A}', \mathbf{A}'']\mathbf{z} = \mathbf{APz} \pmod q.$$

We will show that the probability of E not occurring is negligible. The number of matrices $\mathbf{A}$ with n linearly independent columns is equal to the number of matrices with

n linearly independent rows. For E to occur, the first row may be anything but the zero-row, giving $(q^m - 1)$ possibilities; the second row can be all but multiples of the first, giving $(q^m - q)$ possibilities, and so on. The total number of matrices is q^{nm}, so we get

$$\Pr[\text{not } E] = 1 - q^{-nm} \prod_{i=0}^{n-1} (q^m - q^i) = 1 - \prod_{i=0}^{n-1} (1 - q^{i-m}).$$

Let $c = -2\ln(1/2)$, we bound the probability

$$\begin{aligned} 1 - \prod_{i=0}^{n-1} (1 - q^{i-m}) &= 1 - \exp\left((-1)^2 \ln \Big(\prod_{i=0}^{n-1} (1 - q^{i-m}) \Big) \right) \\ &\overset{(1)}{\leq} \sum_{i=0}^{n-1} - \ln(1 - q^{i-m}) \overset{(2)}{\leq} c\, q^{-m} \sum_{i=0}^{n-1} q^i \\ &= c(q^n - 1)/(q^m(q-1)) \leq c/q^{m-n} \overset{(3)}{\leq} c/2^{\epsilon n}. \end{aligned}$$

Inequality (1) holds because for all real x we have $1 - \exp(-x) \leq x$. Similarly, inequality (2) holds because for all $0 \leq x \leq 1/2$ we have $-\ln(1 - x) \leq cx$. Finally, inequality (3) follows from the conditions stated in the theorem. The resulting function is negligible, which completes the proof. □

In the remainder of the section we will adapt Theorem 4.2.3 to the case of ideal lattices. Throughout this part, let ζ_n be a sequence of algebraic integers such that the ring $R_n = \mathbb{Z}[\zeta_n]$ is a $\mathbb{Z}$-module of rank n, i.e., $R_n \cong \mathbb{Z}^n$ as an additive group. Since $R_n = [1, \zeta_n, \ldots, \zeta_n^{n-1}]\, \mathbb{Z}^n$, we can use any norm on ring elements by transforming them to integral coefficient vectors of this power basis. In order to apply norms on tuples of ring elements, we take the norm of the vector consisting of the norms of each element, so for $\widehat{\mathbf{z}} \in R_n^m$ we have $\|\widehat{\mathbf{z}}\| = \|(\|\mathbf{z}_1\|, \ldots, \|\mathbf{z}_m\|)\|$. We use the shorthand $R_{n,q} = R_n/\langle q \rangle = \mathbb{Z}_q[\zeta_n]$.

Definition 4.2.4 (IdealSIS)**.** Given integers n, m, q, a tuple $\widehat{\mathbf{a}} = [\mathbf{a}_1, \ldots, \mathbf{a}_m] \in R_{n,q}^m$, and a real β, the *ideal shortest vector problem* $\mathsf{IdealSIS}(n, q, m, \beta)$ is to find a nonzero vector $\widehat{\mathbf{z}} = [\mathbf{z}_1, \ldots, \mathbf{z}_m] \in R_n^m \setminus \{\mathbf{0}\}$, such that

$$\sum_{i=1}^{m} \mathbf{a}_i\, \mathbf{z}_i = \mathbf{0} \qquad \text{and} \qquad \|\widehat{\mathbf{z}}\| \leq \beta.$$

Analogous to the case of general lattices, the compact problem is:

Definition 4.2.5 (IdealSIS′)**.** Given integers n, m, q, a tuple $\widehat{\mathbf{a}}' = [\mathbf{a}_1, \ldots, \mathbf{a}_{m-1}] \in R_{n,q}^m$, and a real β, the *compact ideal shortest vector problem* $\mathsf{IdealSIS}'(n, q, m, \beta)$ is to find a nonzero vector $\widehat{\mathbf{z}} = [\mathbf{z}_1, \ldots, \mathbf{z}_m] \in R_n^m \setminus \{\mathbf{0}\}$, such that

$$\mathbf{z}_1 + \sum_{i=1}^{m-1} \mathbf{a}_i\, \mathbf{z}_{i+1} = \mathbf{0} \qquad \text{and} \qquad \|\widehat{\mathbf{z}}\| \leq \beta.$$

Again this problem corresponds to the previous one with $\widehat{\mathbf{a}} = [\mathbf{1}, \widehat{\mathbf{a}}']$. In this case we save $n \log(q)$ description bits and once more, we will show that the new problem is as hard as the well known one.

Theorem 4.2.6. *Let n, $m(n) \in \Omega(\log(n))$ be positive integers, $q(n) \in \omega(n)$ be prime, and $\beta(n) > 0$ be real, then the corresponding* $\mathsf{IdealSIS}$ *reduces to* $\mathsf{IdealSIS}'$.

Proof. Given $\widehat{\mathbf{a}}$, an instance of $\mathsf{IdealSIS}(n, q(n), m(n), \beta(n))$, let E be the event that there is an index i such that $\mathbf{a}' = \mathbf{a}_i$ is invertible mod $q(n)$. For brevity, we will write q, m, β instead of $q(n)$, $m(n)$, $\beta(n)$.

Assuming E holds, there is a permutation $\mathbf{P} \in \{0,1\}^{m \times m}$, such that $\widehat{\mathbf{a}}\mathbf{P} = [\mathbf{a}', \widehat{\mathbf{a}}'']$ and $\mathbf{a}'$ is invertible mod q. We let the $\mathsf{IdealSIS}'$ oracle solve the instance $(q, [\mathbf{1}, \mathbf{a}'^{-1}\widehat{\mathbf{a}}''], \beta)$. This instance is distributed according to $\mathsf{IdealSIS}'$, when the tuple $\mathbf{a}'^{-1}\widehat{\mathbf{a}}''$ is distributed according to $U(R_{n,q}^{m-1})$. This is the case, since $\widehat{\mathbf{a}}''$ was distributed this way and $\mathbf{a}'^{-1}$ is invertible mod q, so it is a permutation on the elements $R_{n,q}$ which does not effect the uniform distribution. From the $\mathsf{IdealSIS}'$ oracle, we obtain a solution $\widehat{\mathbf{z}}$. The vector $\mathbf{P}\widehat{\mathbf{z}}$ solves our $\mathsf{IdealSIS}$ instance:

$$\mathbf{0} = \Big[\mathbf{1}, \mathbf{a}'^{-1}\widehat{\mathbf{a}}''\Big]\widehat{\mathbf{z}} = \Big[\mathbf{a}', \widehat{\mathbf{a}}''\Big]\widehat{\mathbf{z}} = \widehat{\mathbf{a}}\mathbf{P}\widehat{\mathbf{z}} \pmod{q}.$$

We will show that the probability of E not occurring is negligible. Let f be the minimal polynomial of ζ_n, and $f_1, \ldots, f_k$ be the irreducible factors of f over $\mathbb{Z}_q$. Since q is prime, for any invertible element $\mathbf{a} \in R_{n,q}$, it is necessary and sufficient that $\mathbf{a} \bmod f_i \neq \mathbf{0}$. So, the number of invertible elements is $|R_{n,q}^*| = \prod_{i=1}^k (q^{\deg(f_i)} - 1)$. The total number of ring elements is $|R_{n,q}| = q^n$. For E to occur, only one of the m ring elements must be invertible, so we get

$$\Pr[\text{not } E] = \left(1 - q^{-n} \prod_{i=1}^k \left(q^{\deg(f_i)} - 1\right)\right)^m = \left(1 - \prod_{i=1}^k \left(1 - q^{-\deg(f_i)}\right)\right)^m$$

Let $c = -2\ln(1/2)$, we bound $(\Pr[\text{not } E])^{1/m}$

$$\begin{aligned} 1 - \prod_{i=1}^k \left(1 - q^{-\deg(f_i)}\right) &\overset{(1)}{\leq} \sum_{i=1}^k -\ln\left(1 - q^{-\deg(f_i)}\right) \overset{(2)}{\leq} c\sum_{i=1}^k q^{-\deg(f_i)} \\ &= ck/q \leq cn/q \overset{(3)}{\in} 1/\omega(1). \end{aligned}$$

Inequality (1) holds because for all real x we have $1 - \exp(-x) \leq x$. Similarly, inequality (2) holds because for all $0 \leq x \leq 1/2$ we have $-\ln(1-x) \leq cx$. Finally, (3) follows from the conditions stated in the theorem. Since $m(n) \in \Omega(\log(n))$, we know $\Pr[\text{not } E]$ is negligible. □

4.3 SWIFFT Compression Functions

The SWIFFT compression function family was proposed by Lyubashevsky *et al.* at FSE 2008 [LMPR08]. They showed that its efficiency is comparable to SHA-2. It was already shown in 2006 by Lyubashevsky and Micciancio [LM06] that the collision resistance of generalized compact knapsacks, of which SWIFFT is one particular instantiation, is asymptotically based on *worst-case* computational problems in ideal lattices. The arguments given later in 2006 by Peikert and Rosen in [PR06] can also be adapted to prove collision resistance of generalized compact knapsacks with a *tighter* connection to the same worst-case problem.

Specifically, for a set of integer parameters (n, m, p), in the SWIFFT case $(64, 16, 257)$, they use the polynomial $f(x) = x^n + 1$, the ring $R_{p,n} = \mathbb{Z}_p[x]/\langle f \rangle$ and the subset

$$D_n = \{d_0 + d_1 x + \cdots d_{n-1}x^{n-1} \in R_{p,n} : (d_i \bmod p) \in \{0, 1\} \text{ for } i = 0, \ldots, n-1\},$$

to define the family

$$\mathcal{H}_{n,m,p} = \left\{ h_{\widehat{\mathbf{a}}} \colon D_n^m \ni \widehat{\mathbf{x}} \longmapsto \sum_{i=1}^{m} \mathbf{a}_i \mathbf{x}_i \pmod{p} : (\mathbf{a}_1, \ldots, \mathbf{a}_m) = \widehat{\mathbf{a}} \in R_{p,n}^m \right\}.$$

These functions can be computed efficiently. Let $\omega_0, \ldots, \omega_{n-1}$ be the roots of f in $\mathbb{Z}_p$ in any order, and $\mathbf{V}$ be the Vandermonde matrix generated by them

$$\mathbf{V} = \begin{pmatrix} 1 & \omega_0 & \ldots & \omega_0^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_{n-1} & \ldots & \omega_{n-1}^{n-1} \end{pmatrix}.$$

Applying the *Fast Fourier Transform* over $\mathbb{Z}_p$ to SWIFFT, we get

$$\mathbf{z} \equiv \sum_{i=0}^{m-1} \mathbf{a}_i \mathbf{x}_i \bmod f \equiv \mathbf{V}^{-1} \left(\sum_{i=0}^{m-1} \mathbf{V}\mathbf{a}_i \odot \mathbf{V}\mathbf{x}_i \right) \pmod{p}, \tag{4.3.1}$$

where $\odot$ is the pointwise multiplication in $\mathbb{Z}_p^n$. Since $\mathbf{V}$ is invertible, we may use $\mathbf{z}' = \mathbf{V}\mathbf{z}$ as hash, instead of $\mathbf{z}$. Since the compression function key $\widehat{\mathbf{a}}$ is fixed, we may precompute $\mathbf{a}'_i = \mathbf{V}\mathbf{a}_i$ for all i. So evaluating the compression function amounts to computing all n components of $\mathbf{z}'$ with

$$z'_j = \sum_{i=0}^{m-1} a'_{i,j}\, x'_{i,j} \bmod p, \qquad\qquad x'_{i,j} = \sum_{l=0}^{n-1} \omega_j^l\, x_{i,l} \bmod p.$$

Due to the form of f we can set $\omega_j \leftarrow \omega^{2j+1}$ for any element ω of order $2n$ in $\mathbb{Z}_p$. We insert the parameters and split up the indices $j = j_0 + 8j_1$ and $l = l_0 + 8l_1$.

$$\begin{aligned} x'_{i,j_0+8j_1} &= \sum_{l_0=0}^{7} \sum_{l_1=0}^{7} \omega^{(l_0+8l_1)(2(j_0+8j_1)+1)}\, x_{i,l_0+8l_1} \bmod p \\ &= \sum_{l_0=0}^{7} \omega^{16 l_0 j_1} \cdot \underbrace{\omega^{l_0(2j_0+1)}}_{m_{l_0,j_0}} \cdot \underbrace{\sum_{l_1=0}^{7} \omega^{8l_1(2j_0+1)}\, x_{i,l_0+8l_1} \bmod p}_{t_{l_0,j_0}} \end{aligned} \tag{4.3.2}$$

The quantities t_{l_0,j_0} for all 2^8 possible values of x_{i,l_0+8l_1} and m_{l_0,j_0} can be precomputed. The SWIFFT authors recommend using $\omega = 42$, because then $\omega^{16} \bmod p = 4$, so some multiplications in the last expression can be realized with bit-shifts. A single $\mathbf{x}'_i$, i.e., the last expression for all j, can then be evaluated with a total of 64 multiplications and $8 \cdot 24$ additions/subtractions using an FFT network. The total number of operations (ignoring index calculations and modular reduction) for the standard SWIFFT parameters is

$$\underbrace{16 \cdot 64}_{\text{computing } x'_{i,j}} + \underbrace{16 \cdot 64}_{\text{all } a'_{i,j} \cdot x'_{i,j}} = 2048 \text{ multiplications}$$

$$\underbrace{16 \cdot 8 \cdot 24}_{\text{computing } x'_{i,j}} + \underbrace{16 \cdot 64 - 1}_{\text{summing } a'_{i,j} \cdot x'_{i,j}} = 4095 \text{ additions/subtractions}$$

Applying the ideas put forward in Section 4.2, we recommend using functions from

$$\mathcal{H}'_{n,m,p} = \left\{ h_{\widehat{\mathbf{a}}} \colon D_n^m \ni \widehat{\mathbf{x}} \longmapsto \mathbf{x}_1 + \sum_{i=1}^{m-1} \mathbf{a}_i \mathbf{x}_{i+1} \pmod p : (\mathbf{a}_1, \ldots, \mathbf{a}_{m-1}) = \widehat{\mathbf{a}} \in R_{p,n}^{m-1} \right\}.$$

This results in a slightly more efficient scheme which uses less memory. Recall that all entries in $\widehat{\mathbf{a}}' = (\mathbf{1}, \mathbf{a}_1, \ldots, \mathbf{a}_{m-1})$ can be precomputed in practice and having one of them equal to $\mathbf{1}$ saves some multiplications during evaluation depending on the implementation. In Equation 4.3.1, if we computed $\mathbf{z}$ instead of $\mathbf{z}'$ the speed-up would be $1/m$. For $m = 16$ this is $\approx 6\%$ and it may be further increased with the sliding window method used for NTRU [BDL08]. However, at the moment it is more efficient to compute $\mathbf{z}'$. In this case, we save n multiplications, which is about 1% of all operations for standard SWIFFT parameters.

4.3.1 The SWIFFT Lattice

Let $\widehat{\mathbf{a}} \in R_{p,n}$. Consider the function $h_{\widehat{\mathbf{a}}} \in \mathcal{H}_{n,m,p}$ and extend the domain to $R_n = \mathbb{Z}[x]/\langle f \rangle$. The coefficient vectors of *periods* of this function form the set

$$\Lambda_p^{\perp}(\widehat{\mathbf{a}}) = \left\{ (v_1, \ldots, v_{nm}) \in \mathbb{Z}^{nm} : h_{\widehat{\mathbf{a}}} \left(\sum_{i=0}^{n-1} v_{i+1} x^i, \ldots, \sum_{i=0}^{n-1} v_{m(i+1)} x^i \right) = \mathbf{0} \right\}.$$

This is a lattice of dimension nm, since the extended $h_{\widehat{\mathbf{a}}}$ is R_n-linear. A basis for this lattice can be found efficiently using a method described by Buchmann *et al.* in [BLR08]. Collisions in the original (unextended) function $h_{\widehat{\mathbf{a}}}$ correspond exactly to vectors in this lattice with ℓ_∞-norm bounded by 1. Therefore we refer to these lattices as *SWIFFT lattices*.

A *pseudo-collision* is a vector in this lattice with Euclidean norm less than $\sqrt{nm}$, i.e., all vectors in the smallest ball containing all collisions. So every collision corresponds to a pseudo-collision, but not vice versa.

4.4 Parameter Generation

In the previous section, we have seen that SWIFFT, as presented by the authors, has only one set of parameters. In this section we will show that several sets may be used without losing much in terms of efficiency.

Let $k > 0$ be some integer, p be prime and $n = \varphi(k)$, where φ is Euler's totient function. Furthermore, let f be the k-th cyclotomic polynomial, i.e.,

$$f(x) = \prod_{d|k} (1 - x^{k/d})^{\mu(d)}, \text{with} \quad \mu(d) = \begin{cases} (-1)^{\ell} & \text{if } d \text{ is a product of } \ell \text{ distinct primes} \\ 1 & \text{if } d = 1 \\ 0 & \text{otherwise.} \end{cases}$$

It's known that this is always an integer polynomial. It is also monic, irreducible over the integers, and has degree equal to n [Lan65]. Using the same structures as in the Section 4.3 with this new f, i.e., the ring $R_{p,n}$ and subset D_n, we can construct the same compression function family and the asymptotic security argument given in [PR06, LM06] still holds. In order to apply FFT as before, we need to ensure that elements of order k exist in $\mathbb{Z}_p$. This is guaranteed whenever $k \mid (p - 1)$.

We now describe an algorithm for generating parameter sets (n, m, p). For the polynomial f we will use the k-th cyclotomic polynomial such that $n = \varphi(k)$. However, there are cases where multiple polynomials are possible, for example, $n = 256 = \varphi(257) = \varphi(512)$. In these cases, we choose k such that the resulting bitlength of the output is shorter, i.e., the one with smaller p.

```
Input: Integer $n$, s.t. $n = \varphi(k), k > 0$
Output: Parameters $(n, m, p)$
$l \leftarrow 1$
$p \leftarrow k + 1$
while not isPrime(p) do
    $l \leftarrow l + 1$
    $p \leftarrow l \cdot k + 1$
end
$m \leftarrow \lceil 1.99 \cdot \log_2(p) \rceil$
```

Algorithm 1: Parameter generation for $n = \varphi(k), k > 0$.

To illustrate the differences between these parameters, we compute several additional quantities for each set and list them in Figure 4.1.

For example, the output bitlength of the resulting compression function is given by $out = n(\lfloor \log_2(p) \rfloor + 1)$ and its compression rate is $cr = m / \log_2(p)$. To indicate the collision resistance of each parameter set, we state the Hermite factor δ that needs to be achieved in order to find pseudo-collisions, and the minimal lattice dimension d where we can expect to find such pseudo-collisions. The two latter values are computed in the following fashion.

k	n	m	p	out	cr	δ	d
128	64	16	257	513	1.999	1.0084	206
67	66	17	269	529	2.106	1.0084	211
71	70	19	569	631	2.076	1.0073	248
73	72	17	293	577	2.074	1.0077	231
79	78	17	317	625	2.046	1.0072	251
83	82	15	167	575	2.032	1.0075	237
89	88	15	179	617	2.004	1.0071	255
97	96	18	389	769	2.092	1.0061	308
101	100	19	607	901	2.055	1.0056	340
103	102	19	619	919	2.049	1.0055	348
107	106	19	643	955	2.037	1.0053	361
109	108	21	1091	1081	2.081	1.0049	392
113	112	16	227	785	2.044	1.0058	325
127	126	18	509	1009	2.002	1.0047	408
256	128	16	257	1025	1.999	1.0051	373

Figure 4.1: Parameters for $64 \leq n \leq 128$, k prime or a power of two.

