UNIVERSITY OF CALIFORNIA, SAN DIEGO

Towards Practical Lattice-Based Cryptography

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Vadim Lyubashevsky

Committee in charge:

Professor Daniele Micciancio, Chair
Professor Sanjoy Dasgupta
Professor Russell Impagliazzo
Professor Alexander Vardy
Professor Nolan Wallach

2008

The dissertation of Vadim Lyubashevsky is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2008

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

My graduate school experience has been a road littered with successes and failures. As the completion of this arduous journey is coming to an end, I have been retrospecting on the many decisions made along its way and wondering whether I would make the same choices again. While there are certainly things that I would have liked to change, it brings me great happiness to know that the most important decision would, unequivocally, remain the same. It has been an honor and a pleasure to work under my advisor, Daniele Micciancio. Most of everything that I have learned and accomplished came as a result of his inspired guidance, enthusiasm, and patience. For this, I will always remain grateful.

I am also thankful to Russell Impagliazzo whose deep insights always resulted in me learning much more than just the answers to the questions I asked of him. I would like to thank Sanjoy Dasgupta for teaching some wonderful AI classes and being a really good guy to just talk to. I want to thank Nolan Wallach and Alex Vardy for being on my thesis committee. I want to thank Alon Orlitsky for being on my thesis proposal committee, even though it resulted in my indentured servitude as his TA. I want to additionally thank Sanjoy and Alon for being the ones who, by their wonderful teaching of undergraduate classes while I was their TA, had great influence on the development of my teaching style.

I have to certainly acknowledge my friends in the department and outside of it with whom I have spent many enjoyable non-working hours. It gives me great pleasure to thank you all in alphabetical order. Thank you, Nuno, Chris, Sashka, Anjum, Ragesh, Kirill, Yi-Kai, Tita, Petros, Sara, Anton, Saurabh, Tom, Todor, Nate, Sarah, Panos, and Scott.

Finally, I am deeply indebted to Adriana, whose perpetual love and kindness made all the problems and disappointments encountered in my studies appear irrelevant. When I think back on my happiest times, you are always there.

Chapter 3 and Appendix A are, in part, a reprint, of the paper “Generalized Compact Knapsacks Are Collision Resistant” co-authored with Daniele Micciancio and appearing in the proceedings of ICALP 2006. The dissertation author was the primary investigator and author of this paper.

Chapter 4 is, in part, a reprint of the paper “Asymptotically Efficient Lattice-Based Digital Signatures” co-authored with Daniele Micciancio and appearing in the proceeding of TCC 2008. The dissertation author was the primary investigator and author of this paper.

Chapter 5 is an extension of the results of the paper “Lattice-Based Identification Schemes Secure Under Active Attacks” appearing in the proceedings of PKC 2008. The dissertation author was the primary investigator and author of this paper.

VITA

2002	Bachelor of Science in Computer Science Columbia University, New York, NY, USA
2008	Doctor of Philosophy in Computer Science University of California, San Diego, CA, USA

ABSTRACT OF THE DISSERTATION

Towards Practical Lattice-Based Cryptography

by

Vadim Lyubashevsky
Doctor of Philosophy in Computer Science
University of California, San Diego, 2008
Professor Daniele Micciancio, Chair

Lattice-based cryptography began with the seminal work of Ajtai (Ajtai '96) who showed that it is possible to build families of cryptographic functions in which breaking a *randomly chosen* element of the family is as hard as solving *worst-case* instances of lattice problems. This work generated great interest and resulted in constructions of many other cryptographic protocols with security based on worst-case lattice problems. An additional advantage of lattice-based primitives is that, unlike their counterparts based on factoring and discrete log, they are conjectured to be secure in the advent of quantum computing. The main disadvantage of lattice-based constructions is that they generally involve operations on, and storage of, large $n \times n$ matrices. This resulted in the schemes being rather inefficient and unsuitable for practical use. To cope with this inherent inefficiency, Micciancio proposed to build lattice-based primitives based on the worst-case hardness of lattices that have some additional structure. In (Micciancio '02), he showed how to build *one-way functions*, computable in almost linear time, with security based on worst-case problems on such lattices.

While interesting from a theoretical perspective, one-way functions are not very useful in practice. Our goal in this thesis is to present constructions of practical and *efficient* cryptographic protocols whose security is based on worst-case hardness of lattice problems. We first show how to build collision-resistant hash functions whose security is based on the hardness of lattice problems in all lattices with a special structure. The special structure that the lattices possess is that they are *ideals* of certain polynomial rings. The hash functions that we build have almost linear running time, and in practice turn out to be essentially as efficient as ad-hoc constructions that have no provable security. We also give constructions of provably-secure identification and signature schemes whose asymptotic running times are almost linear (up to poly-logarithmic factors), and so these schemes are much more efficient than comparable primitives with security based on factoring and discrete log. Thus our work implies that by considering *ideal lattices*, it is possible to have the best of both worlds: security based on worst-case problems and optimal efficiency.

1

Introduction

Lattice-based cryptography began with the seminal work of Ajtai [Ajt96], who showed that random instances of a certain problem are at least as hard to solve as worst-case instances of lattice problems. The initial construction of a one-way function in [Ajt96] as well as the numerous subsequent constructions of other cryptographic primitives, such as collision-resistant hash functions [Ajt96,GGH96,MR07], identification schemes [MV03], and encryption schemes [AD97,Reg03,Reg05,PW08] are very interesting from a theoretical point of view because they are essentially the only problems for which such a worst-case / average-case connection is known. Additionally, schemes based on lattices are conjectured to remain secure in the advent of quantum computers, which is in sharp contrast to schemes based on factoring and discrete log which would become completely insecure [Sho97]. Despite such security advantages, the cryptographic functions proposed in the above works are not efficient enough to be practical. The source of impracticality is that working with lattices requires the storage of, and operations on, large $n \times n$ integer matrices. This results in cryptographic functions with key size and computation time at least quadratic in the security parameter n.

A first step in the direction of creating cryptographic functions based on worst-case hardness that are efficient in practice, and still provably secure, was taken by Micciancio in [Mic07]. In that paper, the author showed how to create a family of efficiently computable *one-way functions* whose security is based on a certain problem for a particular class of lattices, called cyclic lattices. These lattices admit a much more compact representation than general ones, and the resulting functions can be described and evaluated in time almost linear in n. However, one-wayness is a rather weak security property, interesting mostly from a theoretical point of view. By contrast, the (less efficient) functions based on general lattices discussed in the previous paragraph perform a myriad of considerably stronger and much more useful cryptographic tasks.

In this thesis, we take the next step in creating efficient cryptographic functions with security based on worst-case hardness assumptions. We show how to create efficient collision-resistant hash functions, signatures, and identification schemes whose security is based on standard lattice problems for a special class of lattices, that we call *ideal lattices*. In addition to the strong security guarantees, some of the schemes that we present are also more efficient (in an asymptotic sense) than all other known schemes based on any hardness assumption.

1..1 The Average-case / Worst-case Connection

A major draw of lattice-based cryptographic constructions is that schemes can be built based on the hardness of worst-case problems. This is rather different from the average-case hardness security guarantees that accompany practically all other schemes. We will now briefly explain the advantage of having security based on worst-case problems. Consider, for example, any cryptographic scheme in which one can prove that breaking the scheme implies factoring some number N. The question now is: how should such a hard-to-factor N be chosen? We certainly cannot ask someone to provide us with such an N, because this third party may know a factorization and then can break our scheme. So the only choice we have is to generate it ourselves, but how? Choosing a random N in some range is certainly a bad idea because half the numbers are even, and are thus easy to factor. Perhaps just choosing two primes and then multiplying them together would produce an N that's hard to factor, but one must be careful in how the primes are chosen so as to not make their product vulnerable to specialized factorization algorithms. In short, it is not enough to believe that factoring is hard in the worst-case, we also need to know of a *distribution* over which factoring is hard.

Lattice-based schemes, on the other hand, don't have this problem. Ajtai showed that if *uniformly random* instances of a certain problem A can be solved, then certain other problems can be solved for *all* lattices [Ajt96]. Notice that coming up with a hard instance of problem A is now trivial – it's just generated uniformly at random. Now one can build cryptographic schemes based on the hardness of random instances of problem A, and be sure that the cryptographic scheme is as hard to break as worst-case lattice problems.

1..2 The Source of Efficiency

Ajtai's collision-resistant hash function can be roughly described as follows: the function family $\mathcal{H}$ consists of functions $h_{\mathbf{A}}$ indexed by $n \times k$ matrices $\mathbf{A} \in \mathbb{Z}_p^{n\times k}$, where $k > n \log p$ and the inputs to the function are $\{0,1\}$-vectors in $\mathbb{Z}^k$. The output $h_{\mathbf{A}}(\mathbf{y})$ is simply the product $\mathbf{Ay} \bmod p$. Ajtai showed that finding two distinct vectors $\mathbf{y}, \mathbf{y}'$ such that $\mathbf{Ay} \bmod p = \mathbf{Ay}' \bmod p$ for random $\mathbf{A}$ is as hard as solving certain lattice problems for *all* lattices [Ajt96, GGH96]. This implies that $\mathcal{H}$ is a family of collision-resistant hash functions. But notice that because $\mathbf{A}$ is an $n \times k$ matrix, the multiplication of $\mathbf{Ay}$ requires $nk > n^2$ operations, and so the functions are somewhat inefficient.

One way to increase efficiency is to somehow pick the $n \times k$ matrix $\mathbf{A}$ so that the multiplications $\mathbf{Ay}$ take less than $O(n^2)$ time. While it's certainly easy to construct matrices $\mathbf{A}$ such that multiplication takes linear time (consider, for example, having all columns of the matrix be the same), the difficult part is in ensuring that it's still hard to find $\mathbf{y}, \mathbf{y}' \in \{0,1\}^k$ such that $\mathbf{Ay} \bmod p = \mathbf{Ay}' \bmod p$. An idea of Micciancio [Mic07] was to construct $\mathbf{A}$ by picking the first column at random, and then have the next $n-1$ columns be rotations of the first. that is, if the first column is $(a_1, \ldots, a_n)^T$, then the i^{th} column will be $(a_i, \ldots, a_n, a_1, \ldots, a_{i-1})$. Then the $n+1^{st}$ column of $\mathbf{A}$ is chosen at random again, and the next $n-1$ columns are rotations of that column. This procedure is repeated until all k columns of $\mathbf{A}$ are filled. Notice that in order to populate the entire matrix $\mathbf{A}$, we only need to create k/n random columns, and so the matrix can be represented with just k elements in $\mathbb{Z}_p$ instead of nk, as was previously required. The multiplication of $\mathbf{Ay}$ can now be interpreted as the sum of k/n products of polynomials of degree $n-1$ (see Section 2.D). Each polynomial multiplication can be performed in time $\tilde{O}(n)$ using the Fast Fourier Transform, and then we just need to sum up the $k/n = O(\log n)$ resulting vectors, for a total running time of $\tilde{O}(n)$ (the notation, $\tilde{O}(n)$ means $O(n \log^c n)$ for any constant c). In [Mic07], Micciancio was able to show that in addition to being efficient, the above function enjoys some security properties as well. In particular, he showed that if lattice problems for a certain class of lattices, he called *cyclic lattices* are hard in the worst case, then the above function family is one-way, but not (as we will show) collision-resistant. One of the results of this thesis will be showing how to create the matrix $\mathbf{A}$ such that computing $\mathbf{Ay}$ takes $\tilde{O}(n)$ time and finding collisions implies solving lattice problems for all lattices in a certain class.

The idea of imposing algebraic structure on the description of the function in order to speed up its computation has long been used in the context of codes (e.g. cyclic codes), and even in lattice-based cryptography in the context of the NTRU cryptosystem [HPS98]. But what is unique about the results in [Mic07] and the results in this thesis is that the cryptographic constructions have a proof of security. Indeed, while the NTRU cryptosystem remains resistant to attacks, a proposed version of the NTRU signature scheme was recently completely broken [NR06]. Thus it is our belief that provable security is of great practical value.

1..3 Thesis Outline

In Chapter 2, we present the relevant definitions that we will be using throughout the work. Most important is Section 2.F, which describes the hash function family that we will be proving collision-resistant. The hardness of finding collisions for functions in this family, while interesting in its own right, is also what the security of our signature and identification schemes is based on. We prove the collision-resistance of the function family in Chapter 3, and the only result that we will need from that chapter for the later chapters is Theorem 3.1, which describes the connection between finding collisions in random members of the hash function family and solving lattice problems for all lattices of a certain class. In Chapter 4, we build an asymptotically efficient one-time signature whose security is based on the hardness of finding collisions in our hash function (which in turn is based on the worst-case hardness of lattice problems). This one-time signature can then be combined with a standard tree construction to achieve a full-fledged digital signature scheme whose efficiency is only a factor of $O(\log n)$ worse. In Chapter 5, we construct a provably secure identification protocol which is likewise asymptotically efficient, and in Chapter 6, we present a signature scheme that is secure in the random oracle model, that would seem to be more efficient in practice than the signature scheme from Chapter 4.

1.A Collision-Resistant Hash Functions

Collision-resistant hash functions are functions h that map some domain D to a smaller range such that it is computationally hard to find two elements $x_1, x_2 \in D$ with the property that $h(x_1) = h(x_2)$. One use of collision-resistant functions is in digital signatures, where rather than signing a long message x, we may instead sign a much shorter digest $h(x)$. Because of the collision-resistance of h, it is computationally infeasible to find another message y such that $h(y) = h(x)$, and therefore a signature of $h(x)$ is just as binding as a signature of x. The standardized hash functions of today are all designed in an ad-hoc fashion and lack provable security, and recently people have demonstrated some weaknesses in their constructions [WLF$^+$05,WY05,BCJ$^+$05]. We therefore believe that it may be a good time to consider building efficient collision-resistant hash functions that have a proof of security.

1.A.1 Our Contributions and Related Work

In [Mic07], it was shown how to create an efficient *one-way function* based on worst case hardness of problems for lattices which can be thought of as ideals in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n - 1\rangle$. In our work, we show how to construct *collision-resistant hash functions* based on the hardness of problems for lattices that can be represented as ideals of the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f}\rangle$ where $\mathbf{f}$ can be one of infinitely many other polynomials (including $\mathbf{x}^n - 1$). Thus our result has two desirable features: it weakens the complexity assumption while strengthening the cryptographic primitive.

Our hash functions also admit extremely efficient implementations. The time complexity for computing the image of an element in the domain is roughly $\tilde{O}(n)$, and a practical instantiation of our function [LMPR08] is essentially of the same complexity as the ad-hoc hash functions (e.g. SHA-256).

We now give an informal description of the hash function families that we will be proving collision resistant. Given a ring $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$ (with the usual polynomial addition and multiplication operations) where $\mathbf{f} \in \mathbb{Z}[\mathbf{x}]$ is some monic, irreducible polynomial of degree n and p is an integer of order roughly $n^{1.5}$, generate m random elements $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R$ where m is some small (logarithmic or constant) number. The ordered m-tuple $h = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in R^m$ is our hash function. It will map elements from D^m, where D is a strategically chosen subset of R, to R. For an element $\widehat{\mathbf{b}} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in D^m$, the hash is $h(\widehat{\mathbf{b}}) = \sum\limits_{i=1}^{m} \mathbf{a}_i \mathbf{b}_i$. Notice that the size of the key (the hash function) is $O(mn \log p) = \tilde{O}(n)$, and the operation $\mathbf{a}_i \mathbf{b}_i$ can be performed in time $\tilde{O}(n)$ by using the Fast Fourier Transform. Since m is at most $O(\log n)$, we can hash a message in time $m\tilde{O}(n) = \tilde{O}(n)$. Then to prove that our hash function family is collision resistant, we will show that if there is a polynomial time algorithm that (for a randomly chosen hash function $h \in R^m$,) succeeds with non-negligible probability in finding $\widehat{\mathbf{b}} \neq \widehat{\mathbf{b}}' \in D^m$ such that $h(\widehat{\mathbf{b}}) = h(\widehat{\mathbf{b}}')$, then the Shortest Vector Problem (SVP) is solvable in polynomial time for *every* lattice that corresponds to an ideal of the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$.

There is very little known about the hardness of SVP for lattices that are ideals in $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$. Another result of our work is a connection between finding short vectors in such lattices and computational problems from algebraic number theory that appeared to have been studied before, but with little success. This at least gives some evidence as to the hardness of SVP for such lattices.

Concurrently with, and independently from our work, Peikert and Rosen [PR06] have shown how to construct collision resistant hash functions based on the hardness of finding the shortest vector for lattices which correspond to ideals in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n - 1 \rangle$. While our more general result is interesting from a purely theoretical standpoint, it turns out that choices of certain $\mathbf{f}$ other than $\mathbf{x}^n - 1$ (such as

$\mathbf{x}^n + 1$) result in somewhat more efficient hash functions (see [LMPR08]), making our generalization also of practical use. Also, our hardness assumptions are formulated in a way that leads to natural connections with algebraic number theory.

There have been many proposed cryptographic primitives whose hardness relied on the knapsack problem (e.g. [MH78], [Dam], [CR88]) but the attacks against them (e.g. [Sha84], [JG94], [Vau01]) rendered the primitives impractical. The attacks, though, all attack a group-based knapsack problem, and it is unclear how to apply them to our ring-based one. Also, none of those primitives had a reduction to worst-case instances of lattice problems. Of course, the hardness of our primitive is based on worst-case problems for *ideal* lattices, and very little is known about them. Still, it seems as if there are currently no algorithms that are able take advantage of the ring structure that they possess and we think that determining the worst-case hardness of lattice problems for these lattices is a very interesting open problem.

1.B Signatures

Digital signature schemes, initially proposed in Diffie and Hellman's seminal paper [DH76] and later formalized by Goldwasser, Micali and Rivest, [GMR88], are among the most important and widely used cryptographic primitives. Still, our understanding of these intriguing objects is somehow limited.

The definition of digital signatures clearly fits within the public key cryptography framework. However, efficiency considerations aside, the existence of secure digital signatures schemes can be shown to be equivalent to the existence of conventional (symmetric) cryptographic primitives like pseudorandom generators, one-way hash functions, private key encryption, or even just one-way functions [NY89,Rom90]. There is a big gap, both theoretical and practical, between the efficiency of known constructions implementing public-key and private-key cryptography. In the symmetric setting, functions are often expected to run in time which is linear or almost linear in the security parameter k. However, essentially all known public key encryption schemes with a supporting proof of security are based on algebraic functions that take at least $\Omega(k^2)$ time to compute, where 2^k is the conjectured hardness of the underlying problem. For example, all factoring based schemes must use keys of size approximately $\Theta(k^3)$ to achieve k bits of security to counter the best known sub-exponential time factoring algorithms, and modular exponentiation raises the time complexity to over $\omega(k^4)$ even when restricted to small k-bit exponents and implemented with an asymptotically fast integer multiplication algorithm.

When efficiency is taken into account, digital signatures seem much closer to public key encryption schemes than to symmetric encryption primitives. Most signature schemes known to date employ the same set of number theoretic techniques commonly used in the construction of public key encryption schemes, and result in similar complexity. Digital signatures based on arbitrary one-way hash functions have also been considered, due to the much higher speed of conjectured one-way functions (e.g., instantiated with common block ciphers as obtained from ad-hoc constructions) compared to the cost of modular squaring or exponentiation operations typical of number theoretic schemes. Still, the performance advantage of one-way functions is often lost in the process of transforming them into digital signature schemes: constructions of signature schemes from non-algebraic one-way functions almost invariably rely on Lamport and Diffie's [DH76] one-time signature scheme (and variants thereof) which requires a number of one-way function applications essentially proportional to the security parameter. So, even if the one-way function can be computed in linear time $O(k)$, the complexity of the resulting signature scheme is again at least quadratic $\Omega(k^2)$.

Therefore, a question of great theoretical and practical interest, is whether digital signature schemes can be realized at essentially the same cost as symmetric key cryptographic primitives. While a generic construction that transforms any one-way function into a signature scheme with similar efficiency is impossible [BMG07], one may wonder if there are specific complexity assumptions that allow to build more efficient digital signature schemes than currently known. Ideally, are there digital signature schemes with $O(k)$ complexity, which can be proved as hard to break as solving a computational problem which is believed to require $2^{\Omega(k)}$ time?

1.B.1 Results and Techniques

Tree-Based Signature Scheme

The main result of Chapter 4 is a construction of a provably secure one-time digital signature scheme (i.e., a signature scheme that allows to securely sign a single message) with key size and computation time almost linear (up to poly-logarithmic factors) in the security parameter. In other words, we give a new one-time digital signature scheme with complexity $O(k \log^c k)$ which can be proved to be as hard to break as a lattice problem which is conjectured to require $2^{\Omega(k)}$ time to solve.

A full-fledged signature scheme can then be constructed via a standard transformation from one-time signatures to general signature schemes. We remark that the same transformation from one-time signatures to unrestricted signature schemes was also employed by virtually all previous constructions of digital signatures from arbitrary one-way functions (e.g., [Mer89,NY89,Rom90]). This transformation, which combines one-time signatures together with a tree structure, is relatively efficient and allows one to sign messages with only a logarithmic number of applications of a hash function and a one-time signature scheme [Szy04]. The bottleneck in one-way function based signature schemes is the construction of one-time signatures from one-way functions. The reason for the slowdown is that the one-way function is typically used to sign a k-bit message one bit at a time, so that the entire signature requires k evaluations of the one-way function. In fact, a recent result of Barak and Mahmoody-Ghidary [BMG07] states that it is impossible to convert one-way functions, treated as black boxes, into one-time signatures (that have the same security) with less than $\Omega(k)$ calls to the one-way function. Therefore any construction that hopes to be more efficient must use the one-way functions in a non-black-box way. In this paper we give a direct construction of one-time signatures, where each signature requires just two applications of the lattice based collision-resistant hash function that is described in this work. The same lattice based hash function can then be used to efficiently transform the one-time signature into an unrestricted signature scheme via a hash-tree with only a logarithmic loss in performance.

The high level structure of our lattice-based one-time signature scheme is easily explained. Let h be the collision-resistant function described in Section 1.A. When the user wants to generate a key for the one-time signature scheme, he simply picks two "random" inputs $\widehat{\mathbf{k}}, \widehat{\mathbf{l}} \in D^m$, and computes their images under the hash function $(h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}))$. (The key $(\mathbf{a}_1, \ldots, \mathbf{a}_m)$ to the hash function h can also be individually chosen by the user, or shared among all the users of the signature scheme.) The secret key is the pair $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$, while the public key is given by their hashes $(h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}))$. Then, the signature of a message $\mathbf{z}$ is simply obtained as a "linear combination" $\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{y}}$ of the two secret vectors (the multiplication $\widehat{\mathbf{k}}\mathbf{z}$ is defined as the ring multiplication of each coordinate of $\widehat{\mathbf{k}}$ by $\mathbf{z}$). Signatures can be easily verified using the homomorphic properties of the lattice based hash function $h(\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}) = h(\widehat{\mathbf{k}})\mathbf{z} + h(\widehat{\mathbf{l}})$. If the domain D^m were closed under addition and multiplication, then one could show that the public key $h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}})$ and signature $\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$ do not reveal any information about the secret key $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$, and a forgery relative to a different secret key yields a collision to the hash function. The crucial point is that D^m being closed would guarantee the existence of a different secret key that could have been used to sign $\mathbf{z}$. But since the domain is restricted, there is a possibility that the signer's secret key was the only one that could have produced $h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}})$ and signature $\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$. This turns out to be the main difficulty in carrying out our proof.

We solve this technical problem by choosing the secret key elements $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$ according to a carefully crafted (non-uniform) probability distribution, which can be intuitively thought as a "fuzzy" subset of the full domain R^m. It turns out that if the appropriate distribution on D^m is used, then h is one-way when the input is chosen according to the distribution on D^m, and the distribution is closed under the ring operations in an approximate probabilistic sense.

Random Oracle Based Signature Scheme

In Chapter 6, we show that by using a random oracle, we can construct a signature scheme that does not require the use of a tree, thus increasing the efficiency of the scheme in Chapter 4 by a factor of $\log n$. One way to view our signature scheme is as a transformation of the one-time signature scheme constructed in Chapter 4 into an ID-scheme (described in Chapter 5) and then into a signature scheme via the

Fiat-Shamir heuristic. Converting an ID-scheme into a signature scheme via the Fiat-Shamir heuristic is a very popular technique that has been used in the construction of many well-known signature schemes (e.g. [FS86, Sch91, GQ88, Oka92a, GPS06]). One could also argue that the identification schemes in the preceding protocols were fairly straightforward conversions from one-time signature schemes (see [BS07] for a discussion of the relationship between one-time signatures and ID-schemes). So it seems that it is plausible that using similar techniques, our one-time signature can be converted into a signature scheme that's secure in the random oracle model. But a difficulty arises because the one-time signature "leaks" some information about the secret key, which didn't cause a problem in the one-time signature because the key is used only once. But using the secret key many times in an identification scheme would result in the complete leakage of the key. We will explain how we overcome this problem in Chapter 5, where we build the identification scheme. Then in Chapter 6, we give a self-contained proof of security of the signature scheme.

1.B.2 Related Work

Lamport showed the first construction of a one-time signature based on the existence of one-way functions. In that scheme, the public key consists of the values $f(x_0), f(x_1)$, where f is a one-way function and x_0, x_1 are randomly chosen elements in its domain. The elements x_0 and x_1 are kept secret, and in order to sign a bit i, the signer reveals x_i. This construction requires one application of the one-way function for every bit in the message. Since then, more efficient constructions have been proposed in (e.g. [Mer87,BC92,BM84,EGM96,BM96a,BM96b]), but there was always an inherent limitation in the number of bits that could be signed efficiently with one application of the one-way function.

Preceding our work, there have been other lattice-based signature scheme proposals: most notably the GGH scheme [GGH97] and the NTRU signature scheme [HHGP$^+$03]. Unfortunately, neither of these schemes possessed a proof of security, and recently they were completely broken by Nguyen and Regev [NR06]. Another construction of a lattice-based signature scheme was recently discovered by Gentry et.al [GPV08]. Their construction, while having the same flavor as the GGH scheme, is provably secure in the random oracle model and it resists the attacks of [NR06]. An instantiation of the scheme in [GPV08] with security, in the random oracle model, based on the hardness of SVP in ideal lattices requires $\tilde{O}(n^2)$ time for signing and $\tilde{O}(n)$ time for verification. So even when instantiated with ideal lattices, the scheme of [GPV08] is not as efficient as ours.

1.C Identification Schemes

Public key identification (ID) protocols allow a party holding a secret key to prove its identity to any other entity holding the corresponding public key. The minimum security of such protocols should be that a passive observer who sees the interaction should not then be able to perform his own interaction and successfully impersonate the prover. In a more realistic model, the adversary should first be allowed to interact with the prover in a "dishonest" way in hopes of extracting some information, and then try to impersonate the prover. Identification schemes resistant to such impersonation attempts are said to be secure in the active attack model [FFS88], and this is currently the *de facto* security notion.

Since Fiat and Shamir's seminal paper [FS86], there have been many proposals for constructing secure ID protocols. With a few notable exceptions, most of these protocols (e.g. [GQ88, Sch91, Oka92b, Sho99, Poi00, GPS06]) are based on problems from number theory, and as such, they require fairly costly multiplication and exponentiation operations. Another potential problem, as mentioned earlier, is that the security of these protocols is based on problems that are easy if (when) practical quantum computers become reality [Sho97]. Thus it is prudent to have viable alternative schemes based on different hardness assumptions.

The identification protocols not based on number theory problems (e.g. [Sha89, Ste96]) are generally combinatorial in nature. Because of this lack of algebraic structure, these combinatorial schemes all seem to have an inherent shortcoming in that they require a lot more rounds of communication than their algebraic counterparts. This problem arises because the proof of security is established by showing that the schemes are zero-knowledge proofs of knowledge. It is shown that the prover (or adversary) who successfully proves his identity, actually "knows" the secret (as defined in [FFS88]), yet the protocol is zero-knowledge, and as such, the prover doesn't reveal anything about his secret key. The problem is that in order for the protocol to have negligible soundness error, it must be repeated a polynomial number of times. But zero-knowledge is not preserved under parallel-repetition, and so the protocol has to be run sequentially in order for it to maintain the claimed security.

1.C.1 Our Results

In Chapter 5, we present an asymptotically-efficient ID scheme with security based on the hardness of problems on ideal lattices. We prove security by showing that an adversary who successfully attacks our scheme can be used to find collisions in the hash function described in 1.A.

We believe that the technical details of our ID protocol may also be of independent interest. While our scheme has the structure of a standard 3-move commit-challenge-response protocol, for security reasons, an honest prover sometimes "aborts" the protocol during the response stage. It can be shown that if the prover always responds to the verifier, then his secret key is leaked to even a passive observer. On the other hand, by strategically refusing to reply, each round of the protocol can be shown to be *witness-indistinguishable*. And since witness-indistinguishability is preserved under parallel-composition, all the rounds can be performed in parallel.

1.C.2 Related Work

The one place in the literature that mentions constructions of lattice-based identification schemes is the work of Micciancio and Vadhan [MV03] on statistical zero knowledge relating to lattice problems. In this work, the authors show an efficient-prover SZK proof system for certain lattice problems and mention that one can convert the proof system into an identification scheme. But there does not seem to be a way to make the scheme of [MV03] as efficient as the one presented here. It seems that in order to achieve $2^{\Omega(n)}$ security, the identification protocol would have to run in time $\tilde{O}(n^2)$.

1.D Open Problems and Future Directions

We have shown that some very efficient cryptographic protocols derive their security from the hardness of solving problems for ideal lattices. Thus a very important direction of future research is on establishing some hardness results for these problems. At the present it is not even known whether SVP is NP-hard for ideal lattices (whereas the NP-hardness of SVP under randomized reductions for general lattices has been known for a while). One of our results concerns a connection between ideal lattices and algebraic number fields, and we believe that algebraic number theory could play a prominent role in helping us understand more about these lattices. Following our work, Peikert and Rosen explored the connection between lattices and algebraic number theory further, and were able to establish some interesting connections between average-case problems and worst-case problems in algebraic number theory [PR07]. But the hardness of problems for ideal lattices still remains wide open.

One thing that will not be discussed much in this work is practical instantiations of our protocols. At the present, we have only implemented our hash function and it seems to be very competitive (see [LMPR08]), but it is unclear how efficient our other protocols are in practice. For example, our signature scheme is computationally very efficient, but the signatures will be much longer than the signatures outputted by number-theoretic schemes. Also, the fact that our scheme is provably secure only guarantees that the structure of our scheme is sound, but says very little about what are the minimum parameters needed for the schemes to be secure in practice. For establishing practical security, it is necessary to understand the best algorithms that attack our schemes. Our schemes are based on the hardness of lattice problems, and we believe that algorithms that look for short vectors in lattices are the best forms for attack against our schemes. Unfortunately, little is known about the practical performance of such algorithms. Recently, there have been some experimental results by Nguyen and Gama that showed a somewhat tight bound on the performance of the most widely-used lattice reduction algorithm [GN08b], but the same authors later discovered a theoretically better algorithm [GN08a] which happens to perform worse in practice. The one thing that is evident is that there is still much to learn about the hardness of lattice reduction, and therefore it seems prudent to put off proposing specific parameters for our schemes.

A problem that is of more imminent interest is whether the hardness assumptions needed for our signature and identification scheme can be weakened to the point that they are the same as for our hash function. It will turn out that the hardness assumption that is needed for our hash function is essentially finding vectors within a factor n of the shortest vector in ideal lattices. For our signature and identification schemes, however, the security assumption is finding vectors within a factor n^c of the shortest vector (where c is between 2 and 3 depending on the scheme). Figuring out whether it is somehow possible to reduce this c to 1 (via new constructions) is an important open problem that will have consequences on the practical instantiations of the schemes.

Of course, another direction that we believe should be pursued is constructing other efficient protocols based on the hardness of ideal lattices. We have demonstrated that, at least in an asymptotic sense, ideal lattice-based protocols are extremely efficient. Thus we believe that with some additional novel techniques, it should be possible to construct practically efficient protocols based on the worst-case hardness of ideal lattice problems. At the present, it is only known how to construct collision-resistant hash functions [LM06, PR06], identification schemes [MV03, Lyu08], and signatures [LM08, GPV08], but we are very hopeful that the future will bring many other constructions.