Consider the function $len(d) = p^{n/d}\delta^d$. According to Gama and Nguyen [GN08][2] this is the Euclidean size of the smallest vector we are likely to find when reducing a sublattice with dimension d of any SWIFFT lattice $\Lambda_p^{\perp}(\widehat{\mathbf{a}})$. Micciancio and Regev observed in [MR09] that this function takes its minimal value

$$len(d_{min}) = \delta^{2\sqrt{n\log(p)/\log(\delta)}} \qquad \text{for} \qquad d_{min} = \sqrt{n\log(p)/\log(\delta)}.$$

We recall that a pseudo-collision is a vector in $\Lambda_p^{\perp}(\widehat{\mathbf{a}})$ with Euclidean norm at least $\sqrt{nm}$. In order to find such a vector, we need a δ, such that $len(d_{min}) \leq \sqrt{nm}$. We say this is the Hermite factor required for finding pseudo-collisions, and the corresponding d_{min} is the minimal dimension where we can expect to find a pseudo-collision. Note that these minimal dimensions are about *5 times* smaller than the corresponding dimensions of the SWIFFT lattices. To give an intuition, Gama and Nguyen state that the best lattice reduction algorithms known today can achieve a Hermite factor of roughly $\delta = 1.01$ in high dimensions within acceptable time.

4.4.1 Recommended Parameters

We will give arguments in Section 4.5.3 that parameters sets with $d \geq 260$ correspond to SWIFFT instances where finding pseudo-collisions should be considered infeasible. The smallest set of such parameters in Figure 4.1 are

$$(n, m, p) = (96, 18, 389).$$

[2] Their experiments were performed on random lattices following a different distribution, but experimentally their results apply here as well.

We will see that finding pseudo-collisions for these parameters is as hard as breaking a 127-bit symmetric cipher, whereas, for the original SWIFFT parameters, finding such collisions only compares to breaking a 68-bit symmetric cipher.

Note that most of the efficiency improvements we outlined in Section 4.4 for the original SWIFFT function are still possible with these parameters. Recall Equation 4.3.2, since $k = 97$ is prime we can set $\omega_j \leftarrow \omega^{j+1}$ for any element ω of order k in $\mathbb{Z}_p$. To achieve optimal performance, we recommend to split up the indices $l = l_0 + 8l_1$, where $0 \leq l_0 \leq 7$ and $0 \leq l_1 \leq 11$, similarly for the index j, and finally use $\omega = 275$, since multiplying with $\omega^8 = 16$ can then be realized with bit-shifts. Corresponding to Equation 4.3.2 we get

$$x'_{i,j_0+8j_1} = \sum_{l_0=0}^{7} \omega^{8l_0 j_1} \cdot \underbrace{\omega^{l_0(j_0+1)}}_{m_{l_0,j_0}} \cdot \underbrace{\sum_{l_1=0}^{11} \omega^{l_1(8j_0+64j_1+8)}\, x_{i,l_0+8l_1}}_{t_{l_0,j_0,j_1}} \bmod p.$$

Note that the precomputed t part depends on j_1 now, and needs to be available for 2^{12} possible $x_{i,l}$. So this part will need $12 \cdot 2^4 = 192$ times the space it did before. Doing the same reasoning as before, the number of operations (are their relative increase in percent) is:

$$\underbrace{18 \cdot 64}_{\text{computing } x'_{i,j}} + \underbrace{18 \cdot 96}_{\text{all } a'_{i,j} \cdot x'_{i,j}} = 2880 \text{ (+40\%) multiplications}$$

$$\underbrace{18 \cdot 12 \cdot 24}_{\text{computing } x'_{i,j}} + \underbrace{18 \cdot 96 - 1}_{\text{summing } a'_{i,j} \cdot x'_{i,j}} = 6911 \text{ (+68\%) additions/subtractions}$$

Overall, we have a huge increase in security, almost doubling the effort required to find pseudo-collisions and a comparatively small loss in efficiency.

4.5 Security Analysis

The collision resistance of SWIFFT has the desirable property of being reducible from a worst-case computational problem. In particular, this means an algorithm which breaks random instances of SWIFFT compression functions with main parameter n can also be used to find short nonzero vectors in *all* ideals of the ring $\mathbb{Z}[x]/\langle x^n + 1\rangle$. Finding such vectors is assumed to be infeasible for large n. However, for the standard SWIFFT parameter, $n = 64$, exhaustive search algorithms find these short vectors in less than one hour. In the lattice challenge [BLR08] open for all enthusiasts, similar problems have been solved[3] up to $n = 108$. Gama and Nguyen even state that finding the *shortest* vector in n-dimensional lattices for $n \leq 70$ should be considered easy [GN08]. So, the resulting lower bound on the attacker's runtime is insignificant. However, not attacking the underlying worst-case problem, but a concrete SWIFFT instance is much harder.

We will analyze the *practical* security of SWIFFT. As we have seen in Section 4.3.1, collisions in the SWIFFT compression functions naturally correspond to vectors with ℓ_∞-norm

[3] See `http://www.latticechallenge.org`

bounded by 1 in certain lattices. These may be recovered with lattice basis reduction algorithms. Since these algorithms are highly optimized to find small vectors in the Euclidean norm, it is reasonable to analyze the computational problem of finding pseudo-collisions instead of collisions. These are vectors in the *smallest ball* which contains all vectors corresponding to collisions, so an algorithm which minimizes the Euclidean norm cannot distinguish between the two. In this section, we give experimental evidence that according to a well-known heuristic by Lenstra and Verheul [LV01], finding pseudo-collisions for the standard SWIFFT parameters is comparable to breaking a 68-bit symmetric cipher. In comparison, all other attacks analyzed by the SWIFFT authors take 2^{106} operations and almost as much space.

In their original proposal of SWIFFT, Lyubashevsky *et al.* provide a first analysis of all standard attacks. When it comes to attacks using lattice reduction however, they state that the dimension 1024 of SWIFFT lattices is too big for current algorithms. We start by showing that reducing sublattices of dimension 251, which corresponds to $m = 4$, is sufficient to find pseudo-collisions and dimension 325 ($m = 5$) is sufficient for collisions. Beyond this point, as Micciancio and Regev observe in [MR09], “the problem [SVP] cannot become harder by increasing m”. This means if we find a pseudo-collision in dimension 251, we can pad it with zeros to obtain a pseudo-collision for SWIFFT. In practice, even dimension $d = 205$ is sufficient to find pseudo-collisions (cf. Figure 4.1). In particular, this means that SWIFFTX, where internally SWIFFT is used with $m = 32$, is not more secure than SWIFFT.

4.5.1 Existence of (Pseudo-)Collisions in d-Dimensional Sublattices

Naturally, lattice basis reduction algorithms run much faster on lattices of small dimension. So, we want to find the minimal dimension d such that a d-dimensional sublattice of the SWIFFT lattice still contains a vector corresponding to a pseudo-collision.

The method we have given in Section 4.4 for choosing the dimension d is heuristic, because it is based on an observation by Gama and Nguyen that is experimentally sound but unproven. We will now give a related result independent of experiments but dependent on the structure of SWIFFT sublattices. It is a pigeonhole argument, which gives an upper bound on the best subdimension d.

Let $h_{\widehat{\mathbf{a}}}$ be a random SWIFFT compression function with parameters (n, m, p). The range of this function has size $|R| = p^n$.

Furthermore, let D be the set of all vectors in a d-dimensional submodule of $\mathbb{Z}^{nm}$ that have Euclidean norm less than $r = \sqrt{nm}/2$. Clearly, if D where a subspace of $\mathbb{R}^{nm}$, then, under the same conditions, its volume would be that of a d-dimensional ball with radius r, i.e. $r^d \pi^{d/2}/\Gamma(d/2 + 1)$. So, counting only integer points, we come to approximately the same number and get

$$|D| \approx r^d \pi^{d/2}/\Gamma(d/2 + 1).$$

This approximation is tight for $d > 100$.

We change the domain of $h_{\widehat{\mathbf{a}}}$ to D. Now, any collision in the modified $h_{\widehat{\mathbf{a}}}$ corresponds to a pseudo-collision of the SWIFFT function we started with, because whenever two elements $v_1, v_2 \in D$ are mapped to the same output under $h_{\widehat{\mathbf{a}}}$, then $v_1 - v_2$ is mapped $\mathbf{0}$ and by the triangle inequality, $\|v_1 - v_2\| \leq 2r = \sqrt{nm}$.

Comparing the sizes of domain and range, we find that these collisions exist by the pigeonhole principle for all $d \geq 251$. So the dimension $d = 205$ suggested by the heuristic looks too optimistic, but remember that this argument only gives an upper bound on the required d and does not take into account the randomness in the choice of $\widehat{\mathbf{a}}$.

The situation for proper collisions is similar. Here, we shrink the input to all vectors that have d coefficients in $\{0, 1\}$ and the rest equal to 0. Whichever coefficients we pick, the size of this domain is always $|D| = 2^d$. Collisions of this restricted function now correspond to proper collisions of the SWIFFT function we started with. By the same arguments as before, those collisions exist for $d \geq 513$.

However, a different analysis is possible here. It is one which does take into account the randomness of $\widehat{\mathbf{a}}$ and reveals that proper collisions exist for all $d \geq 325$.

4.5.2 Existence of Collisions in d-Dimensional Sublattices

In this section, we analyze how probable it is that vectors with trinary coefficients exist in a d-dimensional sublattice of the SWIFFT lattice. These correspond to proper collisions of the SWIFFT function. Unlike before, we will now take into account the randomness introduced by the choice of the compression function key $\widehat{\mathbf{a}}$. Let $\mathbf{A}$ be such that the SWIFFT function extended to the full domain $R_{n,p}$ is $h_{\widehat{\mathbf{a}}}(\mathbf{v}) = \mathbf{A}\mathbf{v} \bmod p$. Then, the SWIFFT lattice is given by the kernel of $\mathbf{A}$ over $\mathbb{Z}_p$ and the d-dimensional sublattices correspond to kernels of submatrices of $\mathbf{A}$ with d columns.

If, for example, $\widehat{\mathbf{a}}$ is chosen such that the i-th column in the corresponding $\mathbf{A}$ matrix is zero, then there is a binary vector in every d-dimensional sublattice where the corresponding submatrix contains the i-th column of $\mathbf{A}$. So, the randomness of $\widehat{\mathbf{a}}$ has some influence on the existence of vectors that correspond to collisions.

For simplicity, we deal with the case that the key defining the hash function, now given by the matrix $\mathbf{A}$, is unstructured and chosen uniformly at random from $\mathbb{Z}_p^{n \times nm}$. The following lemma gives the probability that a randomly chosen SWIFFT instance has *no collisions.*

Lemma 4.5.1. *Let $T = \{0, \pm 1\}^d \setminus \{\mathbf{0}\}$ and $\mathbf{A} \in \mathbb{Z}_q^{n \times d}$ be chosen uniformly at random, then*

$$\Pr[\,\forall \mathbf{v} \in T, \mathbf{A}\mathbf{v} \bmod q \neq \mathbf{0}] = \prod_{i=0}^{d-1} \Big(\max\{q^n - 3^i, 0\}/q^n \Big).$$

Proof. Consider the columns of $\mathbf{A}$ being drawn consecutively. We count the number of cases where the condition we check for holds. Certainly the condition is true if and only if the first drawn column is nonzero, giving $(q^n - 1)$ positive cases. Let the first column we drew be $\mathbf{a}_1$. For the condition to remain true, the second column must not be in the set $\{0, \pm 1\}\mathbf{a}_1$, giving $(q^n - 3)$ positive cases. Similarly, the third column must not be in $\{0, \pm 1\}\mathbf{a}_1 + \{0, \pm 1\}\mathbf{a}_2$, which yields $q^n - 3^2$ positive cases. An induction on d validates the given formula. □

Some exemplary probabilities for the existence of trinary vectors in a d-dimensional SWIFFT sublattice are:

n	m	p	δ	d
64	16	29	1.0140	125
64	16	33	1.0135	130
64	16	37	1.0131	134
64	16	41	1.0127	138
64	16	45	1.0124	141
64	16	49	1.0121	144
64	16	53	1.0119	147
64	16	57	1.0117	150
64	16	61	1.0115	152

Figure 4.2: Parameters used for our experiments.

d	273	$\cdots$	299	$\cdots$	325
Pr	2^{-80}		2^{-39}		1

4.5.3 Lattice Basis Reduction Experiments

For our experiments we chose the sublattice dimension where lattice basis reduction algorithms like LLL/BKZ behave optimal in practice (see Section 4.4). We then proceeded to compare the following lattice basis reduction algorithms to see which one performs best in practice on the lattices in our experiment. BKZ as implemented in version 5.5.1 of the "Number Theory Library"(NTL) by Shoup [Sho], Primal-Dual (PD) as implemented by Filipović[4] and Koy, and finally RSR as implemented by Ludwig[5]. Both latter algorithms are available on request from the authors. It became apparent that Primal-Dual runs much slower than both competitors, so for the main experiment we omitted it.

For our experiments, we fixed $n = 64$, $m = 16$ to their standard values and chose the third parameter p variable. This results in a steady decrease in the Hermite factor and increase in the dimension required to find pseudo-collisions (see Figure 4.2). We found that for smaller values of p, corresponding to smaller values of d, pseudo-collisions were found too fast to make sensible measurements.

For each of these 9 parameter sets, we created 10 random SWIFFT lattices using the pseudorandom number generator, which is part of NTL. We then proceeded to break all instances with the NTL floating-point variant of BKZ (bkzfp), by increasing the BKZ parameter β until a pseudo-collision was found and recording the total time taken in each case. We also broke all instances with a floating-point variant of Schnorr's Random Sampling Reduction (RSR) algorithm [Sch03] (rsrfp) implemented by Ludwig [BL06] using the parameters $\delta = 0.9, u = 22$ and again increasing β until a pseudo-collision was found.

In all cases, we computed the average runtime of both algorithms and plotted the base 2 log of this value relative to the dimension d. We also plotted a conservative extrapolation (assuming linear growth in logscale) for the average runtime of both algorithms (see Figure

[4] PD, Bartol Filipović, `bartol.filipovic@sit.fraunhofer.de`

[5] PSR, Christoph Ludwig, `cludwig@cdc.informatik.tu-darmstadt.de`

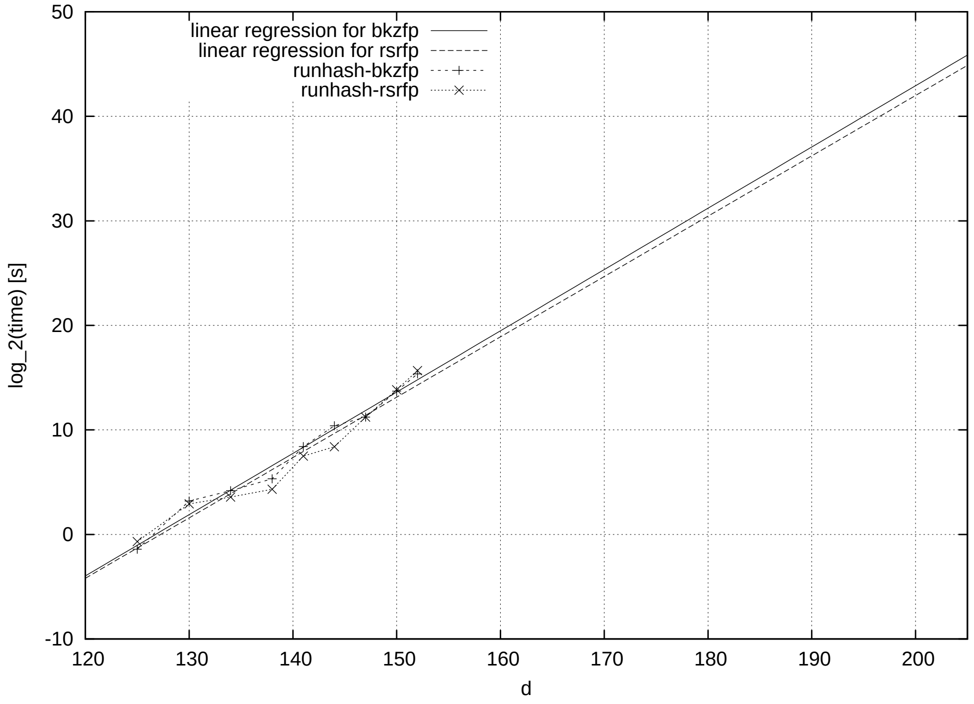

Figure 4.3: Average runtimes of our experiments.

4.3). The same growth assumption has often been made when analyzing NTRU lattices [HHHGW09]. The resulting approximation for the runtime t of rsrfp, the experimentally faster algorithm, is

$$\log_2(t) = 0.577254 \cdot d - 73.459. \tag{4.5.1}$$

All our experiments were run on a single 2.3 GHz AMD Opteron processor. According to the predictions of Lenstra and Verheul [LV01], the computational hardness of a problem solved after t seconds on such a machine is comparable to breaking a k-bit symmetric cipher, where

$$k = \log_2(t) + \log_2(2300) - \log_2(60 \cdot 60 \cdot 24 \cdot 365.25) - \log_2(5 \cdot 10^5) + 56. \tag{4.5.2}$$

Using the data in Figure 4.3, we can compute the security level k corresponding to the average runtime of each algorithm relative to the dimension d for each parameter set.

As we have seen in Figure 4.1, the rightmost side of Figure 4.3 ($d = 206$) corresponds to a *real* SWIFFT lattice. The extrapolated symmetric bit security for finding pseudo-collisions on these lattices is $k = 68.202$. Inserting Equation (4.5.1) into Equation (4.5.2), we find that any parameter set where $d \geq 260$ would correspond to a cipher with symmetric bit-security at least 100 according to our extrapolation. Parameters realizing this paradigm are given in Section 4.4.1.

5 Lattice-Based Zero-Knowledge Identification

Zero-knowledge identification schemes solve the problem of authenticating one party to another via an insecure channel without disclosing any additional information that might be used by an impersonator. In this chapter we propose a scheme whose security relies on the existence of a commitment scheme and on the hardness of worst-case lattice problems. To this end, we adapt a code-based identification scheme by Cayrel and Véron which constitutes an improvement of another code-based scheme by Stern. Our solution sports analogous improvements over the lattice adaption of Stern's scheme which Kawachi *et al.* presented at ASIACRYPT 2008 [KTX08]. Specifically, due to a smaller cheating probability close to 1/2 and a similar communication cost, any desired level of security will be achieved in fewer rounds. Compared to Lyubashevsky's scheme presented at ASIACRYPT 2009 [Lyu09], our proposal, like Kawachi's, offers a milder security assumption: namely, the hardness of (Ideal)SIS for trinary solutions. The same assumption was used for the SWIFFT hash function, which is secure for much smaller parameters than those required by Lyubashevsky.

This chapter is based on a joint work with Pierre-Louis Cayrel, Markus Rückert, and Rosemberg André da Silva [CLRS10a]. It will be presented in October at the *Fourth International Conference on Provable Security* (ProvSec 2010) in Malacca, Malaysia.

5.1 Introduction

One of the main objectives in cryptography is to provide means of access control, and identification (ID) schemes are typically applied in order to reach this goal. These schemes describe interactive protocols between a designated prover and verifier with the purpose of demonstrating that the prover knows a secret that is associated with his identity. In zero-knowledge schemes, no information about this secret is revealed, except the fact that the prover knows it. Besides, using hard lattice problems as security basis allows for very mild assumptions in the sense that they are worst-case instead of average-case and provide resistance against quantum adversaries.

There is an efficient generic construction due to Fiat and Shamir that transforms any ID scheme into a signature scheme, in the random oracle model [FS86]. Therefore, having an efficient ID solution from lattices gives rise to a similarly efficient signature construction, keeping the same hardness assumption. One of the main hardness assumption for ID schemes based on lattices is the short integer solution (SIS) problem. One is given an average case instance $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $m = \Omega(n \log(n))$, and a norm bound b. Then, the task is to find a nonzero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\mathbf{Av} \equiv \mathbf{0} \pmod{q}$ and $\|\mathbf{v}\|_\infty \leq b$. This is hard to accomplish as long as there is at least one single n-dimensional lattice, where solving the approximate shortest vector problem is hard for approximation factors $\gamma \geq b \cdot \tilde{O}(1)$. Hence, it is desirable to build an ID scheme based on SIS with the least possible norm bound b, which is $b = 1$.

The most relevant ID schemes based on number theoretic problems, e.g., [FS86] and [FFS87], do not resist quantum computer attacks that use Shor's algorithm [Sho94]. One of the first schemes to resist such attacks was proposed by Stern [Ste93]. It relies on the syndrome decoding problem and uses of a 3-pass zero-knowledge proof of knowledge (ZK-PoK) with a soundness error of 2/3 and perfect completeness. Recently, Kawachi, Tanaka, and Xagawa [KTX08] were able to change the security assumption of Stern's scheme to SIS with norm bound 1. With their work, Kawachi et al. provide a more efficient alternative to Lyubashevsky's scheme [Lyu08a, Lyu09], which uses a stronger assumption, namely SIS with norm bound $O(n^2 \log(n))$. In contrast to typical zero-knowledge schemes, Lyubashevsky's construction is based on a witness-indistinguishable (not zero-knowledge) proof of knowledge. Furthermore, it has no soundness error, but instead a completeness error of $1-1/e$, which leads to increased communication costs and the undesirable scenario of having an honest prover being rejected by the verifier.

In code-based cryptography, there is also the scheme proposed by Cayrel and Véron [CV10] that improves the Stern's scheme by reducing the soundness error to $q/(2(q-1)) \approx 1/2$. This improvement leads to lower the communication cost, when comparing both schemes for a given security level. Currently, in terms of efficiency, there is no practical lattice-based construction that is comparable to that put forward by Cayrel and Véron.

We propose such a scheme with a soundness error of $(q+1)/2q \approx 1/2$ and perfect completeness. It is based on the same efficient version of the SIS problem that is used by Kawachi et al. or by the SWIFFT compression function [LMPR08]. Both the small soundness error and the mild assumption make our scheme more efficient than previous lattice-based ones. Moreover, by transferring code-based constructions to lattices, we can exploit efficiency improvements using ideal lattices without losing provable security. As a result, our scheme has smaller public keys and more efficient operations than those associated with the current code-based ID schemes.

For a comparison with the most recent lattice-based ID schemes, see Figure 5.1, which assumes that the parameters listed in Figure 5.4 are used, and that a soundness error of 2^{-16} (one of the values recommended in the norm ISO/IEC 9798) is specified. We computed that Lyubashevky's scheme takes 11 rounds to reach a completeness error below 1%, when it is using the most efficient parameters listed in [Lyu09].

The content of this chapter is organized as follows. We present the concepts that are

Scheme	Secret key [Kbyte]	Public key [Kbyte]	Rounds	Communication [Kbyte]	SIS Norm Bound
Lyubashevsky [Lyu09]	0,25	2,00	11	110,00	$\tilde{O}(n^2)$
Kawachi et al. [KTX08]	0,25	0,06	27	58,67	1
Section 5.3	0,25	0,06	17	37,50	1

Figure 5.1: Comparison of lattice-based identification schemes.

used in the construction of the identification scheme in Section 5.2. Later, in Section 5.3, we describe our proposed identification scheme as well as those from which it is derived, namely the original identification schemes by Stern, and an improved code-based variant by Cayrel and Véron. Afterwards, We give a detailed description of the algorithms that comprise the new scheme, and discuss the decisions that were made from a performance and security point of view in Section 5.4. Then, we provide a comprehensive analysis of both theoretical and practical security, and propose concrete parameters for our scheme in Section 5.4.

5.2 Setup

We generally follow the security notions and notations put forward by Kawachi *et al.* in [KTX08]. They give a well-founded introduction to the field of zero-knowledge identification and also present a lattice-based scheme. In this section, we give quick overview of some important concepts.

String commitment schemes. A string commitment scheme is a protocol between two parties, sender and receiver. It works as follows: Both parties agree on a deterministic commitment function Com from a suitable family. This can be be realized, e.g., with a trusted third party. The scheme runs in two phases named committing and revealing.

In the commitment phase, the sender commits to a string s by choosing string ρ uniformly at random and computing $c \longleftarrow \mathsf{Com}(s, \rho)$, which he sends to the receiver. In the revealing phase, which may take place much later, the sender sends both the string s and his chosen randomness ρ to the receiver. Then the receiver checks if $c = \mathsf{Com}(s, \rho)$ holds.

For our main protocol, we will use a commitment scheme which is secure in the sense that it is both hiding and binding. Informally, we say the scheme is statistically hiding, if a computationally unbounded attacking receiver has no noticeable advantage when assigning two commitments c, c' to their respective strings s, s' correctly. We say the scheme is computationally binding, if an attacking sender running in polynomial-time cannot change the commitment c to another value which passes the check in the revealing phase. Refer to e.g., [HM96] for a formal definition.

Security Model. In our model, we employ a string commitment scheme in the trusted setup model, according to which a trusted party honestly sets up the system parameters for the sender and the receiver.

We will show that our scheme is secure against, impersonation under concurrent attacks. This implies that we allow the adversary to play the role of a cheating verifier prior to impersonation, possibly interacting with many different prover clones concurrently. Such clones share the same secret key, but have independent coins and keep their own state. As stated in [BP02], security against this kind of attack implies security against impersonation under active attacks.

Throughout this chapter, we will use the concept of zero-knowledge interactive proof of knowledge systems. In this context, an entity called prover P has as goal to convince a probabilistic polynomial-time (PPT) verifier V that a given string x belongs to a language L, without revealing any more information.

This kind of proof system must satisfy three properties:

Completeness. For any string x belonging to the language, the prover can successfully convince the verifier. That is,

$$\forall x \in L \;\; \Pr\left[(P,V)\,[x] = \mathrm{YES}\right] \geq 1 - \mathrm{negligible}(k).$$

Here, (P,V) denotes the protocol describing the interaction between prover and verifier, and negligible(k) is a negligible function on some security parameter k.

Soundness. For any string x not in the language, any malicious PPT prover cannot convince the verifier with probability better than 1/2. That is,

$$\forall x \notin L \; \forall P' \;\; \Pr\left[(P',V)\,[x] = \mathrm{YES}\right] \leq 1/2.$$

Zero-knowledge. Anything one could learn by listening to P, one could also have simulated by oneself. That is, $\forall V'_{PPT} \; \exists S_{PPT} \; \forall x \in L \; \mathrm{VIEW}_{P,V'}(x)$ close to $S(x)$. Here, VIEW represents the distribution of the transcript of the communication between prover and verifier, and $S(x)$ represents the distribution of the simulation of such interaction. Depending on the proximity of $\mathrm{VIEW}_{P,V'}(x)$ and $S(x)$, as defined in [GMR85], one can have:

- Perfect Zero-knowledge, if the distributions produced by the simulator and the proof protocol are exactly the same.
- Statistical Zero-knowledge, if the statistical distance between the distributions produced by the simulator and the proof protocol is a negligible function.
- Computational Zero-knowledge, if the distributions produced by the simulator and the proof protocol are indistinguishable to any efficient algorithm.

5.3 Identification Schemes

An identification scheme is a collection of algorithms (Setup, KeyGen, Prover, Verifier) meant to provide a proof of identity. The Setup algorithm takes as input a security parameter and generates structures (such as a lattice or code basis) to be used by the other algorithms. The KeyGen algorithm takes as input the parameters generated by Setup and derives key pairs (private, public) to be associated with a set of users. The

Prover and Verifier algorithms correspond to a protocol that is executed by entities P and V, respectively, such that the former convinces the latter of its identity's authenticity, by proving to know the solution to a hard problem, which establishes the relation between the components of P's key pair.

Stern's Identification Scheme. The first practical code-based identification scheme was proposed by Stern [Ste93]. Its basic algorithm uses a hash function h, a pair of keys $(\mathbf{i}, \mathbf{s})$ related by $\mathbf{i} = \mathbf{H}^T\mathbf{s}$, where $\mathbf{H}$ is a public parity check matrix of a given code, $\mathbf{s}$ is a private binary vector of Hamming weight p, and $\mathbf{i}$ is its public syndrome. In a given round, $\mathbf{y}$ is chosen uniformly at random from the same space as $\mathbf{s}$, a permutation σ of the integers $\{1, \dots, \dim(\mathbf{y})\}$ is similarly chosen, and the commitments are calculated by the prover as follows

$$\begin{aligned} c_1 = \quad & h(\sigma \| \mathbf{H}^T\mathbf{y}) \\ c_2 = \quad & h(\sigma(\mathbf{y})) \\ c_3 = \quad & h(\sigma(\mathbf{y} \oplus \mathbf{s})). \end{aligned}$$

Upon receipt of a challenge b chosen uniformly at random from $\{0, 1, 2\}$, the prover reveals the information that enables the verifier to check the correctness of the commitments as below:

$$\begin{aligned} b = 0: \quad & \text{Reveal } \mathbf{y} \text{ and } \sigma. \text{ Check } c_1 \text{ and } c_2. \\ b = 1: \quad & \text{Reveal } \mathbf{y} \oplus \mathbf{s} \text{ and } \sigma. \text{ Check } c_1 \text{ and } c_3. \\ b = 2: \quad & \text{Reveal } \sigma(\mathbf{y}) \text{ and } \sigma(\mathbf{s}). \text{ Check } c_2,\ c_3, \text{ and } \mathrm{wt}(\sigma(\mathbf{s})) = p \end{aligned}$$

This scheme has a soundness error of 2/3. In order to reach a confidence level L on the authenticity of the prover, it has to be repeated a number r of times, so that $1 - (2/3)^r \geq L$.

In the same work, Stern also proposed a few variants of the basic scheme focusing on specific goals, such as: minimize computing load, minimize number of rounds, apply identity-based construction, and employ an analogy of modular knapsacks. For the minimization of number of rounds, he suggested the following solution:

1. The private key $\mathbf{s}$ is replaced by the generators $\{\mathbf{s}_1, \dots, \mathbf{s}_m\}$ of a simplex code.

2. Two commitments $c_1 = h(\sigma \| \mathbf{H}^T\mathbf{y})$ and $c_2 = h\big(\sigma(\mathbf{y}) \| \sigma(\mathbf{s}_1) \| \dots \| \sigma(\mathbf{s}_n)\big)$ are used.

3. The prover computes $z = \sigma(\mathbf{y} \oplus \bigoplus_{j=1}^{m} b_j \mathbf{s}_j)$ using a binary vector $\{b_1, \dots, b_m\}$ received from the verifier.

4. Upon challenge 0, the prover reveals σ, and the verifier checks c_1.

5. Upon challenge 1, the prover discloses $\{\sigma(\mathbf{s}_1), \dots, \sigma(\mathbf{s}_m)\}$, and the verifier checks that c_2 is correct and that the code generated by $\{\mathbf{s}_1, \dots, \mathbf{s}_m\}$ is simplex with the required weight.

This solution replaces the 3-pass approach by a 5-pass one, but it is not effective as far as communication costs are regarded. A more efficient solution is shown in the following paragraph. It also corresponds to the underlying approach for our lattice-based solution.

Cayrel and Véron's Identification Scheme. The identification scheme proposed by Stern [Ste93] was based on the hardness of the syndrome decoding problem. An improvement over this scheme, using the dual construction, was proposed by Véron [Vér96], achieving lower communication costs and better efficiency. Like the basic Stern construct, however, a dishonest prover can have success with probability up to 2/3 in any given round.

By modifying the way the commitments are computed, incorporating a value chosen at random by the verifier, Cayrel and Véron [CV10] were able to bound the cheating probability within a given round close to 1/2, with similar communication costs. The approach followed will be outlined later for the case of our scheme in Algorithm 5.3, where the syndrome decoding problem is replaced by the shortest vector problem as hardness assumption. It involves a 5-pass solution, similar to Stern's construction. However, it avoids the heavy payload associated with transmitting the whole basis of a code (or of a lattice).

Another scheme suggested by Gaborit requires smaller storage for public data [GG07]. Given that the schemes we have seen are dealing with codes, this usually implies that a generator matrix or a parity check matrix is needed to fully characterize them. The idea applied by Gaborit was to use double-circulant matrices for a compact representation.

In our work, we point out that a combination of these two approaches can be used in the lattice context, namely ideal lattices. These allow a very compact representation, as efficient as double-circulant matrices, for an identification scheme structure with soundness error close to 1/2. With this, we manage to have the lowest communication costs and lowest public data storage needs.

Taking Cayrel and Véron's basic idea as basis and changing the main security assumption from the syndrome decoding problem (code-based) to the short integer solution problem for ideal lattices (IdealSIS), we obtain a new identification scheme. The transformation is non-trivial, since low-weight codewords that are required in one setting are not necessarily short vectors as required in the other and vice versa.

We begin by describing the new identification scheme and then give arguments regarding all major properties such as completeness, soundness, and zero-knowledge as well as performance.

Our Identification Scheme. The scheme consists of two main parts: a key generation algorithm (Figure 5.2) and an interactive identification protocol (Figure 5.3).

The key generation algorithm receives as input parameters (n, m, q), e.g., $(64, 2048, 257)$. In Section 5.4.2 we will argue that this is a sensible choice. It chooses a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ uniformly at random and selects as private key a binary vector $\mathbf{x} \in \{0,1\}^m$ of Hamming weight $m/2$. The public key consists of an n-dimensional vector $\mathbf{y} = \mathbf{A}\mathbf{x} \bmod q$, the random matrix $\mathbf{A}$, and a commitment function Com. To instantiate the algorithm, we need to select a family of statistically hiding and computationally binding commitment functions $\mathcal{F}$.

For the time being we recommend the commitment functions used by Kawachi *et al.* since they merely require a lattice-based collision resistant, regular hash function, in our case SWIFFT, which allows us to have a single security assumption. The commitment functions Com that we use are deterministic algorithms, which get as second input a nonce r that is

```
KeyGen:
    $\mathbf{x} \xleftarrow{\$} \{0,1\}^m$, s.t. $\mathrm{wt}(\mathbf{x}) = m/2$
    $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$
    $\mathbf{y} \longleftarrow \mathbf{A}\mathbf{x} \bmod q$
    $\mathsf{Com} \xleftarrow{\$} \mathcal{F}$, suitable family of commitment functions
    **Output** $(sk, pk) = (\mathbf{x}, (\mathbf{y}, \mathbf{A}, \mathsf{Com}))$
```

Figure 5.2: Key generation algorithm, parameters n, m, q are public.

Prover $\mathcal{P}(sk, pk)$		Verifier $\mathcal{V}(pk)$
	$(sk, pk) = (\mathbf{x}, (\mathbf{y}, \mathbf{A}, \mathsf{Com})) \longleftarrow \mathsf{KeyGen}$	
$\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, \sigma \xleftarrow{\$} S_m, \mathbf{z} \longleftarrow \mathbf{P}_\sigma \mathbf{x}$		
$\mathbf{r}_0 \xleftarrow{\$} \{0,1\}^n, \mathbf{r}_1 \xleftarrow{\$} \{0,1\}^n$		
$c_0 \longleftarrow \mathsf{Com}(\sigma \parallel \mathbf{A}\mathbf{u}; \mathbf{r}_0)$		
$c_1 \longleftarrow \mathsf{Com}(\mathbf{z} \parallel \mathbf{P}_\sigma \mathbf{u}; \mathbf{r}_1)$	$\xrightarrow{c_0, c_1}$	
	$\xleftarrow{\alpha}$	$\alpha \xleftarrow{\$} \mathbb{Z}_q$
$\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u} + \alpha \mathbf{x})$	$\xrightarrow{\beta}$	
	$\xleftarrow{\text{Challenge } b}$	$b \xleftarrow{\$} \{0,1\}$
If $b = 0$:	$\xrightarrow{\sigma, \mathbf{r}_0}$	**Check** $c_0 \overset{?}{=} \mathsf{Com}(\sigma \parallel \mathbf{A}\mathbf{P}_\sigma^{-1}\beta - \alpha \mathbf{y}; \mathbf{r}_0)$ $\sigma \overset{?}{\in} S_m$
Else:	$\xrightarrow{\mathbf{z}, \mathbf{r}_1}$	**Check** $c_1 \overset{?}{=} \mathsf{Com}(\mathbf{z} \parallel \beta - \alpha \mathbf{z}; \mathbf{r}_1)$ $\mathbf{z} \overset{?}{\in} \{0,1\}^m, \mathrm{wt}(\mathbf{z}) \overset{?}{=} m/2$

Figure 5.3: Identification protocol

assumed to be chosen uniformly at random from a set big enough to guarantee the hiding property of the commitment.

The identification protocol in Figure 5.3 describes the interaction between prover and verifier in order to convince the second party about the identity of the first. All computation in the protocol is performed modulo q, and we use the following notations. The set of all permutations on m elements is S_m. Any permutation $\sigma \in S_m$ is a linear operation and the associated $m \times m$ binary matrix is $\mathbf{P}_\sigma$.

The protocol is an adaptation of the code-based identification scheme [CV10] which represents a major improvement to Véron's [Vér96] and Stern's [Ste93] schemes. In the same way our protocol represents an improvement over the lattice adaptations of Stern's scheme by Kawachi *et al.* [KTX08]. Like Kawachi's, our adaptation to the lattice setting is non-trivial, since we need to ensure that a binary secret key is used (regardless of the

Hamming weight). This needs to be guaranteed throughout the protocol, which entails some change in the β that is used. Similarly to the coding-based scheme, a cheating prover, not knowing the secret key, can lead a verifier to believe that he actually knows that secret value with a probability up to 1/2 in an individual round of execution. Therefore, in order to diminish the success rate of such an impersonation, the protocol has to be repeated a number of times, which is a function of the degree of confidence requested by the application that is using the scheme. This will be discussed further in Section 5.4, where we argue the soundness.

In the commitment phase, the prover commits to two values c_0, c_1, where c_0 is comprised of the random choices he made and c_1 contains information about his secret key. An adversary that can also correctly compute them with overwhelming probability either is able to break the commitment or to solve the hard problem that makes it possible to obtain a private key from its public counterpart. Those commitments are sent to the verifier, who responds in the second phase with value α taken uniformly at random from $\mathbb{Z}_q$. Upon receipt of the this value, the prover is supposed to multiply it by the private key, add to a permuted masking value u (uniformly chosen at random from $\mathbb{Z}_q^m$) and make a permutation over the sum. Since $\mathbf{u}$ was random, β can be seen as a random variable with uniform distribution over $\mathbb{Z}_q^m$, leaking no information about the private key x.