2

Definitions and Preliminaries

2.A Lattices

A full-rank n-dimensional *lattice* is the set of all integer combinations

$$\left\{\sum_{i=1}^{n} x_i\mathbf{b}_i \colon x_i \in \mathbb{Z}\right\}$$

of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^n$. The set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a *basis* for the lattice, and can be compactly represented by the matrix $\mathbf{B} = [\mathbf{b}_1|\ldots|\mathbf{b}_n] \in \mathbb{R}^{n\times n}$ having the basis vectors as columns. The lattice generated by $\mathbf{B}$ is denoted $\mathcal{L}(\mathbf{B})$. For any basis $\mathbf{B}$, we define the fundamental parallelepiped $\mathcal{P}(\mathbf{B}) = \{\mathbf{B}\mathbf{x}\colon \forall i. 0 \leq x_i < 1\}$. The following lemma states that one can sample lattice points uniformly at random from the fundamental parallelepiped associated to a given sublattice.

Lemma 2.1 ([MG02, Proposition 8.2])**.** *There is a probabilistic polynomial time algorithm that on input a lattice basis* $\mathbf{B}$ *and a full rank sublattice* $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$*, outputs a lattice point* $\mathbf{x} \in \mathcal{L}(\mathbf{B}) \cap \mathcal{P}(\mathbf{S})$ *chosen uniformly at random.*

The lattices that are most relevant to us are *integer lattices*, i.e., lattices $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}^n$ all of whose vectors have integer coordinates. The dual of a lattice $\mathcal{L}(\mathbf{B})$ (denoted $\mathcal{L}(\mathbf{B})^*$) is the lattice generated by the matrix $\mathbf{B}^{-T}$, and consists of all vectors that have integer scalar product with all lattice vectors. For any vector $\mathbf{x} = (x_1, \ldots, x_n)^T$, define the cyclic rotation $rot(\mathbf{x}) = (x_n, x_1, \ldots, x_{n-1})^T$. A lattice

$\mathcal{L}(\mathbf{B})$ is cyclic if it is closed under the rotation operation, i.e., if $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ implies $rot(\mathbf{x}) \in \mathcal{L}(\mathbf{B})$.

The *minimum distance* of a lattice $\mathcal{L}(\mathbf{B})$ is the minimum distance between any two (distinct) lattice points and equals the length of the shortest nonzero lattice vector. The minimum distance can be defined with respect to any norm. For any $p \geq 1$, the ℓ_p norm of a vector $\mathbf{x}$ is defined by $\|\mathbf{x}\|_p = \sqrt[p]{\sum_i |x_i|^p}$ and the corresponding minimum distance is denoted

$$\lambda_1^p(\mathcal{L}(\mathbf{B})) = \min\{\|\mathbf{x} - \mathbf{y}\|_p : \mathbf{x} \neq \mathbf{y} \in \mathcal{L}(\mathbf{B})\} = \min\{\|\mathbf{x}\|_p : \mathbf{x} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}\}.$$

Each norm gives rise to a corresponding computational problem SVP_γ^p (the γ-approximate *Shortest Vector Problem* in the ℓ_p norm): given a lattice $\mathcal{L}(\mathbf{B})$, find a nonzero vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}\|_p \leq \gamma\lambda_1^p(\mathcal{L}(\mathbf{B}))$. We also consider the restriction of SVP to specific classes of lattices. The restriction of SVP to some class of lattices C is denoted C-SVP. (E.g, [Mic07] considers *Cyclic*-SVP).

The notion of minimum distance can be generalized to define the i^{th} successive minimum (in the ℓ_p norm) $\lambda_i^p(\mathcal{L}(\mathbf{B}))$ as the smallest radius r such that the closed sphere $\bar{\mathcal{B}}_p(r) = \{\mathbf{x}\colon \|\mathbf{x}\|_p \leq r\}$ contains i linearly independent lattice points: $\lambda_i^p(\mathcal{L}(\mathbf{B})) = \min\{r : \dim(\mathrm{span}(\mathcal{L}(\mathbf{B}) \cap \bar{\mathcal{B}}_p(r))) \geq i\}$.

In this work, we focus on the infinity norm $\|\mathbf{x}\|_\infty = \lim_{p\to\infty} \|\mathbf{x}\|_p = \max_i |x_i|$ since it is the most natural and convenient norm when dealing with polynomials, but most of our results are easily translated to other norms as well. The shortest vector problem in the infinity norm $\mathrm{SVP}_\gamma^\infty$ was proved to be NP-hard by van Emde Boas for $\gamma = 1$ [van81] and shown to be NP-hard for factor up to $\gamma(n) = n^{1/\log\log n}$ by Dinur [Din02], where n is the dimension of the lattice. The asymptotically fastest algorithm for computing the shortest vector exactly takes time $2^{O(n)}$ [AKS01, BN07] and the best polynomial time algorithm approximates the shortest vector to within a factor of $2^{O(\frac{n\log\log n}{\log n})}$ [AKS01, Sch87, LLL82]. It is conjectured that approximating the shortest vector to within a polynomial factor is a hard problem, although it is shown that (under standard complexity assumptions) for small polynomial factors it is not NP-hard [AR05, GG00].

2.B Algebra and Conventions

Let $\mathbb{Z}[\mathbf{x}]$ and $\mathbb{R}[\mathbf{x}]$ be the sets of polynomials (in an indeterminate variable $\mathbf{x}$) with integer and real coefficients respectively. A polynomial is *monic* if the coefficient of the highest power of $\mathbf{x}$ is one. A polynomial (in $\mathbb{Z}[\mathbf{x}]$) is *irreducible* if it cannot be represented as a product of lower degree polynomials (in $\mathbb{Z}[\mathbf{x}]$). In this paper we identify polynomials (of degree less than n) with the corresponding n-dimensional vectors having the coefficients of the polynomial as coordinates. This allows to translate notation and definitions from one setting to the other. E.g., we define the ℓ_p norm $\|\mathbf{g}\|_p$ of a polynomial $\mathbf{g} \in \mathbb{Z}[\mathbf{x}]$ as the norm of the corresponding vector, and the product of two n-dimensional vectors $\mathbf{xy}$ as the $(2n-1)$-dimensional vector associated to the product of the corresponding polynomials.

Let R be a ring. An *ideal* I of R is an additive subgroup of R closed under multiplication by arbitrary ring elements. The smallest ideal of R containing a subset $S \subseteq R$ is denoted $\langle S \rangle$. In particular, for any ring element $\mathbf{f} \in R$, $\langle \mathbf{f} \rangle$ denotes the set of all multiples of $\mathbf{f}$. Two ring elements $\mathbf{g}, \mathbf{h} \in R$ are equivalent modulo an ideal $I \subseteq R$ if $\mathbf{g} - \mathbf{h} \in I$. When $I = \langle \mathbf{f} \rangle$ is the ideal generated by a single ring element $\mathbf{f}$, then we say that $\mathbf{g}$ and $\mathbf{h}$ are equivalent modulo $\mathbf{f}$. The quotient R/I is the set of all equivalence classes $(\mathbf{g} + I)$ of R modulo I.

Much of our work deals with the rings $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ where $\mathbf{f}$ is monic and irreducible. When $\mathbf{f}$ is a monic polynomial of degree n, every equivalence class $(\mathbf{g} + \langle \mathbf{f} \rangle) \in \mathbb{Z}[x]/\langle \mathbf{f} \rangle$ has a unique representative $\mathbf{g}' \in (\mathbf{g} + \langle \mathbf{f} \rangle)$ of degree less than n. This representative is denoted $(\mathbf{g} \bmod \mathbf{f})$ and can be efficiently computed using the standard division algorithm. The same holds true if we are dealing with the ring $\mathbb{R}[\mathbf{x}]/\langle \mathbf{f} \rangle$. That is, every equivalence class $(\mathbf{g} + \langle \mathbf{f} \rangle) \in \mathbb{R}[x]/\langle \mathbf{f} \rangle$ has a unique representative $\mathbf{g}' \in (\mathbf{g} + \langle \mathbf{f} \rangle)$ of degree less than n which is similarly denoted $(\mathbf{g} \bmod \mathbf{f})$. To avoid cumbersome notation, when we refer to elements $\mathbf{g} \in \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ (or $\mathbb{R}[\mathbf{x}]/\langle \mathbf{f} \rangle$), we will always be assuming that $\mathbf{g}$ is reduced modulo $\mathbf{f}$ (i.e. the degree of $\mathbf{g}$ is at most $n-1$). Another shorthand that we use is denoting the quotient ring $\mathbb{Z}[x]/\langle p, \mathbf{f} \rangle$ for some positive integer p and polynomial $\mathbf{f}$ as $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$.

When multiplying two polynomials $\mathbf{g}, \mathbf{h}$ is some ring, we will always assume that the product $\mathbf{gh}$ automatically gets reduced into that ring. So for example, if $\mathbf{g}, \mathbf{h} \in \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$, then we will assume that the product $\mathbf{gh}$ is reduced modulo $\mathbf{f}$. If, on the other hand, the domains of the multiplicands are different, we will assume that the product belongs to the larger domain. So for example, if $\mathbf{g} \in \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$, while $\mathbf{h} \in \mathbb{R}[\mathbf{x}]$, then the product $\mathbf{gh}$ will be a polynomial in $\mathbb{R}[\mathbf{x}]$ that is *not* reduced modulo $\mathbf{f}$.

Irreducible polynomials that satisfy a certain addtional property play a major role in our work. This additional property will be discussed in detail in Section 2.D, but here we will just prove the irreducibility of these polynomials.

The below lemma states that $\mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1$ is irreducible whenever n is a prime. This is a classical result, the proof of which can be found in standard algebra textbooks.

Lemma 2.2. *The polynomial* $\mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1$ *is irreducible over the integers if and only if* n *is prime.*

The next lemma states that the polynomial $\mathbf{x}^n + 1$ is irreducible if and only if n is a power of two. The following proof of this fact was communicated to us by Chris Peikert [Pei].

Lemma 2.3. *The polynomial* $\mathbf{x}^n + 1$ *is irreducible over the integers if and only if* $n = 2^k$ *for any* $k \geq 0$.

Proof. If $\Phi_d(\mathbf{x})$ represents the d^{th} cyclotomic polynomial[1], then any polynomial of the form $\mathbf{x}^n - 1$ can be written as

$$\mathbf{x}^n - 1 = \prod_{d|n} \Phi_d(\mathbf{x}).$$

Using the above, we can rewrite $\mathbf{x}^n + 1$ as

$$\mathbf{x}^n + 1 = \frac{\mathbf{x}^{2n} - 1}{\mathbf{x}^n - 1} = \frac{\prod_{d|2n} \Phi_d(\mathbf{x})}{\prod_{d|n} \Phi_d(\mathbf{x})} = \prod_{\{d: d|2n \wedge d \nmid n\}} \Phi_d(\mathbf{x})$$

Since all cyclotomic polynomials are irreducible, the above equality implies that $\mathbf{x}^n + 1$ will be irreducible if and only if the set $\{d : d|2n \wedge d \nmid n\}$ contains exactly one element, and this happens exactly when n is a power of 2. □

[1] The d^{th} cyclotomic polynomial is a polynomial whose roots are exactly the primitive d^{th} roots of unity with multiplicity 1

2.C The Polynomial Ring R

We will now describe the polynomial ring that will be central throughout this work. Let $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$ be a ring where p is some small odd prime (i.e. $p = poly(n)$) and $\mathbf{f}$ is a monic polynomial irreducible over $\mathbb{Z}$. For the rest of this thesis, the variables n and p will always be associated with the ring R. We will denote elements in R by bold letters and elements of R^m, for some positive integer m, by a bold letter with a hat. That is, $\widehat{\mathbf{a}} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in R^m$ when all the $\mathbf{a}_i$'s are in R. For an element $\widehat{\mathbf{a}} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in R^m$ and an element $\mathbf{z} \in R$, we define $\widehat{\mathbf{a}}\mathbf{z} = (\mathbf{a}_1\mathbf{z}, \ldots, \mathbf{a}_m\mathbf{z})$. For two elements $\widehat{\mathbf{a}}, \widehat{\mathbf{b}} \in R^m$, addition is defined as $\widehat{\mathbf{a}} + \widehat{\mathbf{b}} = (\mathbf{a}_1 + \mathbf{b}_1, \ldots, \mathbf{a}_m + \mathbf{b}_m)$ and the dot product as $\widehat{\mathbf{a}} \odot \widehat{\mathbf{b}} = \mathbf{a}_1\mathbf{b}_1 + \ldots + \mathbf{a}_m\mathbf{b}_m$.

Notice that with the operations that we defined, the set R^m is an R-module. That is, R^m is an abelian additive group such that for all $\widehat{\mathbf{a}}, \widehat{\mathbf{b}} \in R^m$ and $\mathbf{r}, \mathbf{s} \in R$, we have

1. $(\widehat{\mathbf{a}} + \widehat{\mathbf{b}})\mathbf{r} = \widehat{\mathbf{a}}\mathbf{r} + \widehat{\mathbf{b}}\mathbf{r}$
2. $(\widehat{\mathbf{a}}\mathbf{r})\mathbf{s} = \widehat{\mathbf{a}}(\mathbf{r}\mathbf{s})$
3. $\widehat{\mathbf{a}}(\mathbf{r} + \mathbf{s}) = \widehat{\mathbf{a}}\mathbf{r} + \widehat{\mathbf{a}}\mathbf{s}$

We will now give a definition for the "length" of elements in R. To do so, we will first need to specify their representations in the ring. For our application, we will represent elements in R by a polynomial of degree $n - 1$ having integer coefficients in the range $[-\frac{p-1}{2}, \frac{p-1}{2}]$, and so when we talk about reduction modulo p, we mean finding an equivalent element modulo p in the aforementioned range. For an element $\mathbf{a} = a_0 + a_1\mathbf{x} + \ldots + a_{n-1}\mathbf{x}^{n-1} \in R$, we define $\|\mathbf{a}\|_\infty = max_i(|a_i|)$. Similarly, for $\widehat{\mathbf{a}} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in R^m$, we define $\|\widehat{\mathbf{a}}\|_\infty = \max_i (\|\mathbf{a_i}\|_\infty)$. Notice that $\|\cdot\|_\infty$ is not exactly a norm because $\|\alpha\mathbf{a}\|_\infty \neq \alpha\|\mathbf{a}\|_\infty$ for all integers α (because of the reduction modulo p), but it still holds true that $\|\mathbf{a} + \mathbf{b}\|_\infty \leq \|\mathbf{a}\|_\infty + \|\mathbf{b}\|_\infty$ and $\|\alpha\mathbf{a}\|_\infty \leq \alpha\|\mathbf{a}\|_\infty$.

While putting an upper-bound on $\|\mathbf{a} + \mathbf{b}\|_\infty$ is straight-forward, it turns out that upper-bounding $\|\mathbf{ab}\|_\infty$ is somewhat more involved. Suppose that we are trying to determine the upper bound on $\|\mathbf{ab}\|_\infty$. For a moment, let's pretend that

$\mathbf{a}$ and $\mathbf{b}$ are polynomials in $\mathbb{Z}[x]$. Then, the product $\mathbf{ab}$ will have degree at most $2n-2$ and the absolute value of the maximum coefficient of $\mathbf{ab}$ will be at most $n\|\mathbf{a}\|_\infty\|\mathbf{b}\|_\infty$. Reducing $\mathbf{ab}$ modulo p will not increase the absolute value of the maximum coefficient, but reducing modulo the polynomial $\mathbf{f}$ can (and usually does). So if we want to upper bound $\|\mathbf{ab}\|_\infty$, we need to account for the increase in the coefficient size when we reduce a polynomial in $\mathbb{Z}[x]$ of degree $2n-2$ modulo $\mathbf{f}$. Dealing with this issue is the main topic of the next section.

2.D The Parameter $\theta(\mathbf{f})$, and Bounding $\|\mathbf{ab} \bmod \mathbf{f}\|_\infty$

In the previous section, we were interested in bounding the maximum coefficient of the product $\mathbf{ab}$ in $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$. In this section, we will deal with this general question. In other words, given two polynomials $\mathbf{a}, \mathbf{b} \in \mathbb{R}[\mathbf{x}]$ of degree n, and a monic polynomial $\mathbf{f} \in \mathbb{Z}[\mathbf{x}]$, what is the maximum that $\|\mathbf{ab} \bmod \mathbf{f}\|_\infty$ can be? It turns out that for certain $\mathbf{f}$, the product of $\|\mathbf{ab} \bmod \mathbf{f}\|_\infty$ could actually be exponentially larger than $\|\mathbf{a}\|_\infty$ and $\|\mathbf{b}\|_\infty$. For example, if $\mathbf{f} = \mathbf{x}^n - 2\mathbf{x}^{n-1}$, then if $\mathbf{a} = \mathbf{b} = \mathbf{x}^{n-1}$, we have $\|\mathbf{ab} \bmod \mathbf{f}\|_\infty = 2^{n-1}\mathbf{x}^{n-1}$. Throughout our work, it will be very important that the polynomials $\mathbf{f}$ do not behave in such a way, and in this section we will explain how to test whether a polynomial exhibits such behavior. Throughout this section, we will assume that $\mathbf{a}, \mathbf{b} \in \mathbb{R}[\mathbf{x}]$, but all the results apply equally well to polynomials $\mathbf{a}, \mathbf{b}$ in other rings such as $\mathbb{Z}[\mathbf{x}], \mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$, and $\mathbb{Z}_p[\mathbf{x}]/\langle\mathbf{f}\rangle$.

It turns out that the behavior of multiplication of two polynomials modulo $\mathbf{f}$ very much depends on the behavior of the multiplication of a polynomial with a power of $\mathbf{x}$.

Definition 2.4.

$$\theta(\mathbf{f}) = \min\{j : \forall \mathbf{a} \in \mathbb{R}[\mathbf{x}] \textit{ of degree} < n \textit{ and } 0 \le i \le n-1, \|\mathbf{ax}^i \bmod \mathbf{f}\|_\infty \le j\|\mathbf{a}\|_\infty\}$$

It is not clear from the above definition that we can determine the value $\theta(\mathbf{f})$ when given an $\mathbf{f}$, but in the next lemma we show that we can obtain an upper-bound for $\theta(\mathbf{f})$ by performing just a few modular reductions.

Lemma 2.5.

$$\theta(\mathbf{f}) \leq n \cdot \max_{0 \leq i \leq 2n-2} [\|\mathbf{x}^i \bmod \mathbf{f}\|_\infty]$$

Proof. For any $\mathbf{a} \in \mathbb{R}[\mathbf{x}]$ of degree less than n, we can rewrite $\|\mathbf{a}\mathbf{x}^\mathbf{i} \bmod \mathbf{f}\|_\infty$, for any $0 \leq i \leq n-1$ as

$$\|\mathbf{a}\mathbf{x}^\mathbf{i} \bmod \mathbf{f}\|_\infty = \|a_0\mathbf{x}^i + \ldots + a_{n-1}\mathbf{x}^{i+n-1} \bmod \mathbf{f}\|_\infty \leq n\|\mathbf{a}\|_\infty \max_{0 \leq j \leq 2n-2} \|\mathbf{x}^j \bmod \mathbf{f}\|_\infty,$$

and by definition of $\theta(\mathbf{f})$, we have the claim in the lemma. □

The above bound on $\theta(\mathbf{f})$ is very general, and does not take into account any specific structure that the polynomial $\mathbf{f}$ might have, and it turns out that we can obtain better bounds for some polynomials which will be important throughout the thesis. For example, if $\mathbf{f} = \mathbf{x}^n - 1$, Lemma 2.5 tells us that $\theta(\mathbf{f}) \leq n$, but in fact $\theta(\mathbf{f}) = 1$. This looseness in the bound exists for other polynomials that will be used in our work, such as $\mathbf{x}^n + 1$ and the cyclotomic polynomial $\mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1$. The next lemmas give tight bound on the value of $\theta(\mathbf{f})$ for these polynomials.

Lemma 2.6. $\theta(\mathbf{x}^n - 1) = \theta(\mathbf{x}^n + 1) = 1$.

Proof. Notice that when $\mathbf{f} = \mathbf{x}^n - 1$, the product $\mathbf{b}\mathbf{x} \bmod \mathbf{f}$ for any polynomial $\mathbf{b} = b_0 + \ldots + b_{n-2}\mathbf{x}^{n-2} + b_{n-1}\mathbf{x}^{n-1}$ is $b_{n-1} + b_0\mathbf{x} + \ldots + b_{n-2}\mathbf{x}^{n-1}$, and so $\|\mathbf{b}\mathbf{x} \bmod \mathbf{f}\|_\infty = \|\mathbf{b}\|_\infty$. Thus $\|\mathbf{b}\mathbf{x}^\mathbf{i} \bmod \mathbf{f}\|_\infty = \|\mathbf{b}\|_\infty$. The proof for when $\mathbf{f} = \mathbf{x}^n + 1$ is very similar. Note that $\mathbf{b}\mathbf{x} \bmod \mathbf{f}$ for any polynomial $\mathbf{b} = b_0 + \ldots + b_{n-2}\mathbf{x}^{n-2} + b_{n-1}\mathbf{x}^{n-1}$ is $-b_{n-1} + b_0\mathbf{x} + \ldots + b_{n-2}\mathbf{x}^{n-1}$, and so $\|\mathbf{b}\mathbf{x} \bmod \mathbf{f}\|_\infty = \|\mathbf{b}\|_\infty$. □

Lemma 2.7. *If* $\mathbf{f} = \mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1$*, then* $\theta(\mathbf{f}) = 2$.

Proof. Let $\mathbf{a}$ be any polynomial in $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$. We need to show that for all $0 \leq i \leq n-2$, $\|\mathbf{a}\mathbf{x}^\mathbf{i} \bmod \mathbf{f}\|_\infty \leq 2\|\mathbf{a}\|_\infty$. Notice that since $\mathbf{f}$ is a factor of $\mathbf{x}^n - 1$, we can write

$$\mathbf{a}\mathbf{x}^i \bmod \mathbf{f} = \left(\mathbf{a}\mathbf{x}^i \bmod \mathbf{x}^n - 1\right) \bmod \mathbf{f}.$$

If we define $\mathbf{a}' = \mathbf{a}\mathbf{x}^i \bmod \mathbf{x}^n - 1$, then the degree of $\mathbf{a}'$ is at most $n-1$, and by Lemma 2.6 we know that $\|\mathbf{a}'\|_\infty = \|\mathbf{a}\|_\infty$. Now we just need to reduce $\mathbf{a}'$ modulo $\mathbf{f}$. If a_{n-1} is the coefficient of $\mathbf{a}'$ corresponding to the term $\mathbf{x}^{n-1}$, then $\mathbf{a}' \bmod \mathbf{f} = \mathbf{a}' - a_{n-1}\mathbf{f}$. Therefore

$$\|\mathbf{a}' \bmod \mathbf{f}\|_\infty \leq \|\mathbf{a}'\|_\infty + |a_{n-1}|\|\mathbf{f}\|_\infty \leq \|\mathbf{a}'\|_\infty + \|\mathbf{a}'\|_\infty = 2\|\mathbf{a}'\|_\infty,$$

and so $\|\mathbf{ax^i}\|_\infty \leq 2\|\mathbf{a}\|_\infty$. □

We are now almost ready to put a bound on the product of two polynomials modulo $\mathbf{f}$, but first we note a convenient way of looking at the multiplication of two polynomials modulo $\mathbf{f}$. A multiplication of two polynomials $\mathbf{a} = a_0 + a_1\mathbf{x} + \ldots + a_{n-1}\mathbf{x}^{n-1}$ and $\mathbf{b} = b_0 + b_1\mathbf{x} + \ldots + b_{n-1}\mathbf{x}^{n-1}$ in $\mathbb{R}[\mathbf{x}]$ modulo $\mathbf{f}$ can written as

$$a_0\mathbf{b} + a_1\mathbf{bx} + \ldots + a_{n-1}\mathbf{bx}^{n-1} \bmod \mathbf{f}, \tag{2.1}$$

and thus it can be written as a vector/matrix multiplication over $\mathbb{R}$ between a $1 \times n$ vector consisting of the coefficients of $\mathbf{a}$ and an $n \times n$ matrix whose i^{th} row corresponds to the polynomial $\mathbf{bx}^i \bmod \mathbf{f}$. Using this interpretation of polynomial multiplication, we can give a bound on $\|\mathbf{ab} \bmod \mathbf{f}\|_\infty$.

Lemma 2.8.

$$\|\mathbf{ab} \bmod \mathbf{f}\|_\infty \leq n\theta(\mathbf{f})\|\mathbf{a}\|_\infty\|\mathbf{b}\|_\infty$$

Proof. Using equation (2.1), we can write

$$\begin{aligned}\|\mathbf{ab} \bmod \mathbf{f}\|_\infty &= \|a_0\mathbf{b} + a_1\mathbf{bx} + \ldots + a_{n-1}\mathbf{bx}^{n-1} \bmod \mathbf{f}\|_\infty \\ &\leq n\|\mathbf{a}\|_\infty \max_{0\leq i\leq n-1} \|\mathbf{bx^i} \bmod \mathbf{f}\|_\infty.\end{aligned}$$

By definition of $\theta(\mathbf{f})$, we know that $\max_{0\leq i\leq n-1} \|\mathbf{bx^i} \bmod \mathbf{f}\|_\infty \leq \theta(\mathbf{f})\|\mathbf{b}\|_\infty$, and so we have

$$\|\mathbf{ab} \bmod \mathbf{f}\|_\infty \leq n\theta(\mathbf{f})\|\mathbf{a}\|_\infty\|\mathbf{b}\|_\infty,$$

and we have the claim in the lemma. □

We now turn to the problem in which one of the multiplicands ($\mathbf{a}$ or $\mathbf{b}$) is chosen at random. We will show that if the coefficients of $\mathbf{b}$ are chosen randomly with mean 0, then with high probability, $\|\mathbf{ab}\|_\infty \approx \sqrt{n}\theta(\mathbf{f})\|\mathbf{a}\|_\infty\|\mathbf{b}\|_\infty$. Before proceeding, we will state the well-known Hoeffding bound as well as another lemma that will also be useful in other contexts.

Lemma 2.9 (Hoeffding Bound)**.** *Let $X_1, \ldots, X_n$ be independent random variables with mean μ taking values in the real interval $[a, b]$ and let $X = X_1 + \ldots + X_n$. Then for any k, we have*

$$Pr[|X - \mu n| \geq k] \leq 2e^{\frac{-2k^2}{n(b-a)^2}}$$

Lemma 2.10. *Let* $\mathbf{z}$ *be a polynomial of degree at most* $2n-2$ *in* $\mathbb{R}[\mathbf{x}]$*. Then*

$$\|\mathbf{z} \bmod \mathbf{f}\|_\infty \leq (1+\theta(\mathbf{f}))\|\mathbf{z}\|_\infty$$

Proof. We can rewrite $\mathbf{z} = \mathbf{z}_1 + \mathbf{z}_2$ where $\mathbf{z}_1 = z_0 + z_1\mathbf{x} + \ldots + z_{n-1}\mathbf{x}^{n-1}$ and $\mathbf{z}_2 = z_n\mathbf{x}^n + \ldots + z_{2n-2}\mathbf{x}^{2n-2}$. Now we can write,

$$\begin{aligned}
\|\mathbf{z} \bmod \mathbf{f}\|_\infty = \|\mathbf{z}_1 + \mathbf{z}_2 \bmod \mathbf{f}\|_\infty &\leq \|\mathbf{z_1}\|_\infty + \|\mathbf{z_2} \bmod \mathbf{f}\|_\infty \\
&= \|\mathbf{z_1}\|_\infty + \|(z_n\mathbf{x} + \ldots + z_{2n-2}\mathbf{x}^{n-1})\mathbf{x}^{n-1} \bmod \mathbf{f}\|_\infty \\
&\leq \|\mathbf{z_1}\|_\infty + \theta(\mathbf{f})\|z_n\mathbf{x} + \ldots + z_{2n-2}\mathbf{x}^{n-1}\|_\infty \\
&= \|\mathbf{z_1}\|_\infty + \theta(\mathbf{f})\|\mathbf{z_2}\|_\infty \\
&\leq (1+\theta(\mathbf{f}))\|\mathbf{z}\|_\infty
\end{aligned}$$

□

Lemma 2.11. *Let* $\mathbf{a}$ *be any polynomial in* $\mathbb{R}[\mathbf{x}]$ *of degree less than* n*. Let* $\mathbf{b}$ *be a polynomial in* $\mathbb{R}[\mathbf{x}]$ *of degree less than* n *where every coefficient of* $\mathbf{b}$ *is uniformly and independently distributed in the range* $[-b,b]$ *with mean* 0*. Then*

$$Pr[\|\mathbf{ab} \bmod \mathbf{f}\|_\infty \geq \theta(\mathbf{f})b\|\mathbf{a}\|_\infty\sqrt{n}\log n] \leq 4ne^{\frac{-\log^2 n}{8}}$$

Proof. The product of $\mathbf{a} = a_0 + a_1\mathbf{x} + \ldots + a_{n-1}\mathbf{x}^{n-1}$ and $\mathbf{b} = b_0 + b_1\mathbf{x} + \ldots + b_{n-1}\mathbf{x}^{n-1}$ will have the form $\mathbf{z} = z_0 + z_1\mathbf{x} + \ldots + z_{n-2}\mathbf{x}^{n-2}$ where

$$z_i = \sum_{j=0}^{i} a_j b_{i-j} \text{ (we define } a_i, b_i = 0 \text{ for } i \geq n).$$

Since every coefficient of $\mathbf{b}$ is an independent random variable in the range $[-b,b]$ with mean 0, it implies that every z_i is a sum of at most n independent random variables in the range $[-b\|\mathbf{a}\|_\infty, b\|\mathbf{a}\|_\infty]$ with mean 0. Applying the Hoeffding Bound, we get that for every i,

$$Pr[|z_i| \geq \frac{1}{2}b\|\mathbf{a}\|_\infty\sqrt{n}\log n] \leq 2e^{\frac{-\log^2 n}{8}}.$$

Taking the union bound over all i, we have,

$$Pr[\exists i, |z_i| \geq \frac{1}{2}b\|\mathbf{a}\|_\infty\sqrt{n}\log n] \leq 4ne^{\frac{-\log^2 n}{8}}.$$

So we have determined that all the coefficients of $\mathbf{z}$ will have small coefficients, and now we need to show that $\mathbf{z} \bmod \mathbf{f}$ has small coefficients as well. For this, we apply Lemma 2.10, and obtain that with probability at least $1 - 4ne^{\frac{-\log^2 n}{8}}$,

$$\|\mathbf{z} \bmod \mathbf{f}\|_\infty \leq (1 + \theta(\mathbf{f}))\|\mathbf{z}\|_\infty \leq 2\theta(\mathbf{f})\|\mathbf{z}\|_\infty \leq \theta(\mathbf{f})b\|\mathbf{a}\|_\infty \sqrt{n} \log n.$$

□

2.E Ideal Lattices

While general integer lattices are just additive subgroups of $\mathbb{Z}^n$, ideal lattices are lattices that are also closed under a multiplication operation. So while general lattices are subgroups of a group, ideal lattices are ideals of some ring. The ring that we will be using is the polynomial ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ where $\mathbf{f}$ is a monic polynomial of degree n. If we have an ideal $I = \langle \mathbf{g}_1, \ldots, \mathbf{g}_m \rangle$ in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$, then we can also represent the set I as the set of all integer combinations of the elements

$$\mathbf{g}_1, \mathbf{g}_1\mathbf{x}, \ldots, \mathbf{g}_1\mathbf{x}^{n-1}, \ldots, \mathbf{g}_m, \mathbf{g}_m\mathbf{x}, \ldots, \mathbf{g}_m\mathbf{x}^{n-1},$$

and using the Hermite Normal Form algorithm (cf. [Coh96]), we can find a set of at most n linearly-independent elements of I that form a basis for I . And so we can think of I as being an integer lattice of dimension n.

We will say that a vector $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{Z}^n$ *corresponds* to a polynomial $\mathbf{w} = w_0 + w_1\mathbf{x} + \ldots + w_{n-1}\mathbf{x}^{n-1} \in \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ if for all i, $v_i = w_i$. Similarly, a lattice Λ and an ideal I *correspond* to each other if every vector $(v_0, \ldots, v_{n-1})$ is in Λ, if and only if the polynomial $v_0 + v_1\mathbf{x} + \ldots + v_{n-1}\mathbf{x}^{n-1}$ is in I. Thus an *ideal lattice* is a lattice that corresponds to an ideal in some specified ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$.

An important feature of ideal lattices in $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ for irreducible $\mathbf{f}$ is that they have to be full-rank.