Upon receipt of this value, the verifier makes a challenge to the prover, picking a value uniformly at random from the set $\{0, 1\}$. The prover responds to it by revealing some piece of information that allows the verifier to compute and check the commitments. An honest prover will always be able to respond to either challenge. Besides checking the correctness of the commitments, the verifier must also check that the values disclosed by the prover are well-formed, although in practice this would be solved by defining a suitable encoding for the data.

We will see in Section 5.4.1 how an impersonator can always cheat with a success probability of 1/2, and that no better strategy is possible under our hardness assumptions. So in order to reach a prescribed level of security, the interaction proposed here must be repeated an appropriate number of times.

Ideal lattices. The present construction makes no assumptions about the structure of the SIS matrix $\mathbf{A}$. Therefore, the space necessary for storing this matrix is $\tilde{O}(n^2)$, which is too big for practical purposes. Using ideal lattices, one can reduce such space requirements to $\tilde{O}(n)$ and simultaneously increase computation speed of matrix vector products in the form $\mathbf{Ax}$ to $\tilde{O}(n)$ operations. This has been proposed and performed many times, perhaps most elegantly in the case of the SWIFFT compression function [LMPR08].

Signature via Fiat-Shamir heuristics. If the verifier is replaced by a random oracle, one can derive signature schemes from identification counterparts. As pointed out by Lyubashevsky when comparing his lattice-based identification scheme [Lyu09] with Kawachi's solution [KTX08], the latter does not result in an efficient signature scheme due to the fact that every bit of the challenge (thus, each bit of a message digest when we consider a signature application) results in a reasonable amount of data sent by the prover. For a 240-bit message digest, for example, Kawachi's scheme would result in a signature of over two million bits when applying Fiat-Shamir heuristics.

Our identification scheme, however, has some characteristics of Lyubashevsky's, in the sense that we can relate a message digest with the variable α that the verifier sends to the prover in "pass 2" of Algorithm 5.3, instead of doing it with the challenge bits. Thus, we can make the field from which such a variable is defined to have a width that better suits the signature scheme needs, circumventing the drawback pointed out above. At the same time, we need to ensure that the total number of rounds we run the scheme is bigger than the desired bit-security level of the resulting signature. This is because an attacker who can correctly guess the challenge bits for each round can generate a signature.

Our identification scheme has successfully been used with an extension of the usual Fiat-Shamir transform to derive a threshold ring signature scheme [CLRS10b].

5.4 Security Analysis

In this section we show that the protocol in Figure 5.3 corresponds to a zero-knowledge interactive proof of knowledge of the predicate defined below. Let $I = \{\mathbf{A}, \mathbf{y}, m, q\}$ be public data shared by the parties A and B. Consider the predicate $P(I, \mathbf{x})$ as "$\mathbf{x}$ is a binary vector of Hamming weight $m/2$ satisfying the equation $\mathbf{Ax} = \mathbf{y} \bmod q$".

We provide below proofs for the completeness, soundness and zero-knowledge properties of the identification scheme described in Figure 5.3. In particular, soundness holds even against concurrent attacks, i.e., an adversary may try to impersonate a given identity after having access to polynomially many verifier instances in parallel. Each of the verifier instances has the same secret key but is run with a different random tape. The challenge is to simulate the environment of the attacker during these interactions *and* still being able to extract "useful" information from the adversary during the impersonation phase. The required assumptions are that Com is a statistically hiding and computationally binding commitment scheme, e.g., based on SIS (cf. [KTX08]), and the hardness of the SIS problem.

Completeness. Given that an honest prover has knowledge of the private key $\mathbf{x}$, the blending mask $\mathbf{u}$ and the permutations $\mathbf{P}_{\boldsymbol{\sigma}}$, he will always be able to derive the commitments c_0 and c_1, and reveal to the verifier the information necessary to verify that they are correct. He can also show that the private key in his possession has the appropriate Hamming weight. So the verifier will always accept the honest prover's identity in any given round. This implies perfect completeness.

Zero-Knowledge. We give a demonstration of the zero-knowledge property for the identification protocol shown in Figure 5.3. Here, we require the commitment function Com to be statistically hiding, i.e., $\mathsf{Com}(x; r)$ is indistinguishable from uniform for a uniform $r \in \{0, 1\}^n$.

Theorem 5.4.1. *Let q be prime. The described protocol is a statistically zero-knowledge proof of knowledge if the employed commitment scheme is statistically-hiding.*

Proof. To prove the zero-knowledge property of our protocol, we construct a simulator S that outputs a protocol view $\text{VIEW} = (c_0, c_1, \alpha, \beta, b, (\sigma, r_0), (\mathbf{z}, r_1))$ without knowing the secret $\mathbf{x}$, such that VIEW is indistinguishable from an interaction of the honest prover and verifier. It has access to a cheating verifier V^*, which contributes α and b. Therefore, S generates r_1, r_2 according to protocol and it gets $(\mathbf{A}, \mathbf{y}, \mathsf{Com})$ as input. The simulator has to guess b before talking to V^*. For the moment, let us assume the guess is correct.

If $b = 0$, the simulator selects $\mathbf{u}$ and σ according to protocol and solves $\mathbf{Ax} \equiv \mathbf{y} \pmod q$ for $\mathbf{x}$, which does not need to be short. With this pseudo secret key, the simulator computes c_0 and c_1 according to the protocol. The deviation in c_1 is not recognized because Com is statistically hiding. Then, S computes $\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u}+\alpha\mathbf{x})$ after obtaining α from $V^*(c_1, c_2)$. The result is uniform because $\mathbf{u}$ is chosen uniformly at random. As a result, S can reveal (σ, r_0), which passes the verification for $b = 0$.

If $b = 1$, the simulator needs to play against the second verification branch. It selects a binary $\mathbf{x}$ with Hamming weight $m/2$ and selects σ as per protocol. It computes c_1, c_2 and obtains $\alpha \longleftarrow V^*(c_1, c_2)$. Then, it computes $\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u} + \alpha\mathbf{x})$. As a result, S can reveal $\mathbf{P}_\sigma\mathbf{x}$ that passes verification.

In consequence, the simulator outputs a correct view with probability 1/2. Since the simulator has access to V^*, it can restart the verifier whenever the guess b was incorrect. The result is a statistically close simulation if Com is statistically hiding. □

Soundness. We now show that a dishonest prover is able to cheat a verifier to accept his identity with a probability limited by $(q + 1)/2q \approx 1/2$. The number of possible queries sent by the verifier to a prover is given by all combinations of challenge bits $b \in \{0, 1\}$ and $\alpha \in \{0, \ldots, q - 1\}$ Hence, there are $2q$ possible queries. Say the dishonest prover wants to answer all challenges where $b = 0$, then he computes an alternate secret key $\mathbf{x}'$ with large entries such that $\mathbf{Ax}' = \mathbf{y}$. This can be done with Gaussian elimination, for example. At the same time, when $\alpha = 0$ he can also answer in the case $b = 1$ by sending a random $\mathbf{z}$. Since $\alpha = 0$ this is not checked in the commitment.

Note that the $\alpha = 0$ query issue cannot be resolved by removing 0 from the set that α is drawn from, because the dishonest verifier can effectively shift the values of α by changing his protocol. Say he wants some fix α_0 to take the place of 0 in the unmodified scheme, then he changes both the computations of the commitments and β to:

$$\begin{aligned} c_0 &\longleftarrow \mathsf{Com}(\sigma \parallel \mathbf{Au} - \alpha_0\mathbf{y}; r_0), & \beta &\longleftarrow \mathbf{P}_\sigma(\mathbf{u} + (\alpha - \alpha_0)\mathbf{x}), \\ c_1 &\longleftarrow \mathsf{Com}(\mathbf{z} \parallel \mathbf{P}_\sigma\mathbf{u} - \alpha_0\mathbf{z}; r_1). \end{aligned}$$

In effect, he can answer both challenge bits $b = 0, 1$ for $\alpha = \alpha_0$ now.

Thus, in total, the adversary can answer correctly for $q + 1$ out of $2q$ queries. In the proof, we show that if an adversary is able to answer more queries, it is also able to break one of the underlying assumptions, i.e., solve SIS or break the commitment.

Theorem 5.4.2. *If an honest verifier accepts a dishonest prover with probability $Pr \geq (q+1)/2q+\epsilon(n)$, with $\epsilon(n)$ non-negligible, then there exists a polynomial time probabilistic machine M which breaks the binding property of the commitment* Com *or solves the* SIS *problem with non-negligible probability.*

Proof. Given the SIS problem instance $(n, m, q, \mathbf{A})$ as input and a challenge commitment function Com, we need to simulate the adversary's environment in two phases: a verification phase and an impersonation phase. In order to correctly prove knowledge of a valid secret key $\mathbf{x}$ during the verification phase, we choose $\mathbf{x}$ and $\mathbf{y}$ as in the key generation protocol and run the adversary $\mathcal{A}$ on public parameters (as per protocol).

Therefore, in the verification phase, we can perfectly simulate the prover. Since the protocol is statistically zero-knowledge, the adversary does not learn any information about $\mathbf{x}$ and the output distribution is the same as for all alternative secret keys $\mathbf{x}' \neq \mathbf{x}$.

After the first phase, we let $\mathcal{A}$ play the role of the cheating prover. First, we receive the commitments c_0, c_1. Then, because q is polynomial in n, we challenge the adversary with all $2q$ challenge pairs (α, b) and record successes as "1" and failures as "0" in a table with column labels "$b = 0$", "$b = 1$" and row labels "$\alpha = 0$",...,"$\alpha = q - 1$". This is done by rewinding the adversary appropriately.

For the moment, let us assume that there exist two rows, for α and α', such that both columns contain "1". Let $(\beta, \sigma, \mathbf{r}_0)$ and $(\beta', \sigma', \mathbf{r}_0')$ be the outcomes for challenge $(\alpha, 0)$ and $(\alpha', 0)$, respectively. Furthermore, let $(\beta, \mathbf{z}, r_1)$ and $(\beta', \mathbf{z}', r_1')$ be the outcomes for challenges $(\alpha, 1)$ respectively $(\alpha', 1)$.

Since the commitment Com is binding, we infer that $r_0 = r_0'$, $r_1 = r_1'$, and

$$\begin{aligned} \sigma \parallel \mathbf{A}\mathbf{P}_\sigma^{-1}\beta - \alpha\mathbf{y} &= \sigma' \parallel \mathbf{A}\mathbf{P}_{\sigma'}^{-1}\beta' - \alpha'\mathbf{y}\,, && (5.4.1) \\ \mathbf{z} \parallel \beta - \alpha\mathbf{z} &= \mathbf{z}' \parallel \beta' - \alpha'\mathbf{z}'\,. && (5.4.2) \end{aligned}$$

Equation 5.4.1 implies $\sigma = \sigma'$. Similarly, Equation 5.4.2 shows that the binary vectors $\mathbf{z}, \mathbf{z}'$ of weight $m/2$ are equal. Now, we turn to extracting $\mathcal{A}$'s secret key by rearranging parts of Equations 5.4.1 and 5.4.2. We get

$$\begin{aligned} \mathbf{A}\mathbf{P}_\sigma^{-1}(\beta - \beta')(\alpha - \alpha')^{-1} &\equiv \mathbf{y} \pmod{q}\,, && (5.4.3) \\ (\beta - \beta')(\alpha - \alpha')^{-1} &\equiv \mathbf{z} \pmod{q}\,. && (5.4.4) \end{aligned}$$

This proves that $\mathbf{x}' := \mathbf{P}_\sigma^{-1}\mathbf{z}$ is a valid secret key and the reduction outputs the short lattice vector $\mathbf{v} = \mathbf{x} - \mathbf{x}'$. Notice that we have $\beta \neq \beta'$ because we have Equation 5.4.1, $\alpha \neq \alpha'$, and $\sigma = \sigma'$. The extracted secret key is also different from the one of the simulator because the function $\mathbf{A}\mathbf{x} \bmod q$ compresses the set of valid secret keys and statistically hides them in the sense that the protocol is also witness indistinguishable. Hence, the adversary cannot learn the simulator's key but with probability $\leq 1/2 + n^{-\omega(1)}$

What is left to show is that such a pair (α, α') exists. To see this, we apply a simple counting argument (cf. [OO98]). We know that $\mathcal{A}$ can answer correctly for $> q + 1$ challenges. W.l.o.g., assume that it succeeds $\geq c$ times for $b = 0$ and $> q + 1 - c$ times for $b = 1$. Thus, there are $\geq c$ many "1" entries in column "$b = 0$" and $> q + 1 - c$ many "1" entries in column "$b = 1$".

By way of contradiction, assume that there is no such pair (α, α') for which $\mathcal{A}$ succeeds for the challenges $(\alpha, 0)$, $(\alpha, 1)$, $(\alpha', 0)$, and $(\alpha', 1)$. In other words, assume that the above extraction procedure breaks down. Then there must be at least $c - 1$ zeros in column "$b = 0$". In consequence, the total number of entries in the second column is $> c-1+q+1-c$. Since this is $> q$, we arrive at the desired contradiction and conclude that the knowledge extractor succeeds with non-negligible probability if $\epsilon(n)$ is non-negligible. □

Given that the scheme is a zero-knowledge proof of knowledge, it is also witness indistinguishable with respect to the secret $\mathbf{x}$. Fortunately, witness-indistinguishability is preserved under parallel composition. Thus, our scheme can be run many, i.e., $\omega(\log(n))$ times in parallel to achieve a negligible soundness error but without increasing the number of rounds.

5.4.1 Practical Attacks

The code-based identification scheme proposed by Cayrel and Véron and that serves as starting point for this work has very good performance characteristics. Its security is based on the assumption that selecting a random generator or parity check matrix will result in hard instances of the q-ary syndrome decoding problem, though. When adapting this scheme to use lattices, on the other hand, one achieves a construct based on the hardness of the SIS problem, and that in turn has a worst-case/average-case reduction.

As pointed out in the description of the algorithms, ideal lattices can also be used in the scheme to improve performance and reduce the amount of public data. The precautions regarding (a) the irreducibility of the polynomial that characterizes the ring upon which the lattice is defined and (b) its expansion factor must be observed, as recommended in [LM06]. This ensures that finding short vectors in such lattice is still hard to perform.

The present scheme is also secure against active attacks. Thus, an attacker is allowed to interact with a prover prior to attempting to impersonate him to a verifier. As consequence of the zero-knowledge property, however, no adversary that interacts with a real prover is able to obtain any knowledge that can be used later on to impersonate the prover.

We now prove that our scheme is secure against concurrent attacks, by showing that a public key corresponds to multiple secret keys and that the protocol is witness indistinguishable. It is a standard procedure, as seen in [FS90].

First, the existence of multiple secret keys associated with a given public key is assured by the parameter choice (see Equation 5.4.5). Second, given that our protocol is a zero-knowledge interactive proof, it is also witness indistinguishable [KP01].

The most efficient way to attack this scheme, but probably the most difficult one, consists in solving the inhomogeneous short integer solution (ISIS) problem that is defined by the public key $\mathbf{y}$ and the public matrix $\mathbf{A}$, expressed as $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$, where $\mathbf{x}$ is expected to be binary, with dimension m and Hamming weight $m/2$. This equation can be re-written as $\mathbf{A}'\mathbf{x}' = 0 \bmod q$, with $\mathbf{A}' = [\mathbf{A}|\mathbf{y}]$ and $\mathbf{x}' = [\mathbf{x}|-1]^T$. Lattice basis calculation and reduction can then be applied in this second lattice to try to find a solution. The approximation factor, however, is $\widetilde{O}(n)$, making the task hard.

5.4.2 Concrete Parameters

In order to guarantee with overwhelming probability that there are other solutions to $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$, besides the private key possessed by the prover (which is pivotal in the demonstration of security against concurrent attacks), one can make q and m satisfy the relation below

$$q^n \ll \left|\{\mathbf{x} \in \mathbb{Z}_2^m : \mathrm{wt}(\mathbf{x}) = m/2\}\right|. \tag{5.4.5}$$

Besides, q is bounded by the following theorem, which Micciancio and Regev proved in [MR04].

Theorem 5.4.3. *For any polynomially bounded functions $\beta(n), m(n), q(n) = n^{O(1)}$, with $q(n) \geq 4\sqrt{m(n)}n^{1.5}\beta(n)$ and $\gamma(n) = 14\pi\sqrt{n}\beta(n)$, there is a probabilistic polynomial time reduction from solving GapCVP_γ in the worst-case to solving $\mathsf{SIS}_{q,m,\gamma}$ on the average with non-negligible probability. In particular, for any $m = \Theta(n \log n)$, there exists $q(n) = O(n^{2.5} \log n)$ and $\gamma = O(n\sqrt{\log n})$, such that solving $\mathsf{SIS}_{q,m}$ on the average is at least as hard as solving GapSVP_γ in the worst case.*

Taking as reference the state-of-the-art lattice reduction algorithms studied in [GN08], the length of the shortest vector that can currently be found by the reduction algorithms is given (for $\delta \approx 1.011$) by:

$$length = \min\{q, q^{n/m}\delta^m\} \tag{5.4.6}$$

Simply choosing the same parameters that were recommended for SWIFFT, our scheme We propose the set of parameters below, in Figure 5.4, which are comparable to those used by the SWIFFT hash function. The best attack on our scheme The best combinatorial attack for finding short lattice vectors [Wag02] has a computational complexity above 2^{100} (generalized birthday attack, dividing in 16 groups at each turn). This means that our security level is 100 bits. In addition to that, the best lattice reduction algorithms return vectors with Euclidean norm above 42, taking into account our set of parameters. Given that the private keys resulting from our parameters have Euclidean norm 32, the choice made is safe. Besides, we can also see that the selected parameters satisfy both Theorem 5.4.3 and the restriction given by Equation 5.4.5.

n	m	q	Commitment Length (bits)
64	2048	257	256

Figure 5.4: Concrete parameters

6 Lattice-Based Encryption

In this chapter, we analyze the concrete security and key sizes of theoretically sound lattice-based encryption schemes based on the "learning with errors" (LWE) problem. Our main contributions are: (1) a new lattice attack on LWE that combines basis reduction with an enumeration algorithm admitting a time/success tradeoff, which performs better than the simple distinguishing attack considered in prior analyses; (2) concrete parameters and security estimates for an LWE-based cryptosystem that is more compact and efficient than the well-known schemes from the literature. Our new key sizes are up to 10 times smaller than prior examples, while providing even stronger concrete security levels.

This chapter is based on a joint work with Chris Peikert [LP11]. It was accepted for the *Cryptographers' Track of the RSA Conference* (CT-RSA) 2011 in San Francisco, USA.

6.1 Introduction

Recent years have seen significant progress in theoretically sound lattice-based cryptography, resulting in solutions to many tasks of wide applicability. In the realm of encryption alone, for example, we now have public-key cryptosystems [AD97, Reg03, Reg05b] with chosen-ciphertext security [PW08, Pei09], identity-based encryption [GPV08, CHKP10, ABB10], and a fully homomorphic cryptosystem [Gen09]. Much of this progress has been greatly aided by the use of simple and flexible average-case problems — namely, the *short integer solution* (SIS) introduced by Ajtai [Ajt96b] and the *learning with errors* (LWE) problem of Regev [Reg05b] — that are provably as hard as certain lattice problems in the *worst case*, and appear to require time exponential in the main security parameter to solve.