Lemma 2.12. *Let Λ be a lattice corresponding to a non-zero ideal in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ where $\mathbf{f}$ is a monic, irreducible polynomial. Then Λ is a full-rank lattice of dimension n.*

Proof. Let $I = \langle \mathbf{g}_1, \ldots, \mathbf{g}_m \rangle$ be the ideal of $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ that the lattice Λ corresponds to. One of the $\mathbf{g}_i$'s must be non-zero, so assume it's $\mathbf{g}_1$. We will show that the vectors corresponding to the ring elements $\mathbf{g}_1, \mathbf{g}_1\mathbf{x}, \ldots, \mathbf{g}_1\mathbf{x}^{n-1}$ are linearly independent over $\mathbb{Z}$. This will show that the lattice corresponding to I contains n linearly independent vectors, and thus must have dimension n.

If $\mathbf{g}_1, \mathbf{g}_1\mathbf{x}, \ldots, \mathbf{g}_1\mathbf{x}^{n-1}$ are linearly dependent, then $\mathbf{g}_1(a_0 + a_1\mathbf{x} + \ldots + a_{n-1}\mathbf{x}^{n-1}) = \mathbf{fh} \in \langle \mathbf{f} \rangle$ for some polynomial $\mathbf{h} \in \mathbb{Z}[\mathbf{x}]$. Since $\mathbf{f}$ is irreducible and $\mathbb{Z}[\mathbf{x}]$ is a unique factorization domain, $\mathbf{f}$ is also prime. Thus either $\mathbf{f}|\mathbf{g}_1$ or $\mathbf{f}|a_0 + a_1\mathbf{x} + \ldots + a_{n-1}\mathbf{x}^{n-1}$. But both of those polynomials have degree less than $\mathbf{f}$, and since $\mathbf{f}$ is irreducible, this cannot be unless either $\mathbf{g}_1$ or $a_0 + a_1\mathbf{x} + \ldots + a_{n-1}\mathbf{x}^{n-1}$ is 0. □

Notice that in the proof of the previous theorem, we made an observation that the vectors corresponding to polynomials $\mathbf{g}, \mathbf{gx}, \ldots, \mathbf{gx}^{n-1}$ are linearly independent for any non-zero polynomial $\mathbf{g}$. So if $\mathbf{g}$ happens to correspond to the shortest vector of the lattice, and $\theta(\mathbf{f})$ is small, then all the vectors $\mathbf{g}, \mathbf{gx}, \ldots, \mathbf{gx}^{n-1}$ are short, and so $\lambda_n(\Lambda)$ will be small as well. This observation leads to the following lemma.

Lemma 2.13. *For all lattices Λ corresponding to some ideal of $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$, where $\mathbf{f}$ is a monic, irreducible polynomial of degree n, we have $\lambda_n^\infty(\Lambda) \leq \theta(\mathbf{f})\lambda_1^\infty(\Lambda)$*

Proof. Let $\mathbf{g}$ be the polynomial n such that $\|\mathbf{g}\|_\infty = \lambda_1^\infty(\Lambda)$. Then the polynomials $\mathbf{g}, \mathbf{gx}, \ldots, \mathbf{gx}^{n-1}$ are linearly independent. And by definition of $\theta(\mathbf{f})$, we have that

$$\lambda_n^\infty(\Lambda) \leq \max_i \|\mathbf{gx^i}\|_\infty \leq \theta(\mathbf{f})\|\mathbf{g}\|_\infty \leq \theta(\mathbf{f})\lambda_1^\infty(\Lambda)$$

□

We now define the shortest vector problem when restricted to lattices corresponding to ideals in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ for some monic polynomial $\mathbf{f}$.

Definition 2.14. *For any $\gamma \geq 1$, monic polynomial $\mathbf{f}$, and a lattice Λ corresponding to an ideal in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$, the $\mathbf{f}$-SVP$_\gamma(\Lambda)$ problem asks to find an element $\mathbf{g}$ in Λ such that $\|\mathbf{g}\|_\infty \leq \gamma\lambda_1^\infty(\Lambda)$.*

Even though ideal lattices have more structure than general lattices, there are no known algorithms that can take any significant advantage of it (see [Mic07] for a discussion of this). Therefore it is reasonable to make the conjecture that solving $\mathbf{f}$-SVP is as hard as solving SVP.

Conjecture 2.1. *For any monic polynomial* $\mathbf{f}$ *of degree* n *and any constant* c*, solving the* $\mathbf{f}$*-SVP*$_\gamma$ *problem in the worst case requires* $2^{\Omega(n)}$ *time when* $\gamma = O(n^c)$*.*

2.F The Hash Function and the Collision Problem

In this section, we will present the hash function whose collision-resistance we will prove in Chapter 3. This hash function will also serve as a building block for all the other cryptographic constructions in this thesis.

Definition 2.15. *For any* $D \subseteq R$*, the function family* $\mathcal{H}(R, D, m)$ *mapping* D^m *to* R *is defined as*

$$\mathcal{H}(R, D, m) = \{h_{\widehat{\mathbf{a}}} : \widehat{\mathbf{a}} \in R^m\}, \text{ where for any } \widehat{\mathbf{z}} \in D^m,\ h_{\widehat{\mathbf{a}}}(\widehat{\mathbf{z}}) = \widehat{\mathbf{a}} \odot \widehat{\mathbf{z}}.$$

If $D = R$*, we will simply denote the functions as* $\mathcal{H}(R, m)$*. Notice that the families* $\mathcal{H}(R, m)$ *and* $\mathcal{H}(R, D, m)$ *are exactly the same, and so the* D *in* $\mathcal{H}(R, D, m)$ *is merely used for convenience to remind the reader that we are considering the restriction of the domain to* D^m*.*

Throughout the paper, we will write h rather than $h_{\widehat{\mathbf{a}}}$ with the understanding that there is an $\widehat{\mathbf{a}}$ associated with the function h.

The efficiency of all the cryptographic primitives that we will describe in this thesis is due to the fact that for any $h \in \mathcal{H}(R, m)$, computing $h(\widehat{\mathbf{z}})$ takes time $m\tilde{O}(n)$.

Claim 2.16.

1. *For any* $\mathbf{f}$ *and* $p = n^{O(1)}$*, any two elements in* $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$ *can be multiplied in time* $\tilde{O}(n)$*.*

2. *For any* $\mathbf{f}$ *and* $p = n^{O(1)}$ *and* $h \in \mathcal{H}(R, m)$ *for* $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$, $h(\widehat{\mathbf{z}})$ *can be computed in time* $m\tilde{O}(n)$.

Proof. Computing $h(\widehat{\mathbf{z}})$ requires m additions of the products $\mathbf{a}_i\mathbf{z}_i$. Since $\mathbf{a}_i$ and $\mathbf{z}_i$ are polynomials of degree n with logarithmic-length coefficients, their product can be computed in time $\tilde{O}(n)$ using the Fast Fourier Transform. □

We now make the observation that the functions in $\mathcal{H}(R, m)$ are module homomorphisms.

Claim 2.17. *$\mathcal{H}(R, m)$ is a set of module homomorphisms. That is, for every $\widehat{\mathbf{y}}, \widehat{\mathbf{z}} \in R^m$, $\mathbf{c} \in R$, and $h \in \mathcal{H}(R, m)$, the following two conditions are satisfied:*

1. $h(\widehat{\mathbf{y}} + \widehat{\mathbf{z}}) = h(\widehat{\mathbf{y}}) + h(\widehat{\mathbf{z}})$

2. $h(\widehat{\mathbf{y}}\mathbf{c}) = h(\widehat{\mathbf{y}})\mathbf{c}$

Proof. By the definition of the hash function h, we have

1. $h(\widehat{\mathbf{y}} + \widehat{\mathbf{z}}) = \widehat{\mathbf{a}} \odot (\widehat{\mathbf{y}} + \widehat{\mathbf{z}}) = \widehat{\mathbf{a}} \odot \widehat{\mathbf{y}} + \widehat{\mathbf{a}} \odot \widehat{\mathbf{z}} = h(\widehat{\mathbf{y}}) + h(\widehat{\mathbf{z}})$

2. $h(\widehat{\mathbf{y}}\mathbf{c}) = \widehat{\mathbf{a}} \odot (\mathbf{y}_1\mathbf{c}, \ldots, \mathbf{y}_m\mathbf{c}) = \mathbf{a}_1\mathbf{y}_1\mathbf{c} + \ldots + \mathbf{a}_m\mathbf{y}_m\mathbf{c} = (\mathbf{a}_1\mathbf{y}_1 + \ldots + \mathbf{a}_m\mathbf{y}_m)\mathbf{c} = (\widehat{\mathbf{a}} \odot \widehat{\mathbf{y}})\mathbf{c} = h(\widehat{\mathbf{y}})\mathbf{c}$

□

Another observation that will be useful in Chapter 4 is that the kernel of h contains many elements that have small norm.

Lemma 2.18. *For every positive integer c and for every $h \in \mathcal{H}(R, m)$, there exist at least c^{mn} elements $\widehat{\mathbf{y}} \in R^m$ such that $\|\widehat{\mathbf{y}}\|_\infty \leq cp^{1/m}$ and $h(\widehat{\mathbf{y}}) = \mathbf{0}$.*

Proof. Let S be the set containing all elements in R^m with coefficients between 0 and $cp^{1/m}$. Since $|S| = (\lfloor cp^{1/m} \rfloor + 1)^{mn} > c^{mn}p^n$ and $|R| = p^n$, by the pigeonhole principle, there exists a $\mathbf{t} \in R$ and a subset $S' \subseteq S$ such that $|S'| > c^{mn}$ and for all $\widehat{\mathbf{s}}' \in S'$, $h(\widehat{\mathbf{s}}') = \mathbf{t}$. Let $S' = \{\widehat{\mathbf{s}}'_1, \widehat{\mathbf{s}}'_2, \ldots, \widehat{\mathbf{s}}'_k\}$ and consider the set $Y = \{\widehat{\mathbf{s}}'_1 - \widehat{\mathbf{s}}'_1, \widehat{\mathbf{s}}'_1 - \widehat{\mathbf{s}}'_2, \ldots, \widehat{\mathbf{s}}'_1 - \widehat{\mathbf{s}}'_k\}$ of size $|S'|$. Note that for each $\widehat{\mathbf{y}} \in Y$, $\|\widehat{\mathbf{y}}\|_\infty \leq cp^{1/m}$ and $h(\widehat{\mathbf{y}}) = \mathbf{0}$ because of the homomorphic property of h. □

We will define the collision problem Col(h, D) as follows:

Definition 2.19. *Given an element $h \in \mathcal{H}(R, m)$, the collision problem Col(h, D) (where $D \subseteq R$) asks to find two distinct elements $\widehat{\mathbf{z}}, \widehat{\mathbf{z}}' \in D^m$ such that $h(\widehat{\mathbf{z}}) = h(\widehat{\mathbf{z}}')$.*

2.G Statistical Distance

Let X and Y be random variables over a set A with probability density functions δ_X and δ_Y respectively. The statistical distance between X and Y, denoted $\Delta(X, Y)$, is

$$\Delta(X, Y) = \frac{1}{2} \int_{a \in A} |\delta_X(a) - \delta_Y(a)| da.$$

If the set A is a discrete set, then the statistical distance $\Delta(X, Y)$ can be rewritten as

$$\Delta(X, Y) = \frac{1}{2} \sum_{a \in A} |Pr[X = a] - Pr[Y = a]|.$$

The statistical distance satisfies the following useful properties:

$$\begin{aligned} \Delta(f(X), f(Y)) &\leq \Delta(X, Y) && (2.2) \\ \Delta((X_1, \ldots, X_k), (Y_1, \ldots, Y_k)) &\leq \sum_{i=1}^{k} \Delta(X_i, Y_i) && (2.3) \\ & && (2.4) \end{aligned}$$

for any function f and independent random variables $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_k$.

2.H Algorithms and Asymptotic Notation

An *algorithm* is a sequence of steps that produces some output. If A is an algorithm, then $A(x)$ denotes the output of A when it is given the input x. A *randomized (or probabilistic)* algorithm A is an algorithm that is allowed to have access to a purely random bit-generator during its execution, or equivalently, it is a deterministic algorithm that is given a uniformly random bit-string ρ as an auxiliary input. We will sometimes write $A(x; \rho)$ to denote that the input is x and the random string is ρ. We will use the notation $y \overset{\$}{\leftarrow} S$ to denote that the value for y is being chosen uniformly at random from the set S.

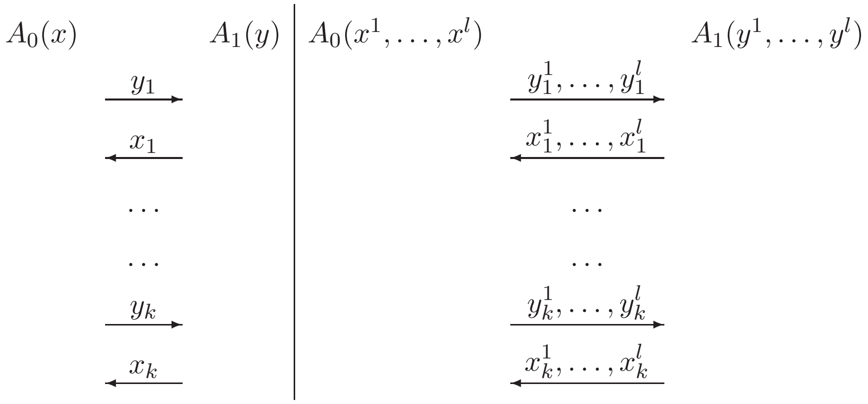

Figure 2.1 On the left is an interactive protocol between algorithms A_0 and A_1 whose inputs are x and y, respectively. On the right is a parallel execution of l protocols between A_0 and A_1.

An *interactive algorithm* is an algorithm that, before producing its final output, may produce some intermediate outputs or wait for additional inputs. An interactive protocol between two interactive algorithms A_0 and A_1, denoted (A_0, A_1), is a protocol where the intermediate outputs of A_i are used as intermediate inputs of algorithm A_{1-i}. The output of the protocol consists of all the outputs produced by the two algorithms. If algorithms A_0 and A_1 are given inputs x and y, respectively, at the start of the protocol, then the protocol is denoted as $(A_0(x), A_1(y))$. Running l copies of a protocol between A_0 and A_1*sequentially* means performing the interactions $(A_0(x^1), A_1(y^1)), \dots, (A_0(x^l), A_1(x^l))$ one after the other. A *parallel* composition of l copies of a protocol between A_0 and A_1 involves outputting all intermediate results for all l copies of the protocol at every step where intermediate results are outputted (see Figure 2.1).

We use standard asymptotic notation symbols O, o, Ω, ω, and Θ to measure the running-time complexity of algorithms. We recall their definitions here.

Definition 2.20.

- $f(x) = O(g(x))$ *if* $\lim\limits_{x\to\infty} \frac{f(x)}{g(x)} \neq \infty$.
- $f(x) = o(g(x))$ *if* $\lim\limits_{x\to\infty} \frac{f(x)}{g(x)} = 0$.
- $f(x) = \Omega(g(x))$ *if* $\lim\limits_{x\to\infty} \frac{f(x)}{g(x)} \neq 0$.

- *$f(x) = \omega(g(x))$ if $\lim\limits_{x\to\infty} \frac{f(x)}{g(x)} = \infty$.*
- *$f(x) = \Theta(g(x))$ if $\lim\limits_{x\to\infty} \frac{f(x)}{g(x)} = c$, where c is some constant.*

We will often use the "soft-Oh" (i.e., $\tilde{O}$) notation to suppress poly-logarithmic factors. For example, if $f(n) = O(n^2 \log^3 n)$, we may simply write $f(n) = \tilde{O}(n^2)$.

2.I Witness Indistinguishability

The concept of *witness indistinguishability* was introduced by Feige and Shamir in [FS90]. For a string x and relation R, a witness set $W_R(x)$ consists of all strings w such that $R(w, x) = 1$. For example, x could be a boolean formula and the relation R could be defined as $R(x, w) = 1$ iff w is an assignment that makes x evaluate to 1. Then the set $W_R(x)$ is the set of all assignments that make x evaluate to 1. In our case, the witness will correspond to the secret key and the string x is the public key.

Let $\mathcal{P}$ and $\mathcal{V}$ be two randomized interactive algorithms and $(\mathcal{P}, \mathcal{V})$ be a protocol between $\mathcal{P}$ and $\mathcal{V}$. We denote by $\mathcal{V}_{\mathcal{P}(x,w)}(x, y)$ the output of $\mathcal{V}$ after participating in the protocol $(\mathcal{P}, \mathcal{V})$. We say that $(\mathcal{P}, \mathcal{V})$ is *statistically* witness-indistinguishable if for all $\mathcal{V}'$, all large enough x, any y, and any two $w, w' \in W_R(x)$,

$$\Delta\left(\mathcal{V}'_{\mathcal{P}(x,w)}(x, y), \mathcal{V}'_{\mathcal{P}(x,w')}(x, y)\right) < 2^{-\omega(\log|x|)}.$$

In other words, every cheating verifier $\mathcal{V}'$ with any auxiliary input y, cannot distinguish whether the witness that $\mathcal{P}$ is using in the protocol is w or w'. An important feature of witness indistinguishability is that it is closed under parallel composition. In other words, if the protocol $(\mathcal{P}, \mathcal{V})$ is witness-indistinguishable, then running polynomially-many copies of the protocol in parallel is witness-indistinguishable as well. If

$$\Delta\left(\mathcal{V}'_{\mathcal{P}(x,w)}(x, y), \mathcal{V}'_{\mathcal{P}(x,w')}(x, y)\right) = 0,$$

we say that the protocol $(\mathcal{P}, \mathcal{V})$ is *perfectly* witness-indistinguishable.

2.J Gaussian Distributions

In this work, we will be using techniques developed in [AR05,Reg03,MR07] that involve Gaussian distributions over lattices. In this section we recall all the relevant definitions and results from [MR07]. We will also present some new results about gaussian distributions on lattices.

For any vectors $\mathbf{c}, \mathbf{x}$ and any $s > 0$, let $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2}$ be a Gaussian function centered in $\mathbf{c}$ scaled by a factor of s. The total measure associated to $\rho_{s,\mathbf{c}}$ is $\int_{\mathbf{x}\in\mathbb{R}^n} \rho_{s,\mathbf{c}}(\mathbf{x})d\mathbf{x} = s^n$. So, $\int_{\mathbf{x}\in\mathbb{R}^n}(\rho_{s,\mathbf{c}}(\mathbf{x})/s^n)d\mathbf{x} = 1$ and $\rho_{s,\mathbf{c}}/s^n$ is a probability density function.

The distribution $\rho_{s,\mathbf{c}}/s^n$ can be efficiently approximated using standard techniques (see [MR07]), so in the rest of the paper we make the simplifying assumption that we can sample from $\rho_{s,\mathbf{c}}/s^n$ exactly and work with real numbers.

Functions are extended to sets in the usual way; e.g., $\rho_{s,\mathbf{c}}(A) = \sum_{\mathbf{x}\in A} \rho_{s,\mathbf{c}}(\mathbf{x})$ for any countable set A. For any $s, \mathbf{c}$ and lattice Λ, define the discrete probability distribution (over the lattice Λ) $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}$, where $\mathbf{x} \in \Lambda$. Intuitively, $D_{\Lambda,s,\mathbf{c}}$ is the conditional probability[2] that $(\rho_{s,\mathbf{c}}/s^n) = \mathbf{x}$ given $(\rho_{s,\mathbf{c}}/s^n) \in \Lambda$. For brevity, we sometimes omit s or $\mathbf{c}$ from the notation $\rho_{s,\mathbf{c}}$ and $D_{\Lambda,s,\mathbf{c}}$. When $\mathbf{c}$ or s are not specified, we assume that they are the origin and 1 respectively.

In [MR07], Gaussian distributions are used to define a new lattice invariant called the *smoothing parameter*.

Definition 2.21. *For an n-dimensional lattice Λ, and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest s such that $\rho_{1/s}(\Lambda^* \setminus \{0\}) \le \epsilon$.*

In [MR07], many important properties of the smoothing parameter are established. Here we only need the following few bounds. The first one shows that the smoothing parameter is the amount of Gaussian noise that needs to be added to a lattice in order to get an almost uniform distribution.

Lemma 2.22 ([MR07, Lemma 4.1])**.** *Let $\rho_s/s^n \bmod \mathbf{B}$ be the distribution obtained by sampling a point according to the probability density function ρ_s/s^n and reduc-*

[2] We are conditioning on an event that has probability 0; this can be made rigorous by standard techniques.

ing the result modulo $\mathbf{B}$. *For any lattice* $\mathcal{L}(\mathbf{B})$, *the statistical distance between* $\rho_s/s^n \bmod \mathbf{B}$ *and the uniform distribution over* $\mathcal{P}(\mathbf{B})$ *is at most* $\frac{1}{2}\rho_{1/s}(\mathcal{L}(\mathbf{B})^* \setminus \{0\})$. *In particular, if* $s \geq \eta_\epsilon(\mathcal{L}(\mathbf{B}))$, *then the distance* $\Delta(\rho_s/s^n \bmod \mathbf{B}, U(\mathcal{P}(\mathbf{B})))$ *is at most* $\epsilon/2$.

It's useful to note that since every coset of $\mathbb{R}^n/\mathcal{L}(\mathbf{B})$ has a unique representative in $\mathcal{P}(\mathbf{B})$, we can think of the distribution of ρ_s/s^n, for $s \geq \eta_\epsilon(\mathcal{L}(\mathbf{B}))$, to be an almost uniform distribution on the cosets of $\mathbb{R}^n/\mathcal{L}(\mathbf{B})$.

The next property bounds the smoothing parameter in terms of λ_n. In [MR07], the authors are working with the l_2 norm,while we will be primarily concentrating on the l_∞ norm. This is why the next lemma differs from the one in [MR07] by a factor of $\sqrt{n}$.

Lemma 2.23 ([MR07, Lemma 3.3])**.** *For any n-dimensional lattice* Λ *and positive real* $\epsilon > 0$,

$$\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}} \cdot \lambda_n^2(\Lambda) \leq \sqrt{\frac{n\ln(2n(1+1/\epsilon))}{\pi}} \cdot \lambda_n^\infty(\Lambda).$$

The next lemma states that if $\mathbf{x} \sim D_{\Lambda,s,\mathbf{c}}$ for a large enough s, then $\mathbf{x}$ is not concentrated on one value. The lemma is even stronger in [MR07], but we do not need its full power here[3].

Lemma 2.24 ([MR07])**.** *Let* Λ *be any n-dimensional lattice and let s be such that* $s > 2\eta_\epsilon(\Lambda)$ *for* $\epsilon \leq 1/100$, *and let* $\mathbf{c} \in \mathbb{R}^n$ *be any point. Then for all* $\mathbf{x}' \in \Lambda$, $Pr_{\mathbf{x}\sim D_{\Lambda,s,\mathbf{c}}}[\mathbf{x} = \mathbf{x}'] \leq 99/100$.

2.J.1 New Lemmas for the Gaussian Distribution Over Lattices

In this subsection, we state a new result for Gaussian distributions over lattices which strengthens a result from [MR07], and thus might be of independent interest. In [MR07], the authors showed that for any $\mathbf{c}$ and a large enough s, the first few moments of the distribution $D_{\Lambda,s,\mathbf{c}}$ behave essentially the same as the moments of the continuous Gaussian distribution ρ_s/s^n. In this work, though, we need much higher moments of $D_{\Lambda,s,\mathbf{c}}$. In appendix A we prove that *all* the moments of $D_{\Lambda,s,\mathbf{c}}$

[3] This lemma only appears in the conference version of [MR07]

behave like the moments of ρ_s/s^n plus a little error. The precise statement of this is given in Lemma A.6. This result allows us to prove the following lemma, whose proof can also be found in Appendix A. We remark that following our work [LM06], Peikert was able to obtain the same results in the context of a more general analysis of the distributions of sums of discrete gaussians [Pei07]. (In this section, the notation $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the vector dot product of $\mathbf{a}$ and $\mathbf{b}$.)

Lemma 2.25. *For any n-dimensional lattice Λ, point $\boldsymbol{c} \in \mathbb{R}^n$, a vector $\boldsymbol{u}$ such that $\|\boldsymbol{u}\| = 1$, positive real $s > 2\eta_\epsilon(\Lambda)$ where $\epsilon < (\log n)^{-2\log n}$,*

$$Pr_{\boldsymbol{x} \sim D_{\Lambda,s,\boldsymbol{c}}}[|\langle \boldsymbol{x} - \boldsymbol{c}, \boldsymbol{u} \rangle| \geq s \log n] = n^{-\omega(1)}$$

Lemma 2.26. *For any n-dimensional lattice Λ, positive reals $\epsilon < (\log n)^{-2\log n}$, $s > 2\eta_\epsilon(\Lambda)$, polynomials $\mathbf{c}, \mathbf{z} \in \mathbb{R}[\mathbf{x}]$ of degree less than n, and any monic polynomial $\mathbf{f}$ of degree n,*

$$Pr_{\mathbf{d} \sim D_{\Lambda,s,\mathbf{c}}}[\|(\mathbf{d} - \mathbf{c})\mathbf{z} \bmod \mathbf{f}\|_\infty \geq 2\theta(\mathbf{f})\|\mathbf{z}\|_\infty s\sqrt{n} \log n] = n^{-\omega(1)}$$

Proof. Define n-dimensional vectors $\mathbf{z}^{(i)}$ as follows:

$$\mathbf{z}^{(i)} = \begin{cases} (z_i, z_{i-1}, \ldots, z_0, 0, \ldots, 0) & \text{for } 0 \leq i \leq n-1 \\ (0, \ldots, 0, z_{n-1}, \ldots, z_{i+2-n}, z_{i+1-n}) & \text{for } n \leq i \leq 2n-2 \end{cases}$$

With the above notation, the polynomial product of $(\mathbf{d} - \mathbf{c})\mathbf{z}$ in $\mathbb{R}[\mathbf{x}]$ can be written as

$$(\mathbf{d} - \mathbf{c})\mathbf{z} = \sum_{i=0}^{2n-2} \langle \mathbf{d} - \mathbf{c}, \mathbf{z}^{(i)} \rangle \mathbf{x}^i$$

Thus,

$$\begin{aligned} \|(\mathbf{d} - \mathbf{c})\mathbf{z}\|_\infty &= \max_i |\langle \mathbf{d} - \mathbf{c}, \mathbf{z}^{(i)} \rangle| = \max_i \left| \|\mathbf{z}^{(i)}\| \left\langle \mathbf{d} - \mathbf{c}, \frac{\mathbf{z}^{(i)}}{||\mathbf{z}^{(i)}||} \right\rangle \right| \\ &\leq \|\mathbf{z}\| \max_i \left| \left\langle \mathbf{d} - \mathbf{c}, \frac{\mathbf{z}^{(i)}}{||\mathbf{z}^{(i)}||} \right\rangle \right| \leq \sqrt{n}\|\mathbf{z}\|_\infty \max_i \left| \left\langle \mathbf{d} - \mathbf{c}, \frac{\mathbf{z}^{(i)}}{||\mathbf{z}^{(i)}||} \right\rangle \right| \end{aligned}$$

By Lemma 2.25 and the union bound, we get

$$Pr_{d \sim D_{\Lambda,s,c}} \left[\max_i \left| \left\langle \mathbf{d} - \mathbf{c}, \frac{\mathbf{z}^{(i)}}{||\mathbf{z}^{(i)}||} \right\rangle \right| \geq s \log n \right] \leq 2n \cdot n^{-\omega(1)} = n^{-\omega(1)}$$

and we can now apply Lemma 2.10 to obtain that with probability $1 - n^{-\omega(1)}$

$$\|(\mathbf{d}-\mathbf{c})\mathbf{z} \bmod \mathbf{f}\|_\infty \leq 2\theta(\mathbf{f})\|(\mathbf{d}-\mathbf{c})\mathbf{z}\|_\infty \leq 2\theta(\mathbf{f})\sqrt{n}\|\mathbf{z}\|_\infty s \log n$$

□

2.K Cryptographic Primitives

2.K.1 Digital Signatures

We recall the definitions of signature schemes and what it means for a signature scheme to be secure.

Definition 2.27. *A* signature scheme *consists of a triplet of polynomial-time (possibly probabilistic) algorithms (G, S, V) such that for every pair of outputs (s, v) of $G(1^n)$ and any n-bit message m,*

$$Pr[V(v, m, S(s, m)) = 1] = 1$$

where the probability is taken over the randomness of algorithms S and V.

In the above definition, G is called the key-generation algorithm, S is the signing algorithm, V is the verification algorithm, and s and v are, respectively, the signing and verification keys.

A signature scheme is said to be secure if there is only a negligible probability that any forger, after seeing signatures of messages of his choosing, can sign a message whose signature he has not already seen [GMR88].

Definition 2.28. *A signature scheme (G, S, V) is said to be secure if for every polynomial-time (possibly randomized) forger $\mathcal{F}$, the probability that after seeing the public key and $\{(\mu_1, S(s, \mu_1)), \ldots, (\mu_q, S(s, \mu_q))\}$ for any q messages μ_i of its choosing (where q is polynomial in n), $\mathcal{F}$ can produce $(\mu \neq \mu_i, \sigma)$ such that $V(v, \mu, \sigma) = 1$, is negligibly small. The probability is taken over the randomness of G, S, V, and $\mathcal{F}$.*

A weaker notion of security, called "one-time security" means that a forger, after seeing a signature of only a *single* message of his choosing, cannot produce a valid signature of a different message.

Definition 2.29. *A signature scheme* (G, S, V) *is said to be* one-time secure *if for every polynomial-time (possibly randomized) forger* $\mathcal{F}$*, the probability that after seeing the public key and* $(m, S(s, m))$ *for any message* m *of its choosing,* $\mathcal{F}$ *can produce* $(m' \neq m, \sigma')$ *such that* $V(v, m', \sigma') = 1$*, is negligibly small. The probability is taken over the randomness of* G*,* S*,* V*, and* $\mathcal{F}$*.*

In the standard security definition of a signature scheme, the forger should not be able to produce a signature of a new message. A stronger notion of security, called *strong unforgeability* requires that in addition to the above, a forger shouldn't even be able to come up with a different signature for a message whose signature he has already seen. The schemes presented in this paper satisfies this stronger notion of unforgeability.

2.K.2 Identification Schemes

An identification scheme consists of a key-generation algorithm and a description of an interactive protocol between a prover, possessing the secret key, and verifier possessing the corresponding public key. In general, it is required that the verifier accepts the interaction with a prover who behaves honestly with probability one. In this work, though, we need to relax this definition, and only require that the verifier accepts an honest prover with probability negligibly close to one (i.e $1 - 2^{-\omega(\log n)}$).

The standard active attack model against identification schemes proceeds in two phases [FFS88]. In the first phase, the adversary interacts with the prover in an effort to obtain some information. In the second stage, the adversary plays the role of the prover and tries to make a verifier accept the interaction. We remark that in the second stage, the adversary no longer has access to the honest prover. The adversary succeeds if he is able to make an honest verifier accept with some non-negligible probability.

2.K.3 The Random Oracle Model

A *random oracle* is a truly random function, and proving security of cryptographic protocols in the *random oracle model* assumes that all the parties participating in the protocol have access to a random oracle. While accessing a truly random function is practically infeasible, protocols proven secure in the random oracle model are still secure if the random function is replaced by a trusted third party. Relying on trusted third parties, however, is usually the very antithesis of cryptography's purpose, and so in practice, the random oracle methodology involves repalcing the trusted third party with a cryptographic hash function (such as SHA). The effect of this is that the security proof no longer applies and there are in fact constructions of "artificial" cryptographic protocols that can be proven secure in the random oracle model, yet are insecure whenever any real function is substituted for the trusted third party [CGH04]. Nevertheless, starting with the work of Bellare and Rogaway [BR93], who advocated the use of the random oracle methodology in the design of cryptographic protocols, many useful and seemingly secure primitives have been constructed in this way. While the role of random oracles in provable security remains somewhat controversial, it is generally accepted that a proof in the random oracle model is much better than no proof at all.

The General Forking Lemma

The following lemma of Bellare and Neven [BN06] will be very useful for proving that our signature scheme in Chapter 6 is secure in the random oracle model. We point out that the lemma itself has no mention of signatures or random oracles, but simply gives a bound on the probability that an algorithm will output something when run twice on slightly different inputs.

We now give a very rough overview of the lemma's statement. Suppose that A is an algorithm that takes as input $(x, h_1, \ldots, h_q; \rho)$, where all the h_i's are chosen randomly from some set H, and A is required to output one of the h_i's. If this algorithm outputs h_I, where $1 \leq I \leq q$, then the probability that $h_I \neq h'_I$ and the output of $A(x, h_1, \ldots, h_{I-1}, h'_I, h'_{I+1}, \ldots, h'_q; \rho)$ (where $h'_i \overset{\$}{\leftarrow} H$) will be h'_I, is not too small.