For *practical* parameters, however, the concrete hardness of the SIS and LWE problems against algorithmic attacks is still far from a settled issue. This makes it difficult to assess the actual security and efficiency of cryptographic schemes that are based on these problems. The purpose of this chapter is to shed further light on this issue, by considering new variants of known schemes and attacks, and analyzing their consequences in terms of key sizes and estimated security.

We analyze the concrete security and efficiency of modern lattice-based cryptographic schemes, with a focus on LWE and public-key encryption. To start, we describe an LWE-based cryptosystem that has substantially smaller keys and ciphertexts than the more well-known systems in the literature (namely, the original system of Regev [Reg05b] and its more efficient amortized variants [PVW08, GPV08]). Our scheme incorporates several techniques and perspectives from recent works; in particular, it is an instance of an abstract system described by Micciancio [Mic10] that generalizes all the schemes of [Reg05b, PVW08, GPV08], and the system's design and security proof (under the LWE assumption) combine a variety of techniques from recent works [Ale03, MR09, LPS10, Pei10] to yield asymptotic and concrete improvements in key size. While there are not any new techniques involved, to our knowledge the literature lacks a full description and analysis of the system, despite it now being an important target of study.

Our second main contribution is a new and stronger way of using existing algorithmic attack tools, such as lattice basis reduction and bounded-distance decoding with preprocessing, to analyze the concrete security of recent lattice-based cryptosystems. Our attack is directed specifically at the LWE problem, and exploits some of its structural properties in ways that have not been attempted before in a cryptanalytic context. (Our attack also does not seem immediately applicable to other lattice problems, such as the unique shortest vector problem, that have been used for public-key encryption [AD97, Reg03, AD07].) Therefore, we believe that our analysis gives a more accurate assessment of LWE's concrete hardness than estimates derived from prior lattice attacks.

Applying our attack to the improved cryptosystem, we then propose concrete parameters and (conservative) runtime estimates for modern commodity hardware. Despite our improved attacks, the resulting key sizes are still smaller than prior example parameters by factors as large as 10, even for stronger security levels. (See Section 6.6 for full details.) For example, using parameters that can encrypt a 128-bit payload and appear to be at least as secure as AES-128, we obtain public key sizes of about $1,120$ kilobits, or about 400 kilobits assuming a public source of trusted randomness.

Clearly, the above key sizes are still too large for many applications, but this is a consequence of the quadratic overhead inherent to the use "standard" LWE. By using the compact "ring-based" variant of LWE and cryptosystem from [LPR10] (which is related to the heuristic NTRU scheme [HPS98] and the theoretically sound line of works initiated in [Mic02]), we can immediately shrink the above key sizes by a factor of at least 200. The resulting sizes of 2-5 kilobits are comparable to modern recommendations for RSA, and the cryptosystem itself is many times faster on modern hardware.

6.2 Discrete Gaussians and LWE

In this section, we recall some basic facts about the discrete Gaussian distribution and give a detailed account of its relation to the LWE problem. With this background, we will be ready to discuss LWE-based encryption schemes.

6.2.1 Discrete Gaussians

For a lattice Λ and a positive real $s > 0$, the *discrete Gaussian* distribution $D_{\Lambda,s}$ over Λ with parameter s is the probability distribution having support Λ that assigns a probability proportional to $\exp(-\pi\|\mathbf{x}\|^2/s^2)$ to each $\mathbf{x} \in \Lambda$. For $\Lambda = \mathbb{Z}^n$, it is easy to see (by orthonormality of its standard basis) that the discrete Gaussian $D_{\mathbb{Z}^n,s}$ is simply the product distribution of n independent copies of $D_{\mathbb{Z},s}$. There are efficient algorithms for sampling from a distribution within negligible statistical distance of $D_{\mathbb{Z},s}$, given any $s > 0$. (See, e.g., [GPV08]: for arbitrary s there is a rejection sampling algorithm, and for small s one can compute a close approximation to the cumulative distribution function.).

We will need two tail bounds on discrete Gaussians.

Lemma 6.2.1 ([Ban93, Lemma 1.5])**.** *Let $c \geq 1$ and $C = c \cdot \exp(\frac{1-c^2}{2}) < 1$. Then for any real $s > 0$ and any integer $n \geq 1$, we have*

$$\Pr\left[\|D_{\mathbb{Z}^n,s}\| \geq c \cdot \tfrac{1}{\sqrt{2\pi}} \cdot s\sqrt{n}\right] \leq C^n.$$

Lemma 6.2.2 ([Ban95, Lemma 2.4])**.** *For any real $s > 0$ and $T > 0$, and any $\mathbf{x} \in \mathbb{R}^n$, we have*

$$\Pr\left[|\langle \mathbf{x}, D_{\mathbb{Z}^n,s}\rangle| \geq T \cdot s\|\mathbf{x}\|\right] < 2\exp(-\pi \cdot T^2).$$

6.2.2 Learning with Errors

The *learning with errors* (LWE) problem was introduced by Regev [Reg05b] as a generalization of the well-known ‘learning parity with noise’ problem, to larger moduli. The problem is parameterized by a dimension $n \geq 1$ and an integer modulus $q \geq 2$, as well as an error distribution χ over $\mathbb{Z}$ (or its induced distribution over $\mathbb{Z}_q$). In this work we will be concerned only with discrete Gaussian error distributions $\chi = D_{\mathbb{Z},s}$ over the integers, where $\alpha := s/q \in (0,1)$ is often called the (relative) *error rate*.

For an $\mathbf{s} \in \mathbb{Z}_q^n$, the LWE distribution $A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing a uniformly random $\mathbf{a} \in \mathbb{Z}_q^n$ and error term $e \leftarrow \chi$, and outputting the pair $(\mathbf{a}, t = \langle \mathbf{a}, \mathbf{s}\rangle + e \bmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The *search* version of the LWE problem is, given any desired number of independent samples $(\mathbf{a}_i, t_i) \leftarrow A_{\mathbf{s},\chi}$, to find $\mathbf{s}$. The *decision* version of LWE is to distinguish, with non-negligible advantage, between any desired number of independent samples $(\mathbf{a}_i, t_i) \leftarrow A_{\mathbf{s},\chi}$ (for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$), and the same number of independent samples drawn from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. It is often convenient to write these problems in matrix form as follows: collecting the vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ as the columns of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and the (implicit) error terms $e_i \in \mathbb{Z}$ and values $t_i \in \mathbb{Z}_q$ as the entries of vectors $\mathbf{e} \in \mathbb{Z}^m$, $\mathbf{t} \in \mathbb{Z}_q^m$ respectively, we are given the input

$$\mathbf{A}, \quad \mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e} \bmod q$$

and are asked to find $\mathbf{s}$, or to distinguish the input from a uniformly random $(\mathbf{A}, \mathbf{t})$. The LWE problem may also be viewed as an average-case ‘bounded-distance decoding’ problem on a certain family of lattices: for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define the lattice

$$\Lambda(\mathbf{A}^t) = \{\mathbf{z} \in \mathbb{Z}^m : \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ such that } \mathbf{z} = \mathbf{A}^t\mathbf{s} \bmod q\}.$$

Then the $\mathbf{t}$ component of the LWE input may be seen as a perturbed lattice point in $\Lambda(\mathbf{A}^t)$, to be decoded.

Hardness of LWE. We recall several facts from the literature about the provable hardness of LWE. The first is that for error distribution $\chi = D_{\mathbb{Z},\alpha\cdot q}$ where $\alpha \cdot q \geq 2\sqrt{n}$, the search version of LWE is at least as hard as *quantumly* approximating certain worst-case problems on n-dimensional lattices to within $\tilde{O}(n/\alpha)$ factors [Reg05b].[1] Moreover, for similar parameters and large enough q, search-LWE is at least as hard as *classically* approximating the decision shortest vector problem and variants [Pei09]. For moduli q that are sufficiently 'smooth' (i.e., products of small enough primes), the decision form of LWE is at least as hard as the search form [Reg05b, Pei09].

A particularly important fact for our purposes is that decision-LWE becomes no easier to solve even if the secret $\mathbf{s}$ is chosen from the error distribution χ, rather than uniformly at random [MR09, ACPS09]. This may be seen as follows: given access to $A_{\mathbf{s},\chi}$, we can draw many samples to obtain

$$\mathbf{A}^t = \begin{bmatrix} \mathbf{A}_1^t \\ \mathbf{A}_2^t \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^t \\ \mathbf{A}_2^t \end{bmatrix} \mathbf{s} + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix} = \mathbf{A}^t\mathbf{s} + \mathbf{e} \bmod q,$$

where $\mathbf{A}_2$ is uniform, $\mathbf{e}$ is drawn from χ, and $\mathbf{A}_1 \in \mathbb{Z}_q^{n\times n}$ is *square* and *invertible.* (This follows by forming $\mathbf{A}_1$ by greedily drawing samples that can form an invertible matrix, and disposing of any others until $\mathbf{A}_1$ is complete.) We can then transform $\mathbf{A}$ and $\mathbf{t}$ into

$$\bar{\mathbf{A}}^t := -\mathbf{A}_2^t \cdot \mathbf{A}_1^{-t} \bmod q, \quad \bar{\mathbf{t}} := \bar{\mathbf{A}}^t\mathbf{t}_1 + \mathbf{t}_2 = \bar{\mathbf{A}}^t\mathbf{e}_1 + \mathbf{e}_2 \bmod q,$$

where $\bar{\mathbf{A}}$ is uniform; therefore, we have effectively replaced $\mathbf{s}$ with the error vector $\mathbf{e}_1$. On the other hand, note that when $\mathbf{A}, \mathbf{t}$ are uniformly random, then so are $\bar{\mathbf{A}}, \bar{\mathbf{t}}$.

In terms of lattices, the above may be interpreted as follows: using the bijection $\mathbf{s} \mapsto \mathbf{A}_1^t\mathbf{s}$ from $\mathbb{Z}_q^n$ to itself, we can see that the lattice $\Lambda(\mathbf{A}^t)$ defined above has as a basis the matrix

$$\mathbf{H} = \begin{bmatrix} & \mathbf{I} \\ q\mathbf{I} & -\bar{\mathbf{A}}^t \end{bmatrix}.$$

(This basis $\mathbf{H}$ is a canonical representation of $\Lambda(\mathbf{A}^t)$ known as the Hermite normal form. We have ordered the basis vectors so that the Gram-Schmidt vectors of $\mathbf{H}$ are integer multiples of the standard basis vectors, where the first several have length q, and the remainder have length 1.) Because $\mathbf{A}^t\mathbf{s} \bmod q \in \Lambda(\mathbf{A}^t)$, we have $\mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e} = \mathbf{e} \bmod \mathbf{H}$, which is

$$\mathbf{e} - \mathbf{H}\mathbf{e}_1 = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_2 + \bar{\mathbf{A}}^t\mathbf{e}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{t}} \end{bmatrix} \bmod q.$$

In conclusion, $\bar{\mathbf{t}} = \bar{\mathbf{A}}^t\mathbf{e}_1 + \mathbf{e}_2$ is the unique canonical representative of $\mathbf{e}$ modulo the lattice $\Lambda(\mathbf{A}^t)$. Finally, assuming hardness of decision-LWE, a standard hybrid argument over the columns of $\mathbf{E}$ (see, e.g., [PW08]) shows that $(\bar{\mathbf{A}}, \bar{\mathbf{A}}^t\mathbf{E}_1 + \mathbf{E}_2)$ is indistinguishable from uniform, where the entries of $\mathbf{E} = \left[\begin{smallmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \end{smallmatrix}\right]$ are chosen independently from χ.

[1] It is important to note that the original hardness result of [Reg05b] is for a *continuous* Gaussian error distribution, which when rounded naively to the nearest integer does not produce a true discrete Gaussian. Fortunately, a suitable randomized rounding method does so [Pei10].

6.3 LWE-Based Encryption

Here we describe an LWE-based cryptosystem that is more space-efficient than the ones commonly known in the literature. It is an instance of an abstract system described by Micciancio [Mic10] that generalizes all the schemes of [Reg05b, PVW08, GPV08], though a full description and analysis of the generalized system has not appeared in the literature. The security proof combines a number of techniques and perspectives from recent works [MR09, LPS10, Pei10] for the purpose of improved efficiency and a tight analysis. For completeness, we also briefly describe an efficient ring-based analogue of the system, which is described in full generality in the full version of [LPR10].

Despite being a generalization of prior LWE-based cryptosystems, the present scheme can actually be instantiated to have keys and ciphertexts that are smaller by a factor of about $\lg q$, while simultaneously *improving* the concrete security! The improved security comes from the smaller keys (for given security parameter n), which allows for a relatively larger noise rate that makes the LWE problem harder. The smaller keys come from a different style of security proof, which is very similar to the proofs for the coding-based cryptosystem of Alekhnovich [Ale03] and the subset sum-based cryptosystem of Lyubashevsky, Palacio, and Segev [LPS10]. In brief, the proof uses the LWE assumption *twice* (first on the public key, and then again on the ciphertext) to show that the adversary's view in a passive attack is indistinguishable from uniformly random. By contrast, the proofs for prior LWE-based schemes involve a statistical argument on either the public key or ciphertext, but this requires larger keys. We point out that statistical arguments still appear necessary for many advanced applications of LWE, such as identity-based encryption [GPV08] and others that use a 'trapdoor basis,' and we do not know whether comparably small keys and ciphertexts can be obtained for these schemes.

6.3.1 Cryptosystem

The cryptosystem involves a few parameters: an integer modulus $q \geq 2$ and integer dimensions $n_1, n_2 \geq 1$, which relate to the underlying LWE problems; Gaussian parameters s_k and s_e for key generation and encryption, respectively; and a message alphabet Σ (for example, $\Sigma = \{0, 1\}$) and message length $\ell \geq 1$.

We also require a simple error-tolerant encoder and decoder, given by functions

$$\mathsf{encode}\colon \Sigma \to \mathbb{Z}_q \text{ and } \mathsf{decode}\colon \mathbb{Z}_q \to \Sigma,$$

such that for some large enough threshold $t \geq 1$, $\mathsf{decode}(\mathsf{encode}(m)+e \bmod q) = m$ for any integer $e \in [-t, t)$. For example, if $\Sigma = \{0, 1\}$, then we can define $\mathsf{encode}(m) := m \cdot \lfloor \frac{q}{2} \rfloor$, and $\mathsf{decode}(\bar{m}) := 0$ if $\bar{m} \in \big[-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor\big) \subset \mathbb{Z}_q$, and 1 otherwise. This method has error tolerance $t = \lfloor \frac{q}{4} \rfloor$. We also extend encode and decode to vectors, component-wise.

To get the smallest public keys, our system makes use of a uniformly random public matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n_1 \times n_2}$ that is generated by a trusted source, and is used by all parties in the system. If there is no trusted source, then $\bar{\mathbf{A}}$ may be chosen by the user herself as part of key generation, and included in the public key.

- $\mathsf{Gen}(\bar{\mathbf{A}}, 1^\ell)$: choose $\mathbf{R}_1 \leftarrow D_{\mathbb{Z},s_k}^{n_1 \times \ell}$ and $\mathbf{R}_2 \leftarrow D_{\mathbb{Z},s_k}^{n_2 \times \ell}$, and let $\mathbf{P} = \mathbf{R}_1 - \bar{\mathbf{A}} \cdot \mathbf{R}_2 \in \mathbb{Z}_q^{n_1 \times \ell}$. The public key is $\mathbf{P}$ (and $\bar{\mathbf{A}}$, if needed), and the secret key is $\mathbf{R}_2$.

In matrix form, the relationship between the public and secret keys is:

$$\begin{bmatrix} \bar{\mathbf{A}} & \mathbf{P} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix} = \mathbf{R}_1 \bmod q. \tag{6.3.1}$$

- $\mathsf{Enc}(\bar{\mathbf{A}}, \mathbf{P}, \mathbf{m} \in \Sigma^\ell)$: choose $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in \mathbb{Z}^{n_1} \times \mathbb{Z}^{n_2} \times \mathbb{Z}^\ell$ with each entry drawn independently from $D_{\mathbb{Z},s_e}$. Let $\bar{\mathbf{m}} = \mathsf{encode}(\mathbf{m}) \in \mathbb{Z}_q^\ell$, and compute the ciphertext

$$\mathbf{c}^t = \begin{bmatrix} \mathbf{c}_1^t & \mathbf{c}_2^t \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1^t & \mathbf{e}_2^t & \mathbf{e}_3^t + \bar{\mathbf{m}}^t \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathbf{A}} & \mathbf{P} \\ \mathbf{I} & \\ & \mathbf{I} \end{bmatrix} \in \mathbb{Z}_q^{1 \times (n_2 + \ell)}. \tag{6.3.2}$$

  (Note that the first ciphertext component $\mathbf{c}_1^t$ can be precomputed before $\mathbf{P}$ and $\mathbf{m}$ are known.)

- $\mathsf{Dec}(\mathbf{c}^t = [\mathbf{c}_1^t, \mathbf{c}_2^t], \mathbf{R}_2)$: output $\mathsf{decode}(\mathbf{c}_1^t \cdot \mathbf{R}_2 + \mathbf{c}_2^t)^t \in \Sigma^\ell$.

  Using Equation (6.3.2) followed by Equation (6.3.1), we are applying decode to

$$\begin{bmatrix} \mathbf{c}_1^t & \mathbf{c}_2^t \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix} = (\mathbf{e}^t + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \bar{\mathbf{m}}^t \end{bmatrix}) \cdot \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix} = \mathbf{e}^t \cdot \mathbf{R} + \bar{\mathbf{m}}^t,$$

  where $\mathbf{R} = \left[\begin{smallmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I} \end{smallmatrix}\right]$. Therefore, decryption will be correct as long as each $|\langle \mathbf{e}, \mathbf{r}_j \rangle| < t$, the error threshold of decode. (We give a formal analysis in Section 6.3.2 below.)

For another perspective on this scheme as an (approximate) key-agreement mechanism, let $\ell = 1$ for simplicity. By the discussion in Section 6.2.2, we can interpret key generation as reducing a Gaussian error vector $\mathbf{r}$ modulo a lattice defined by $\bar{\mathbf{A}}$, and publishing the result $\bar{\mathbf{A}}\mathbf{r}_2 - \mathbf{r}_1 \bmod q$. Likewise, we can view encryption as reducing a Gaussian error vector $\mathbf{e}$ modulo the *dual* of the same lattice, and publishing the result $\mathbf{e}_1^t \bar{\mathbf{A}} + \mathbf{e}_2^t \bmod q$. Using their respective private error vectors and the other party's public message, the sender and receiver can both (approximately) compute $\mathbf{e}_1^t \bar{\mathbf{A}} \mathbf{r}_2 \in \mathbb{Z}_q$, whereas a passive adversary cannot. A formal proof of security appears below in Section 6.3.3.

Ring-based analogue. We briefly describe a very similar scheme that is based on the decision *ring*-LWE problem [LPR10]. For messages of length any $\ell \leq n = n_1 = n_2$, and using the same values of n and q as above, the public and secret keys are up to an n factor smaller than in the above system, namely $n \lg q$ or $2n \lg q$ bits at most, depending on the availability of a common trusted string. (The ciphertext size is the same, namely $2n \lg q$ bits.)

Let $R = \mathbb{Z}[x]/f(x)$ be a polynomial ring for some monic polynomial $f(x)$ that is irreducible over $\mathbb{Z}$; common choices include *cyclotomic* polynomials such as $f(x) = x^n + 1$ for n a power of 2. (See [LPR10] for efficiency and security properties of this and other cyclotomic polynomials, including degrees n that are not powers of 2.) Let $q \in \mathbb{Z}$ be a sufficiently large integer modulus for which $f(x)$ splits into linear (or very low-degree) factors modulo q, and let $R_q = R/q = \mathbb{Z}_q[x]/f(x)$. Let χ_k, χ_e be error distributions over R that are concentrated on 'small' elements of R; see [LPR10] for what error distributions enable rigorous security proofs.

Let Σ be a message alphabet. The message encoder and decoder are functions

$$\mathsf{encode}\colon \Sigma^n \to R_q \text{ and } \mathsf{decode}\colon R_q \to \Sigma^n,$$

such that $\mathsf{decode}(\mathsf{encode}(m) + e \bmod q) = m$ for any 'small enough' $e \in R$, e.g., one whose coefficients as a polynomial in $\mathbb{Z}[x]/f(x)$ are all in $[-t, t)$ for some integer threshold $t \geq 1$.

As above, the system uses a uniformly random $a \in R_q$ that can be generated by a trusted source, or chosen by the user.

- $\mathsf{Gen}(a)$: choose $r_1, r_2 \leftarrow \chi_k$, and let $p = r_1 - a \cdot r_2 \in R_q$. The public key is p (and a, if needed), and the secret key is r_2.
- $\mathsf{Enc}(a, p, m \in \Sigma^n)$: choose $e_1, e_2, e_3 \leftarrow \chi_e$. Let $\bar{m} = \mathsf{encode}(m) \in R_q$, and compute the ciphertext $[c_1 = a \cdot e_1 + e_2,\ c_2 = p \cdot e_1 + e_3 + \bar{m}] \in R_q^2$.
- $\mathsf{Dec}(c = [c_1, c_2], r_2)$: output $\mathsf{decode}(c_1 \cdot r_2 + c_2) \in \Sigma^n$. By a straightforward calculation, decryption will be correct as long as $e_1 \cdot r_1 + e_2 \cdot r_2 + e_3$ is within the error threshold of decode; this holds with high probability when χ_k, χ_e are sufficiently concentrated.

The proof of security, under the decision ring-LWE assumption for noise distributions χ_k and χ_e, is essentially identical to the proof of Theorem 6.3.2.

6.3.2 Parameters for Correctness

Here we give an upper bound on the Gaussian parameters s_k, s_e in terms of the desired per-symbol error probability δ. For reasonably small values of δ, correctness for the entire message can effectively be guaranteed by way of a simple error-correcting code.

One small subtlety is that if a portion of the random vector $\mathbf{e}$ used for encryption happens to be 'too long,' then the probability of decryption error for every symbol can be unacceptably large. We address this by giving a bound on $\mathbf{e}$, in Equation (6.3.4) below, which is violated with probability at most $2^{-\kappa}$ for some statistical parameter κ (say, $\kappa = 40$ for concreteness). We then calculate the error probabilities assuming that the bound holds; the overall decryption error probability is then no more than $2^{-\kappa}$ larger. One can also modify the Enc algorithm to reject and resample any $\mathbf{e}$ that violates Equation (6.3.4); the adversary's advantage can increase by at most $2^{-\kappa}$.

Lemma 6.3.1 (Correctness). *In the cryptosystem from Section 6.3.1, the error probability per symbol (over the choice of secret key) is bounded from above by any desired $\delta > 0$, as long as*

$$s_k \cdot s_e \leq \frac{\sqrt{2}\pi}{c} \cdot \frac{t}{\sqrt{(n_1 + n_2) \cdot \ln(2/\delta)}}. \tag{6.3.3}$$

Here $c \geq 1$ is a value that depends (essentially) only on $n_1 + n_2$; representative values are given in Figure 6.1.

(n_1+n_2)	$c \geq$	$(s_k \cdot s_e)/t \leq$
256	1.35	0.08936
384	1.28	0.07695
512	1.25	0.06824
640	1.22	0.06253

Figure 6.1: Bounds on parameters for Lemma 6.3.1 using a per-symbol error probability of $\delta = 0.01$, where c is determined so that the probability of choosing a 'bad' encryption vector $\mathbf{e}$ is at most 2^{-40}.

Proof. As shown above in the specification of the decryption algorithm, the jth symbol of the message decrypts correctly if $|\langle \mathbf{e}, \mathbf{r}_j \rangle| < \lfloor \frac{q}{4} \rfloor$. Recall that the entries of $\mathbf{e} \in \mathbb{Z}^{n_1+n_2+\ell}$ are independent and have distribution $D_{\mathbb{Z},s_e}$, and $\mathbf{r}_j \in \mathbb{Z}^{n_1+n_2+\ell}$ is the jth column of $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix}$, where the entries of $\mathbf{R}_1$ and $\mathbf{R}_2$ are drawn independently from $D_{\mathbb{Z},s_k}$.

To bound the error probability, let $\bar{\mathbf{e}} \in \mathbb{Z}^{n_1+n_2}$ consist of the first n_1+n_2 entries of $\mathbf{e}$. Then by Lemma 6.2.1, there is a $c \geq 1$ such that

$$\|\bar{\mathbf{e}}\| \leq c \cdot \frac{1}{\sqrt{2\pi}} \cdot s_e \sqrt{n_1+n_2} \tag{6.3.4}$$

except with very small probability (concrete values of c are given in Figure 6.1). For any fixed $\bar{\mathbf{e}}$ satisfying the above bound, observe that each $\langle \mathbf{e}, \mathbf{r}_j \rangle$ is independent and distributed essentially as $\langle \bar{\mathbf{e}}, D_{\mathbb{Z},s_k}^{n_1+n_2} \rangle$.[2] By Lemma 6.2.2, for any $T \geq 0$ we have

$$\Pr\left[\left|\langle \bar{\mathbf{e}}, D_{\mathbb{Z},s_k}^{n_1+n_2} \rangle\right| \geq T \cdot s_k \|\bar{\mathbf{e}}\|\right] < 2\exp(-\pi \cdot T^2).$$

Letting $T = t/(s_k\|\bar{\mathbf{e}}\|)$, where t is the error tolerance of our message encoding, and using the bound on $\|\bar{\mathbf{e}}\|$ from above, we get the bound on $s_k \cdot s_e$ from the lemma statement. □

6.3.3 Security Proof

Theorem 6.3.2. *The cryptosystem from Section 6.3.1 is CPA-secure, assuming the hardness of decision-*LWE *with modulus q for: (i) dimension n_2 with error distribution $D_{\mathbb{Z},s_k}$, and (ii) dimension n_1 with error $D_{\mathbb{Z},s_e}$.*

Proof. It suffices to show that the entire view of the adversary in an IND-CPA attack is computationally indistinguishable from uniformly random, for any encrypted message $\mathbf{m} \in \Sigma^\ell$. The view consists of $(\bar{\mathbf{A}}, \mathbf{P}, \mathbf{c})$, where $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n_1 \times n_2}$ is uniformly random, $\mathbf{P} \leftarrow \mathsf{Gen}(\bar{\mathbf{A}}, 1^\ell)$, and $\mathbf{c}^t \leftarrow \mathsf{Enc}(\bar{\mathbf{A}}, \mathbf{P}, \mathbf{m})$. First, $(\bar{\mathbf{A}}, \mathbf{P})$ is computationally indistinguishable from uniformly random $(\bar{\mathbf{A}}, \mathbf{P}^*) \in \mathbb{Z}_q^{n_1 \times (n_2+\ell)}$ under assumption (i) in the lemma statement, because $\mathbf{P} = (\bar{\mathbf{A}}^t)^t \cdot (-\mathbf{R}_2) + \mathbf{R}_1$, and $\bar{\mathbf{A}}^t$ is uniform while the entries of both

[2] We ignore the one additional term drawn from $D_{\mathbb{Z},s_e}$, which is compensated for by some slack in our final choice of parameters.

$-\mathbf{R}_2$ and $\mathbf{R}_1$ are drawn from $D_{\mathbb{Z},s_k}$. So the adversary's view is indistinguishable from $(\mathbf{A},\mathbf{c})$ where $\mathbf{A} = (\bar{\mathbf{A}},\mathbf{P}^*)$ is uniformly random and $\mathbf{c} \leftarrow \mathsf{Enc}(\mathbf{A},\mathbf{m})$. Now $(\mathbf{A},\mathbf{c})$ is also computationally indistinguishable from uniformly random $(\mathbf{A},\mathbf{c}^*)$ under assumption (ii) in the lemma statement, because $\mathbf{c} = (\mathbf{A}^t\mathbf{e}_1 + [\begin{smallmatrix}\mathbf{e}_2\\ \mathbf{e}_3\end{smallmatrix}]) + [\begin{smallmatrix}\mathbf{0}\\ \mathbf{m}\end{smallmatrix}]$, and $\mathbf{A}$ is uniform while the entries of $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are drawn from $D_{\mathbb{Z},s_e}$. □

It should be noted that for some settings of the parameters, one of the two assumptions in Theorem 6.3.2 may be true *information-theoretically* for the number of LWE samples exposed by the system in an attack. For instance, if $n_2 \geq n_1 \lg q$ and $s_k \geq \omega(\sqrt{\log n_1})$, then the public key $(\bar{\mathbf{A}},\mathbf{P})$ is within a negligible (in n_1) statistical distance of uniformly random (by a suitable version of the leftover hash lemma), whereas the corresponding ciphertexts are statistically far from uniform. These properties are important in, for example, the 'dual' cryptosystem and identity-based encryption scheme of [GPV08]. Conversely, the applications found in [PVW08, BHY09, ACPS09] have public keys that are far from uniform, but require that encryption under a 'malformed' (uniformly random) public key produces a ciphertext that is statistically independent of the encrypted message. These properties are achieved when $n_1 \geq n_2 \lg q$ and $s_e \geq \omega(\sqrt{\log n_2})$, again by the leftover hash lemma.

6.4 Lattice Decoding Attacks

The most promising practical attacks on the cryptosystem from Section 6.3, and more generally on LWE itself, use lattice-basis reduction followed by a decoding phase using the reduced basis.[3] In this section we analyze the performance of decoding as it relates to the quality of a given reduced basis. Then in Section 6.5 we analyze the effort required to obtain bases of a desired quality.

Before proceeding, we briefly explain how our *decoding* attack on LWE differs from the *distinguishing* attacks considered in other works [MR09, RS10]. In the latter, the adversary distinguishes (with some noticeable advantage) an LWE instance $(\mathbf{A},\mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e})$ from uniformly random, which is typically enough to break the semantic security of an LWE-based cryptosystem with the same advantage. To do this, the adversary finds a short nonzero integral vector $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = \mathbf{0} \bmod q$, which may be seen as a short vector in the (scaled) dual of the LWE lattice $\Lambda(\mathbf{A}^t)$. (Equivalently, the points of $\Lambda(\mathbf{A}^t)$ may be partitioned into hyperplanes orthogonal to $\mathbf{v}$, successively separated by distance $q/\|\mathbf{v}\|$.) The adversary then simply tests whether the inner product $\langle \mathbf{v},\mathbf{t}\rangle$ is "close" to zero modulo q. When $\mathbf{t}$ is uniform, the test accepts with probability exactly 1/2, but when $\mathbf{t} = \mathbf{A}^t\mathbf{s}+\mathbf{e}$ for Gaussian $\mathbf{e}$ with parameter s, we have $\langle \mathbf{v},\mathbf{t}\rangle = \langle \mathbf{v},\mathbf{e}\rangle \bmod q$, which is essentially a Gaussian (reduced mod q) with parameter $\|\mathbf{v}\| \cdot s$. When this parameter is not much larger than q, the Gaussian (mod q) can be distinguished from uniform with advantage very close to $\exp(-\pi \cdot (\|\mathbf{v}\| \cdot s/q)^2)$. For example, when $\|\mathbf{v}\| = 4q/s$ the distinguishing advantage is about 2^{-72}. However, to distinguish (and hence decrypt a ciphertext) with

[3] There are also purely combinatorial attacks on LWE [BKW03, Wag02] that may perform asymptotically better than lattice reduction, but so far not in practice. Also, these attacks generally require more LWE samples than our cryptosystem exposes, and an exponentially large amount of space.

high confidence, one needs $\|\mathbf{v}\| \leq q/(2s)$ or so, which usually requires a great deal more effort to obtain.

It is customary to include the inverse distinguishing advantage in the total 'cost' of an attack, so the computational effort and advantage need to be carefully balanced. For practical parameters, the optimal total cost of the distinguishing attack typically involves a very small distinguishing advantage (see Section 6.6), which may not be very useful in some settings, such as hybrid encryption.

Our decoding attack is stronger than the distinguishing attack in that it can actually recover the secret error vector in the LWE instance (and hence decrypt the ciphertext) with the same or better advantage, while using lower-quality vectors. For all the parameter settings that we investigated, our attack yields a better total effort as a ratio of time/advantage, and it is significantly more efficient in the high-advantage regime. (See Section 6.6 and Figure 6.5 in particular for details.) The attack works by using an entire reduced basis (not just one vector), and by expending some additional post-reduction effort to find the LWE solution. We also point out that unlike in basis reduction, the post-reduction effort is fully parallelizable.

The attack. Recall from Section 6.2.2 that an LWE instance $(\mathbf{A}, \mathbf{t} = \mathbf{A}^t\mathbf{s}+\mathbf{e})$ may be seen as a bounded-distance decoding problem on a certain lattice $\Lambda = \Lambda(\mathbf{A}^t)$, where $\mathbf{A}^t\mathbf{s} \in \Lambda$.

The standard method for solving a bounded-distance decoding problem on lattices is the recursive NearestPlane algorithm of Babai [Bab85]. The input to the algorithm is some lattice basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ (which for best results should be as reduced as possible) and a target point $\mathbf{t} \in \mathbb{R}^m$, and the output is a lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ that is 'relatively close' to $\mathbf{t}$. The precise guarantee is that for any $\mathbf{t} \in \operatorname{span}(\mathbf{B})$, $\mathsf{NearestPlane}(\mathbf{B}, \mathbf{t})$ returns the unique $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\mathbf{t} \in \mathbf{v} + \mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$. In other words, if $\mathbf{t} = \mathbf{v} + \mathbf{e}$ for some $\mathbf{v} \in \mathcal{L}(\mathbf{B})$, the algorithm outputs $\mathbf{v}$ if and only if $\mathbf{e}$ happens to lie in $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$.

The main drawback of this approach in attacking LWE is that in a reduced basis $\mathbf{B}$, the last several Gram-Schmidt vectors of $\mathbf{B}$ are typically very short, whereas the first few are relatively long. In such a case, the parallelepiped $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$ is very 'long and skinny,' and so the Gaussian error vector $\mathbf{e}$ is very unlikely to land in it, causing NearestPlane to produce an incorrect answer.

We address this issue by giving a generalized algorithm that admits a time/success tradeoff. It works just as NearestPlane does, except that it can recurse on some $d_i \geq 1$ distinct planes in the ith level of the recursion. In essence, the multiple recursion has the effect of making the parallelepiped $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$ wider in the direction of $\widetilde{\mathbf{b}}_i$ by a factor of exactly d_i.[4] To capture the most probability mass of the Gaussian error distribution of $\mathbf{e}$, one should choose the multiples d_i so as to maximize $\min_i(d_i \cdot \|\widetilde{\mathbf{b}}_i\|)$.[5]

[4] The algorithm of Klein [Kle00] also can recurse on more than one plane per iteration. Klein's algorithm solves the general bounded-distance decoding problem, and selects the planes at each stage probabilistically (though it can also be derandomized); its guarantee is related solely to the shortest Gram-Schmidt vector in the basis. Our algorithm is tailored specifically to the setting where we know the distribution of the offset vector; this allows the algorithm to recurse on exactly those planes that maximize the probability of success (over the choice of the error vector).

[5] One could further generalize the algorithm to search within an approximate *ball* made up of 'bricks' that are copies of $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$, thus capturing even more of the Gaussian without adding much more to the

The input to our NearestPlanes algorithm is a lattice basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \subset \mathbb{R}^m$, a vector $\mathbf{d} = (d_1, \ldots, d_k) \in (\mathbb{Z}^+)^k$ of positive integers, and a target point $\mathbf{t} \in \mathbb{R}^m$. It outputs a set of $\prod_{i\in[k]} d_i$ distinct lattice vectors in $\mathcal{L}(\mathbf{B})$, as follows:

1. If $k = 0$, return $\mathbf{0}$. Else, let $\mathbf{v}$ be the projection of $\mathbf{t}$ onto span$(\mathbf{B})$.
2. Let $c_1, \ldots, c_{d_k} \in \mathbb{Z}$ be the d_k distinct integers closest to $\langle \widetilde{\mathbf{b}_k}, \mathbf{v}\rangle / \langle \widetilde{\mathbf{b}_k}, \widetilde{\mathbf{b}_k}\rangle$.
3. Return $\bigcup\limits_{i\in[d_k]} (c_i \cdot \mathbf{b}_k + \mathsf{NearestPlanes}(\{\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}\}, (d_1, \ldots, d_{k-1}), \mathbf{v} - c_i \cdot \mathbf{b}_k)$.

Note that the recursive calls to NearestPlanes can be run entirely in parallel. The following lemma is an immediate extension of the analysis from [Bab85].

Lemma 6.4.1. *For* $\mathbf{t} \in$ span$(\mathbf{B})$, $\mathsf{NearestPlanes}(\mathbf{B}, \mathbf{d}, \mathbf{t})$ *returns the set of all* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *such that* $\mathbf{t} \in \mathbf{v} + \mathcal{P}_{1/2}(\widetilde{\mathbf{B}} \cdot \mathbf{D})$*, where* $\mathbf{D} = \mathrm{diag}(\mathbf{d})$*. The running time is essentially* $\prod_{i\in[k]} d_i$ *times as large as that of* $\mathsf{NearestPlane}(\mathbf{B}, \mathbf{t})$*.*

Note that the columns of $\widetilde{\mathbf{B}} \cdot \mathbf{D}$ from the lemma statement are the orthogonal vectors $d_i \cdot \widetilde{\mathbf{b}}_i$, so $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}} \cdot \mathbf{D})$ is a rectangular parallelepiped with axis lengths $d_i \cdot \|\widetilde{\mathbf{b}}_i\|$.

Success probability of NearestPlanes. When $\mathbf{t} = \mathbf{v} + \mathbf{e}$ for some $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ and a *continuous* Gaussian $\mathbf{e} \leftarrow D_s$ for some $s > 0$, the probability that $\mathbf{v}$ is in the output set of $\mathsf{NearestPlanes}(\mathbf{B}, \mathbf{d}, \mathbf{t})$ is

$$\Pr\left[\mathbf{e} \in \mathcal{P}_{1/2}(\widetilde{\mathbf{B}} \cdot \mathrm{diag}(\mathbf{d}))\right] = \prod_{i=1}^{m} \Pr\left[|\langle \mathbf{e}, \widetilde{\mathbf{b}}_i\rangle| < d_i \cdot \langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i\rangle / 2\right] = \prod_{i=1}^{m} \mathrm{erf}\left(\frac{d_i \cdot \|\widetilde{\mathbf{b}}_i\| \sqrt{\pi}}{2s}\right), \tag{6.4.1}$$

which follows by the independence of the values $\langle \mathbf{e}, \widetilde{\mathbf{b}}_i\rangle$, due to the orthogonality of the Gram-Schmidt vectors $\widetilde{\mathbf{b}}_i$. When $\mathbf{e}$ is drawn from a sufficiently wide *discrete* Gaussian over the integer lattice (in practice, a parameter of 6 or more suffices), the above is an extremely close approximation to the true probability.