Lemma 2.30 (Lemma 1 of [BN06]). *Fix an integer $q \geq 1$ and a set H of size $h \geq 2$. Let A be a randomized algorithm that on input $x, h_1, \ldots, h_q$ returns a pair, the first element of which is an integer in the range $0, \ldots, q$ and the second element of which* *we refer to as a side output. Let IG be a randomized algorithm that we call the input generator. The accepting probability of A, denoted acc is defined as the probability that $J \geq 1$ in the experiment*

$$x \xleftarrow{\$} IG; h_1, \ldots, h_q \xleftarrow{\$} H; (J, \sigma) \xleftarrow{\$} A(x, h_1, \ldots, h_q).$$

The forking algorithm F_A associated to A is the randomized algorithm that takes input x and proceeds as follows:

$F_A(x)$

```
Pick bit-string $\rho$ for $A$ at random
$h_1, \ldots, h_q \xleftarrow{\$} H$
$(I, \sigma) \xleftarrow{\$} A(x, h_1, \ldots, h_q; \rho)$
if $I = 0$ then
    return $(0, \epsilon, \epsilon)$
end if
$h'_I, \ldots, h'_q \xleftarrow{\$} H$
$(I', \sigma') \leftarrow A(x, h_1, \ldots, h_{I-1}, h'_I, \ldots, h'_q; \rho)$
if $I = I'$ and $h_I \neq h'_I$ then
    return $(1, \sigma, \sigma')$
else
    return $(0, \epsilon, \epsilon)$
end if
```

Let

$$frk = Pr_{x \xleftarrow{\$} IG, (b, \sigma, \sigma') \xleftarrow{\$} F_A(x)}[b = 1].$$

Then

$$frk \geq acc \cdot \left(\frac{acc}{q} - \frac{1}{h} \right).$$

□

3

Hash Function

3.A Introduction

This chapter is devoted to proving the hardness of the collision problem in Definition 2.19. We will be showing that for any ring $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$ and some set $D \subset R$, if there is a polynomial-time algorithm that can solve $\mathrm{Col}(h, D)$ for a random $h \in \mathcal{H}(R, D, m)$ with some non-negligible probability, then there is a polynomial-time algorithm that can find an approximate shortest vector in all lattices that correspond to ideals in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$.

Theorem 3.1. *Let $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$ be a ring where $\mathbf{f}$ is a monic, irreducible polynomial of degree n and define the set $D = \{\mathbf{y} \in R : \|\mathbf{y}\|_\infty \leq d\}$ for some integer d. Let $\mathcal{H}(R, D, m)$ be a hash function family as in Definition 2.15 such that $m > \frac{\log p}{\log 2d}$ and $p \geq 4\theta(\mathbf{f})^2 dmn^{1.5} \log n$. If there is a polynomial-time algorithm that solves Col(h, D) for a random $h \in \mathcal{H}(R, D, m)$ with some non-negligible probability, then there is a polynomial-time algorithm that can solve $\mathbf{f}$-SVP$_\gamma(\Lambda)$ for every lattice Λ corresponding to an ideal in $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$, where $\gamma = 16\theta(\mathbf{f})^2 dmn \log^2 n$.*

To achieve the smallest value for the approximation factor γ, we can set $m = \Theta(\log n + \log \theta(\mathbf{f}))$ and $d = \Theta(\log n)$. This makes $\gamma = \tilde{O}(n)\theta(\mathbf{f})^2$. For purposes of being able to compute the function faster, though, it is useful to have m be smaller than $\Theta(\log n)$. It is possible to make m constant at the expense of being able to approximate the shortest vector only to a factor of $\gamma = \tilde{O}(n^{1+\delta})\theta(\mathbf{f})^2$. To be able to set m to a constant, we can set $d = n^{\delta}$ for some $\delta > 0$. Then we can set $m = \frac{\log(\theta(\mathbf{f}))}{\delta \log n} + \frac{2+\delta}{\delta} + o(1)$.

Note that the factor $\theta(\mathbf{f})$ plays a prominent role in the approximation factor for the solution to $\mathbf{f}$-SVP. Therefore it is prudent to choose polynomials $\mathbf{f}$ for which $\theta(\mathbf{f})$ is small. We saw in Section 2.D that if $\mathbf{f} = \mathbf{x}^n + 1$ or $\mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1$, then $\theta(\mathbf{f}) = 1$ and 2 respectively. It is our belief that the hardness of $\mathbf{f}$-SVP doesn't really depend on the particular $\mathbf{f}$, and so we would recommend choosing an $\mathbf{f}$ such that $\theta(\mathbf{f})$ is as small as possible in order to make the reduction from $\mathbf{f}$-SVP to $\mathrm{Col}(h, D)$ as tight as possible.

3.A.1 Basing Security On All Ideal Lattices

Theorem 3.1 says that being able to solve $\mathrm{Col}(h, D)$ for $h \in \mathcal{H}(R, m)$ implies being able to solve $\mathbf{f}$-SVP_{γ} for a *particular* $\mathbf{f}$. So for example, we can have collision-resistant hash functions based on the hardness of $\mathbf{f}$-SVP_{γ} when $\mathbf{f} = \mathbf{x}^n + 1$, but we will need a different hash function if we want to base security on the hardness of $\mathbf{f}$-SVP_{γ} for $\mathbf{f} = \mathbf{x}^n + \mathbf{x}^{n-1} + \ldots + 1$. So a natural question is whether we can have *one* hash function that is based on the hardness of $\mathbf{f}$-SVP for *every* monic, irreducible polynomial $\mathbf{f}$. In this section, we describe such a function.

The idea is actually very simple. Currently, our hash functions take as inputs polynomials in the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$ and perform all operations in that ring. But now consider the same hash functions performing their operations over the ring $\mathbb{Z}_p[\mathbf{x}]$. In other words, everything stays the same, except we don't perform a reduction modulo $\mathbf{f}$. Formally, consider the function family

$$\mathcal{H}(\mathbb{Z}_p[\mathbf{x}], m) = \{h_{\widehat{\mathbf{a}}} : \widehat{\mathbf{a}} \in \mathbb{Z}_p[\mathbf{x}]\}, \textit{ where for any } \widehat{\mathbf{z}} \in \mathbb{Z}_p[\mathbf{x}]^m,\ h_{\widehat{\mathbf{a}}}(\widehat{\mathbf{z}}) = \widehat{\mathbf{a}} \odot \widehat{\mathbf{z}}.$$

Notice that if the degree of all the $\mathbf{a}_i, \mathbf{z}_i$ in $\widehat{\mathbf{a}}, \widehat{\mathbf{z}}$ is less than n, then the degree of the polynomial $h(\widehat{\mathbf{z}})$ will be at most $2n - 2$, and so the size of the range of h will be p^{2n-1}. We can define the *UCol* (Universal Collision) problem as

Definition 3.2. *In the $UCol(h, D, n)$ problem, we are given an an $h_{\widehat{\mathbf{a}}} \in \mathcal{H}(\mathbb{Z}_p[\mathbf{x}], m)$ where each $h_{\widehat{\mathbf{a}}}$ is defined by m $\mathbf{a}_i$ of degree less than n, and are asked to find distinct $\widehat{\mathbf{z}}, \widehat{\mathbf{z}}' \in D^m$ such that $h(\widehat{\mathbf{z}}) = h(\widehat{\mathbf{z}}')$.*

Lemma 3.3. *If $m > \frac{2\log p}{\log|D|}$, then there is a reduction from solving Col(h, D) where $h \in \mathcal{H}(R, D, m)$ and $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$ to $UCol(d, D, n)$.*

Proof. Given an $h \in \mathcal{H}(R, m)$, we treat it as a hash function in $\mathcal{H}(\mathbb{Z}_p[\mathbf{x}], m)$. Since $m > \frac{2\log p}{\log|D|}$, the function h will be compressing and will therefore contain collisions. Therefore, we can solve the $UCol(d, D, n)$ problem to obtain a $\widehat{\mathbf{z}}, \widehat{\mathbf{z}}' \in D^m$ such that $h(\widehat{\mathbf{z}}) = h(\widehat{\mathbf{z}}')$ and of course, this also means that $h(\widehat{\mathbf{z}} \bmod \mathbf{f}) = h(\widehat{\mathbf{z}}' \bmod \mathbf{f})$, and therefore $\widehat{\mathbf{z}}$ and $\widehat{\mathbf{z}}'$ are solutions to Col(h, D). □

Therefore finding collisions in the hash function family $\mathcal{H}(\mathbb{Z}_p[\mathbf{x}], m)$ is as hard as solving $\mathbf{f}$-SVP for every $\mathbf{f}$. The only downside of using hash functions from $\mathcal{H}(\mathbb{Z}_p[\mathbf{x}], m)$ rather than from $\mathcal{H}(R, m)$ is that the output size of functions from $\mathcal{H}(\mathbb{Z}_p[\mathbf{x}], m)$ is about twice the size of the output of the functions from $\mathcal{H}(R, m)$.

3.B Proof of Theorem 3.1

In this section, we will prove Theorem 3.1. We will be finding the approximate shortest vector by finding incrementally smaller vectors in the lattice. We will show that by having access to an oracle who can solve Col(h, D), we will be able to find a vector that is half the size of the shortest current vector (until a certain point). More precisely, we will be using solutions to Col(h, D) for random $h \in \mathcal{H}(R, m)$ to repeatedly solve the following problem:

Given: a lattice Λ, $\mathbf{g} \in \Lambda$ such that $\mathbf{g} \neq 0$ and $\|\mathbf{g}\|_\infty > 16\theta(\mathbf{f})^2 dmn \log^2 n \lambda_1^\infty(\Lambda)$
Find: $\mathbf{h} \in \Lambda$, such that $\mathbf{h} \neq 0$ and $\|\mathbf{h}\|_\infty \leq \|\mathbf{g}\|_\infty/2$.

Define a number s as

$$s = \frac{\|\mathbf{g}\|_\infty}{16\theta(\mathbf{f})\sqrt{n}\log ndm} \geq \theta(\mathbf{f})\sqrt{n}(\log n)\lambda_1^\infty(\Lambda) \geq \sqrt{n}(\log n)\lambda_n^\infty(\Lambda) \geq \eta_\epsilon(\Lambda)$$

for $\epsilon = (\log n)^{-2\log n}$, where the last inequality follows by Lemma 2.23, and the inequality before that is due to Lemma 2.13. By Lemma 2.22, it follows that if $\mathbf{y} \in \mathbb{R}^n$ where $\mathbf{y} \sim \rho_s/s^n$, then $\Delta(\mathbf{y} + \Lambda, U(\mathbb{R}^n/\Lambda)) \leq (\log n)^{-2\log n}/2$. (That is, $\mathbf{y}$ is in an almost uniformly random coset of $\mathbb{R}^n/\Lambda$). We will now try to create an

$\mathbf{h} \in \Lambda$ which is smaller than $\mathbf{g}$ using the procedure below. In the procedure, it may not be obvious how each step is performed, and the reader is referred to Lemma 3.4 for a detailed explanation of each step. Also, let I be the ideal in $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ that corresponds to the lattice Λ.

```
1: for  $i = 1$ to $m$ do
2:     generate a uniformly random coset of $I/\langle \mathbf{g} \rangle$ and let $\mathbf{v}_i$ be a polynomial in that coset
3:     generate $\mathbf{y}_i \in \mathbb{R}^n$ such that $\mathbf{y}_i$ has distribution $\rho_s/s^n$ and consider $\mathbf{y}_i$ as a polynomial in $\mathbb{R}[x]$
4:     let $\mathbf{w}_i$ be the unique polynomial in $\mathbb{R}[x]$ of degree less than $n$ with coefficients in the range $[0, p)$ such that $p(\mathbf{v}_i + \mathbf{y}_i) \equiv \mathbf{g}\mathbf{w}_i$ in $\mathbb{R}^n/\langle p\mathbf{g} \rangle$
5:     $\mathbf{a}_i = [\mathbf{w}_i] \bmod p$ (where $[\mathbf{w}_i]$ means round each coefficient of $\mathbf{w}_i$ to the nearest integer)
6:     set $\widehat{\mathbf{a}} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$
7:     use oracle $\mathcal{C}$ to solve $\mathrm{Col}(h_{\widehat{\mathbf{a}}}, D)$, and using its output obtain polynomials $\mathbf{z}_1, \ldots, \mathbf{z}_m$ such that $\|\mathbf{z_i}\|_\infty \le 2d$ and $\sum \mathbf{z}_i \mathbf{a}_i \equiv \mathbf{0}$ in the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$
8: end for
9: output $\mathbf{h} = \sum_{i=1}^{m} \left( \frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])}{p} - \mathbf{y}_i \right) \mathbf{z}_i \bmod \mathbf{f}$.
```

To complete the proof, we will have to show five things: first, we have to prove that the above procedure runs in polynomial time, this is done in Lemma 3.4. Then, in Lemma 3.5, we show that in step (6) we are feeding the oracle $\mathcal{C}$ with an $h \in \mathcal{H}(R, m)$ where the distribution of h is statistically close to uniform over $\mathcal{H}(R, m)$. In Lemma 3.6, we show that the resulting polynomial $\mathbf{h}$ is in the ideal I. We then show that if $\mathcal{C}$ outputted a collision, then with non-negligible probability, $\|\mathbf{h}\|_\infty \le \|\mathbf{g}\|_\infty/2$ and that $\mathbf{h} \ne \mathbf{0}$. This is done in Lemmas 3.7 and 3.8 respectively. If we happen to fail, we can just repeat the procedure again. Since each run of the procedure is independent, we will obtain a shorter vector in polynomial time.

Lemma 3.4. *The above procedure runs in polynomial time.*

Proof. We will show that each step in the algorithm takes polynomial time. In step (2), we need to generate a random element of $I/\langle g \rangle$. By Lemma 2.12, the ideals I

and $\langle g \rangle$ can be thought of as $\mathbb{Z}$-modules of dimension n. Since $\langle \mathbf{g} \rangle \subseteq I$, the group $I/\langle \mathbf{g} \rangle$ is finite. Thus by Lemma 2.1, we can efficiently generate a random element of $I/\langle \mathbf{g} \rangle$. Step (4) of the algorithm will be justified in Lemma 3.5. In step (5), we are just rounding each coefficient of $\mathbf{w}_i$ to the nearest integer and then reducing modulo p. Now each $\mathbf{a}_i$ can be thought of as an element of $\mathbb{Z}_p[x]/\langle \mathbf{f} \rangle$, so $h_{\widehat{\mathbf{a}}}$ in step (6) is an element of R^m, and we ask oracle $\mathcal{C}$ to solve $\mathrm{Col}(h_{\widehat{\mathbf{a}}}, D)$. The oracle will return $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m), (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m)$ where $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i \in \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ such that $\|\boldsymbol{\alpha}_\mathbf{i}\|_\infty, \|\boldsymbol{\beta}_\mathbf{i}\|_\infty \leq d$ and $\sum \mathbf{a}_i \boldsymbol{\alpha}_i \equiv \sum \mathbf{a}_i \boldsymbol{\beta}_i$ in the ring $\mathbb{Z}_p[x]/\langle \mathbf{f} \rangle$. Thus if we set $\mathbf{z}_i = \boldsymbol{\alpha}_i - \boldsymbol{\beta}_i$, we will have $\|\mathbf{z}_\mathbf{i}\|_\infty \leq 2d$ and $\sum \mathbf{z}_i \mathbf{a}_i \equiv \mathbf{0}$ in the ring $\mathbb{Z}_p[x]/\langle \mathbf{f} \rangle$. □

Lemma 3.5. *Consider the polynomials $\mathbf{a}_i$ as elements in $\mathbb{Z}_p^n$. Then,*

$$\Delta((\mathbf{a}_1, \ldots, \mathbf{a}_m), U(\mathbb{Z}_p^{n \times m})) \leq m\epsilon/2.$$

Proof. We know that $\mathbf{v}_i$ is in a uniformly random coset of $I/\langle \mathbf{g} \rangle$ and let's assume for now that $\mathbf{y}_i$ is in a uniformly random coset of $\mathbb{R}^n/I$. This means that $\mathbf{v}_i + \mathbf{y}_i$ is in a uniformly random coset of $\mathbb{R}^n/\langle \mathbf{g} \rangle$ and thus the distribution of $p(\mathbf{v}_i + \mathbf{y}_i)$ is in a uniformly random coset of $\mathbb{R}^n/\langle p\mathbf{g} \rangle$. A basis for the additive group $\langle p\mathbf{g} \rangle$ is $p\mathbf{g}, p\mathbf{g}\mathbf{x}, \ldots, p\mathbf{g}\mathbf{x}^{n-1}$, thus every element of $\mathbb{R}^n/\langle p\mathbf{g} \rangle$ has a unique representative of the form $\alpha_0 p\mathbf{g} + \alpha_1 p\mathbf{g}\mathbf{x} + \ldots + \alpha_{n-1} p\mathbf{g}\mathbf{x}^{n-1} = \mathbf{g}(p\alpha_0 + p\alpha_1\mathbf{x} + \ldots + p\alpha_{n-1}\mathbf{x}^{n-1})$ for $\alpha_i \in [0, 1)$. So step (4) of the algorithm is justified, and since $p(\mathbf{v}_i + \mathbf{y}_i)$ is in a uniformly random coset of $\mathbb{R}^n/\langle p\mathbf{g} \rangle$, the coefficients of the polynomial $\mathbf{w}_i = p\alpha_0 + p\alpha_1\mathbf{x} + \ldots + p\alpha_{n-1}\mathbf{x}^{n-1}$ are uniform over the interval $[0, p)$, and thus the coefficients of $[\mathbf{w}_i]$ are uniform over the integers modulo p. The caveat is that $\mathbf{y}_i$ is not really in a uniformly random coset of $\mathbb{R}^n/I$, but is very close to it. By our choice of s, we have that $\Delta(\rho_s/s^n + I, U(\mathbb{R}^n/I)) \leq \epsilon/2$, and since $\mathbf{a}_i$ is a function of $\mathbf{y}_i$, by equation 2.2 we have that $\Delta(\mathbf{a}_i, U(\mathbb{Z}_p^n)) \leq \epsilon/2$. Since all the $\mathbf{a}_i$'s are independent, by equation 2.3, we have that $\Delta((\mathbf{a}_1, \ldots, \mathbf{a}_m), U(\mathbb{Z}_p^{n \times m})) \leq m\epsilon/2$. □

Lemma 3.6. $\mathbf{h} \in I$

Proof. In step (4) of the algorithm, assume that $p(\mathbf{v}_i + \mathbf{y}_i) + \mathbf{k}_i \mathbf{g} p = \mathbf{g}\mathbf{w}_i$ for some

$\mathbf{k}_i \in \mathbb{Z}[\mathbf{x}]$. Then,

$$\begin{aligned}\mathbf{h} &= \sum_{i=1}^{m} \left(\frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])}{p} - \mathbf{y}_i \right) \mathbf{z}_i \bmod \mathbf{f} \\ &= \sum_{i=1}^{m} (\mathbf{v}_i + \mathbf{y}_i + \mathbf{g}\mathbf{k}_i - \mathbf{g}\mathbf{a}_i/p - \mathbf{y}_i)\mathbf{z}_i \bmod \mathbf{f} \\ &= \sum_{i=1}^{m} (\mathbf{v}_i + \mathbf{g}\mathbf{k}_i)\mathbf{z}_i \bmod \mathbf{f} - \frac{\mathbf{g}\sum \mathbf{a}_i\mathbf{z}_i}{p} \bmod \mathbf{f}\end{aligned}$$

Since $\mathbf{v}_i \in I$ and $\mathbf{g} \in I$, we have that $\mathbf{v}_i + \mathbf{g}\mathbf{k}_i \bmod \mathbf{f} \in I$ and therefore $\sum(\mathbf{v}_i + \mathbf{g}\mathbf{k}_i)\mathbf{z}_i \bmod \mathbf{f} \in I$. Also, since $\sum \mathbf{a}_i\mathbf{z}_i \bmod \mathbf{f} \equiv \mathbf{0} (\bmod\ p)$, we have that $\frac{\sum \mathbf{a}_i\mathbf{z}_i}{p} \in \mathbb{Z}[x]$, and since $\mathbf{g} \in I$, we have that $\frac{\mathbf{g}\sum \mathbf{a}_i\mathbf{z}_i}{p} \bmod \mathbf{f} \in I$. □

Lemma 3.7. *With probability negligibly different from* 1, $\|\mathbf{h}\|_\infty \leq \frac{\|\mathbf{g}\|_\infty}{2}$.

Proof. We rewrite $\|\mathbf{h}\|_\infty$ as,

$$\begin{aligned}\|\mathbf{h}\|_\infty &= \left|\left| \sum_{i=1}^{m} \left(\frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])}{p} - \mathbf{y}_i \right) \mathbf{z}_i \bmod \mathbf{f} \right|\right|_\infty \\ &\leq \sum_{i=1}^{m} \left|\left| \left(\frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])}{p} \right) \mathbf{z}_i \bmod \mathbf{f} \right|\right|_\infty + \sum_{i=1}^{m} ||\mathbf{y}_i\mathbf{z}_i \bmod \mathbf{f}||_\infty\end{aligned}$$

We will first bound the term on the left using Lemma 2.8.

$$\left|\left| \left(\frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])}{p} \right) \mathbf{z}_i \bmod \mathbf{f} \right|\right|_\infty \leq \frac{n\theta(\mathbf{f})}{p} ||\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i]) \bmod \mathbf{f}||_\infty ||\mathbf{z}_i||_\infty$$

Assume for a moment that the coefficients of $\mathbf{w}_i$ are independently, uniformly distributed in the range $[0, p)$. Thus the coefficients of $\mathbf{w}_i - [\mathbf{w}_i]$ are independently, uniformly distributed in the range $[-1/2, 1/2]$. We also notice that $\mathbf{w}_i$ is completely independent from $\mathbf{g}$. Thus we can apply Lemma 2.11 and conclude that with probability negligibly close to 1,

$$\|\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i]) \bmod \mathbf{f}\|_\infty \leq \theta(\mathbf{f})\|\mathbf{g}\|_\infty \sqrt{n} \log n.$$

The preceding is all based on the assumption that the distribution of the coefficients of $\mathbf{w}_i$ is uniform, and the coefficients are independent, but in Lemma 3.5, we showed that the distribution of the n coefficients of $\mathbf{w}_i$ is statistically close to uniform over

$[0,p)^n$. So, the preceding inequality still holds with probability negligibly close to 1. Thus, with probability negligibly close to 1,

$$\sum_{i=1}^{m} \left|\left| \left(\frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])}{p} \right) \mathbf{z}_i \bmod \mathbf{f} \right|\right|_\infty \leq \frac{\|\mathbf{g}\|_\infty \theta(\mathbf{f})^2 dmn^{1.5} \log n}{p} < \frac{\|\mathbf{g}\|_\infty}{4}$$

where the last inequality follows because of our choice of p.

Now we will bound $\sum \|\mathbf{y}_i \mathbf{z}_i \bmod \mathbf{f}\|_\infty$. We will show

$$Pr_{\mathbf{y}_i \sim \rho_s / s^n}[||\mathbf{y}_i\mathbf{z}_i \bmod \mathbf{f}||_\infty > 2\theta(\mathbf{f})\|\mathbf{z}_i\|_\infty s\sqrt{n}\log n | (a_1, \ldots, a_m), (z_1, \ldots, z_m)] \quad (3.1)$$

$$= n^{-\omega(1)} \quad (3.2)$$

for each i. First, we will make the following observation. For any fixed coset of $\mathbb{R}^n/I$, call it $\mathbf{y}'_i + I$, the distribution of a_i given $\mathbf{y}_i$ is the same for all $\mathbf{y}_i \in \mathbf{y}'_i + I$. Thus, given that $\mathbf{y}_i \in \mathbf{y}'_i + I$, the distribution of $\mathbf{y}_i$ is independent of $(\mathbf{a}_1, \ldots, \mathbf{a}_m)$ because $\mathbf{a}_i$ is a randomized function of $\mathbf{y}'_i + I$ and $\mathbf{a}_{j \neq i}$ is independent of $\mathbf{y}_i$. And thus given that $\mathbf{y}_i \in \mathbf{y}'_i + I$, the distribution of $\mathbf{y}_i$ is also independent of $(z_1, \ldots, z_m)$ because $(\mathbf{z}_1, \ldots, \mathbf{z}_m)$ is a (randomized) function of $(\mathbf{a}_1, \ldots, \mathbf{a}_m)$. So we have

$$Pr[\mathbf{y}_i | \mathbf{y}_i \in \mathbf{y}'_i + I] = \frac{\rho_s(\mathbf{y}_i)}{\rho_s(\mathbf{y}'_i + I)} = \frac{\rho_{s,-\mathbf{y}'_i}(\mathbf{y}_i - \mathbf{y}'_i)}{\rho_{s,-\mathbf{y}'_i}(I)}$$

and so the conditional distribution of $(\mathbf{y}_i - \mathbf{y}'_i) \in I$ is $D_{I,s,-\mathbf{y}'_i}$. Thus, we have

$$\begin{aligned} &Pr_{\mathbf{y}_i \sim \rho_s / s^n}[||\mathbf{y}_i\mathbf{z}_i \bmod \mathbf{f}||_\infty > 2\theta(\mathbf{f})||\mathbf{z}_i||_\infty s\sqrt{n}\log n | \mathbf{y}_i \in \mathbf{y}'_i + I] \\ &= Pr_{(\mathbf{y}_i - \mathbf{y}'_i) \sim D_{I,s,-\mathbf{y}'_i}}[||((\mathbf{y}_i - \mathbf{y}'_i) - (-\mathbf{y}'_i))\mathbf{z}_i \bmod \mathbf{f}||_\infty \geq 2\theta(\mathbf{f})||\mathbf{z}_i||_\infty s\sqrt{n}\log n] \end{aligned}$$

and by Lemma 2.26, we have

$$Pr_{(\mathbf{y}_i - \mathbf{y}'_i) \sim D_{I,s,-\mathbf{y}'_i}}[||((\mathbf{y}_i - \mathbf{y}'_i) - (-\mathbf{y}'_i))\mathbf{z}_i \bmod \mathbf{f}||_\infty \geq 2\theta(\mathbf{f})||\mathbf{z}_i||_\infty s\sqrt{n}\log n] = n^{-\omega(1)}$$

The bound on Equation (3.1) follows by averaging over all possible $\mathbf{y}'_i + I$. Summing for all i, we get

$$\Pr\left[\sum_{i=1}^{m} ||\mathbf{y}_i\mathbf{z}_i \bmod \mathbf{f}||_\infty \geq 4\theta(\mathbf{f})dms\sqrt{n}\log n\right] = n^{-\omega(1)}$$

And since $\|\mathbf{g}\|_\infty = 16\theta(\mathbf{f})dms\sqrt{n}\log n$, we get that with probability negligibly close to 1, $\|\mathbf{h}\|_\infty < \frac{\|\mathbf{g}\|_\infty}{2}$. □

Lemma 3.8. $Pr[\mathbf{h} \neq \mathbf{0} | (\mathbf{a}_1, \ldots, \mathbf{a}_m), (\mathbf{z}_1, \ldots, \mathbf{z}_m)] = \Omega(1)$

Proof. Since some $\mathbf{z}_i$ has to be non-zero, assume without loss of generality that $\mathbf{z}_1$ is a non-zero polynomial. Then $\mathbf{h} = \mathbf{0}$ if and only if

$$\mathbf{y}_1\mathbf{z}_1 \bmod \mathbf{f} = \sum_{i=1}^{m} \frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])\mathbf{z}_i}{p} \bmod \mathbf{f} - \sum_{i=2}^{m} \mathbf{y}_i\mathbf{z}_i \bmod \mathbf{f}$$

Notice that as in Lemma 3.7, if we are given the coset of $\mathbb{R}^n/I$ that $\mathbf{y}_1$ belongs to (call it $\mathbf{y}_1' + I$), then $\mathbf{y}_1$ is independent of all $\mathbf{a}_i$ and $\mathbf{z}_i$ and all $\mathbf{y}_{i>1}$. So we want to bound

$$Pr_{\mathbf{y}_1 \sim \rho_s/s^n}\left[\mathbf{y}_1\mathbf{z}_1 \bmod \mathbf{f} = \sum_{i=1}^{m} \frac{\mathbf{g}(\mathbf{w}_i - [\mathbf{w}_i])\mathbf{z}_i}{p} \bmod \mathbf{f} - \sum_{i=2}^{m} \mathbf{y}_i\mathbf{z}_i \bmod \mathbf{f} \middle| y_1 \in y_1' + I\right] \tag{3.3}$$

and averaging over all $\mathbf{y}_1' + I$ will give us the final result. Notice that if $\mathbf{y}_1\mathbf{z}_1 = \mathbf{c}$, then for each given $\mathbf{z}_1$, there is only one value that y_1 can have because the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ is an integral domain and we assumed that $\mathbf{z}_1 \neq 0$. Thus equation 3.3 is equivalent to

$$Pr_{\mathbf{y}_1 \sim \rho_s/s^n}[\mathbf{y}_1 | \mathbf{y}_1 \in \mathbf{y}_1' + I] = \frac{\rho_s(\mathbf{y}_1)}{\rho_s(\mathbf{y}_1' + I)} = \frac{\rho_{s,-\mathbf{y}_i'}(\mathbf{y}_i - \mathbf{y}_i')}{\rho_{s,-\mathbf{y}_i'}(I)}$$

which is the probability that $\mathbf{x} = \mathbf{y}_1 - \mathbf{y}_1'$ given that $\mathbf{x} \sim D_{I,s,-\mathbf{y}_1'}$. By Lemma 2.24, this probability is at most 99/100. Thus with probability $\Omega(1)$, $\mathbf{h} \neq \mathbf{0}$. □

3.C Finding Collisions In $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n - 1 \rangle$

In this section we show how to find collisions if the family of hash functions $\mathcal{H}(R, m)$ is instantiated with a polynomial $\mathbf{f} = \mathbf{x}^n - 1$. This answers an open problem posed in [Mic07] as well as illustrates the value of provable security. Indeed, there is only a seemingly minor difference between having the function be computed over the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1 \rangle$ and $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n - 1 \rangle$, yet in one case the function is provably secure, and in the other case it is completely insecure, as we will now show.

The intuitive reason we can find collisions is that the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n - 1 \rangle$ has an ideal that is small and consists of elements with small norms. That ideal is $J = \langle \mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1 \rangle$. It's not hard to see that $|J| = p$ and that all elements of J have the form $\alpha(\mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1)$ for integers $0 \leq \alpha \leq p-1$. So the idea for solving $\mathrm{Col}(h, D)$ (for any set $D \subset R$) is to choose $(\mathbf{y}_1, \ldots, \mathbf{y}_m) \neq (\mathbf{z}_1, \ldots, \mathbf{z}_m)$ such that $\mathbf{y}_i, \mathbf{z}_i \in J$ and $\mathbf{y}_i, \mathbf{z}_i \in D$. This would force both $h(\mathbf{y}_1, \ldots, \mathbf{y}_m)$ and $h(\mathbf{z}_1, \ldots, \mathbf{z}_m)$ to be in J. There are $|D|$ possibilities for each $\mathbf{y}_i$, thus there are a total of D^m possibilities for $(\mathbf{y}_1, \ldots, \mathbf{y}_m)$. Thus if $|D|^m \geq p$, then a collision is guaranteed to exist and will take time on the order of p to find. But in order for h to be a hash function, we needed $|D|^{nm}$ to be greater than p^n, and thus $|D|^m > p$, which is exactly the condition we need to find a collision in J.

3.D Cyclic Lattices

A cyclic lattice is a lattice corresponding to an ideal of the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n - 1 \rangle$. In a way, they are the simplest and most natural lattices that correspond to ideals because they are simply lattices where if $(v_1, \ldots, v_n)$ is a vector, then $(v_n, v_1, \ldots, v_{n-1})$ must be a vector as well. It is precisely on the hardness of the shortest vector problem of these lattices that Micciancio one-way function had its security based on. But we showed that Micciancio's one-way functions are not collision-resistant, and so it's not clear whether collision-resistant hash functions can be built based on the hardness of cyclic lattice problems. In this section, we show that we can indeed build hash functions with such security.

Theorem 3.9. *For a prime n, let $\Phi_n(\mathbf{x}) = \mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1$. Then,*

$$(\mathbf{x}^n - 1)\text{-}SVP_{4\gamma} \leq \Phi_n(\mathbf{x})\text{-}SVP_{\gamma}$$

Proof. Let Λ be any cyclic lattice and let I be the ideal that corresponds to Λ. Also, let $\mathbf{v}$ be a polynomial in I such that $\|\mathbf{v}\|_\infty = \lambda_1^\infty(I)$. There are two cases to consider. The first case is that $\mathbf{v}$ is in the ideal generated by $\Phi_n(\mathbf{x})$. Since the ideal $\langle \Phi_n(\mathbf{x}) \rangle$ is generated by a polynomial of degree $n-1$, the intersection $\langle \Phi_n(\mathbf{x}) \rangle \cap I$ has dimension 1, and so we can easily find the vector $\mathbf{v}$ in the lattice (of dimension 1) corresponding to the ideal $\langle \Phi_n(\mathbf{x}) \rangle \cap I$.

In the second case, we have $\mathbf{v} \notin \langle \Phi_n(\mathbf{x}) \rangle$. This means that the polynomial $\mathbf{v}' = \mathbf{v} \bmod \Phi_n(\mathbf{x}) \neq \mathbf{0}$, and therefore if we define the ideal $I' = I \bmod \Phi_n(\mathbf{x})$, the polynomial $\mathbf{v}'$ will be a non-zero polynomial in the ideal I'. Notice that since the degree of $\mathbf{v}$ is n and the degree of $\Phi_n(\mathbf{x})$ is $n-1$, we have $\|\mathbf{v}'\|_\infty \leq 2\|\mathbf{v}\|_\infty$. Also observe that I' is an ideal of $\mathbb{Z}[\mathbf{x}]/\langle\Phi_n(\mathbf{x})\rangle$ (because $\Phi_n(\mathbf{x})|\mathbf{x}^n-1$), and therefore our algorithm that solves $\Phi_n(\mathbf{x})$-SVP_γ will find a non-zero polynomial $\mathbf{w}'$ in I' such that

$$\|\mathbf{w}'\|_\infty \leq \gamma\lambda_1^\infty(I') \leq \gamma\|\mathbf{v}'\|_\infty \leq 2\gamma\|\mathbf{v}\|_\infty = 2\gamma\lambda_1^\infty(I).$$

Now we notice that since $\mathbf{w}' \in I'$, the polynomial $\mathbf{w}' \cdot (\mathbf{x}-1)$ must be in I. This is because $\mathbf{w}' = \mathbf{w} \bmod \Phi_n(\mathbf{x}) = \mathbf{w} - \boldsymbol{\alpha}\Phi_n(\mathbf{x})$ (for some polynomial $\boldsymbol{\alpha}$), and so $\mathbf{w}'(\mathbf{x}-1) = \mathbf{w}(\mathbf{x}-1) - \boldsymbol{\alpha}(\mathbf{x}^n-1) \in I$. Also, the degree of $\mathbf{w}'$ is at most $n-2$, so $\mathbf{w}'(\mathbf{x}-1)$ is not $\mathbf{0}$ in the ring $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{x}^n-1\rangle$. Therefore the polynomial $\mathbf{w}'(\mathbf{x}-1)$ is in I and

$$\|\mathbf{w}'(\mathbf{x}-\mathbf{1})\|_\infty \leq \|\mathbf{w}'\mathbf{x}\|_\infty + \|\mathbf{w}'\|_\infty \leq \theta(\mathbf{x}^n-1)\|\mathbf{w}'\|_\infty + \|\mathbf{w}'\|_\infty = 2\|\mathbf{w}'\|_\infty \leq 4\gamma\lambda_1^\infty(I).$$

□

Corollary 3.10. *Let n be a prime, let $\mathbf{f} = \mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \ldots + 1$, and let $\mathcal{H}(R, D, m)$ be a family of hash functions for the parameters in Theorem 3.1. If we can solve Col(h, D) where h is chosen uniformly at random from $\mathcal{H}(R, D, m)$, then we can solve $(\mathbf{x}^n-1)$-SVP_γ for $\gamma = \tilde{O}(n)$.*

3.E Connection with Algebraic Number Theory

In this subsection, we relate the problem of finding the shortest polynomial in an ideal to a certain problem from algebraic number theory. Our goal will be to show that finding collisions in h implies finding certain elements in number fields. The connection between algebraic number theory and the ring $\mathbb{Z}[x]/\langle\mathbf{f}\rangle$ comes from the following lemma.

Lemma 3.11. *If $\mathbf{f} \in \mathbb{Z}[\mathbf{x}]$ is monic and is the minimum polynomial*[1] *of θ, then $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle \cong \mathbb{Z}[\theta]$.*

[1] *the minimum polynomial of θ is the monic polynomial in $\mathbb{Z}[\mathbf{x}]$ with the smallest degree that has θ as a root*

Proof. Let the degree of $\mathbf{f}$ be n and assume $\alpha \in \mathbb{Z}[\theta]$ is represented as an integer combination of powers of θ. That is, $\alpha = \alpha_0 + \alpha_1\theta + \ldots + \alpha_{n-1}\theta^{n-1}$. Then the function $\sigma : \mathbb{Z}[\theta] \to \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ which maps α to $\alpha_0 + \alpha_1\mathbf{x} + \ldots + \alpha_{n-1}\mathbf{x}^{n-1}$ is an isomorphism. We will not prove this, but it is not hard to show using basic algebraic number theory. □

Definition 3.12. *Let θ be an algebraic integer of degree n. Then for any $\alpha \in \mathbb{Q}(\theta)$ where $\alpha = \alpha_0 + \alpha_1\theta + \ldots + \alpha_{n-1}\theta^{n-1}$, define the function maxCoeff$_\theta(\alpha)$ to be* $\max(|\alpha_0|, \ldots, |\alpha_{n-1}|)$.

From Lemma 3.11, we can see that finding an element with the smallest norm in an ideal I of $\mathbb{Z}[x]/\langle \mathbf{f} \rangle$ is equivalent to finding the element α in the ideal $\sigma^{-1}(I)$ of $\mathbb{Z}[\theta]$ (where θ is a zero of $\mathbf{f}$) such that $\text{maxCoeff}_{\mathbb{Q}(\theta)}(\alpha)$ is the smallest of all the $\alpha' \in \sigma^{-1}(I)$. This is not too interesting of a problem because it is exactly the shortest vector problem in ideal lattices with the indeterminate $\mathbf{x}$ replaced by θ. A more interesting result is relating the norm of elements in $\mathbb{Z}[x]/\langle \mathbf{f} \rangle$ to the conjugates of elements in $\mathbb{Z}[\theta]$.

Definition 3.13. *For any $\alpha \in \mathbb{C}$, define the function $maxConj(\alpha)$ to be* $\max(|\phi_1|, \ldots, |\phi_n|)$ *where ϕ_i are the zeros of the minimum polynomial of α over $\mathbb{Q}$.*

Notice that $\text{maxCoeff}_\theta(\alpha)$ depends on the particular representation of α, while $maxConj(\alpha)$ does not. Now we define the smallest conjugate problem.

Definition 3.14. *Let θ be an algebraic integer of degree n. Let $K = \mathbb{Q}(\theta)$ be a number field, and let $\mathbb{Z}[\theta]$ be a subring of K. Let I be any ideal of $\mathbb{Z}[\theta]$. In the approximate Smallest Conjugate Problem $SCP_\gamma(I)$, we are asked to find an element $\alpha \in I$ such that $maxConj(\alpha) \leq \gamma \cdot maxConj(\alpha')$ for all $\alpha' \in I$.*

The problem of finding elements with small conjugates is somewhat related to the "Polynomial Reduction Problem" in [Coh96, Section 4.4.2] for which no polynomial time algorithm seems to be known.

As we did for SVP, we can consider the restriction of SCP to certain classes of ideals. Let $\mathbf{f}$ be an irreducible integer polynomial. We will write $\mathbf{f}$-SCP to mean the problem SCP restricted to ideals of the ring $\mathbb{Z}[\theta]$ where θ is a zero of $\mathbf{f}$.

Now we will prove a theorem relating $\mathbf{f}$-SCP to $\mathbf{f}$-SVP for some values of $\mathbf{f}$ such as $\mathbf{f} = \mathbf{x}^n + \mathbf{x}^{n-1} + \ldots + 1$ and $\mathbf{f} = \mathbf{x}^n + 1$. The key reason that we are able to get such a relationship is that when θ is a zero of such $\mathbf{f}$'s, then for any $\alpha \in \mathbb{Q}(\theta)$, $\mathrm{maxConj}(\alpha)$ and $\mathrm{maxCoeff}_\theta(\alpha)$ differ by at most a factor of n. This is proved by Lemmas 3.17, 3.18, and 3.19. Lemmas 3.17, 3.18 give us the sufficient conditions under which there is such a close relationship, and Lemmas 3.19, 3.20 show that when the minimum polynomial of θ is either $\mathbf{x}^n + \mathbf{x}^{n-1} + \ldots + 1$ or $\mathbf{x}^n + 1$, then those conditions are satisfied.

Theorem 3.15. *Let $\mathbf{f} = \mathbf{x}^n + \mathbf{x}^{n-1} + \ldots + 1$ be irreducible, and let $\sigma : \mathbb{Z}[\theta] \to \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ be an isomorphism as in Lemma 3.11. Then $\mathbf{f}$-$SVP_{\gamma n^2} \le \mathbf{f}$-$SCP_\gamma(\sigma^{-1}(I))$ and $\mathbf{f}$-$SCP_{\gamma n^2} \le \mathbf{f}$-$SVP_\gamma(I)$.*

Proof. Let θ be a zero of $\mathbf{f}$. First, we will show $\mathbf{f}\text{-SCP}_{\gamma n^2} \le \mathbf{f}\text{-SVP}_\gamma$. Consider an ideal I of $\mathbb{Z}[\theta]$ given to us by its generators $g_1, \ldots, g_k$ represented as a linear combination of powers of θ. That is $g_i = g_{i,0} + g_{i,1}\theta + \ldots + g_{i,n-1}\theta^{n-1}$. We use the oracle for $\mathbf{f}\text{-SVP}_\gamma$ to find the element $\mathbf{h} \in \sigma(I)$ whose norm is less than $\gamma\lambda_1^\infty(\sigma(I))$ and let $\alpha = \sigma^{-1}(\mathbf{h})$. Thus $\mathrm{maxCoeff}_\theta(\alpha) \le \gamma \cdot \mathrm{maxCoeff}_\theta(\alpha')$ for all $\alpha' \in I$. And so applying Lemma 3.19 twice, we get

$$\begin{aligned} maxConj(\alpha) &\le n \cdot \mathrm{maxCoeff}_\theta(\alpha) \\ &\le n\gamma \cdot \mathrm{maxCoeff}_\theta(\alpha') \text{ for all } \alpha' \in I \\ &\le n^2\gamma \cdot maxConj(\alpha') \text{ for all } \alpha' \in I \end{aligned}$$

and so we have a γn^2 approximation for $\mathbf{f}$-SCP.

Now we show $\mathbf{f}\text{-SVP}_{\gamma n^2} \le \mathbf{f}\text{-SCP}_\gamma$. Consider an ideal I of $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + \mathbf{x}^{n-1} + \ldots + 1 \rangle$ given to us by its generators $\mathbf{g}_1, \ldots, \mathbf{g}_k$. We use the oracle for $\mathbf{f}\text{-SCP}_\gamma$ to find the element $\alpha \in \sigma^{-1}(I)$ such that $maxConj(\alpha) \le \gamma \cdot maxConj(\alpha')$ for all $\alpha' \in \sigma^{-1}(I)$. And by applying Lemma 3.19 twice, we get

$$\begin{aligned} \mathrm{maxCoeff}_\theta(\alpha) &\le n \cdot maxConj(\alpha) \\ &\le n\gamma \cdot maxConj(\alpha') \text{ for all } \alpha' \in \sigma^{-1}(I) \\ &\le n^2\gamma \cdot \mathrm{maxCoeff}_\theta(\alpha') \text{ for all } \alpha' \in \sigma^{-1}(I) \end{aligned}$$

This means that the infinity norm of $\sigma(\alpha)$ is at most $\gamma n^2 \lambda_1^\infty(I)$, and thus we have a γn^2 approximation of $\mathbf{f}$-SVP. $\square$

Theorem 3.16. *Let $\mathbf{f} = \mathbf{x}^n + 1$ be irreducible, and let $\sigma : \mathbb{Z}[\theta] \rightarrow \mathbb{Z}[\mathbf{x}]/\langle \mathbf{f} \rangle$ be an isomorphism as in Lemma 3.11. Then $\mathbf{f}$-$SVP_{\gamma n} \leq \mathbf{f}$-$SCP_\gamma(\sigma^{-1}(I))$ and $\mathbf{f}$-$SCP_{\gamma n} \leq \mathbf{f}$-$SVP_\gamma(I)$.*

Proof. The proof of this is analogous to the proof of Theorem 3.15, except that instead of using Lemma 3.19 to obtain a connection between the maximum coefficient and the maximum conjugate, we use Lemma Lemma 3.20. $\square$

Lemma 3.17. *Let $\mathbf{f} \in \mathbb{Z}[\mathbf{x}]$ be a monic irreducible polynomial of degree n with zeros $\theta_1, \ldots, \theta_n \in \mathbb{C}$ such that for all i, $|\theta_i^{n-1}| \leq t$. Let $K = \mathbb{Q}(\theta_1)$ and $\alpha = \alpha_0 + \alpha_1\theta_1 + \ldots + \alpha_{n-1}\theta_1^{n-1} \in K$. Then $maxConj(\alpha) \leq nt \cdot maxCoeff_{\theta_1}(\alpha)$.*

Proof. Let $\sigma_1, \ldots, \sigma_n : K \rightarrow \mathbb{C}$ be the n distinct embeddings of K into $\mathbb{C}$. Then the field polynomial of α is $fld_\alpha(x) = \prod_{i=1}^{n}(x - \sigma_i(\alpha))$. Since the field polynomial is a power of the minimum polynomial of α, the set of zeros of the minimal polynomial of α is exactly the set $\{\sigma_i(\alpha)\}$. Since $\sigma_i(\theta_1) = \theta_i$, we have that $\sigma_i(\alpha) = \alpha_0 + \alpha_1\theta_i + \ldots + \alpha_{n-1}\theta_i^{n-1}$. Since $|\theta_i^{n-1}| \leq t$, we have the claim in the lemma. $\square$

Lemma 3.18. *Let $\mathbf{f} \in \mathbb{Z}[\mathbf{x}]$ be a monic, irreducible polynomial of degree n with zeros $\theta_1, \ldots, \theta_n \in \mathbb{C}$. Let $K = \mathbb{Q}(\theta_1)$ be a number field. If there exists an integer $m \geq n$ such that for all $1 \leq i \leq n$ and $j \leq m - 1$ we have $1 \leq |\theta_i^j| \leq t$, and $\left|\sum_{i=1}^{n} \theta_i^m\right| \geq n$ and for all $j \neq 0(mod\ m)$, we have $\left|\sum_{i=1}^{n} \theta_i^j\right| \leq s \leq 1$, then for all $\alpha \in K$, we have $maxCoeff_{\theta_1}(\alpha) \leq \frac{nt}{n(1-s)+s} maxConj(\alpha)$.*

Proof. Let $\sigma_1, \ldots, \sigma_n : K \rightarrow \mathbb{C}$ be the n distinct embeddings of K into $\mathbb{C}$. Then the set of zeros of the minimum polynomial of α is $\{\sigma_i(\alpha)\}$. Let $k = max_i(|\sigma_i(\alpha)|)$. For each $0 \leq j \leq n - 1$, we can set up the following system of n inequalities: for $1 \leq i \leq n$, $|\sigma_i(\alpha)\theta_i^{m-n+j}| \leq tk$. The preceding is true because $|\sigma_i(\alpha)| \leq k$ and $|\theta_i^{m-n+j}| \leq t$. Now we take a closer look at the system of inequalities for a particular

j. Let $\alpha = \alpha_0 + \alpha_1\theta_1 + \ldots + \alpha_{n-1}\theta_1^{n-1}$.

$$
\begin{aligned}
|\sigma_1(\alpha)\theta_1^{m-n+j}| &= |\alpha_0\theta_1^{m-n+j} + \ldots + \alpha_{n-j}\theta_1^{m} + \ldots + \alpha_{n-1}\theta_1^{m+j-1}| \leq kt \\
|\sigma_2(\alpha)\theta_2^{m-n+j}| &= |\alpha_0\theta_2^{m-n+j} + \ldots + \alpha_{n-j}\theta_2^{m} + \ldots + \alpha_{n-1}\theta_2^{m+j-1}| \leq kt \\
\ldots &= \ldots \\
|\sigma_n(\alpha)\theta_n^{m-n+j}| &= |\alpha_0\theta_n^{m-n+j} + \ldots + \alpha_{n-j}\theta_n^{m} + \ldots + \alpha_{n-1}\theta_n^{m+j-1}| \leq kt
\end{aligned}
$$

If we let $A = \sum\limits_{i=1}^{n} |\alpha_i|$ and $S_j = \sum\limits_{i=1}^{n} \theta_i^{m-n+j}$ then

$$
\begin{aligned}
& n|\alpha_{n-j}| - s(A - |\alpha_{n-j}|) = \\
& n|\alpha_{n-j}| - s(|\alpha_0| + \ldots + |\alpha_{n-j-1}| + |\alpha_{n-j+1}| + \ldots + |\alpha_{n-1}|) \leq \\
& |\alpha_{n-j}S_n| - (|\alpha_0 S_j| + \ldots + |\alpha_{n-j-1}S_{n-1}| + |\alpha_{n-j+1}S_{n+1}| + \ldots + |\alpha_{n-1}S_{n-1+j}|) \leq \\
& |\alpha_{n-j}S_n| - |\alpha_0 S_j + \ldots + \alpha_{n-j-1}S_{n-1} + \alpha_{n-j+1}S_{n+1} + \ldots + \alpha_{n-1}S_{n-1+j}| \leq \\
& |\alpha_0 S_j + \ldots + \alpha_{n-j-1}S_{n-1} + \alpha_{n-j}S_n + \alpha_{n-j+1}S_{n+1} + \ldots + \alpha_{n-1}S_{n-1+j}| \leq \\
& |\sigma_1(\alpha)\theta_1^{m-n+j}| + \ldots + |\sigma_n(\alpha)\theta_n^{m-n+j}| \leq nkt
\end{aligned}
$$

So for all α_i, we have the inequality

$$|\alpha_i| \leq \frac{nkt + sA}{n+s}$$

Setting $B = \frac{nkt+sA}{n+s}$, we get that $A \leq nB$, and thus $B \leq \frac{nkt}{n(1-s)+s}$ and since $|\alpha_i| \leq B$, we get the claim in the lemma. □

Lemma 3.19. *Let* $\mathbf{f} = \mathbf{x}^n + \mathbf{x}^{n-1} + \ldots + 1$ *be an irreducible polynomial and* $\theta \in \mathbb{C}$ *be one of its zeros. Let* $K = \mathbb{Q}(\theta)$ *and let* α *be an element of* K*. Then* $maxCoeff_\theta(\alpha) \leq n \cdot maxConj(\alpha)$ *and* $maxConj(\alpha) \leq n \cdot maxCoeff_\theta(\alpha)$.

Proof. To prove that $maxConj(\alpha) \leq n \cdot \text{maxCoeff}_\theta(\alpha)$, we will apply Lemma 3.17. Since $\mathbf{f}$ is the cyclotomic polynomial, all of its zeros have norm 1 and so we apply Lemma 3.17 with $t = 1$ and we obtain the desired inequality.

To show that $\text{maxCoeff}_\theta(\alpha) \leq n \cdot maxConj(\alpha)$, we will need to apply Lemma 3.18. In that lemma, we will set $t = 1$ and $m = n + 1$. If θ is a zero of $x^n + x^{n-1} + \ldots + 1$, then $\theta^{n+1} = (\theta^n + \ldots + 1)(\theta - 1) + 1 = 1$, and so $\left|\sum\limits_{i=1}^{n} \theta_i^m\right| = n$. Since f is a cyclotomic polynomial, it has a zero, call it θ_1, such that $\theta_i = \theta_1^i$ for all i. And since we already showed that $\theta_i^{n+1} = 1$, we know that $\theta_i^j = \theta_i^{j \bmod (n+1)}$. Thus for all j such that $j \bmod (n+1) \neq 0$, we have

$$\left|\sum_{i=1}^{n} \theta_i^j\right| = \left|\sum_{i=1}^{n} \theta_i^{j \bmod (n+1)}\right| = \left|\sum_{i=1}^{n} \theta_1^{i(j \bmod (n+1))}\right| = \left|\sum_{i=1}^{n} \theta_{j \bmod (n+1)}^{i}\right| = |-1| = 1$$

Thus Lemma 3.18 applies with $s = 1$. And so we have

$$\text{maxCoeff}_\theta(\alpha) \leq n \cdot maxConj(\alpha)$$

as claimed. □

Lemma 3.20. *Let* $\mathbf{f} = \mathbf{x}^n + \beta \in \mathbb{Z}[\mathbf{x}]$ *be an irreducible polynomial and* $\theta \in \mathbb{C}$ *be one of its zeros. Let* $K = \mathbb{Q}(\theta)$ *and let* α *be an element of* K*. Then* $maxCoeff_\theta(\alpha) \leq |\beta| \cdot maxConj(\alpha)$ *and* $maxConj(\alpha) \leq |\beta| n \cdot maxCoeff_\theta(\alpha)$*.*

Proof. Let $\theta_1 = \theta, \theta_2, \ldots, \theta_n$ be the zeros of f. To prove that $maxConj(\alpha) \leq n|\beta| \cdot \text{maxCoeff}_\theta(\alpha)$, we will apply Lemma 3.17. For any θ_i, we have $|\theta_i|^n = |\theta_i^n| = |\beta|$. Therefore, $|\theta_i^{n-1}| = |\theta_i|^{n-1} \leq |\beta|$, and we apply Lemma 3.17 with $t = |\beta|$. To show that $\text{maxCoeff}_\theta(\alpha) \leq |\beta| \cdot maxConj(\alpha)$, we will need to apply Lemma 3.18. We will apply that lemma with $t = |\beta|$, $s = 0$, and $m = n$. We already showed that $|\theta_i^j| \leq |\beta|$ for $0 \leq j \leq n-1$, and it's easy to see that $\left|\sum_{i=1}^{n} \theta_i^n\right| = |\beta n| \geq n$ (because $\theta_i^n = -\beta$). Now we will show that for all $j \neq 0(\text{mod } n)$,

$$\sum_{i=1}^{n} \theta_i^j = 0. \tag{3.4}$$

First, assume that $1 \leq j < n$. Then equation 3.4 follows by applying Newton's formulas for symmetric polynomials [Coh96, Proposition 4.3.3]. If $j > n$ and $j \neq 0(\text{mod } n)$, then there exists an integer k such that $1 \leq j - kn \leq n-1$ and we have

$$\sum_{i=1}^{n} \theta_i^j = \sum_{i=1}^{n} (\theta_i^{kn} \theta_i^j - kn) = \sum_{i=1}^{n} (\theta_i^{kn} \theta_i^{j-kn}) = -\beta \sum_{i=1}^{n} \theta_i^{j-kn} = 0$$

Thus Lemma 3.18 applies with $t = |\beta|$, $s = 0$, and $m = n$ and we have the claimed result. □

Chapter 3 is, in part, a reprint, of the paper "Generalized Compact Knapsacks Are Collision Resistant" co-authored with Daniele Micciancio and appearing in the proceedings of ICALP 2006. The dissertation author was the primary investigator and author of this paper.

4

One-time Signature

In this section we present our one-time signature scheme. The security of the scheme will be ultimately based on the worst-case hardness of approximating the shortest vector in all lattices corresponding to ideals in the ring $\mathbb{Z}[x]/\langle \mathbf{f} \rangle$ for any irreducible polynomial $\mathbf{f}$. The key-generation algorithm for the signature scheme allows us to specify the polynomial $\mathbf{f}$ that we want to use for the hardness assumption, and by the same reasoning as in the hash function chapter, we want to pick an $\mathbf{f}$ such that $\theta(\mathbf{f})$ is small. In Figure 4.1, we describe the key-generation algorithm, and in Figure 4.2, we present the procedures for signing a message and verifying a signature.

4.A The One-Time Signature Scheme

The public key of the signature scheme consists of a randomly chosen hash function h from $\mathcal{H}(R, m)$ and the hashes $h(\widehat{\mathbf{k}})$ and $h(\widehat{\mathbf{l}})$ of two appropriately chosen small elements $\widehat{\mathbf{k}}, \widehat{\mathbf{l}} \in R^m$. The secret keys are $\widehat{\mathbf{k}}$ and $\widehat{\mathbf{l}}$. The messages will be polynomials $\mathbf{z} \in R$ such that $\|\mathbf{z}\|_\infty \leq 1$ and the signature of $\mathbf{z}$ is $\widehat{\mathbf{s}} = \widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$. To verify a signature $\widehat{\mathbf{s}}$ of a message $\mathbf{z}$, the verifier checks that $\|\widehat{\mathbf{s}}\|_\infty$ is "small" and also that $h(\widehat{\mathbf{k}})\mathbf{z} + h(\widehat{\mathbf{l}}) = h(\widehat{\mathbf{s}})$.

We would like to draw the reader's attention to the particulars of how the key-generation algorithm generates the secret signing key $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$. Because of the way that the integer j is generated, the secret key $\widehat{\mathbf{k}}$ (resp. $\widehat{\mathbf{l}}$) gets chosen uniformly at random from the set DK_j (resp. DL_j) with probability 2^{-j} for $1 \leq j < \lfloor \log^2 n \rfloor$

```
Key-Generation Algorithm:
Input: $1^n$, irreducible polynomial $\mathbf{f} \in \mathbb{Z}[\mathbf{x}]$ of degree $n$.
Set $p \leftarrow \Theta((\theta(\mathbf{f})n)^3)$, $m \leftarrow \lceil \log n \rceil$, $R \leftarrow \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{f} \rangle$
For all positive $i$, let the sets $DK_i$ and $DL_i$ be defined as:

    $$DK_i = \{\widehat{\mathbf{y}} \in R^m \text{ such that } \|\widehat{\mathbf{y}}\|_\infty \leq 5ip^{1/m}\}$$

    $$DL_i = \{\widehat{\mathbf{y}} \in R^m \text{ such that } \|\widehat{\mathbf{y}}\|_\infty \leq 5in\theta(\mathbf{f})p^{1/m}\}$$

Choose uniformly random $h \in \mathcal{H}(R, m)$
Pick a uniformly random string $r \in \{0,1\}^{\lfloor \log^2 n \rfloor}$
if $r = 0^{\lfloor \log^2 n \rfloor}$ then
    set $j = \lfloor \log^2 n \rfloor$
else
    set $j$ to the position of the first 1 in the string $r$
end if
Pick $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$ independently and uniformly at random from $DK_j$ and $DL_j$
    respectively
Signing Key: $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$. Verification Key: $(h, h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}))$
```

Figure 4.1 **Key-Generation Algorithm**.

and with probability 2^{-j+1} for $j = \lfloor \log^2 n \rfloor$. Since $DK_1 \subset DK_2 \subset \ldots \subset DK_{\lfloor \log^2 n \rfloor}$ and $DL_1 \subset DL_2 \subset \ldots \subset DL_{\lfloor \log^2 n \rfloor}$, the keys $\widehat{\mathbf{k}}$ and $\widehat{\mathbf{l}}$ end up being chosen from the sets $DK_{\lfloor \log^2 n \rfloor}$ and $DL_{\lfloor \log^2 n \rfloor}$, but *not* uniformly at random. Notice that keys with smaller coefficients are more likely to be chosen, and it's also extremely unlikely that we will ever end up with keys that are not in $DK_{\lfloor \log^2 n \rfloor - 1}$ and $DL_{\lfloor \log^2 n \rfloor - 1}$. So with probability negligibly close to 1, there will always be valid secret keys that are "larger" than the ones generated by the key-generation algorithm. This will be crucial to the proof of security.

We will first show that the verification algorithm always accepts the signature generated by the signing algorithm on any message $\mathbf{z} \in R$ with $\|\mathbf{z}\|_\infty \leq 1$. Note

Signing Algorithm:

Input: Message $\mathbf{z} \in R$ such that $\|\mathbf{z}\|_\infty \leq 1$; signing key $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$

Output: $\widehat{\mathbf{s}} \leftarrow \widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$

Verification Algorithm:

Input: Message $\mathbf{z}$; signature $\widehat{\mathbf{s}}$; verification key $(h, h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}))$

```
if $\|\widehat{\mathbf{s}}\|_\infty \leq 10\theta(\mathbf{f})p^{1/m}n\log^2 n$ and $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{k}})\mathbf{z} + h(\widehat{\mathbf{l}})$ then
    "ACCEPT"
else
    "REJECT"
end if
```

Figure 4.2 **Signing and Verification Algorithms**.

that the signing keys $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$ are contained in sets $DK_{\log^2 n}$ and $DL_{\log^2 n}$ respectively. Thus $\|\widehat{\mathbf{k}}\|_\infty \leq 5p^{1/m}\log^2 n$ and $\|\widehat{\mathbf{l}}\|_\infty \leq 5\theta(\mathbf{f})p^{1/m}n\log^2 n$. Therefore,

$$\|\widehat{\mathbf{s}}\|_\infty = \|\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}\|_\infty \leq \|\widehat{\mathbf{k}}\mathbf{z}\|_\infty + \|\widehat{\mathbf{l}}\|_\infty \leq \theta(\mathbf{f})n\|\widehat{\mathbf{k}}\|_\infty\|\mathbf{z}\|_\infty + \|\widehat{\mathbf{l}}\|_\infty \leq 10\theta(\mathbf{f})p^{1/m}n\log^2 n$$

Also, by the homomorphic property of functions $h \in \mathcal{H}(R, m)$,

$$h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}) = h(\widehat{\mathbf{k}})\mathbf{z} + h(\widehat{\mathbf{l}}).$$

We now show that the key-generation, signature, and verification procedures can be executed in time $\tilde{O}(n)$.

Lemma 4.1. *The key-generation, signature, and verification algorithms in Figures 4.1 and 4.2 can be performed in time* $\tilde{O}(n)$

Proof. The most time-intensive computation in the key-generation algorithm is computing $h(\widehat{\mathbf{k}})$ and $h(\widehat{\mathbf{l}})$. Since $\widehat{\mathbf{k}}, \widehat{\mathbf{l}} \in R^m$, by Claim 2.16, computing $h(\widehat{\mathbf{k}})$ and $h(\widehat{\mathbf{l}})$ takes time $m\tilde{O}(n) = \tilde{O}(n)$. Signing requires the multiplication of $\widehat{\mathbf{k}}$ by $\mathbf{z}$, (which just involves m multiplications of elements in R, and thus takes time $m\tilde{O}(n) = \tilde{O}(n)$), as well as adding the result to $\widehat{\mathbf{l}}$, which takes the same amount of time. The verification algorithm requires computing $h(\widehat{\mathbf{s}})$ and some polynomial multiplications, which can all be done in time $\tilde{O}(n)$ □

4.B Proof of Security

We next show that the above signature scheme is secure against forgery. More precisely, we show that forging a signature implies being able to solve the Col(h, D) problem, which in turn implies being able to approximate $\lambda_1(\Lambda)$ for any lattice Λ that corresponds to an ideal in the ring $\mathbb{Z}[x]/\langle \mathbf{f} \rangle$ (Theorem 4.4).

Theorem 4.2. *If there exists a polynomial-time forger that, after seeing the public key $h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}})$, and a signature $\widehat{\mathbf{s}} = \widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$ of an adaptively chosen message $\mathbf{z}$, can output a valid signature of another message $\mathbf{z}'$ with probability $1/poly(n)$, then there exists a polynomial time algorithm that can solve the Col(h, D) problem for $D = \{\mathbf{d} : \|\mathbf{d}\|_\infty \leq 10\theta(\mathbf{f})p^{1/m} n \log^2 n\}$.*

Proof. Let $\mathcal{F}$ be a forger who can break the one-time signature scheme. This means that after seeing a signature for any message of his choice, $\mathcal{F}$ can then successfully sign a different message of his choice.

Before proceeding any further, we point out that a forger who succeeds in forging a signature with non-negligible probability must succeed with non-negligible probability in the case that $j < \lfloor \log^2 n \rfloor$ in the key-generation step. This is because j equals $\lfloor \log^2 n \rfloor$ with probability only $2^{-\lfloor \log^2 n \rfloor + 1}$, and so a forger must also be able to forge signatures for other values of j if he is to have a non-negligible success probability. In the remainder of the proof, we will be assuming that the j generated in the key-generation step was less than $\lfloor \log^2 n \rfloor$. In other words, we'll be assuming that $\widehat{\mathbf{k}} \in DK_{\lfloor \log^2 n - 1 \rfloor}$ and $\widehat{\mathbf{l}} \in DL_{\lfloor \log^2 n - 1 \rfloor}$.