We conclude this section by giving an informal explanation for why the advantage of the decoding attack can potentially be much larger than that of the distinguishing attack above, given vectors of the same quality. In the distinguishing attack, using a vector $\mathbf{v}$ of length (say) $\|\mathbf{v}\| \approx 4q/s$ implies that $\langle \mathbf{v}, \mathbf{t}\rangle \bmod q$ is distributed roughly as D_{4q} modulo q, whose statistical distance is only about 2^{-72} from uniform. A basis $\mathbf{B}$ of $\Lambda(\mathbf{A}^t)$ of equivalent quality has $\|\widetilde{\mathbf{b}_m}\| = q/\|\mathbf{v}\| = s/4$, because $\Lambda(\mathbf{A}^t)$ lies in hyperplanes orthogonal to $\mathbf{v}$ and separated by distance $q/\|\mathbf{v}\|$. So even without using multiple recursion in NearestPlanes (i.e., letting every $d_m = 1$), the corresponding term in Equation (6.4.1) is $\mathrm{erf}(\sqrt{\pi}/8) \approx 0.25$; moreover, the remaining terms typically approach 1 very rapidly, since $\|\widetilde{\mathbf{b}}_i\|$ usually increases quickly as i decreases. Letting $d_i > 1$ increases the overall success probability even more at little added cost, and allows for obtaining a relatively large advantage without needing higher-quality basis vectors.

search space. However, this would significantly complicate the analysis, and we find that the present approach is already very effective.

6.5 Basis Reduction and Experiments

In this section we present an analysis of lattice basis reduction on random q-ary lattices arising from LWE, and results of reduction experiments on various parameters. Our goal is to predict a conservative, but still useful, lower bound on the practical runtime of the lattice decoding attack described in Section 6.4 for a given set of LWE parameters.

We found that the best practical lattice reduction algorithm currently available to us is the BKZ algorithm as implemented by Shoup in the NTL library [Sho], so this is what we used in our experiments. The BKZ algorithm is parameterized by a blocksize k between 2 and the dimension of the lattice to be reduced. As the blocksize increases, the reduced basis improves in quality (i.e., it contains shorter lattice vectors, whose Gram-Schmidt lengths are closer together), but the runtime of BKZ also rapidly increases, becoming practically infeasible for $k \geq 30$ or so.

There has been some recent progress in the development of algorithms for finding short vectors in lattices, which can be used as subroutines to (or entire replacements of) BKZ reduction. For example, Gama, Nguyen, and Regev [GNR10] recently proposed a new method called “Extreme Enum”, which is much faster than its predecessor, the Schnorr-Euchner enumeration [SE94]. There are also single-exponential time algorithms for the Shortest Vector Problem [AKS01, MV10b, MV10a], which can run faster in practice than Schnorr-Euchner enumeration in certain low dimensions; however, these algorithms also require exponential space. We were not able to evaluate the performance and effectiveness of all these approaches, leaving this for future work. The BKZ implementation we use employs Schnorr-Euchner enumeration and, since the BKZ framework uses the enumeration subroutine as a black box, we presume that new algorithms incorporating Extreme Enum and other approaches will soon be available for evaluation. (For a comparison of enumeration algorithms in practice, see the open SVP-challenge website.[6])

In Section 6.5.1, we analyze the main properties of BKZ-reduced bases for q-ary lattices that are relevant to our decoding attack. In Section 6.5.2, we use our experiments to estimate the runtime required to obtain bases of a desired quality. We point out that the rest of our analysis is independent of this estimate, and can easily be applied with other runtime estimates for BKZ variants or other approaches.

6.5.1 Basis Reduction for q-ary Lattices

We begin by reviewing some of the prior work on basis reduction, in particular as applied to the q-ary lattices that arise from LWE.

The analysis of lattice reduction algorithms by Gama and Nguyen [GN08] identified the *Hermite factor* of the reduced basis as the dominant parameter in the runtime of the reduction and the quality of the reduced basis. A basis $\mathbf{B}$ of an m-dimensional lattice Λ has Hermite factor δ^m for $\delta \geq 1$ if $\|\mathbf{b}_1\| = \delta^m \cdot \det(\Lambda)^{1/m}$. For convenience, we call δ the *root-Hermite factor*.

Another important concept is the *Geometric Series Assumption* (GSA), introduced by Schnorr [Sch03]. The GSA says that in a BKZ-reduced basis $\widetilde{\mathbf{B}}$, the lengths $\|\widetilde{\mathbf{b}_i}\|$ of the Gram-Schmidt vectors decay geometrically with i, namely, $\|\widetilde{\mathbf{b}_i}\| = \|\mathbf{b}_1\| \cdot \alpha^{i-1}$ for some

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

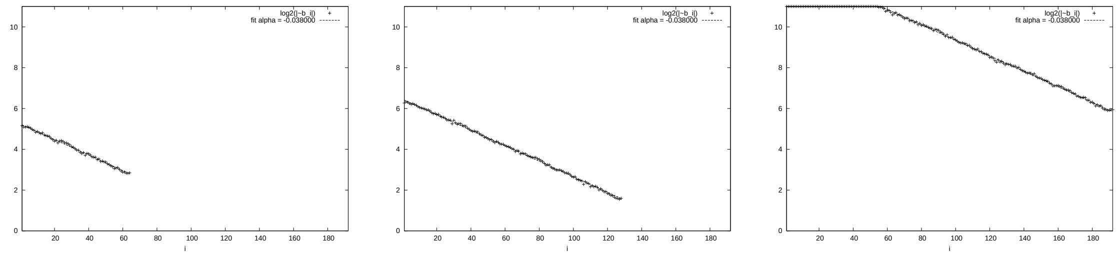

Figure 6.2: Logarithmic GSO lengths of three LWE instances after BKZ-20 reduction, which conform to the GSA assumption (modified with fixed upper and lower bounds on the Gram-Schmidt lengths). In all cases, the observed slope of the plot is very nearly the same, but other parameters vary. Parameters are $n = 32, q = 257, m = 64$ (left); $n = 64, q = 257, m = 128$ (center); $n = 32, q = 2053, m = 192$ (right).

$0 < \alpha < 1$. Our experiments on random q-ary lattices adhere to the GSA very closely, with the exception that the Gram-Schmidt lengths are always upper- and lower-bounded by q and 1 respectively, owing to the special structure of q-ary lattices (see Figure 6.2). For large BKZ blocksizes that correspond to effective attacks on LWE, these exceptional cases do not arise, and our bases conform to the GSA as ordinarily stated.

By combining the notion of Hermite factor with the GSA, we can predict the lengths of *all* Gram-Schmidt vectors in a basis $\mathbf{B}$ (of an m-dimensional lattice Λ) having root-Hermite factor δ. An easy calculation shows that under the GSA,

$$\det(\Lambda) = \prod_{i=1}^{m} \|\widetilde{\mathbf{b}_i}\| = \alpha^{m(m-1)/2} \cdot \delta^{m^2} \cdot \det(\Lambda) \quad \Longrightarrow \quad \alpha = \delta^{-2m/(m-1)} \approx \delta^{-2}, \qquad (6.5.1)$$

where the approximation holds for large m.

We now turn to q-ary lattices that arise from LWE. Recall from Section 6.2.2 that LWE is a bounded-distance decoding problem on the m-dimensional lattice

$$\Lambda(\mathbf{A}^t) = \{\mathbf{z} \in \mathbb{Z}^m : \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ such that } \mathbf{z} = \mathbf{A}^t\mathbf{s} \bmod q\}$$

for some $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with $m \geq n$. Because the LWE problem allows us to ignore some of the rows of $\mathbf{A}^t$ (and the corresponding noisy inner products), a natural and important question is what 'subdimension' m makes a lattice attack most effective. This question was addressed in [MR09], where a simple calculation showed that for a desired root-Hermite factor δ, the subdimension $m = \sqrt{n \lg(q)/\lg(\delta)}$ is optimal in the context of the natural distinguishing attack on LWE (as described at the beginning of Section 6.4). The analysis of [MR09] actually applies to the lattice

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\},$$

which is the *dual* of $\Lambda(\mathbf{A}^t)$ up to scaling by a q factor, and the optimal subdimension m given above minimizes the length of $\widetilde{\mathbf{d}_1} = \mathbf{d}_1$ in a reduced basis $\mathbf{D}$ of $\Lambda^{\perp}(\mathbf{A})$ having root-Hermite factor δ. In our setting, by duality the same choice of m *maximizes* $\|\widetilde{\mathbf{b}_m}\| = q/\|\widetilde{\mathbf{d}_1}\|$, where the basis $\mathbf{B}$ of $\Lambda(\mathbf{A}^t)$ is the dual basis of $\mathbf{D}$ in *reverse* order.

In our decoding attack (and assuming the GSA), the form of the success probability given in Equation (6.4.1) as a product of $\operatorname{erf}(\cdot)$ terms also strongly indicates that we should maximize $\|\widetilde{\mathbf{b}_m}\|$, and hence use the same subdimension $m = \sqrt{n\lg(q)/\lg(\delta)}$ as above. We do not have a fully rigorous proof of this claim, since using a smaller m decreases the number of terms in the product, and hence could potentially increase the success probability. However, it seems unlikely that using a smaller m would improve the success probability by much (if at all). This is because $\|\widetilde{\mathbf{b}_m}\| = q/\|\mathbf{d}_1\|$ decreases rapidly as m decreases (see [MR09]), and $\|\widetilde{\mathbf{b}_{m-i}}\| \approx \|\widetilde{\mathbf{b}_m}\| \cdot \delta^{2(i-1)}$ is a very close approximation for small i, which are the Gram-Schmidt vectors that largely determine the success probability. Likewise, increasing m also appears counterproductive, since it both decreases $\|\widetilde{\mathbf{b}_m}\|$ *and* increases the number of terms in the product.

All of the above assumes that a cryptosystem exposes enough LWE samples (via its public keys and/or ciphertexts) to use the optimal subdimension. While this is always true of prior cryptosystems [Reg05b, PVW08, GPV08], it is not necessarily the case for our cryptosystem in Section 6.3, due to its smaller keys and ciphertexts. In this case, the adversary should use the dimension m corresponding to the actual number of published samples (this rule applies to some of our parameters sets given in Section 6.6).

6.5.2 Extrapolating BKZ Runtimes

In order to assign concrete runtimes to the attacks we put forward, we need to predict the runtime required to achieve a given root-Hermite factor δ in random q-ary lattices.

Gama and Nguyen [GN08] observed that on random lattices generated according to a variety of models, the runtime required to achieve a given root-Hermite factor δ in large dimensions (exceeding 200 or so) is largely determined by δ alone; the lattice dimension and determinant contribute only second-order terms. Our initial experiments confirmed this behavior for random q-ary lattices, and so we extrapolated runtimes using a fixed set of LWE parameters q and n, for a variety of values δ that correspond to sufficiently large optimal subdimensions $m = \sqrt{n\lg(q)/\lg(\delta)} \approx 200$. Our experiments were performed on a single 2.3 GHz AMD Opteron machine, using the single-precision floating-point BKZ implementation from the standard NTL library [Sho]. (Practical attacks on LWE for parameters beyond toy examples would require using at least quadruple precision, which would increase the running times by at least some constant factor, so our extrapolations are somewhat optimistic and hence conservative from a security point of view.)

Figure 6.3 shows the results of our experiments and their extrapolations. Using the rule of thumb that obtaining a 2^k approximation to the shortest vector in an m-dimensional lattice takes time $2^{\tilde{O}(m/k)}$ using BKZ, we conclude that the logarithm of the runtime should grow roughly linearly in $1/\lg(\delta)$. Our limited experiments seem consistent with this behavior, though many more would be needed to confirm it with confidence. Using least-square regression, the best linear fit to our data for $t_{\mathsf{BKZ}}(\delta) := \lg(T_{\mathsf{BKZ}}(\delta))$, the log runtime (in seconds, on our machine) of BKZ as a function of δ, is $t_{\mathsf{BKZ}}(\delta) = 1.806/\lg(\delta) - 91$. Since our experiments were limited by resources and available time, and we expect to see further improvements in basis reduction techniques (such as those in [GNR10]), for

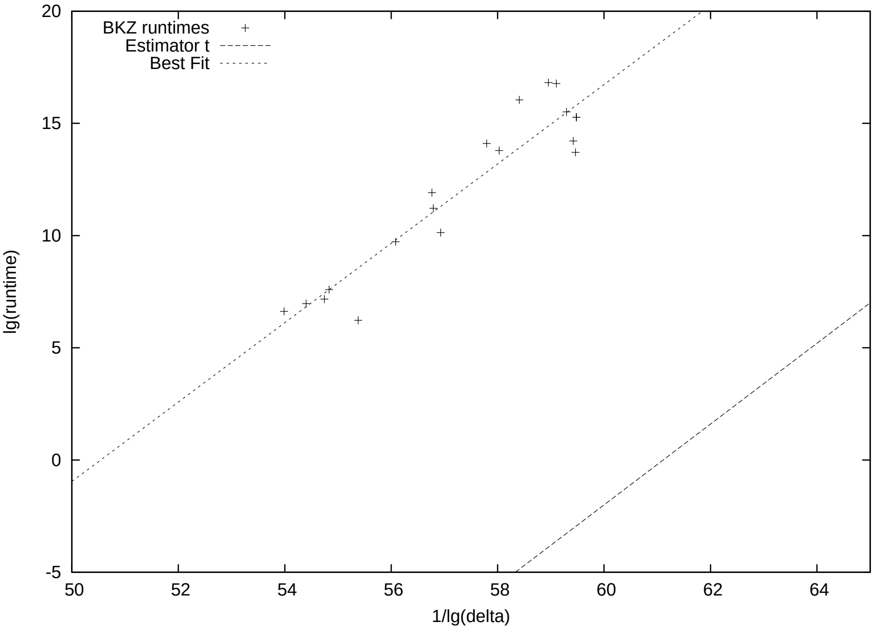

Figure 6.3: Runtime of BKZ experiments on random q-ary lattices, with parameters $n = 72$, $q = 1021$, and $m = \sqrt{n \lg(q)/\lg(\delta_0)}$, i.e., the optimal subdimension with respect to a desired root-Hermite factor δ_0. The vertical axis is $t_{\mathsf{BKZ}}(\delta) := \lg(T_{\mathsf{BKZ}}(\delta))$, the logarithmic runtime required to obtain a vector with root-Hermite factor δ when running BKZ with successively increasing blocksizes. The horizontal axis is $1/\lg(\delta)$ for the actual root-Hermite factor δ achieved by the reduction. For comparison, the graph shows the best-fit estimator $t_{\mathsf{BKZ}}(\delta) = 1.086/\lg(\delta) - 91$, and our conservative lower bound estimate $t_{\mathsf{BKZ}}(\delta) = 1.8/\lg(\delta) - 110$.

analyzing concrete hardness we use a conservative lower bound estimate of

$$t_{\mathsf{BKZ}}(\delta) := \lg(T_{\mathsf{BKZ}}(\delta)) = 1.8/\lg(\delta) - 110. \qquad (6.5.2)$$

Note that in this estimate, the 1.8 factor is very slightly smaller, and the -110 constant term is substantially smaller, than their counterparts in the best-fit function from our experiments. We chose the value 1.8 because our experiments were limited to relatively small block sizes, and the runtimes needed to achieve smaller values of δ very quickly became infeasible, so we believe that the true coefficient on the linear term (even with improved algorithms) is larger than 1.8. Similarly, our choice of -110 provides for some security margin against special-purpose hardware. In conclusion, we believe that our lower bound estimate provides some safety against foreseeable advances in algorithms and hardware, but in any case, our analysis is entirely modular and can be immediately adapted to work with any revised estimator.

n	q	s	Per-User Key ($\mathbf{P}$)	Full Key ($\mathbf{P}$ & $\bar{\mathbf{A}}$)	Ciphertext ($\mathbf{c}$)	Message Expansion
128	2053	6.77	1.8×10^5	3.6×10^5	2.8×10^3	22.0
192	4093	8.87	2.9×10^5	7.4×10^5	3.8×10^3	30.0
256	4093	8.35	4.0×10^5	11.2×10^5	4.6×10^3	36.0
320	4093	8.00	4.9×10^5	17.2×10^5	5.4×10^3	42.0
136	2003	13.01	2.8×10^6	5.8×10^6	2.9×10^3	22.6
214	16381	7.37	2.4×10^6	6.4×10^6	4.8×10^3	18.7

Figure 6.4: Sizes (in bits) of public keys and ciphertexts for the cryptosystem described in Section 6.3; for comparison, the last two rows are for parameters given in [MR09]. In each case, the message size is $\ell = 128$ bits. The "message expansion" factor is the ratio of ciphertext size to plaintext size. Recall that in the ring-based system, the public key sizes are about a factor of n smaller.

6.6 Cryptosystem Parameters

We now estimate the concrete security of, and compute the space requirements for, the LWE-based cryptosystem from Section 6.3 on a variety of parameters, and compare with the example parameters given in [MR09] for the cryptosystem described therein (which is essentially due to [PVW08]). Figure 6.4 gives key and ciphertext sizes and Figure 6.5 provides the security estimates.

Instantiating the parameters. We set the cryptosystem's parameters as $n_1 = n_2 = n$ and $s_k = s_e = s$ for some positive integer n and $s > 0$, so that the two LWE hardness assumptions made in Theorem 6.3.2 are equivalent. In practice, though, distinguishing the public key and ciphertext from uniform are not equally hard, because the public key exposes fewer LWE samples than the ciphertext does. In particular, the adversary cannot use the optimal subdimension in attacking the public key, making it quite a bit harder to break. This fact could allow us to use slightly smaller s_k and correspondingly larger s_e parameters to get slightly stronger overall security, but we elect not to introduce such complications at this point. (And arguably, the secret key ought to be better-protected than any individual ciphertext.)

We choose the modulus q to be just large enough (according to the bounds in Figure 6.1) to allow for a Gaussian parameter $s \geq 8$, so that the discrete Gaussian $D_{\mathbb{Z}^m,s}$ approximates the continuous Gaussian D_s extremely well.[7] Increasing the value of q beyond this threshold appears not to increase the concrete security of our cryptosystem, and (somewhat paradoxically) may even slightly decrease it! This is because the BKZ runtime depends almost entirely on the root-Hermite factor δ, and by the constraints on our parameters (specifically, $s_k = s_e = s = O(\sqrt{q})$), the δ yielding a successful attack on our system grows as $q^{\Theta(1/n)}$, which increases with q (albeit very slowly).

[7] Note that the theoretical worst-case reduction [Reg05b] for LWE asks that $s \geq 2\sqrt{n}$. However, the constant factors are not tight, and here we are concerned with concrete hardness against known attacks.