The algorithm below uses the forger $\mathcal{F}$ to solve the Col(h, D) problem for the parameters specified in Theorem 4.2.

```
Col($h, D$)
1: Run the Key-Generation algorithm (but use the given $h$ instead of generating a
   random one).
2: Receive message $\mathbf{z}$ from $\mathcal{F}$.
3: Send $\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$ to $\mathcal{F}$.
4: Receive message $\mathbf{z}'$ and its signature $\widehat{\mathbf{s}}'$ from $\mathcal{F}$
```

```
5: Output $\widehat{\mathbf{s}}'$ and $\widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}}$
```

We now need to show that the outputs of the above algorithm are a collision for the function h with non-negligible probability. If $\mathcal{F}$ succeeds in forging a signature $\widehat{\mathbf{s}}'$ for $\mathbf{z}'$ (which happens with non-negligible probability), then $\|\widehat{\mathbf{s}}'\|_\infty \leq 10\theta(\mathbf{f})p^{1/m}n\log^2 n$ and $h(\widehat{\mathbf{s}}') = h(\widehat{\mathbf{k}})\mathbf{z}' + h(\widehat{\mathbf{l}}) = h(\widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}})$. So if $\widehat{\mathbf{s}}' \neq \widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}}$, then our algorithm outputted two distinct elements that form a collision for the function h.

On the other hand, if $\widehat{\mathbf{s}}' = \widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}}$, then we do not get a collision. To complete the proof of Theorem 4.2, we will show that it's extremely unlikely that a forger (even one with unlimited computational power) can produce an $\widehat{\mathbf{s}}'$ and a $\mathbf{z}'$ such that $\widehat{\mathbf{s}}' = \widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}}$. This will be done in two steps. In the first step, we show that being able to produce such an $\widehat{\mathbf{s}}'$ and $\mathbf{z}'$ implies uniquely determining the signing key $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$. Then in the second step we show that given the public key $(h, h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}))$ and a signature $\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$ of message $\mathbf{z}$, it is *information theoretically* impossible to determine the signing key $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$. This means that if $\mathcal{F}$ is able to forge a signature $\widehat{\mathbf{s}}'$ for some message $\mathbf{z}'$, then almost certainly $\widehat{\mathbf{s}}' \neq \widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}}$.

We now show that obtaining an $\widehat{\mathbf{s}}'$ and a $\mathbf{z}'$ such that $\widehat{\mathbf{s}}' = \widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}}$ uniquely determines $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$. Since we know that $\widehat{\mathbf{s}} = \widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$ and $\widehat{\mathbf{s}}' = \widehat{\mathbf{k}}\mathbf{z}' + \widehat{\mathbf{l}}$, it follows that $\widehat{\mathbf{s}} - \widehat{\mathbf{s}}' = \widehat{\mathbf{k}}(\mathbf{z} - \mathbf{z}')$. Since $\|\widehat{\mathbf{k}}\|_\infty \leq 5p^{1/m}\log^2 n$ and $\|\mathbf{z} - \mathbf{z}'\|_\infty \leq 2$, multiplying $\widehat{\mathbf{k}}$ by $\mathbf{z} - \mathbf{z}'$ in the ring $\mathbb{Z}_p[x]/\langle\mathbf{f}\rangle$ is the same as multiplying them in the ring $\mathbb{Z}[x]/\langle\mathbf{f}\rangle$ because the coefficients never get big enough to get reduced modulo p. This is because

$$\begin{aligned}
\|\widehat{\mathbf{k}}(\mathbf{z} - \mathbf{z}')\|_\infty &\leq 10\theta(\mathbf{f})p^{1/m}n\log^2 n \\
&= 10\theta(\mathbf{f})\Theta\left((\theta(\mathbf{f})n)^{\frac{3}{\log n}}\right)n\log^2 n \\
&= 10\theta(\mathbf{f})\theta(\mathbf{f})^{\frac{3}{\log n}}\Theta\left(n^{\frac{3}{\log n}}\right)n\log^2 n \\
&= 80\theta(\mathbf{f})^{1+\frac{3}{\log n}}n\log^2 n \\
&= \theta(\mathbf{f})^{1+o(1)} \cdot o(n^2),
\end{aligned}$$

but in order to get reduced modulo p, the absolute value of the coefficients would have to be at least $p/2 = \Theta(\theta(\mathbf{f})^3 n^3)$, which is a much larger quantity. Now, since the ring $\mathbb{Z}[x]/\langle\mathbf{f}\rangle$ is an integral domain and $\mathbf{z} - \mathbf{z}' \neq \mathbf{0}$, there cannot exist another key $\widehat{\mathbf{k}}' \neq \widehat{\mathbf{k}}$ such that $\widehat{\mathbf{k}}'(\mathbf{z}-\mathbf{z}') = \widehat{\mathbf{k}}(\mathbf{z}-\mathbf{z}')$. And so the key $\widehat{\mathbf{k}}$ is uniquely determined (and is equal to $\frac{\widehat{\mathbf{s}}-\widehat{\mathbf{s}}'}{\mathbf{z}-\mathbf{z}'}$), and similarly the key $\widehat{\mathbf{l}} = \widehat{\mathbf{s}} - \widehat{\mathbf{k}}\mathbf{z}$ is also unique.

Now we move on to showing that by knowing only $h, h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}), \mathbf{z}$, and $\widehat{\mathbf{k}}\mathbf{z}+\widehat{\mathbf{l}}$, it is information theoretically impossible to determine the signing key $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$ (and thus, information theoretically impossible to come up with $\widehat{\mathbf{s}}', \mathbf{z}'$ such that $\widehat{\mathbf{s}}' = \widehat{\mathbf{k}}\mathbf{z}'+\widehat{\mathbf{l}}$). The idea is to show that for every $h, h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}), \mathbf{z}, \widehat{\mathbf{k}}\mathbf{z}+\widehat{\mathbf{l}}$ there is an exponential number of signing keys $(\widehat{\mathbf{k}}', \widehat{\mathbf{l}}')$, other than $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$, that satisfy $h(\widehat{\mathbf{k}}) = h(\widehat{\mathbf{k}}')$, $h(\widehat{\mathbf{l}}) = h(\widehat{\mathbf{l}}')$, and $\widehat{\mathbf{k}}\mathbf{z}+\widehat{\mathbf{l}} = \widehat{\mathbf{k}}'\mathbf{z}+\widehat{\mathbf{l}}'$. And in addition, the total probability that one of these other keys was chosen in the key-generation step (conditioned on $h, h(\widehat{\mathbf{k}}), h(\widehat{\mathbf{l}}), \mathbf{z}, \widehat{\mathbf{k}}\mathbf{z}+\widehat{\mathbf{l}}$) is almost one.

We point out that we are not proving *witness-indistinguishability*. It's actually quite possible that for every other key $(\widehat{\mathbf{k}}', \widehat{\mathbf{l}}')$, the probability that it was the key that was used to sign the message is exponentially smaller than the probability that $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$ was the key. What we will be showing is that the sum of probabilities of all other possible keys *combined* being the secret key is exponentially *larger* than the probability that $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$ was the key.

Lemma 4.3. *Let $(h, \mathbf{K}, \mathbf{L})$ be the verification key of the signature scheme and $\widehat{\mathbf{s}}$ is the signature of some message $\mathbf{z}$. Then for any signing key $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$ such that $\widehat{\mathbf{k}} \in DK_{\lfloor \log^2 n - 1 \rfloor}, \widehat{\mathbf{l}} \in DL_{\lfloor \log^2 n-1 \rfloor}$, $h(\widehat{\mathbf{k}}) = \mathbf{K}$, $h(\widehat{\mathbf{l}}) = \mathbf{L}$ and $\widehat{\mathbf{s}} = \widehat{\mathbf{k}}\mathbf{z}+\widehat{\mathbf{l}}$, the probability that this was the actual signing key generated by the key-generation algorithm is negligibly small.*

Proof. We define the set Y to be the elements of the kernel of h that have "small lengths". In particular,

$$Y = \{\widehat{\mathbf{y}} \in R^m \text{ such that } \|\widehat{\mathbf{y}}\|_\infty \leq 5p^{1/m} \text{ and } h(\widehat{\mathbf{y}}) = \mathbf{0}\}.$$

For every $\widehat{\mathbf{y}} \in Y$, consider the elements $\widehat{\mathbf{k}}' = \widehat{\mathbf{k}} - \widehat{\mathbf{y}}$ and $\widehat{\mathbf{l}}' = \widehat{\mathbf{l}} + \widehat{\mathbf{y}}\mathbf{z}$. Notice that

$$h(\widehat{\mathbf{k}}') = h(\widehat{\mathbf{k}} - \widehat{\mathbf{y}}) = h(\widehat{\mathbf{k}}) - h(\widehat{\mathbf{y}}) = \mathbf{K} - \mathbf{0} = \mathbf{K},$$

$$h(\widehat{\mathbf{l}}') = h(\widehat{\mathbf{l}} + \widehat{\mathbf{y}}\mathbf{z}) = h(\widehat{\mathbf{l}}) + h(\widehat{\mathbf{y}})\mathbf{z} = \mathbf{L} + \mathbf{0} = \mathbf{L},$$

$$\widehat{\mathbf{k}}'\mathbf{z} + \widehat{\mathbf{l}}' = (\widehat{\mathbf{k}} - \widehat{\mathbf{y}})\mathbf{z} + \widehat{\mathbf{l}} + \widehat{\mathbf{y}}\mathbf{z} = \widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}} = \widehat{\mathbf{s}}.$$

Thus, for every $\widehat{\mathbf{y}} \in Y$, if $\widehat{\mathbf{k}}'$ happens to be in $DK_{\lfloor \log^2 n \rfloor}$ and $\widehat{\mathbf{l}}'$ happens to be in $DL_{\lfloor \log^2 n \rfloor}$, then $(\widehat{\mathbf{k}}', \widehat{\mathbf{l}}')$ is another valid signing key that could have been used to sign the message $\mathbf{z}$. Since $\|\widehat{\mathbf{y}}\|_\infty \leq 5p^{1/m}$ and $\|\widehat{\mathbf{y}}\mathbf{z}\|_\infty \leq 5n\theta(\mathbf{f})p^{1/m}$, we get the following bounds on the norms of $\widehat{\mathbf{k}}'$ and $\widehat{\mathbf{l}}'$:

$$\|\widehat{\mathbf{k}}'\|_\infty \leq \|\widehat{\mathbf{k}}\|_\infty + \|\widehat{\mathbf{y}}\|_\infty \leq \|\widehat{\mathbf{k}}\|_\infty + 5p^{1/m},$$

$$\|\widehat{\mathbf{l}}'\|_\infty \leq \|\widehat{\mathbf{l}}\|_\infty + \|\widehat{\mathbf{y}}\mathbf{z}\|_\infty \leq \|\widehat{\mathbf{l}}\|_\infty + 5n\theta(\mathbf{f})p^{1/m}.$$

For the remainder of the proof, let i be the smallest integer such that $\widehat{\mathbf{k}}$ and $\widehat{\mathbf{l}}$ are contained in DK_i and DL_i respectively. Then $\widehat{\mathbf{k}}'$ and $\widehat{\mathbf{l}}'$ are definitely contained in DK_{i+1} and DL_{i+1} for every $\widehat{\mathbf{y}} \in Y$. And since we assumed that $\widehat{\mathbf{k}} \in DK_{\lfloor \log^2 n-1 \rfloor}$ and $\widehat{\mathbf{l}} \in DL_{\lfloor \log^2 n-1 \rfloor}$, it turns out that $(\widehat{\mathbf{k}}', \widehat{\mathbf{l}}')$ is a perfectly valid signing key. To prove the lemma, we will need to upper-bound the probability that the generated secret keys were $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$ given that the public keys are $\mathbf{K} = h(\widehat{\mathbf{k}})$ and $\mathbf{L} = h(\widehat{\mathbf{l}})$ and the signature of $\mathbf{z}$ is $\widehat{\mathbf{s}} = \widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$. Let E be the event that the verification key are $\mathbf{K}$ and $\mathbf{L}$ and the signature of $\mathbf{z}$ is $\widehat{\mathbf{s}}$.

$$Pr[\text{signing key} = (\widehat{\mathbf{k}}, \widehat{\mathbf{l}})|E] = \frac{Pr[\text{key} = (\widehat{\mathbf{k}}, \widehat{\mathbf{l}})\,\&\, E]}{Pr[E]} = \frac{Pr[\text{key} = (\widehat{\mathbf{k}}, \widehat{\mathbf{l}})]}{Pr[E]}$$

We now calculate the probability that the keys were $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$. This is computed by noting that $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$ were generated by selecting $j \geq i$ with probability 2^{-j} and then selecting $\widehat{\mathbf{k}}, \widehat{\mathbf{l}}$ from DK_j and DL_j. Since $\widehat{\mathbf{k}}$ and $\widehat{\mathbf{l}}$ are chosen uniformly and independently at random from DK_j and DL_j, the probability that they are both chosen is $\frac{1}{|DK_j| \cdot |DL_j|}$. So,

$$Pr[\text{signing key} = (\widehat{\mathbf{k}}, \widehat{\mathbf{l}})] = \frac{1}{2^i |DK_i||DL_i|} + \frac{1}{2^{i+1}|DK_{i+1}||DL_{i+1}|} + \ldots \tag{4.1}$$

To calculate the probability of event E, we need to figure out the probability that the keys chosen will result in public keys $\mathbf{K}$ and $\mathbf{L}$ and when given the message $\mathbf{z}$, the signature will be $\widehat{\mathbf{s}}$. We have shown above that for every $\widehat{\mathbf{y}} \in Y$, choosing the keys $\widehat{\mathbf{k}} - \widehat{\mathbf{y}}, \widehat{\mathbf{l}} + \widehat{\mathbf{y}}\mathbf{z}$ will produce public keys $\mathbf{K}, \mathbf{L}$ and signature $\widehat{\mathbf{s}}$. Since we know that $\widehat{\mathbf{k}} - \widehat{\mathbf{y}}$ and $\widehat{\mathbf{l}} + \widehat{\mathbf{y}}\mathbf{z}$ are contained in DK_{i+1} and DL_{i+1} respectively, we get

$$Pr[E] > \frac{|Y|}{2^{i+1}|DK_{i+1}||DL_{i+1}|} + \frac{|Y|}{2^{i+2}|DK_{i+2}||DL_{i+2}|} + \ldots \tag{4.2}$$

If we let $q = Pr[\text{signing key} = (\widehat{\mathbf{k}}, \widehat{\mathbf{l}})]$, then combining (4.1) and (4.2) we get

$$Pr[E] > |Y| \left(q - \frac{1}{2^i |DK_i||DL_i|} \right)$$

and so,

$$\begin{aligned} \frac{Pr[\text{signing key} = (\widehat{\mathbf{k}}, \widehat{\mathbf{l}})]}{Pr[E]} &< \frac{q}{|Y| \left(q - \frac{1}{2^i|DK_i||DL_i|} \right)} = \frac{q 2^i |DK_i||DL_i|}{|Y|(q2^i|DK_i||DL_i| - 1)} \\ &= \frac{1}{|Y|} \left(1 + \frac{1}{q2^i|DK_i||DL_i| - 1} \right) \end{aligned}$$

Before proceeding, we will state the following inequality that will be used later,

$$\begin{aligned} \frac{|DK_{i+1}||DL_{i+1}|}{|DK_i||DL_i|} &= \frac{(2 \cdot 5(i+1)p^{1/m})^{mn}(2 \cdot 5(i+1)n\theta(\mathbf{f})p^{1/m})^{mn}}{(2 \cdot 5ip^{1/m})^{mn}(2 \cdot 5in\theta(\mathbf{f})p^{1/m})^{mn}} \\ &= \left(1 + \frac{1}{i} \right)^{2mn} \leq 2^{2mn} = 4^{mn} \end{aligned}$$

Now we use the above inequality to lower bound the quantity $q2^i|DK_i||DL_i|$. Recall that q was defined to be the probability that the signing key is $(\widehat{\mathbf{k}}, \widehat{\mathbf{l}})$, and so from Equation (4.1), we obtain

$$\begin{aligned} q2^i|DK_i||DL_i| &= 2^i|DK_i||DL_i| \left(\frac{1}{2^i|DK_i||DL_i|} + \frac{1}{2^{i+1}|DK_{i+1}||DL_{i+1}|} + \dots \right) \\ &> 2^i|DK_i||DL_i| \left(\frac{1}{2^i|DK_i||DL_i|} + \frac{1}{2^{i+1}|DK_{i+1}||DL_{i+1}|} \right) \\ &= 1 + \frac{|DK_i||DL_i|}{2|DK_{i+1}||DL_{i+1}|} \geq 1 + \frac{1}{2 \cdot 4^{mn}} \end{aligned}$$

Using the above inequality, we obtain

$$\frac{Pr[\text{signing key} = (\widehat{\mathbf{k}}, \widehat{\mathbf{l}})]}{Pr[E]} < \frac{1}{|Y|} \left(1 + \frac{1}{q2^i|DK_i||DL_i| - 1} \right) \leq \frac{1}{|Y|}(1 + 2 \cdot 4^{mn})$$

and since by Lemma 2.18 we know that $|Y| \geq 5^{mn}$, we are done. □

This concludes the proof Theorem 4.2.

□

Strong unforgeability

We now show that our one-time signature scheme also satisfies a stronger notion of security, called *strong unforgeability*. In the previous section we showed that if a forger can produce a signature for an unseen message, then $\mathrm{Col}(h, D)$ can be solved in polynomial time. Now we point out that $\mathrm{Col}(h, D)$ can be solved in polynomial time even if the forger is able to produce a different signature of a message whose signature he has seen. Suppose that after seeing the signature $\widehat{\mathbf{s}} = \widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}$ of a message $\mathbf{z}$, the forger $\mathcal{F}$ sends back another valid signature $\widehat{\mathbf{s}}' \neq \widehat{\mathbf{s}}$ of $\mathbf{z}$. Then $\widehat{\mathbf{s}}$ and $\widehat{\mathbf{s}}'$ form a collision for h. This is because

$$h(\widehat{\mathbf{s}}') = h(\widehat{\mathbf{k}})\mathbf{z} + h(\widehat{\mathbf{l}}) = h(\widehat{\mathbf{k}}\mathbf{z} + \widehat{\mathbf{l}}) = h(\widehat{\mathbf{s}}).$$

Corollary 4.4. *For any monic irreducible $\mathbf{f} \in \mathbb{Z}[\mathbf{x}]$, if there exists a polynomial-time forger that breaks the one-time strong signature scheme described in Figures 4.1 and 4.2, then there is a polynomial-time algorithm that solves $\mathbf{f}$-$SVP_\gamma(\Lambda)$ for $\gamma = \theta(\mathbf{f})^{3+o(1)}\tilde{O}(n^2)$ for every lattice Λ that corresponds to an ideal in $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$.*

Proof. From Theorem 4.2, and the above discussion about strong unforgeability, we know that breaking the signature implies solving $\mathrm{Col}(h, D)$, where

$$D = \{\mathbf{d} : \|\mathbf{d}\|_\infty \leq 10\theta(\mathbf{f})p^{1/m} n \log^2 n = \theta(\mathbf{f})^{1+o(1)}\tilde{O}(n)\}$$

and h is a random function from $\mathcal{H}(R, D, m)$. All the parameters satisfy the conditions of Theorem 3.1, and therefore solving $\mathrm{Col}(h, D)$ implies solving $\mathbf{f}$-SVP_γ. □

Chapter 4 is, in part, a reprint of the paper "Asymptotically Efficient Lattice-Based Digital Signatures" co-authored with Daniele Micciancio and appearing in the proceeding of TCC 2008. The dissertation author was the primary investigator and author of this paper.

5

Identification Scheme

In this chapter, we describe an identification scheme that is asymptotically efficient up to logarithmic factors. We will give a construction of a 3-round public coin ID scheme in which the interaction takes time $\tilde{O}(n)$ with security based on the hardness of solving $\mathbf{f}$-SVP$_\gamma$ for some polynomial factor γ, which is conjectured to be a problem requiring $2^{\Omega(n)}$ time. In order to slightly simplify the presentation, we will only prove the scheme secure for $\mathbf{f} = \mathbf{x}^n + 1$, but it's fairly straightforward to extend our results, with appropriate modifications, for other polynomials $\mathbf{f}$. The parameters used in the ID scheme are described in Table 5.1.

5.A Probabilistic Lemmas

The following simple, yet crucial, lemma will be used to prove that our identification scheme does not abort too often (Lemma 5.3). The lemma basically states that the number of $\widehat{\mathbf{y}} \in D_y^m$ such that $\widehat{\mathbf{w}} + \widehat{\mathbf{y}} \in \mathrm{G}^m$ (for parameters D_y and G in Table 5.1) is the same for every $\widehat{\mathbf{w}}$ that is small enough.

Lemma 5.1. *For any $\widehat{\mathbf{w}}$ such that $\|\widehat{\mathbf{w}}\|_\infty \leq n$,*

$$Pr_{\widehat{\mathbf{y}} \overset{\$}{\leftarrow} D_y^m}[\widehat{\mathbf{w}} + \widehat{\mathbf{y}} \in G^m] = \left(\frac{2(mn^2 - n) + 1}{2mn^2 + 1}\right)^{mn} = \frac{1}{e} - o(1)$$

Proof. Given some $\widehat{\mathbf{w}}$ such that $\|\widehat{\mathbf{w}}\|_\infty \leq n$, let's look at it as a vector of dimension mn with coefficients (call them w_j, for $1 \leq j \leq mn$) having absolute value at most n.

n	integer that is a power of 2
p	prime of order $\Theta(n^4)$
m	$3\log n$
R	field $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n+1\rangle$
D	$\{\mathbf{g}\in R : \lVert\mathbf{g}\rVert_\infty \le mn^2\}$
D_c	$\{\mathbf{g}\in R : \lVert\mathbf{g}\rVert_\infty \le 1\}$
D_s	$\{\mathbf{g}\in R : \lVert\mathbf{g}\rVert_\infty \le 1\}$
D_y	$\{\mathbf{g}\in R : \lVert\mathbf{g}\rVert_\infty \le mn^2\}$
G	$\{\mathbf{g}\in R : \lVert\mathbf{g}\rVert_\infty \le mn^2-n\}$

Figure 5.1 **ID-scheme Variable Definitions**.

The sum $\widehat{\mathbf{w}}+\widehat{\mathbf{y}}$ will be in G^m if for every coefficient w_j, the corresponding coefficient of $\widehat{\mathbf{y}}$ (call it y_j) is in the range

$$[-mn^2+n-w_j, mn^2-n-w_j]. \tag{5.1}$$

Because every coefficient y_j is generated randomly in the range

$$[-mn^2, mn^2],$$

the probability that it is in the range (5.1) is *exactly*

$$\frac{2(mn^2-n+1)}{2mn^2+1}. \tag{5.2}$$

Notice that in the above equation, we made crucial use of the fact that the range in equation (5.1) is completely contained in the range of possible coefficients y_j of $\widehat{\mathbf{y}}$ (this is because $|w_j|\le n$). The probability that $\widehat{\mathbf{w}}+\widehat{\mathbf{y}}\in\mathrm{G}^m$ is just the quantity in equation (5.2) raised to the power mn.

$$\begin{aligned} Pr_{\widehat{\mathbf{y}}\xleftarrow{\$}D_y^m}[\widehat{\mathbf{w}}+\widehat{\mathbf{y}}\in\mathrm{G}^m \,|\, \|\widehat{\mathbf{w}}\|_\infty\le n] &= \left(\frac{2(mn^2-n)+1}{2mn^2+1}\right)^{mn} \\ &> \left(1-\frac{1}{mn}\right)^{mn} = \frac{1}{e}-o(1) \end{aligned}$$

□

The next lemma establishes that when we choose the public key for the identification scheme, there will be (with high probability) at least two different secret keys that could correspond to it.

Private key: $\widehat{\mathbf{s}} \xleftarrow{\$} D_s^m$

Public key: $h \xleftarrow{\$} \mathcal{H}(R, m), \mathbf{S} \leftarrow h(\widehat{\mathbf{s}})$

Prover		Verifier
$\widehat{\mathbf{y}} \xleftarrow{\$} D_y^m$		
$\mathbf{Y} \leftarrow h(\widehat{\mathbf{y}})$		
	$\xrightarrow{\mathbf{Y}}$	
		$\mathbf{c} \xleftarrow{\$} D_c$
	$\xleftarrow{\mathbf{c}}$	
$\widehat{\mathbf{z}} \leftarrow \widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}$		
if $\widehat{\mathbf{z}} \notin \mathrm{G}^m$ then $\widehat{\mathbf{z}} \leftarrow \perp$		
	$\xrightarrow{\widehat{\mathbf{z}}}$	
		$d \leftarrow \begin{cases} 1 & \text{if } \widehat{\mathbf{z}} \in \mathrm{G}^m \text{ and } h(\widehat{\mathbf{z}}) = \mathbf{Sc} + \mathbf{Y} \\ 0 & \text{otherwise} \end{cases}$

Figure 5.2 **Identification Scheme**.

Lemma 5.2. *If we pick an $\widehat{\mathbf{s}}$ uniformly at random from D_s^m, then with probability $1 - 2^{-\Omega(n \log n)}$, there will be another $\widehat{\mathbf{s}}' \in D_s^m$ such that $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{s}}')$.*

Proof. The range of h (which is R) consists of p^n elements. This means that there are at most p^n elements $\widehat{\mathbf{s}} \in D_s^m$ that do not collide with any other element in D_s^m. Since the set D_s^m contains 3^{mn} elements, the probability of randomly selecting a non-colliding element is at most

$$\left(\frac{p}{3^m}\right)^n = 2^{-\Omega(n \log n)}.$$

□

5.B Identification Scheme

We present our identification scheme in Figure 5.2. More precisely, we present *one round* of our scheme, and the full identification scheme will consist of

$\omega(\log n)$ rounds performed in parallel. The prover will only need to succeed in one round of the scheme in order to be accepted by the verifier. The reason that we need $\omega(\log n)$ rounds is for completeness because the prover may not want to respond in every round for security reasons that we will describe shortly.

To generate the keys, we pick a hash function $h \overset{\$}{\leftarrow} \mathcal{H}(R, m)$ and a secret key $\widehat{\mathbf{s}} \overset{\$}{\leftarrow} D_s^m$. The public key consists of the function h and an element $\mathbf{S} \in R$ where $\mathbf{S} = h(\widehat{\mathbf{s}})$. In the first step of the protocol, the prover picks a $\widehat{\mathbf{y}} \in D_y^m$ and sends the commitment $\mathbf{Y} = h(\widehat{\mathbf{y}})$ to the verifier. The verifier sends a random challenge $\mathbf{c} \overset{\$}{\leftarrow} D_c$ and the prover then computes $\widehat{\mathbf{z}} = \widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}$. If $\widehat{\mathbf{z}}$ is in the set G^m, then the prover sends $\widehat{\mathbf{z}}$ to the verifier, but if $\widehat{\mathbf{z}} \notin \mathrm{G}^m$, the prover aborts the protocol and sends $\perp$ to signify his action. If the prover does not abort, then the verifier checks whether $\widehat{\mathbf{z}} \in \mathrm{G}^m$ and $h(\widehat{\mathbf{z}}) = \mathbf{S}\mathbf{c} + \widehat{\mathbf{y}}$.

The reason that the prover needs to sometimes abort is because if he sends a $\widehat{\mathbf{z}} = \widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}$ that is not in G^m, he will be revealing some information about his secret key $\widehat{\mathbf{s}}$ (for example, if $\widehat{\mathbf{s}}$ has many non-zero coefficients, then $\|\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}\|_\infty$ will tend to be large). On the other hand, by sending $\widehat{\mathbf{z}} \in \mathrm{G}^m$, and aborting otherwise, the prover ensures that the protocol will be witness-indistinguishable (Theorem 5.5). That is, it will be impossible to tell which of the many possible $\widehat{\mathbf{s}}$'s that satisfy $h(\widehat{\mathbf{s}}) = \mathbf{S}$ is the secret key. Because witness-indistinguishability is preserved under parallel composition, the protocol can be repeated in parallel and the verifier will accept if and only if in any one round, he set $d = 1$. We will show that the prover will abort approximately $1 - 1/e$ fraction of the time (Lemma 5.3), and therefore the expected number of rounds needed is $1/e$. If the protocol has $\omega(\log n)$ rounds, then the honest prover will be accepted all but a negligible fraction of the time.

Lemma 5.3. *A prover in possession of an* $\widehat{\mathbf{s}} \in D_s^m$ *such that* $h(\widehat{\mathbf{s}}) = \mathbf{S}$ *will be accepted with probability* $1/e - o(1)$*.*

Proof. Because $\|\widehat{\mathbf{s}}\|_\infty = 1$ and $\|\mathbf{c}\|_\infty = 1$, we have $\|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq \theta(\mathbf{x}^n + 1)n = n$. We now apply Lemma 5.1 and conclude that $\widehat{\mathbf{z}} = \widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}$ will be in G^m with probability $1/e - o(1)$. And also

$$h(\widehat{\mathbf{z}}) = h(\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} = h(\widehat{\mathbf{s}})\mathbf{c} + h(\widehat{\mathbf{y}}) = \mathbf{S}\mathbf{c} + \mathbf{Y}.$$

□

Lemma 5.4. *The running-time of the identification protocol is $\tilde{O}(n)$.*

Proof. The first move of the protocol requires the prover to compute $h(\widehat{\mathbf{y}})$, which takes time $\tilde{O}(n)$ (2.16). The response of the prover requires the computation of $\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}$, which also takes $\tilde{O}(n)$ time. The verifier then needs to compute $h(\widehat{\mathbf{z}})$ and $\mathbf{S}\mathbf{c} + \mathbf{Y}$, which again takes $\tilde{O}(n)$ time. □

5.C Proof of Security

We will now present the proof of witness-indistinguishability of the identification scheme in Figure 5.2. In other words, we will show that the distribution of $\mathbf{Y}, \mathbf{c}, \widehat{\mathbf{z}}$ is completely independent of the secret key $\widehat{\mathbf{s}}$ such that $h(\widehat{\mathbf{s}}) = \mathbf{S}$. We will show is that for any two $\widehat{\mathbf{s}}, \widehat{\mathbf{s}}'$ such that $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{s}}')$, no (possibly cheating) verifier $\mathcal{V}$ can distinguish whether the secret key is $\widehat{\mathbf{s}}$ or $\widehat{\mathbf{s}}'$.

The witness-indistinguishability of the protocol will be then used in Theorem 5.6 to show that the adversary cannot know the exact secret key, and therefore cannot control the exact information being extracted from it.

Theorem 5.5. *For every $h \in \mathcal{H}(R, m)$, and every (possibly cheating) verifier $\mathcal{V}$ with any auxiliary input α*

$$\Delta\left(\mathcal{V}_{\mathcal{P}(h,\mathbf{S},\widehat{\mathbf{s}})}(h, \mathbf{S}, \alpha), \mathcal{V}_{\mathcal{P}(h,\mathbf{S},\widehat{\mathbf{s}}')}(h, \mathbf{S}, \alpha)\right) = 0,$$

where $\widehat{\mathbf{s}}$ and $\widehat{\mathbf{s}}'$ are any two elements in D_s^m such that $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{s}}') = \mathbf{S}$.

Proof. Because the value of $\mathbf{c}$ is independent of the particular value of $\widehat{\mathbf{y}} \in h^{-1}(\mathbf{Y})$, we can rewrite the identification protocol so that $\widehat{\mathbf{y}}$ is picked after the verifier picks the challenge $\mathbf{c}$. We consider the following protocol:

1. Prover picks $\widehat{\boldsymbol{\gamma}} \overset{\$}{\leftarrow} D_y^m$ and sends $\mathbf{Y} = h(\widehat{\boldsymbol{\gamma}})$ to the verifier.

2. Verifier picks $\mathbf{c} \in D_c$ and sends it to the prover.

3. Prover picks $\widehat{\mathbf{y}} \overset{\$}{\leftarrow} h^{-1}(\mathbf{Y}) \cap D_y^m$, and computes $\widehat{\mathbf{z}} = \widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}$.
   If $\widehat{\mathbf{z}} \in \mathrm{G}^m$, send $\widehat{\mathbf{z}}$ to the verifier. Otherwise, send $\perp$.