			Adv. ε	(Distinguish)		(Decode)		
n	q	s	$\lg(\varepsilon)$	δ	lg(secs)	δ	lg(#enum)	lg(secs)
			≈ 0	*1.0065	83	1.0089	47	32
128	2053	6.77	-32	1.0115	< 0	1.0116	13	< 0
	(toy)		-64	1.0128	< 0	1.0130	1	< 0
			≈ 0	*1.0045	168	1.0067	87	78
192	4093	8.87	-32	1.0079	49	1.0083	54	42
	(low)		-64	1.0087	34	1.0091	44	29
			≈ 0	*1.0034	258	*1.0052	131	132
256	4093	8.35	-32	1.0061	96	1.0063	87	90
	(medium)		-64	1.0067	77	1.0068	73	75
			≈ 0	*1.0027	353	*1.0042	163	189
320	4093	8.00	-32	1.0049	146	1.0052	138	132
	(high)		-64	1.0054	122	1.0055	117	119
			≈ 0	1.0038	219	1.0071	82	68
136	2003	13.01	-32	1.0088	33	1.0092	42	27
	[MR09]		-64	1.0098	18	1.0102	27	14
			≈ 0	1.0053	126	1.0078	66	52
214	16381	7.37	-32	1.0091	28	1.0094	39	25
	[MR09]		-64	1.0099	17	1.0102	29	14

Figure 6.5: Example parameters and attacks for the LWE-based cryptosystem described in Section 6.3.1, for various adversarial advantages. The cryptosystem parameters are $n = n_1 = n_2$, q, $s = s_k = s_e$, and message length $\ell = 128$ bits. For comparison, the last two parameter settings ($n = 136$, $n = 214$) come from the example parameters of [MR09]. The columns labelled "Distinguish" refer to a distinguishing (i.e., semantic security) attack. These give the root-Hermite factors δ needed to obtain the respective distinguishing advantages (over the random choice of the LWE error vector), and the corresponding logarithmic runtime (in seconds) according to our optimistic estimator from Equation (6.5.2). The columns labelled "Decode" refer to our decoding (i.e., message and randomness recovery) attack. These give example root-Hermite factors and number of NearestPlanes enumerations needed to obtain the respective decoding probability, and the corresponding estimated runtime of the attack. Other trade-offs between δ and the number of enumerations are possible (as δ increases, so does #enum); we chose the largest δ for which the estimated enumeration runtime does not exceed that of basis reduction. *An asterisk on a value of δ indicates that for reduced vectors of lengths required by the attack, the cryptosystem reveals too few LWE samples to allow an optimal choice of subdimension and corresponding root-Hermite factor δ. In such cases, we used the value of δ induced by working with the full dimension $m = n_1 + n_2 + \ell = 2n + 128$.

Estimating the security. We analyze the distinguishing attack and our decoding attack (both described in Section 6.4), estimating the total runtimes for each of a few representative adversarial advantages. The attacks apply to a single key and ciphertext; by a standard hybrid argument, the advantage increases at most linearly in the number of ciphertexts encrypted under a single key.

For analyzing the basic distinguishing attack we rely on calculations from [MR09]. We first compute a bound $\beta = (q/s) \cdot \sqrt{\ln(1/\varepsilon)/\pi}$ on the length of a nonzero vector $\mathbf{v} \in \Lambda^{\perp}(\mathbf{A})$ that would yield the desired distinguishing advantage (taken over the random choice of the LWE error). We then compute the root-Hermite factor $\delta = 2^{(\lg^2 \beta)/(4n \lg q)}$ that would yield such a vector, assuming that the attacker can use the optimal subdimension $m = \sqrt{n \lg(q)/\lg(\delta)}$. (The value of δ follows from the fact that in the optimal subdimension, a root-Hermite factor of δ yields a vector of length $2^{2\sqrt{n \lg q \lg \delta}}$.) If the optimal subdimension for this δ exceeds $n_1 + n_2 + \ell = 2n + 128$ (the number of LWE samples implicitly exposed by a ciphertext), then we discard this δ and instead use the one for which $\delta^m \cdot q^{n/m} = \beta$, where $m = 2n + 128$. (Values of δ computed in this way are indicated in Figure 6.5 by asterisks.) We then calculate a lower bound on the BKZ runtime using our conservative estimator from Equation (6.5.2).

In analyzing our decoding attack, we try various values of δ, computing both the estimated BKZ runtime and the number of enumerations needed (assuming the GSA) to achieve the desired success probability according to Equation (6.4.1). If the number of enumerations does not exceed the BKZ runtime (in seconds) by more than a 2^{16} factor, we consider this to be an acceptable attack. (This 2^{16} factor is somewhat arbitrary, but seems to be a reasonable estimate on the number of NearestPlanes enumerations that can be performed per second, especially with parallelism.) We list the largest value of δ for which we found an acceptable attack, along with the corresponding runtime (which includes both the BKZ and NearestPlanes phases).

7 Conclusion

This thesis contains several contributions to practical lattice-based cryptography.

We analyzed the difference between worst-case problems in ideal lattices and worst-case problems in general lattices. Both are frequently used as security assumptions for lattice schemes. We found that the instances of ideal lattice problems form a negligibly small subclass. Though there is no known attack that employs the additional structure of ideal lattices with notable effect, we still feel that assumptions on ideal lattices should not yet be used for applications where long-term security is desired. After finishing the analysis, we focused on practical cryptographic constructions.

Starting with the compression function SWIFFT and using it as an example, we presented an efficiency improvement that is generally applicable to all known lattice constructions based on worst-case problems; technically, we covered constructions based on (Ideal)SIS and a similar improvement was already known for those based on LWE. We demonstrated how the SWIFFT design can be generalized to a small range of parameters without suffering a significant loss in efficiency. Furthermore, we showed that sublattice attacks are possible against SWIFFT. Using these, we gave evidence that it is feasible to recover ℓ_2-pseudo-collisions for SWIFFT instances in about 2^{50} seconds on a modern machine. Then, we gave replacement parameters for which such pseudo-collisions are harder to find. Since the SWIFFT compression function is slow compared to other constructions seen in the SHA-3 competition, it should be used only for high security applications. In this case, we recommend that our replacement parameters be used to make the recovery of pseudo-collisions computationally hard.

We went on to study zero-knowledge identification schemes. Here, we adapted a recent construction by Cayrel and Véron, which is based on assumptions about error-correcting codes, to lattices. Doing so enabled us to prove security based on worst-case lattice assumptions and provide a slight efficiency improvement. This makes our construction the most efficient one known today amongst all that are secure against attackers with quantum computers. The adaptation of code-based schemes, such as the one presented here, is often fruitful for both fields. In our case, we found that the soundness of the code-based scheme was slightly smaller than previously claimed. In general, we find that collaborations between these fields are a worthwhile endeavor.

Finally, we discussed public-key encryption. Here, we unified some recent progress into a clean representation of a scheme that is substantially more efficient than all predecessors. At the same time, we presented an improved attack for secret-key recovery and decoding that works against our scheme as well as its predecessors. We found that our attack is often more efficient than previous ones, and, at the same time, able to achieve more, because previous attacks only attempt to distinguish ciphertexts from randomness. We purposefully left our analysis modular with respect to the runtime of lattice basis reduction algorithms, because there are unpublished improvements in this area that will have a significant impact. Once these are available, it will be easy to replace the estimator we provide for the runtime of current algorithms with an updated one.

Further research. In each of the last three chapters we demonstrated that lattice-based assumptions are well-suited to realize a specific cryptographic construction. Now, a promising investigation would be to find out specifically how well these or similar constructions scale if massive parallel processing is used. Such parallelization is already a reality today on modern CPUs and graphic cards, and it is reaching smaller devices like smart-phones. Current research is focused around using parallelization in cryptanalysis, i.e., to the attacker's benefit. It seems natural to counter this by using the same technological advance to improve the efficiency of cryptosystems. Moreover, using parallelism for the cryptosystem is more powerful. Assume there is a cryptosystem which, by means of parallelization, may double its main security parameter without a noticeable loss in efficiency. Then, doubling or even increasing the attacker's strength a thousandfold will not even come close to endangering such a system, because the attacker has to overcome a doubling in the exponent.

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572. 2010. Cited on page 45.

[ACPS09] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618. 2009. Cited on pages 48 and 53.

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293. 1997. Cited on pages 45 and 46.

[AD07] M. Ajtai and C. Dwork. The first and fourth public-key cryptosystems with worst-case/average-case equivalence. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(97), 2007. Cited on page 46.

[ADL$^+$08] Y. Arbitman, G. Dogon, V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFTX: A proposal for the SHA-3 standard, 2008. `http://www.eecs.harvard.edu/~alon/PAPERS/lattices/swifftx.pdf`. Cited on page 17.

[Ajt96a] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1996*, pages 99–108. ACM Press, 1996. Cited on pages 1, 8, 11, and 18.

[Ajt96b] M. Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996. Cited on page 45.

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610. 2001. Cited on page 56.

[Ale03] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307. 2003. Cited on pages 46 and 49.

[AR04] D. Aharonov and O. Regev. Lattice problems in np cap conp. In *FOCS*, pages 362–371. IEEE Computer Society, 2004. ISBN 0-7695-2228-9. Cited on page 8.

[Ash10] R. B. Ash. *A Course In Algebraic Number Theory.* Dover Publications, 2010. `http://www.math.uiuc.edu/~r-ash/ANT.html`. Cited on page 15.

[Bab85] L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986. Preliminary version in STACS 1985. Cited on pages 54 and 55.

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993. Cited on page 47.

[Ban95] W. Banaszczyk. Inequalites for convex bodies and polar reciprocal lattices in R^n. *Discrete & Computational Geometry*, 13:217–231, 1995. Cited on page 47.

[BHY09] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35. 2009. Cited on page 53.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. Cited on page 53.

[BL06] J. Buchmann and C. Ludwig. Practical lattice basis sampling reduction. In F. Hess, S. Pauli, and M. E. Pohst, editors, *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 222–237. Springer, 2006. ISBN 3-540-36075-1. Cited on page 29.

[BLR08] J. Buchmann, R. Lindner, and M. Rückert. Explicit hard instances of the shortest vector problem. In J. Buchmann and J. Ding, editors, *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 79–94. Springer, 2008. ISBN 978-3-540-88402-6. Cited on pages 23 and 26.

[BP02] M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. *Advances in Cryptology-Crypto 2002*, pages 149–162, 2002. Cited on page 34.

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552. 2010. Cited on page 45.

[CMR07] C. D. Cannière, F. Mendel, and C. Rechberger. Collisions for 70-step sha-1: On the full cost of collision search. In C. M. Adams, A. Miri, and M. J. Wiener, editors, *Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2007. ISBN 978-3-540-77359-7. Cited on page 17.

[CV10] P.-L. Cayrel and P. Véron. Improved code-based identification scheme, 2010. `http://arxiv.org/abs/1001.3017v1`. Cited on pages 3, 32, 36, and 37.

[DG03] A. M. Daniel Goldstein. On the equidistribution of hecke points. *Forum Mathematicum 2003, 15:2*, pages 165–189, 2003. Cited on page 13.

[FFS87] U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *STOC*, pages 210–217. ACM, 1987. Cited on page 32.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986. Cited on page 32.

[FS90] U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In *STOC*, pages 416–426. ACM, 1990. Cited on page 42.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. 2009. Cited on page 45.

[GG07] P. Gaborit and M. Girault. Lightweight code-based identification and signature. *IEEE Transactions on Information Theory (ISIT)*, pages 186–194, 2007. Cited on page 36.

[GGH96] O. Goldreich, S. Goldwasser, and S. Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996. Cited on pages 11 and 18.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, page 304. ACM, 1985. Cited on page 34.

[GN08] N. Gama and P. Q. Nguyen. Predicting lattice reduction. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 2008. ISBN 978-3-540-78966-6. Cited on pages 3, 4, 25, 26, 43, 56, and 58.

[GNR10] N. Gama, P. Q. Nguyen, and O. Regev. Lattice enumeration using extreme pruning. In *EUROCRYPT*, pages 257–278. 2010. Cited on pages 56 and 58.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. 2008. Cited on pages 11, 18, 45, 46, 47, 49, 53, and 58.

[HHHGW09] P. S. Hirschhorn, J. Hoffstein, N. Howgrave-Graham, and W. Whyte. Choosing NTRUEncrypt parameters in light of combined lattice reduction and MITM approaches. In M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, editors, *ACNS*, volume 5536 of *Lecture Notes in Computer Science*, pages 437–455. 2009. ISBN 978-3-642-01956-2. Cited on page 30.

[HM96] S. Halevi and S. Micali. Practical and provably-secure commitment schemes from collision-free hashing. In N. Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 201–215. Springer, 1996. ISBN 3-540-61512-1. Cited on page 33.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288. 1998. Cited on page 46.

[Kle00] P. N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, pages 937–941. 2000. Cited on page 54.

[KP01] J. Kilian and E. Petrank. Concurrent and resettable zero-knowledge in poly-loalgorithm rounds. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 560–569. ACM, New York, NY, USA, 2001. ISBN 1-58113-349-9. doi:http://doi.acm.org/10.1145/380752.380851. Cited on page 42.

[KTX08] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, pages 372–389. 2008. Cited on pages 3, 31, 32, 33, 37, 38, and 39.

[Lan65] S. Lang. *Algebra*. Addison-Wesley series in mathematics. Addison-Wesley, Reading, Mass. [u.a.], 1965. Cited on page 24.

[Lan73] S. Lang. *Elliptic Functions*. Addison Wesley, 1973. Cited on page 16.

[LM06] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *International Colloquium on Automata, Languages and Programming (ICALP) 2006*, Lecture Notes in Computer Science, pages 144–155. Springer-Verlag, 2006. Cited on pages 2, 9, 12, 21, 24, and 42.

[LM08] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *Theory of Cryptography Conference (TCC) 2008*, Lecture Notes in Computer Science, pages 37–54. Springer-Verlag, 2008. Cited on pages 11, 12, 18, and 19.

[LMPR08] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In *Fast Software Encryption (FSE) 2008*, Lecture Notes in Computer Science, pages 54–72. Springer-Verlag, 2008. Cited on pages 2, 12, 18, 19, 21, 32, and 38.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23. 2010. Cited on pages 2, 9, 46, 49, 50, and 51.

[LPS10] V. Lyubashevsky, A. Palacio, and G. Segev. Public-key cryptographic primitives provably as secure as subset sum. In *TCC*, pages 382–400. 2010. Cited on pages 46 and 49.

[LV01] A. K. Lenstra and E. R. Verheul. Selecting cryptographic key sizes. *J. Cryptology*, 14(4):255–293, 2001. Cited on pages 27 and 30.

[Lyu08a] V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography*, pages 162–179. 2008. Cited on page 32.

[Lyu08b] V. Lyubashevsky. *Towards Practical Lattice-Based Cryptography.* Ph.D. thesis, University of California, San Diego, 2008. Cited on pages 2 and 9.

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616. 2009. Cited on pages 3, 31, 32, 33, and 38.

[ME05] M. R. Murty and J. Esmonde. *Problems in Algebraic Number Theory*, volume 190 of *Graduate Texts in Mathematics.* Springer Verlag, 2005. Cited on page 14.

[MG02] D. Micciancio and S. 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, 2002. Cited on pages 5, 6, and 8.

[Mic02] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Preliminary version in FOCS 2002. Cited on page 46.

[Mic10] D. Micciancio. Duality in lattice cryptography. In *Public Key Cryptography.* 2010. Invited talk. Cited on pages 8, 46, and 49.

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004. Cited on pages 9 and 42.

[MR09] D. Micciancio and O. Regev. *Post Quantum Cryptography*, chapter Lattice-based Cryptography. Springer-Verlag, 2009. Cited on pages 2, 4, 5, 18, 25, 27, 46, 48, 49, 53, 57, 58, 60, 61, and 62.

[MV10a] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In *STOC*, pages 351–358. 2010. Cited on page 56.

[MV10b] D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *SODA*, pages 1468–1480. 2010. Cited on page 56.

[MVO07] M. R. Murty and J. Van Order. Counting integral ideals in a number field. *Expositiones Mathematicae*, 25(1):53–66, 2007. Cited on pages 2 and 14.

[MvOV01] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001. Cited on page 17.

[Ngu09] P. Q. Nguyen. *The LLL Algorithm: Survey and Applications*, chapter Hermite's Constant and Lattice Algorithms, pages 19–70. Information Security and Cryptography. Springer, 2009. Cited on page 5.

[OO98] K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In H. Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 354–369. Springer, 1998. ISBN 3-540-64892-5. Cited on page 41.

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342. 2009. Cited on pages 11, 45, and 48.

[Pei10] C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *CRYPTO*, pages 80–97. 2010. Cited on pages 46, 48, and 49.

[PR06] C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography Conference (TCC) 2006*, Lecture Notes in Computer Science, pages 145–166. Springer-Verlag, 2006. Cited on pages 22 and 24.

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In D. Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008. ISBN 978-3-540-85173-8. Cited on pages 46, 49, 53, 58, and 60.

[PW08] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196. 2008. Cited on pages 45 and 48.

[Reg03] O. Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004. Preliminary version in STOC 2003. Cited on pages 45 and 46.

[Reg05a] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *STOC*, pages 84–93. ACM, 2005. ISBN 1-58113-960-8. Cited on pages 1, 3, 8, and 9.

[Reg05b] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005. Cited on pages 45, 46, 47, 48, 49, 58, and 60.

[Reg09] O. Regev. *The LLL Algorithm: Survey and Applications*, chapter On the Complexity of Lattice Problems with Polynomial Approximation Factors, pages 475–496. Information Security and Cryptography. Springer, 2009. Cited on page 7.

[RS10] M. Rückert and M. Schneider. Selecting secure parameters for lattice-based cryptography. Cryptology ePrint Archive, Report 2010/137, 2010. `http://eprint.iacr.org/`. Cited on page 53.

[Sch68] W. M. Schmidt. Asymptotic formulae for point lattices of bounded determinant and subspaces of bounded height. *Duke Mathematical Journal*, 35(2):327–339, 1968. Cited on page 13.

[Sch03] C.-P. Schnorr. Lattice reduction by random sampling and birthday methods. In H. Alt and M. Habib, editors, *STACS*, volume 2607 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2003. ISBN 3-540-00623-0. Cited on pages 29 and 56.

[SE94] C.-P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathmatical Programming*, 66:181–199, 1994. Cited on page 56.

[Sho] V. Shoup. Number theory library 5.5.2 (NTL) for C++. `http://www.shoup.net/ntl/`. Cited on pages 29, 56, and 58.

[Sho94] P. W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In L. M. Adleman and M.-D. A. Huang, editors, *ANTS*, volume 877 of *Lecture Notes in Computer Science*, page 289. Springer, 1994. ISBN 3-540-58691-1. Cited on pages 1 and 32.

[SLdW07] M. Stevens, A. K. Lenstra, and B. de Weger. Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities. In M. Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2007. ISBN 978-3-540-72539-8. Cited on page 17.

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. Cryptology ePrint Archive, Report 2009/285, 2009. `http://eprint.iacr.org/`. Cited on pages 12 and 19.

[Ste93] J. Stern. A new identification scheme based on syndrome decoding. In D. R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 1993. ISBN 3-540-57766-1. Cited on pages 3, 32, 35, 36, and 37.

[Vér96] P. Véron. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996. Cited on pages 36 and 37.

[Wag02] D. Wagner. A generalized birthday problem. In *CRYPTO*, pages 288–303. 2002. Cited on pages 43 and 53.

[XT08] K. Xagawa and K. Tanaka. A compact signature scheme with ideal lattice, 2008. Asian Assiciation for Algorithms and Computation (AAAC). Cited on pages 18 and 19.