Because the distribution of $\widehat{\gamma}$ in step 1 is identical to the distribution of $\widehat{\mathbf{y}}$ in step 3, the distribution of $\widehat{\mathbf{z}}$ in the above protocol and the protocol in Figure 5.2 is identical as well. Notice that the witness $\widehat{\mathbf{s}}$ only affects $\widehat{\mathbf{z}}$, and so to prove the theorem, we only need to show that the distribution of $\widehat{\mathbf{z}}$ is not affected by the choice of the witness in the domain $h^{-1}(\mathbf{S}) \cap D_s^m$ and the challenge $\mathbf{c} \in D_c$.

Let $\widehat{\mathbf{s}}$ and $\widehat{\mathbf{s}}'$ be any two elements in $h^{-1}(\mathbf{S}) \cap D_s^m$ and $\mathbf{c}$ be any challenge in D_c. Notice that for any $\widehat{\mathbf{y}} \in h^{-1}(\mathbf{Y}) \cap D_y^m$ such that $\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} = \widehat{\mathbf{z}} \in \mathrm{G}^m$, the value $\widehat{\mathbf{s}}'\mathbf{c} + \widehat{\mathbf{y}}'$ for $\widehat{\mathbf{y}}' = \widehat{\mathbf{y}} + \widehat{\mathbf{s}}\mathbf{c} - \widehat{\mathbf{s}}'\mathbf{c}$ is also equal to $\widehat{\mathbf{z}}$. Therefore if $\widehat{\mathbf{y}}'$ is also in $h^{-1}(\mathbf{Y}) \cap D_y^m$, then the fact that for every $\widehat{\mathbf{y}}$ there is a corresponding $\widehat{\mathbf{y}}'$ such that $\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} = \widehat{\mathbf{s}}'\mathbf{c} + \widehat{\mathbf{y}}'$ proves that for every $\widehat{\mathbf{z}} \in \mathrm{G}^m$, the probability that $\widehat{\mathbf{z}}$ will be output by the prover is independent of the $\widehat{\mathbf{s}} \in h^{-1}(\mathbf{S}) \cap D_s^m$. Furthermore, the above also proves that the number of elements $\widehat{\mathbf{y}} \in h^{-1}(\mathbf{Y}) \cap D_y^m$ such that $\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} \notin \mathrm{G}^m$ is equal for every $\widehat{\mathbf{s}} \in h^{-1}(\mathbf{S}) \cap D_s^m$. Therefore proving that $\widehat{\mathbf{y}}' \in h^{-1}(\mathbf{Y}) \cap D_y^m$ will prove the theorem.

We first prove that $\widehat{\mathbf{y}}' \in h^{-1}(\mathbf{Y})$ by showing that $h(\widehat{\mathbf{y}}) = h(\widehat{\mathbf{y}}')$. By the homomorphic property of h, we have

$$h(\widehat{\mathbf{y}}') = h(\widehat{\mathbf{y}} + \widehat{\mathbf{s}}\mathbf{c} - \widehat{\mathbf{s}}'\mathbf{c}) = h(\widehat{\mathbf{y}}) + (h(\widehat{\mathbf{s}}) - h(\widehat{\mathbf{s}}'))\mathbf{c} = h(\widehat{\mathbf{y}}).$$

We now show that $\widehat{\mathbf{y}}' \in D_y^m$. By Lemmas 2.8 and 2.6 we know that

$$\|\widehat{\mathbf{s}}'\mathbf{c}\|_\infty \leq n\theta(\mathbf{x}^n + 1)\|\widehat{\mathbf{s}}'\|_\infty \|\mathbf{c}\|_\infty \leq n,$$

and by the asumption that $\widehat{\mathbf{y}} + \widehat{\mathbf{s}}\mathbf{c} \in \mathrm{G}^m$, we obtain

$$\|\widehat{\mathbf{y}}'\|_\infty = \|\widehat{\mathbf{y}} + \widehat{\mathbf{s}}\mathbf{c} - \widehat{\mathbf{s}}'\mathbf{c}\|_\infty \leq \|\widehat{\mathbf{y}} + \widehat{\mathbf{s}}\mathbf{c}\|_\infty + \|\widehat{\mathbf{s}}'\mathbf{c}\|_\infty \leq mn^2 - n + n = mn^2.$$

Therefore $\widehat{\mathbf{y}}' \in h^{-1}(\mathbf{Y}) \cap D_y^m$. □

We now show that our ID-scheme is secure by showing how to use an adversary who successfully attacks the ID scheme to solve the $\mathrm{Col}(h, D)$ problem.

Theorem 5.6. *If h is any function in $\mathcal{H}(R, m)$ for the parameters defined in Table 5.1 and there exists a polynomial-time adversary who can break the ID scheme with probability q in the active attack model, then there exists a polynomial-time algorithm that solves Col(h, D) with probability at least $\frac{q}{4}\left(q - \frac{1}{|D_c|}\right) - 2^{-\Omega(n \log n)}$.*

Proof. We will now describe how to solve the $\mathrm{Col}(h, D)$ problem when given access to an adversary who breaks the ID scheme. Given a random function $h \in \mathcal{H}(R, m)$, we create a secret key $\widehat{\mathbf{s}} \in D_s^m$, and compute the public key $\mathbf{S} = h(\widehat{\mathbf{s}})$. We give the public key $h, \mathbf{S}$ to the adversary. In the first stage of the attack, when the adversary acts as the verifier, we can perfectly simulate the interaction between him and the prover because we have the secret key. In the second stage of the attack, when it's the adversary's turn to impersonate the honest prover, he will initiate the interaction by sending some $\mathbf{Y}$, and we will respond with a random $\mathbf{c} \in D_c$. The adversary will (with probability at least q) output a $\widehat{\mathbf{z}}$ such that $\widehat{\mathbf{z}} \in \mathrm{G}^m$ and $h(\widehat{\mathbf{z}}) = \mathbf{Sc} + \mathbf{Y}$. We will then rewind the adversary, and send him a different random challenge $\mathbf{c}'$ from D_c. The adversary will output a $\widehat{\mathbf{z}}'$ such that $\widehat{\mathbf{z}}' \in \mathrm{G}^m$ and $h(\widehat{\mathbf{z}}') = \mathbf{Sc}' + \mathbf{Y}$. The following claim concerning the outputs of the adversary will be will be proved after the theorem.

Claim 5.7. *If challenges* $\mathbf{c}$ *and* $\mathbf{c}'$ *are chosen at random from* D_c*, then the probability that* $\mathbf{c} \neq \mathbf{c}'$ *and the adversary produces valid responses* $\widehat{\mathbf{z}}, \widehat{\mathbf{z}}'$ *to both challenges (i.e.* $h(\widehat{\mathbf{z}}) = \mathbf{Sc} + \mathbf{Y}$ *and* $h(\widehat{\mathbf{z}}') = \mathbf{Sc}' + \mathbf{Y}$*) is at least* $\frac{q}{2}\left(q - \frac{1}{|D_c|}\right)$.

Assuming that the adversary gave valid answers both times, we can combine his two responses and obtain

$$h(\widehat{\mathbf{z}}) - \mathbf{Sc} = h(\widehat{\mathbf{z}}') - \mathbf{Sc}'.$$

Using the homomorphic properties of h and the fact that we know an $\widehat{\mathbf{s}}$ such that $h(\widehat{\mathbf{s}}) = \mathbf{S}$, we can obtain the equality

$$h(\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{c}) = h(\widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{c}').$$

Because $\|\widehat{\mathbf{s}}\|_\infty = \|\mathbf{c}\|_\infty = 1$, we have $\|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq n$, and also we know that $\widehat{\mathbf{z}}, \widehat{\mathbf{z}}' \in \mathrm{G}^m$, and therefore $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{c}, \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}'\mathbf{c}' \in D$. So we have a solution to $\mathrm{Col}(h, D)$, except in the case that $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{c} = \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{c}'$. By Lemma 5.2, we know that for a randomly chosen $\widehat{\mathbf{s}}$, there is a $1 - 2^{-\Omega(n \log n)}$ probability that there is another $\widehat{\mathbf{s}}' \in D_s^m$ such that $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{s}}')$. By the perfect witness-indistinguishability property of the protocol established in Theorem 5.5, the adversary cannot know with probability greater than 1/2 whether the secret key was $\widehat{\mathbf{s}}$ or $\widehat{\mathbf{s}}'$. We will now show that if $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{c} = \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{c}'$

	r_1	r_2	r_3	$\ldots$	r_{2^ρ}
$\mathbf{c}_1$	1		1		
$\mathbf{c}_2$		1	1		1
$\mathbf{c}_3$	1	1			1
$\ldots$					
$\mathbf{c}_{\lvert D_c \rvert}$	1		1		1

Figure 5.3 **Interaction Outcomes**.

then $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}'\mathbf{c} \neq \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}'\mathbf{c}'$ and therefore the probability that we will obtain a colision is at least 1/2. For contradiction, assume that $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{c} = \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{c}'$ and $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}'\mathbf{c} = \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}'\mathbf{c}'$. Combining these two equalities, we obtain that

$$(\widehat{\mathbf{s}} - \widehat{\mathbf{s}}')(\mathbf{c} - \mathbf{c}') = 0.$$

Since we are doing the multiplication in the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$, which *is not* an integral domain, we cannot automatically conclude that the preceding equality implies that either $\widehat{\mathbf{s}} - \widehat{\mathbf{s}}'$ or $\mathbf{c} - \mathbf{c}'$ is 0. But we can make the following observation: since all the coordinates of $\widehat{\mathbf{s}}, \widehat{\mathbf{s}}', \mathbf{c}$, and $\mathbf{c}'$ have absolute value at most 1, the differences $\widehat{\mathbf{s}} - \widehat{\mathbf{s}}'$ and $\mathbf{c} - \mathbf{c}'$ have coordinates whose absolute value is at most 2. When we multiply such polynomials together in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$, the absolute values of the coefficients of the product are never above $4n$. In order to be reduced modulo $p = \Theta(n^4)$, the absolute value of the coefficients would have to be greater than $p/2$, but $4n$ is much smaller than that. Therefore if the product $(\widehat{\mathbf{s}} - \widehat{\mathbf{s}}')(\mathbf{c} - \mathbf{c}')$ is $\mathbf{0}$ in the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$, it also must be $\mathbf{0}$ in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$. But because $\mathbf{x}^n + 1$ is irreducible over the integers, $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ *is* an integral domain, and therefore either $\widehat{\mathbf{s}} - \widehat{\mathbf{s}}'$ or $\mathbf{c} - \mathbf{c}'$ is 0. Since we know that $\mathbf{e} \neq \mathbf{e}'$, it must be that $\widehat{\mathbf{s}} = \widehat{\mathbf{s}}'$, and we have a contradiction. □

Proof of Claim 5.7. The adversary impersonating a prover is a randomized algorithm, which can be represented as a deterministic Turing machine with a random tape r consisting of ρ binary digits, where ρ is bounded by the running time of the adversarial prover. We will represent the possible outcomes of the interaction between the adversarial prover and an honest verifier in Figure 5.3. A 1 in a cell at the intersection of $\mathbf{c}_i$ and r_i means that if the prover's random tape is r_i and the challenge is $\mathbf{c}_i$, then the prover succeeds in the impersonation. Because we assumed that the probability (over the choice of the random tape and the challenges) with which the adversary can break the ID scheme is q, we know that a q-fraction of all the cells in the table must have a 1. By a counting argument, we can say that at least half of the columns must have at least $q|D_c|$ ones in them. Therefore the probability that two random challenges are distinct and the prover successfully responds to them when given a randomly chosen random tape is at least

$$\frac{1}{2}\left(\frac{q|D_c|}{|D_c|}\right)\left(\frac{q|D_c|-1}{|D_c|-1}\right) > \frac{q}{2}\left(q-\frac{1}{|D_c|}\right).$$

□

The above theorem together with Theorem 3.1 imply the following corollary, which gives a relationship between breaking the ID scheme and solving $\mathbf{f}$-SVP$_\gamma(\Lambda)$ for all lattices that correspond to ideals in $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$.

Corollary 5.8. *If the identification scheme in Figure 5.2 is insecure against active attacks for the parameters in Table 5.1, then there is polynomial-time algorithm that can solve $\mathbf{f}$-SVP$_\gamma(\Lambda)$ for $\gamma = \tilde{O}(n^3)$ for every lattice Λ corresponding to an ideal in the ring $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$.*

Proof. By Theorem 5.6, we know that breaking the ID scheme implies solving the Col(h, D) problem where h is any function in $\mathcal{H}(R, D, m)$ for the parameters in Table 5.1. It's easy to check that the parameters satisfy Theorem 3.1, and therefore solving Col(h, D) for random $h \in \mathcal{H}(R, D, m)$ implies that we can solve $\mathbf{f}$-SVP$_\gamma(\Lambda)$ for $\gamma = 16\theta(\mathbf{f})^2(mn^2)n\log^2 n = \tilde{O}(n^3)$ for every lattice Λ corresponding to an ideal in $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$. □

Chapter 5 is an extension of the results of the paper "Lattice-Based Identification Schemes Secure Under Active Attacks" appearing in the proceedings of PKC 2008. The dissertation author was the primary investigator and author of this paper.

6

Tree-less Signature Scheme

In this chapter we describe a signature scheme whose security, in the random oracle model, is based on the hardness of solving $\mathbf{f}$-SVP$_\gamma(\Lambda)$ for all lattices corresponding to ideals in the ring $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$. As for the identification scheme in the previous chapter, we will only present the proof for $\mathbf{f} = \mathbf{x}^n + 1$. But with appropriate modifications, it's fairly straightforward to extend our results to other polynomials $\mathbf{f}$. Unlike for the identification scheme, we will not need the ring $\mathbb{Z}_p[\mathbf{x}]/\langle\mathbf{f}\rangle$ to be a field.

Since we already have an identification scheme that we proved to be secure, we can construct a signature scheme by simply applying the Fiat-Shamir heuristic in a black-box fashion (see for example [AABN02]). But it turns out that in our case, we can construct a slightly more efficient signature scheme when we put a little "twist" on the Fiat-Shamir transform. Recall that a single round of the ID scheme from the previous chapter did not have perfect completeness because the prover sometimes chose to abort the interaction. But because there is no interaction in signature schemes, there is no point in having the rounds that lead to an abort be included in the transformation. Thus, the signer may choose as many potential commitments $\widehat{\mathbf{y}}$ as he wants, and get "challenges" $\mathbf{c}$ from the random oracle until he gets a value of $\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}}$ that is in the set G^m, and it's only that "round" of the ID protocol that will be used in the signature.

The parameters used in our signature scheme are given in Table 6.1, and we will prove our scheme secure by showing that the signing algorithm is witness-

n	integer that is a power of 2
p	prime of order $\Theta(n^4)$
m	$3\log n$
R	ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$
D	$\{\mathbf{f} \in R : \Vert\mathbf{f}\Vert_\infty \le mn^{1.5}\log n + \sqrt{n}\log n\}$
D_c	$\{\mathbf{f} \in R : \Vert\mathbf{f}\Vert_\infty \le 1\}$
D_s	$\{\mathbf{f} \in R : \Vert\mathbf{f}\Vert_\infty \le 1\}$
D_y	$\{\mathbf{f} \in R : \Vert\mathbf{f}\Vert_\infty \le mn^{1.5}\log n\}$
G	$\{\mathbf{f} \in R : \Vert\mathbf{f}\Vert_\infty \le mn^{1.5}\log n - \sqrt{n}\log n\}$

Figure 6.1 **Signature Scheme Variable definitions**.

indistinguishable and a proof of knowledge of an element in $\widehat{\boldsymbol{\alpha}} \in D^m$ such that $h(\widehat{\boldsymbol{\alpha}})$ is the public key. Using this knowledge-extractor, we will be able to find two elements in D^m that evaluate to the same value by the hash function h which is randomly chosen from $\mathcal{H}(R, D, m)$. Notice that the set D is smaller than it was in the ID scheme chapter. This savings of $\sqrt{n}$ is mainly due to the fact that in this chapter we are using a random oracle to generate elements that correspond to "challenges", while in the ID scheme, the verifier challenges could have been chosen in an adversarial fashion.

6.A Probabilistic Lemmas

This section contains some auxiliary lemmas that will be used in our proofs. The first lemma is analogous to Lemma 5.1 and states that the number of $\widehat{\mathbf{y}} \in D_y^m$ such that $\widehat{\mathbf{w}} + \widehat{\mathbf{y}} \in \mathrm{G}^m$ is the same for every $\widehat{\mathbf{w}}$ that is small enough.

Lemma 6.1. *For any $\widehat{\mathbf{w}}$ such that $\|\widehat{\mathbf{w}}\|_\infty \le \sqrt{n}\log n$,*

$$Pr_{\widehat{\mathbf{y}} \xleftarrow{\$} D_y^m}[\widehat{\mathbf{w}} + \widehat{\mathbf{y}} \in G^m] = \left(\frac{2(mn^{1.5}\sqrt{n} - \sqrt{n}\log n) + 1}{2mn^{1.5}\log n + 1}\right)^{mn} = \frac{1}{e} - o(1)$$

Proof. Given some $\widehat{\mathbf{w}}$ such that $\|\widehat{\mathbf{w}}\|_\infty \le \sqrt{n}\log n$, let's look at it as a vector of dimension mn with coefficients (call them w_j, for $1 \le j \le mn$) having absolute value at most $\sqrt{n}\log n$. The sum $\widehat{\mathbf{w}} + \widehat{\mathbf{y}}$ will be in G^m if for every coefficient w_j, the corresponding coefficient of $\widehat{\mathbf{y}}$ (call it y_j) is in the range

$$[-mn^{1.5}\log n + \sqrt{n}\log n - w_j, mn^{1.5}\log n - \sqrt{n}\log n - w_j]. \tag{6.1}$$

Because every coefficient y_j is generated randomly in the range

$$[-mn^{1.5}\log n, mn^{1.5}\log n],$$

the probability that it is in the range (6.1) is *exactly*

$$\frac{2(mn^{1.5}\log n - \sqrt{n}\log n) + 1}{2mn^{1.5}\log n + 1}. \tag{6.2}$$

Notice that in the above equation, we made crucial use of the fact that the range in equation (6.1) is completely contained in the range of possible coefficients y_j of $\widehat{\mathbf{y}}$ (this is because $|w_j| \leq \sqrt{n}\log n$). The probability that $\widehat{\mathbf{w}} + \widehat{\mathbf{y}} \in \mathrm{G}^m$ is just the quantity in equation (6.2) raised to the power mn.

$$\begin{aligned} Pr_{\widehat{\mathbf{y}} \overset{\$}{\leftarrow} D_y^m}[\widehat{\mathbf{w}} + \widehat{\mathbf{y}} \in \mathrm{G}^m | \|\widehat{\mathbf{w}}\|_\infty \leq \sqrt{n}\log n] &= \left(\frac{2(mn^{1.5}\sqrt{n} - \sqrt{n}\log n) + 1}{2mn^{1.5}\log n + 1}\right)^{mn} \\ &> \left(1 - \frac{1}{mn}\right)^{mn} = \frac{1}{e} - o(1) \end{aligned}$$

□

The following simple corollary states that if $\mathbf{c}$ and $\widehat{\mathbf{y}}$ are chosen at random from the domains D_c and D_y^m, then there is a high probability that $\widehat{\mathbf{s}}\mathbf{c} + \mathbf{y}$ will be in G^m for any $\widehat{\mathbf{s}} \in D_s^m$. The reason that we need $\mathbf{c}$ to be random is that we can upper-bound $\|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq \sqrt{n}\log n$, whereas if $\mathbf{c}$ were any element in D_c, we could only say that $\|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq n$.

Corollary 6.2. *For any* $\widehat{\mathbf{s}} \in D_s^m$,

$$Pr_{\mathbf{c} \overset{\$}{\leftarrow} D_c, \widehat{\mathbf{y}} \overset{\$}{\leftarrow} D_y^m}[\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} \in G^m] = \frac{1}{e} - o(1)$$

Proof. We lower-bound

$$Pr_{\mathbf{c} \overset{\$}{\leftarrow} D_c, \widehat{\mathbf{y}} \overset{\$}{\leftarrow} D_y^m}[\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} \in \mathrm{G}^m]$$

by

$$\begin{aligned} Pr_{\mathbf{c} \overset{\$}{\leftarrow} D_c, \widehat{\mathbf{y}} \overset{\$}{\leftarrow} D_y^m}[\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} \in \mathrm{G}^m] \geq Pr_{\mathbf{c} \overset{\$}{\leftarrow} D_c}[\|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq \sqrt{n}\log n] && (6.3) \\ \cdot Pr_{\widehat{\mathbf{y}} \overset{\$}{\leftarrow} D_y^m}[\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} \in \mathrm{G}^m | \|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq \sqrt{n}\log n]. && (6.4) \end{aligned}$$

Signing Key: $\widehat{\mathbf{s}} \xleftarrow{\$} D_s^m$

Verification Key: $h \xleftarrow{\$} \mathcal{H}(R, m), \mathbf{S} \leftarrow h(\widehat{\mathbf{s}})$

Random Oracle: $\mathrm{H} : \{0,1\}^* \rightarrow D_c$

```
Sign($\mu, h, \widehat{\mathbf{s}}$)
1: $\widehat{\mathbf{y}} \xleftarrow{\$} D_y^m$
2: $\mathbf{e} \leftarrow \mathrm{H}(h(\widehat{\mathbf{y}}), \mu)$
3: $\widehat{\mathbf{z}} \leftarrow \widehat{\mathbf{s}}\mathbf{e} + \widehat{\mathbf{y}}$
4: if $\widehat{\mathbf{z}} \in \mathrm{G}^m$ then
5:     output $(\widehat{\mathbf{z}}, \mathbf{e})$
6: else
7:     goto step 1
8: end if
```

```
Verify($\mu, \widehat{\mathbf{z}}, \mathbf{e}, h, \mathbf{S}$)
1: if $\widehat{\mathbf{z}} \in \mathrm{G}^m$ and $\mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{S}\mathbf{e}, \mu)$ then
2:     ACCEPT
3: else
4:     REJECT
5: end if
```

Figure 6.2 **Signature Scheme**

By using Lemma 2.11 and the union bound, we obtain that

$$Pr_{\mathbf{c} \xleftarrow{\$} D_c}[\|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq \sqrt{n}\log n] \geq 1 - 4mne^{-\frac{\log^2 n}{8}} = 1 - n^{-\omega(1)}.$$

Then by using Lemma 6.1, we get that

$$Pr_{\widehat{\mathbf{y}} \xleftarrow{\$} D_y^m}[\widehat{\mathbf{s}}\mathbf{c} + \widehat{\mathbf{y}} \in \mathrm{G}^m | \|\widehat{\mathbf{s}}\mathbf{c}\|_\infty \leq \sqrt{n}\log n] = \frac{1}{e} - o(1).$$

Combining the above two results gives us the claim in the corollary.

□

6.B Signature Scheme

We present our signature scheme in Figure 6.2. Notice that steps 1,2 and 3 of the signing protocol are exactly the same as we would have under the normal Fiat-Shamir transform of the ID scheme in Figure 5.2. In step 4, however, we need to check whether it will be safe to send our signature of the message μ. If it is safe, then we send the signature, and otherwise we just generate another signature of μ until we get one that is safe to send. In the next lemma, we will show that all the algorithms associated with the signature scheme run in time $\tilde{O}(n)$.

Lemma 6.3. *The key-generation, signing, and verification algorithms in our signature scheme take time $\tilde{O}(n)$.*

Proof. The key-generation step of the scheme first selects an $\widehat{\mathbf{s}}$ from D_s^m, which simply involves picking $mn = 2n \log n$ random numbers from the set $\{-1, 0, 1\}$, and selecting a hash function $h \in \mathcal{H}(R, m)$, which involves picking mn random numbers from the set $\left\{-\frac{p-1}{2}, -\frac{p-1}{2} + 1, \ldots, \frac{p-1}{2}\right\}$. Then we need to compute $\mathbf{S} = h(\widehat{\mathbf{s}})$, which by Claim 2.16 takes time $\tilde{O}(n)$. The verification step involves evaluating $h(\widehat{\mathbf{z}})$, multiplying $\mathbf{Se}$, subtracting the two quantities, and making one random oracle call. Each of those operations take $\tilde{O}(n)$ time. The signing algorithm generates a $\widehat{\mathbf{y}} \in D_y^m$, computes $h(\widehat{\mathbf{y}})$, makes a random oracle call, and then computes $\widehat{\mathbf{z}} = \widehat{\mathbf{s}}\mathbf{e} + \widehat{\mathbf{y}}$. All the preceding operations take time $\tilde{O}(n)$. But unless $\widehat{\mathbf{z}} \in \mathrm{G}^m$, we will need to repeat those operations again until we do obtain a $\widehat{\mathbf{z}} \in \mathrm{G}^m$. By Corollary 6.2, we know that $\widehat{\mathbf{z}}$ will be in G^m with constant probability $1/e - o(1)$, and so with probability negligibly close to 1, we will have to repeat the operations no more than $\omega(\log n)$ times (while the expected number of repetitions is less than 3). Thus the signing algorithm also takes time $\tilde{O}(n)$. □

In the following lemma, we observe that a valid signature will always be accepted by the Verify algorithm.

Lemma 6.4. *If $(\widehat{\mathbf{z}}, \mathbf{e})$ is a signature of μ produced by running* $\mathrm{Sign}(\mu, h, \widehat{\mathbf{s}})$*, then* $\mathrm{Verify}(\mu, \widehat{\mathbf{z}}, \mathbf{e}, h, \mathbf{S})$ *always accepts.*

Proof. In order for $\mathrm{Verify}(\mu, \widehat{\mathbf{z}}, \mathbf{e}, h, \mathbf{S})$ to accept, we need $\widehat{\mathbf{z}}$ to be in G^m and $\mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{Se}, \mu)$. Since the Sign algorithm always outputs a $\widehat{\mathbf{z}} \in \mathrm{G}^m$, that part is OK. Also, the $\widehat{\mathbf{z}}$ output by the signing algorithm equals $\widehat{\mathbf{s}}\mathbf{e} + \widehat{\mathbf{y}}$, where $\mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{y}}), \mu)$. Thus $\mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e}), \mu) = \mathrm{H}(h(\widehat{\mathbf{z}}) - h(\widehat{\mathbf{s}})\mathbf{e}, \mu) = \mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{Se}, \mu)$. □

6.C Proof of Security

We now move to showing witness-indistinguishability of the scheme. By witness-indistinguishability, we mean that for any secret key $\widehat{\mathbf{s}}$, the signature of any message is statistically indistinguishable whether it's signed using $\widehat{\mathbf{s}}$ or any other key

$\widehat{\mathbf{s}}'$ such that $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{s}}')$. At first, it might seem a little surprising that the signature scheme is witness-indistinguishable because if the norm of $\widehat{\mathbf{s}}\mathbf{e}$ is larger than the norm of $\widehat{\mathbf{s}}'\mathbf{e}$, then the expected norm of $\widehat{\mathbf{s}}\mathbf{e}+\widehat{\mathbf{y}}$ will be larger than that of $\widehat{\mathbf{s}}'\mathbf{e}+\widehat{\mathbf{y}}$. And since we are sending both $\widehat{\mathbf{z}}$ and $\mathbf{e}$, it should be possible to look at the norm and figure out whether the secret key is $\widehat{\mathbf{s}}$ or $\widehat{\mathbf{s}}'$. But our scheme is witness-indistinguishable because we choose the "masking parameter" $\widehat{\mathbf{y}}$ from a large-enough set D_y^m and only send elements from another carefully-chosen set $\mathrm{G}^m \subset D_y^m$ which essentially has the effect of filtering out the values of $\widehat{\mathbf{s}}\mathbf{c}+\widehat{\mathbf{y}}$ that have large norms.

Theorem 6.5. *For any $h \in \mathcal{H}(R,m)$, message μ, and any two $\widehat{\mathbf{s}}, \widehat{\mathbf{s}}' \in D_s^m$ such that $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{s}}')$, we have*

$$\Delta((\widehat{\mathbf{z}},\mathbf{e}),(\widehat{\mathbf{z}}',\mathbf{e})) = n^{-\omega(1)}$$

where $\widehat{\mathbf{z}}$ and $\mathbf{e}$ are the random variables representing the output of $\mathrm{Sign}(\mu, h, \widehat{\mathbf{s}})$*, and $\widehat{\mathbf{z}}'$ and $\mathbf{e}'$ are the random variables representing the output of* $\mathrm{Sign}(\mu, h, \widehat{\mathbf{s}}')$.

Proof. Define the set $D_c(\widehat{\mathbf{s}}, \widehat{\mathbf{s}}')$ as

$$D_c(\widehat{\mathbf{s}}, \widehat{\mathbf{s}}') = \{\mathbf{c} \in D_c : \|\widehat{\mathbf{s}}\mathbf{c}\|_\infty, \|\widehat{\mathbf{s}}'\mathbf{c}\|_\infty \le \sqrt{n}\log n\}.$$

Then by Lemma 2.11 and the union bound, we can conclude that for any two $\widehat{\mathbf{s}}, \widehat{\mathbf{s}}' \in D_s^m$, almost all of the elements of D_c are in $D_c(\widehat{\mathbf{s}}, \widehat{\mathbf{s}}')$. More precisely, we have that

$$\frac{|D_c(\widehat{\mathbf{s}}, \widehat{\mathbf{s}}')|}{|D_c|} = 1 - n^{-\omega(1)} \tag{6.5}$$

We now rewrite $\Delta((\widehat{\mathbf{z}},\mathbf{e}),(\widehat{\mathbf{z}}',\mathbf{e}))$ as

$$\begin{aligned}
\Delta((\widehat{\mathbf{z}},\mathbf{e}),(\widehat{\mathbf{z}}',\mathbf{e})) &= \frac{1}{2}\sum_{\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta}} |Pr[(\widehat{\mathbf{z}},\mathbf{e}) = (\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})] - Pr[(\widehat{\mathbf{z}}',\mathbf{e}') = (\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})]| && (6.6)\\
&= \frac{1}{2}\sum_{\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta}\in D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')} |Pr[(\widehat{\mathbf{z}},\mathbf{e}) = (\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})] - Pr[(\widehat{\mathbf{z}}',\mathbf{e}') = (\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})]| && (6.7)\\
&+ \frac{1}{2}\sum_{\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta}\notin D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')} |Pr[(\widehat{\mathbf{z}},\mathbf{e}) = (\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})] - Pr[(\widehat{\mathbf{z}}',\mathbf{e}') = (\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})]| && (6.8)
\end{aligned}$$

We will first show that the quantity in Equation (6.8) is negligibly small. We write

$$\frac{1}{2}\sum_{\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta}\notin D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')} |Pr[(\widehat{\mathbf{z}},\mathbf{e})=(\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})]-Pr[(\widehat{\mathbf{z}}',\mathbf{e}')=(\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})]| \tag{6.9}$$

$$\leq \frac{1}{2}\sum_{\boldsymbol{\beta}\notin D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')} |Pr[\mathbf{e}=\boldsymbol{\beta}]-Pr[\mathbf{e}'=\boldsymbol{\beta}]| \tag{6.10}$$

$$\leq \sum_{\boldsymbol{\beta}\notin D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')} Pr[\mathbf{e}=\boldsymbol{\beta}] \tag{6.11}$$

$$= Pr[\mathbf{e}\notin D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')] = Pr[\mathrm{H}(h(\widehat{\mathbf{y}}),\mu)\notin D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')] = 1-\frac{|D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')|}{|D_c|} = n^{-\omega(1)} \tag{6.12}$$

where the last equality follows from Equation (6.5) and the equality before that follows from the fact that H is modeled as a random oracle and so the distribution of $\mathrm{H}(h(\widehat{\mathbf{y}}),\mu)$ is uniform over D_c.

We will now proceed with the proof that the value of (6.7) is 0. We will do this by showing that for every $\widehat{\boldsymbol{\alpha}}\in \mathrm{G}^m$ and $\boldsymbol{\beta}\in D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')$, we will have

$$Pr[(\widehat{\mathbf{z}},\mathbf{e})=(\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})] = Pr[(\widehat{\mathbf{z}}',\mathbf{e}')=(\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})]. \tag{6.13}$$

We first rewrite $Pr[(\widehat{\mathbf{z}},\mathbf{e})=(\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})]$ as

$$\begin{aligned}
Pr[(\widehat{\mathbf{z}},\mathbf{e})=(\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})] &= Pr[\widehat{\mathbf{z}}=\widehat{\boldsymbol{\alpha}}\wedge\mathbf{e}=\beta]\\
&= Pr[\mathbf{e}=\boldsymbol{\beta}\wedge\widehat{\mathbf{s}}\mathbf{e}+\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}]\\
&= Pr[\mathbf{e}=\boldsymbol{\beta}\wedge\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}]\\
&= Pr[\mathbf{e}=\boldsymbol{\beta}|\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}]Pr[\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}]\\
&= Pr[\mathrm{H}(h(\widehat{\mathbf{y}}),\mu)=\boldsymbol{\beta}|\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}]Pr[\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}]\\
&= Pr[\mathrm{H}(h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}),\mu)=\boldsymbol{\beta}]Pr[\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}]
\end{aligned}$$

and we similarly write

$$Pr[(\widehat{\mathbf{z}}',\mathbf{e}')=(\widehat{\boldsymbol{\alpha}},\boldsymbol{\beta})] = Pr[\mathrm{H}(h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}),\mu)=\boldsymbol{\beta}]Pr[\widehat{\mathbf{y}}'=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}].$$

To prove the equality in Equation (6.13), we will show that

1. $Pr[\mathrm{H}(h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}),\mu)=\boldsymbol{\beta}] = Pr[\mathrm{H}(h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}),\mu)=\boldsymbol{\beta}]$

2. $Pr[\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}] = Pr[\widehat{\mathbf{y}}'=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}]$

To show (1), we notice that

$$h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta})=h(\widehat{\boldsymbol{\alpha}})-h(\widehat{\mathbf{s}})\boldsymbol{\beta}=h(\widehat{\boldsymbol{\alpha}})-h(\widehat{\mathbf{s}}')\boldsymbol{\beta}=h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}),$$

and so $\mathrm{H}(h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}),\mu)=\mathrm{H}(h(\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}),\mu)$.

To prove (2), we make use of the fact that $\boldsymbol{\beta}\in D_c(\widehat{\mathbf{s}},\widehat{\mathbf{s}}')$, and so we have $\|\widehat{\mathbf{s}}\boldsymbol{\beta}\|_\infty,\|\widehat{\mathbf{s}}'\boldsymbol{\beta}\|_\infty\le\sqrt{n}\log n$. Since $\widehat{\boldsymbol{\alpha}}\in\mathrm{G}^m$, we have $\|\widehat{\boldsymbol{\alpha}}\|_\infty\le mn^{1.5}\log n-\sqrt{n}\log n$, and so

$$\|\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}\|_\infty,\|\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}\|_\infty\le mn^{1.5}\log n.$$

Notice that both $\widehat{\mathbf{y}}$ and $\widehat{\mathbf{y}}'$ are chosen uniformly at random from D_y^m, which contains the entire range of possibilities for the values of both $\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}$ and $\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}$. Thus

$$Pr[\widehat{\mathbf{y}}=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}\boldsymbol{\beta}]=Pr[\widehat{\mathbf{y}}'=\widehat{\boldsymbol{\alpha}}-\widehat{\mathbf{s}}'\boldsymbol{\beta}]=\frac{1}{|D_y|^m}$$

and this completes the proof of the theorem. □

In the following theorem, we will show that if there exists a forger who is able to obtain a forgery with some non-negligible probability, then there exists an algorithm that can solve the $\mathrm{Col}(h,D)$ problem with non-negligible probability. The proof uses the "forking lemma" [PS00, BN06] in order to obtain two signatures using the same random oracle query. By using these two signatures and the fact that the protocol is witness-indistinguishable, we will be able to obtain a solution to $\mathrm{Col}(h,D)$. We remind the reader that we are claiming that the scheme is strongly unforgeable, which means that a valid forgery may either consist of a signature of an unseen message, or it may consist of a different signature of a message whose signature the forger has already seen.

Theorem 6.6. *Suppose there exists a polynomial-time forger $\mathcal{F}$ who makes at most ζ queries to the signer, ψ queries to the random oracle H, and succeeds in forging with probability δ. Then for a randomly-chosen $h\in\mathcal{H}(R,m)$, there exists an algorithm of the same time-complexity that outputs a solution to $Col(h,D)$ with probability at least*

$$\left(\frac{1}{2}-n^{-\omega(1)}\right)\left(\delta-\frac{1}{|D_c|}\right)\left(\frac{\delta-1/|D_c|}{\psi+\omega(\log n)\zeta}-\frac{1}{|D_c|}\right).$$

Proof. Given an $h \in \mathcal{H}(R,m)$, we pick a secret signing key $\widehat{\mathbf{s}} \leftarrow D_s^m$, and then compute and publish the corresponding public verification keys $h, \mathbf{S} = h(\widehat{\mathbf{s}})$. Define q as the bound on the number of times the the random oracle H is called during $\mathcal{F}$'s attack. A random oracle query can be made by the forger directly, or by the signing algorithm when the forger asks to see a signature of some message. In Lemma 6.3, we observed that the signing algorithm will need to make at most $\omega(\log n)$ queries to H to produce a signature, and so the value q is bounded by $\psi + \omega(\log n)\zeta$. We then pick random coins ρ for the forger and σ for the signer, and we also pick $\mathbf{r}_1, \ldots, \mathbf{r}_q \overset{\$}{\leftarrow} D_c$, which will correspond to the responses of the random oracle. We now consider a subroutine $\mathcal{A}$, which takes as input $(h, \widehat{\mathbf{s}}, \rho, \sigma, \mathbf{r}_1, \ldots, \mathbf{r}_q)$. The subroutine $\mathcal{A}$ initializes $\mathcal{F}$ by giving it the public key $(h, \mathbf{S})$ and the random coins ρ, and then proceeds to run $\mathcal{F}$. Whenever $\mathcal{F}$ wants some message signed, $\mathcal{A}$ uses the secret key $\widehat{\mathbf{s}}$ and the signer's random coins σ to produce a valid signature. During signing, queries to H will have to be made, and the response of H will be first $\mathbf{r}_i$ in the list $(\mathbf{r}_1, \ldots, \mathbf{r}_q)$ that hasn't been used yet. Of course, $\mathcal{A}$ will have to keep a table of all the queries to H, so in case the same query is made twice, it will have to reply with the previously answered $\mathbf{r}_i$. The forger $\mathcal{F}$ can also make queries to the random oracle, in which case the reply will similarly be the first unused $\mathbf{r}_i$ in the list $(\mathbf{r}_1, \ldots, \mathbf{r}_q)$ (unless the query is not being made for the first time). Once $\mathcal{F}$ finishes running and outputs a forgery (with probability δ), our subroutine $\mathcal{A}$ simply outputs $\mathcal{F}$'s output.

With probability δ, $\mathcal{F}$ will output a message μ and its signature $(\widehat{\mathbf{z}}, \mathbf{e})$ such that $\mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{S}\mathbf{e}, \mu)$. Notice that if $\mathcal{F}$ did not query H on the input $(h(\widehat{\mathbf{z}}) - \mathbf{S}\mathbf{e}, \mu)$, then it only has a $1/|\mathrm{H}|$ chance of producing an $\mathbf{e}$ such that $\mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{S}\mathbf{e}, \mu)$. Thus with probability $1 - 1/|D_c|$, $\mathbf{e}$ must be one of the $\mathbf{r}_i$'s, and so the probability that $\mathcal{F}$ succeeds in a forgery and $\mathbf{e}$ is one of the $\mathbf{r}_i$'s, is at least $\delta - 1/|D_c|$. Let j be such that $\mathbf{e} = \mathbf{r}_j$. There are two possibilities: $\mathbf{r}_j$ was a response to a random oracle query made by $\mathcal{F}$, or it was a response to a query made during signing. We will deal with the latter case first because in this case we do not need to "fork".

Suppose that the signer made a random oracle query $\mathrm{H}(h(\widehat{\mathbf{y}}'), \mu')$ and got output $\mathbf{e} = \mathbf{r}_j$ and then computed $\widehat{\mathbf{z}}' = \widehat{\mathbf{s}}\mathbf{e} + \widehat{\mathbf{y}}'$. Then one of two things could have happened: either $\widehat{\mathbf{z}}' \in \mathrm{G}^m$ or $\widehat{\mathbf{z}}' \notin \mathrm{G}^m$. If $\widehat{\mathbf{z}}' \in \mathrm{G}^m$, then the the signer would output the signature $\widehat{\mathbf{z}}', \mathbf{e}$. In this case, if the forger outputs a valid forgery $(\mu, \widehat{\mathbf{z}}, \mathbf{e})$, then either $\mu \neq \mu'$ or $\widehat{\mathbf{z}} \neq \widehat{\mathbf{z}}'$ (or both are unequal), because otherwise the forger is just outputting a signature and a message that he has already seen. If $\mu \neq \mu'$, then we have a collision in the random oracle H. This is because

$$\mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{Se}, \mu) = \mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}}') - \mathbf{Se}, \mu'). \tag{6.14}$$

If $\mu = \mu'$, then we must have $\widehat{\mathbf{z}} \neq \widehat{\mathbf{z}}'$ and

$$\mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{Se}, \mu) = \mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}}') - \mathbf{Se}, \mu). \tag{6.15}$$

So we see that either $h(\widehat{\mathbf{z}}) - \mathbf{Se} \neq h(\widehat{\mathbf{z}}') - \mathbf{Se}$, in which case we have a collision for H, or we have that $h(\widehat{\mathbf{z}}) - \mathbf{Se} = h(\widehat{\mathbf{z}}') - \mathbf{Se}$, in which case we have $h(\widehat{\mathbf{z}}) = h(\widehat{\mathbf{z}}')$, which is a collision for h. And since $\widehat{\mathbf{z}}, \widehat{\mathbf{z}}' \in \mathrm{G}^m \subset D^m$, we have found a solution to $\mathrm{Col}(h, D)$. We now deal with the case that $\widehat{\mathbf{z}}' \notin \mathrm{G}^m$. In this case, if the forger outputs a valid forgery $(\mu, \widehat{\mathbf{z}}, \mathbf{e})$, then either $\mu \neq \mu'$ or $\widehat{\mathbf{z}} \neq \widehat{\mathbf{z}}'$ (or both are unequal), because $(\widehat{\mathbf{z}}', \mathbf{e})$ is not a valid signature of μ (since $\widehat{\mathbf{z}}' \notin \mathrm{G}^m$). If $\mu \neq \mu'$, then we have a collision in the random oracle H because of (6.14). Similarly, if $\mu = \mu'$ and $\widehat{\mathbf{z}} \neq \widehat{\mathbf{z}}'$, then by (6.15), we either have a collision in H or we have $h(\widehat{\mathbf{z}}) = h(\widehat{\mathbf{z}}')$, which is a collision for h. We also know that $\widehat{\mathbf{z}} \in \mathrm{G}^m \subset D^m$, and

$$\|\widehat{\mathbf{z}}'\|_\infty = \|\widehat{\mathbf{s}}\mathbf{r_j} + \widehat{\mathbf{y}}\|_\infty \leq \|\widehat{\mathbf{s}}\mathbf{r_j}\|_\infty + \|\widehat{\mathbf{y}}\|_\infty \leq \sqrt{n}\log n + mn^{1.5}\log n,$$

and so $\widehat{\mathbf{z}}'$ is also in D^m, and we have a solution to $\mathrm{Col}(h, D)$. Notice that after q queries to the random oracle, the probability of getting a collision in H is less than $q/|D_c|$, and so the probability that the forger $\mathcal{F}$ outputs a forgery which is a solution to $\mathrm{Col}(h, D)$ is at least $\delta - q/|D_c|$. But we will see below that the probability of obtaining a solution to $\mathrm{Col}(h, D)$ when $\mathbf{r}_j$ was a response to a random oracle query made by $\mathcal{F}$ is smaller than $\delta - q/|D_c|$, and so it will serve as the lower bound of the success probability of solving $\mathrm{Col}(h, D)$.

We now turn to the case that $\mathbf{r}_j$ was a response to a random oracle query made by $\mathcal{F}$. In this case, we first record the output $(\mu, \widehat{\mathbf{z}}, \mathbf{r}_j)$ of $\mathcal{F}$, and then generate fresh random elements $\mathbf{r}'_j, \ldots, \mathbf{r}'_q \xleftarrow{\$} D_c$. We then run the subroutine $\mathcal{A}$ again with inputs $(h, \widehat{\mathbf{s}}, \rho, \sigma, \mathbf{r}_1, \ldots, \mathbf{r}_{j-1}, \mathbf{r}'_j, \ldots, \mathbf{r}'_q)$. By Lemma 2.30, we obtain that the probability that $\mathbf{r}'_j \neq \mathbf{r}_j$ and the forger uses the random oracle response $\mathbf{r}'_j$ (and the query associated to it) in its forgery is at least

$$\left(\delta - \frac{1}{|D_c|}\right)\left(\frac{\delta - 1/|D_c|}{q} - \frac{1}{|D_c|}\right),$$

and thus with the above probability, $\mathcal{F}$ outputs a signature $(\widehat{\mathbf{z}}', \mathbf{e}')$ of the message μ where $\mathbf{e}' = \mathbf{r}'_j$ and $h(\widehat{\mathbf{z}}') - \mathbf{S}\mathbf{e}' = h(\widehat{\mathbf{z}}) - \mathbf{S}\mathbf{e}$. Because we know the secret key $\widehat{\mathbf{s}}$ such that $h(\widehat{\mathbf{s}}) = \mathbf{S}$, we can use the homomorphic properties of h to obtain the equality

$$h(\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e}) = h(\widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{e}') \tag{6.16}$$

So if $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e} \neq \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{e}'$, then we have a collision for h. This is where we use the witness-indistinguishability of the signature scheme. By Lemma 5.2, there is another possible secret key $\widehat{\mathbf{s}}' \in D_s^m$ such that $h(\widehat{\mathbf{s}}) = h(\widehat{\mathbf{s}}')$ and from Theorem 6.5, we know that it's statistically impossible to tell with probability greater than $1/2 + n^{-\omega(1)}$ which of the two secret keys are being used by the signer. Thus there is a $1/2 - n^{-\omega(1)}$ probability that the secret key was not $\widehat{\mathbf{s}}$, but rather some other $\widehat{\mathbf{s}}'$. We will now show that if $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e} = \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{e}'$, then for any other $\widehat{\mathbf{s}}'$, we will have $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}'\mathbf{e} \neq \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}'\mathbf{e}'$. For contradiction assume that

$$\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e} = \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{e}' \text{ and } \widehat{\mathbf{z}} - \widehat{\mathbf{s}}'\mathbf{e} \neq \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}'\mathbf{e}'.$$

By subtracting the two equations, we obtain that $(\widehat{\mathbf{s}} - \widehat{\mathbf{s}}')(\mathbf{e} - \mathbf{e}') = 0$. Since we are doing the multiplication in the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$, which *is not* an integral domain, we cannot automatically conclude that the preceding equality implies that either $\widehat{\mathbf{s}} - \widehat{\mathbf{s}}'$ or $\mathbf{e} - \mathbf{e}'$ is 0. But we can make the following observation: since all the coordinates of $\widehat{\mathbf{s}}, \widehat{\mathbf{s}}', \mathbf{e}$, and $\mathbf{e}'$ have absolute value at most 1, the differences $\widehat{\mathbf{s}} - \widehat{\mathbf{s}}'$ and $\mathbf{e} - \mathbf{e}'$ have coordinates whose absolute value is at most 2. When we multiply such polynomials together in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$, the absolute values of the coefficients of the product are never above $4n$. In order to be reduced modulo $p = \Theta(n^4)$, the absolute value of the coefficients would have to be greater than $p/2$, but $4n$ is much smaller than that. Therefore if the product $(\widehat{\mathbf{s}} - \widehat{\mathbf{s}}')(\mathbf{e} - \mathbf{e}')$ is 0 in the ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$, it also must be $\mathbf{0}$ in the ring $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$. But because $\mathbf{x}^n + 1$ is irreducible over the integers, $\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ *is* an integral domain, and therefore either $\widehat{\mathbf{s}} - \widehat{\mathbf{s}}'$ or $\mathbf{e} - \mathbf{e}'$ is 0. Since we know that $\mathbf{e} \neq \mathbf{e}'$, it must be that $\widehat{\mathbf{s}} = \widehat{\mathbf{s}}'$, and we have a contradiction. Thus $\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e} \neq \widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{e}'$ with probability at least $1/2 - n^{-\omega(1)}$ and we have a collision for h. All that's left to check is that $\|\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e}\|_\infty$ and $\|\widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{e}'\|_\infty$ are in the set D^m. Since all the random oracle replies $\mathbf{r}_i$ are are uniformly random in D_c, we can use Lemma 2.11 and the union bound to conclude that with probability $1 - n^{-\omega(1)}$ the norm $\|\widehat{\mathbf{s}}\mathbf{r_i}\|_\infty$ for all $\mathbf{r}_i$ is at most $\sqrt{n}\log n$. And since $\widehat{\mathbf{z}}, \widehat{\mathbf{z}}' \in \mathrm{G}^m$, we have $\|\widehat{\mathbf{z}} - \widehat{\mathbf{s}}\mathbf{e}\|_\infty, \|\widehat{\mathbf{z}}' - \widehat{\mathbf{s}}\mathbf{e}'\|_\infty \leq mn^{1.5}\log n$. This concludes the proof of the theorem.

□

Corollary 6.7. *If the signature scheme in Figure 6.2 for the parameters in Table 6.1 is not strongly unforgeable, then there is a polynomial-time algorithm that can solve* $\mathbf{f}$*-SVP*$_\gamma(\Lambda)$ *for* $\gamma = \tilde{O}(n^{2.5})$ *for every lattice* Λ *corresponding to an ideal in the ring* $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$.

Proof. By Theorem 6.6, we know that breaking the signature scheme implies solving the $\mathrm{Col}(h, D)$ problem where h is chosen at random from $\mathcal{H}(R, D, m)$ for the parameters in Table 6.1. It's straightforward to check that the parameters satisfy Theorem 3.1, and therefore solving $\mathrm{Col}(h, D)$ for random $h \in \mathcal{H}(R, D, m)$ means that we can solve $\mathbf{f}$-$\mathrm{SVP}_\gamma(\Lambda)$ for $\gamma = 16\theta(\mathbf{f})^2(mn^{1.5}\log n + \sqrt{n}\log n)n\log^2 n = \tilde{O}(n^{2.5})$ for every lattice Λ corresponding to an ideal in $\mathbb{Z}[\mathbf{x}]/\langle\mathbf{f}\rangle$. □

Appendix A

New Bounds on Gaussian Distributions Over Lattices

In this appendix, we will provide a proof of Lemma 2.25. In all that follows, let ρ be defined the same way as in subsection 2.J, and let $\widehat{\rho}$ be the fourier transform of ρ. That is, for vectors $\mathbf{x}$ and $\mathbf{y}$, $\widehat{\rho}(\mathbf{y}) = \int\limits_{-\infty}^{\infty} \rho(\mathbf{x}) e^{-2\pi i \langle x,y \rangle} d\mathbf{x}$. Next, we state some general properties of the fourier transform. If h is defined by $h(\mathbf{x}) = g(\mathbf{x}+\mathbf{v})$ for some function g and vector $\mathbf{v}$ then

$$\hat{h}(\mathbf{w}) = e^{2\pi i \langle \mathbf{v},\mathbf{w} \rangle} \hat{g}(\mathbf{w}). \tag{A.1}$$

Another important fact is that the Gaussian is its own Fourier transform, i.e., $\hat{\rho} = \rho$. More generally, for any $s > 0$ it holds that $\widehat{\rho_s} = s^n \rho_{1/s}$. We use the following formulation of the Poisson summation formula.

Lemma A.1. *For any lattice Λ and any*[1] *function $f : \mathbb{R}^n \to \mathbb{C}$, $f(\Lambda) = \det(\Lambda^*) \hat{f}(\Lambda^*)$ where $\hat{f}$ denotes the Fourier transform of f.*

The below proposition is just the value of the m^{th} moment of a standard normal gaussian. We do not provide a proof for it, although it is easily proved by integrating by parts.

[1] *For this formula to hold, f needs to satisfy certain niceness assumptions. These assumptions always hold in our applications. See [Ebe02] for more details.*

Proposition A.2.

$$\int_{-\infty}^{\infty} x^m e^{-\pi x^2} dx = \begin{cases} \frac{m!}{(m/2)!(4\pi)^{m/2}} & \textit{if } m \textit{ is even,} \\ 0 & \textit{if } m \textit{ is odd.} \end{cases}$$

In the next lemma, we state the closed form of the fourier transform of the m^{th} moment of the standard normal gaussian.

Lemma A.3. *For all values of y and integers $m \geq 0$, we have*

$$\int_{-\infty}^{\infty} x^m e^{-\pi x^2} e^{-2\pi i x y} dx = \left((-i)^m m! \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \frac{(-1)^j y^{m-2j}}{j!(m-2j)!(4\pi)^j} \right) \widehat{\rho}(y)$$

(Note that when $y = 0$ and m is even, the term 0^0 will appear in the sum. But since when $y = 0$ proposition A.2 applies, in order to make this lemma include proposition A.2, we'll assume that $0^0 = 1$.)

Proof. The proof is by induction. We will need to establish base cases for $m = 0$ and $m = 1$. For $m = 0$, the equality clearly holds. For $m = 1$, we need to show that

$$\int_{-\infty}^{\infty} x e^{-\pi x^2} e^{-2\pi i x y} dx = -iy\widehat{\rho}(y) \tag{A.2}$$

It's not difficult to show the above by integrating by parts.

Now we assume that the lemma is true for all values of y and all $k < m + 2$. We will prove that

$$\int_{-\infty}^{\infty} x^{k+2} e^{-\pi x^2} e^{-2\pi i x y} dx = \left((-i)^{k+2} (k+2)! \sum_{j=0}^{\lfloor \frac{k+2}{2} \rfloor} \frac{(-1)^j y^{k+2-2j}}{j!(k+2-2j)!(4\pi)^j} \right) \widehat{\rho}(y) \tag{A.3}$$

Integrating the the above by parts and using the induction hypothesis, we get

$$\int_{-\infty}^{\infty} x^{k+2} e^{-\pi x^2} e^{-2\pi ixy} dx \tag{A.4}$$

$$= \frac{k+1}{2\pi} \int_{-\infty}^{\infty} x^{k} e^{-\pi x^2} e^{-2\pi ixy} dx - iy \int_{-\infty}^{\infty} x^{k+1} e^{-\pi x^2} e^{-2\pi ixy} dx \tag{A.5}$$

$$= \frac{k+1}{2\pi} \left((-i)^k k! \sum_{j=0}^{\lfloor \frac{k}{2} \rfloor} \frac{(-1)^j y^{k-2j}}{j!(k-2j)!(4\pi)^j} \right) \widehat{\rho}(y) \tag{A.6}$$

$$- iy \left((-i)^{k+1} (k+1)! \sum_{j=0}^{\lfloor \frac{k+1}{2} \rfloor} \frac{(-1)^j y^{k+1-2j}}{j!(k+1-2j)!(4\pi)^j} \right) \widehat{\rho}(y) \tag{A.7}$$

$$= (-i)^{k+2}(k+2)! \tag{A.8}$$

$$\left(\frac{-1}{2\pi(k+2)} \sum_{j=0}^{\lfloor \frac{k}{2} \rfloor} \frac{(-1)^j y^{k-2j}}{j!(k-2j)!(4\pi)^j} + \frac{1}{k+2} \sum_{j=0}^{\lfloor \frac{k+1}{2} \rfloor} \frac{(-1)^j y^{k+2-2j}}{j!(k+1-2j)!(4\pi)^j} \right) \widehat{\rho}(y) \tag{A.9}$$

We will show that equation (A.9) is equivalent to the right side of equation (A.3) by showing that the coefficients of like powers of y are equivalent. The $(-i)^{k+2}(k+2)!\widehat{\rho}(y)$ part is the same in both equations, so we'll be ignoring it. Notice that to get the coefficient of the term y^{k+2-2l}, we need to look at the coefficient of the term we get for $j = l - 1$ in the first sum of equation (A.9) and for $j = l$ in the second sum. Some special cases occur when $l = 0$ or $l = \lfloor \frac{k+2}{2} \rfloor$ (then $j = l - 1$ and $j = l$ may not exist as terms in both sums) but let's first handle the general case first (i.e. the coefficient of y^{k+2-2l} comes from both terms of equation (A.9)). We need to show that

$$\frac{-1}{2\pi(k+2)} \cdot \frac{(-1)^{l-1} y^{k-2(l-1)}}{(k-2(l-1))!(l-1)!(4\pi)^{l-1}} + \frac{1}{k+2} \cdot \frac{(-1)^l y^{k+2-2l}}{(k+1-2l)!l!(4\pi)^l} \tag{A.10}$$

$$= \frac{(-1)^l y^{k-2l+2}}{(k+2-2l)!l!(4\pi)^l} \tag{A.11}$$

The above equality is not too hard to show with a little algebra manipulation. Now we come to the special cases. If $l = 0$, then the coefficient of y^{k+2-2l} comes entirely from the second sum of equation (A.9). Plugging in, we get

$$\frac{1}{k+2} \cdot \frac{(-1)^0 y^{k+2-2\cdot 0}}{0!(k+1-2\cdot 0)!(4\pi)^0} = \frac{y^{k+2}}{(k+2)!}$$

and thus the coefficients of the y^{k+2} term are the same in equations (A.9) and (A.3). Now we consider the case when $l = \lfloor \frac{k+2}{2} \rfloor$. Here, two subcases arise. The simple one is if k is odd. In this subcase, $\lfloor\frac{k+2}{2}\rfloor = \lfloor\frac{k+1}{2}\rfloor$, and thus the coefficient of y^{k+2-2l} comes from both sums of equation (A.9) and this case has been already handled by equation (A.10). In the other subcase, $\lfloor\frac{k+2}{2}\rfloor \neq \lfloor\frac{k+1}{2}\rfloor$, and so k must be even, and thus $l = \frac{k}{2}+1$. In this subcase, the coefficient of $y^{k+2-2l} = y^0$ comes from only the first sum of equation (A.9). That coefficient is what we get when $j = \frac{k}{2}$, and it's

$$\frac{-1}{2\pi(k+2)} \cdot \frac{(-1)^{\frac{k}{2}}}{(\frac{k}{2})!(4\pi)^{\frac{k}{2}}} = \frac{(-1)^{\frac{k}{2}+1}}{4\pi(\frac{k}{2}+1)(\frac{k}{2})!(4\pi)^{\frac{k}{2}}} = \frac{(-1)^{\frac{k}{2}+1}}{(\frac{k}{2}+1)!(4\pi)^{\frac{k}{2}+1}}$$

which is exactly the term in equation (A.3) when $j = \frac{k}{2}+1$. □

In the next two lemmas, we define the function $g_m(\mathbf{x}) = (x_1 - c_1)^m \rho_{\mathbf{c}}(\mathbf{x})$ (where x_1 and c_1 are the first coordinates of $\mathbf{x}$ and $\mathbf{c}$ respectively) and will bound the absolute value of its fourier transform. The reason for doing this will become clear in Lemma A.6

Lemma A.4. *If* $g_m(\boldsymbol{x}) = (x_1 - c_1)^m \rho_{\boldsymbol{c}}(\boldsymbol{x})$, *then*

$$\widehat{g_m}(\boldsymbol{y}) = \left((-i)^m m! \sum_{j=0}^{\lfloor\frac{m}{2}\rfloor} \frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j}\right) \widehat{\rho_{\boldsymbol{c}}}(\boldsymbol{y})$$

(The same caveat applies here as in Lemma A.3, i.e. if $y_1 = 0$ *and* m *is even, then* 0^0 *will appear in the sum. And again for notational convenience, let* $0^0 = 1$ *in this case.)*

Proof. Define the function

$$f_m(\mathbf{x}) = g_m(\mathbf{x}+\mathbf{c}) = x_1^m \rho_{\mathbf{c}}(\mathbf{x}+\mathbf{c}) = x_1^m \rho(\mathbf{x})$$

This means that the fourier transform of $g_m(\mathbf{x})$ is

$$\widehat{g_m}(\mathbf{y}) = \widehat{f_m}(\mathbf{y}) e^{-2\pi i\langle\mathbf{c},\mathbf{y}\rangle} \tag{A.12}$$

Define $\mathbf{x}'$ to be the vector $\mathbf{x}$ with the first coordinate removed, and similarly let $\mathbf{y}'$ be the vector $\mathbf{y}$ with the first coordinate removed So,

$$f_m(\mathbf{x}) = x_1^m \rho(\mathbf{x}) = x_1^m \rho(x_1)\rho(\mathbf{x}') \tag{A.13}$$

and

$$\widehat{f_m}(\mathbf{y}) = \left(\int_{-\infty}^{\infty} x_1^m e^{-\pi x_1^2} e^{-2\pi i x_1 y_1} dx_1\right) \widehat{\rho}(\mathbf{y}') \tag{A.14}$$

$$= \left((-i)^m m! \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j}\right) \widehat{\rho}(y_1)\widehat{\rho}(\mathbf{y}') \tag{A.15}$$

$$= \left((-i)^m m! \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j}\right) \widehat{\rho}(\mathbf{y}) \tag{A.16}$$

where the second equality follows from Lemma A.3. And since

$$\widehat{\rho_c}(\mathbf{y}) = \widehat{\rho}(\mathbf{y}) e^{-2\pi i \langle \mathbf{c}, \mathbf{y} \rangle}$$

we combine equations (A.12) and (A.16) to obtain the claim in the lemma. □

Lemma A.5.

$$|\widehat{g_m}(\boldsymbol{y})| \le \begin{cases} \frac{m!}{(m/2)!(4\pi)^{m/2}} & \textit{if } m \textit{ is even and } \boldsymbol{y} = \boldsymbol{0}, \\ 0 & \textit{if } m \textit{ is odd and } \boldsymbol{y} = \boldsymbol{0}, \\ m^{2m}\rho_2(\boldsymbol{y}) & \textit{in all other cases.} \end{cases}$$

Proof. Since $|\widehat{\rho_{\mathbf{c}}}(\mathbf{y})| = \rho(\mathbf{y})$, we have by Lemma A.4,

$$|\widehat{g_m}(\mathbf{y})| = \left|(-i)^m m! \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j}\right| \rho(\mathbf{y}) \tag{A.17}$$

Now we will quickly dispatch of the case where $\mathbf{y} = \mathbf{0}$. In this case $\rho(\mathbf{y}) = 1$ and all the terms in the sum in equation (A.17) will cancel out except possibly $y_1^{m-2\lfloor \frac{m}{2} \rfloor}$ (because remember that we assumed that $0^0 = 1$). If m is odd, then the exponent will not be 0, thus the sum will be 0, and if m is even, then the exponent will be 0. Thus, the sum will have the value of the term when $j = \frac{m}{2}$, which is what is claimed in the lemma. Now we will handle an easy subcase of the "all other cases." The subcase is when $\mathbf{y} \neq \mathbf{0}$ but $y_1 = 0$. In this subcase, the sum in equation (A.17) is equal to 0 when m is odd and is equal to $\frac{m!}{(m/2)!(4\pi)^{m/2}}$ when m is even. Either way, the product of this sum with $\rho(\mathbf{y})$ is less than $m^{2m}\rho_2(\mathbf{y})$. Now we will handle all the remaining cases (i.e. when $y_1 \neq 0$).

$$|\widehat{g_m}(\mathbf{y})| = \left|(-i)^m m! \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j}\right| \rho(\mathbf{y}) \tag{A.18}$$

$$\leq m! \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \left| \frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j} \right| \rho(\mathbf{y}) \tag{A.19}$$

Note that if $|y_1| \leq 1$, then $\left|\frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j}\right| \leq 1$ and thus equation (A.19) is at most $(\lfloor \frac{m}{2} \rfloor + 1) m! \rho(\mathbf{y})$ which is less than $m^{2m}\rho_2(\mathbf{y})$. So let's now assume that $|y_1| \geq 1$. Then we have

$$\begin{aligned}
|\widehat{g_m}(\mathbf{y})| &\leq m! \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \left| \frac{(-1)^j y_1^{m-2j}}{j!(m-2j)!(4\pi)^j} \right| \rho(\mathbf{y}) \\
&\leq \left(\frac{m}{2} + 1\right) m! y_1^m \rho(\mathbf{y}) \\
&= \left(\frac{m}{2} + 1\right) m! m^{2m/3} \frac{y_1^m}{m^{2m/3}} \rho(y_1)\rho(\mathbf{y}') \\
&\leq m^{2m} \frac{y_1^m}{m^{2m/3}} \rho(y_1)\rho_2(\mathbf{y}')
\end{aligned}$$

where we recall that $\mathbf{y}'$ is defined as the vector $\mathbf{y}$ with the first component removed. So all that is left to complete the proof of the lemma is to show that

$$\frac{y_1^m}{m^{2m/3}} \rho(y_1) \leq \rho_2(y_1) \tag{A.20}$$

Consider the case where $y_1 \leq m^{2/3}$. Then equation (A.20) is clearly true. In the case where $y_1 > m^{2/3}$, we need to show that

$$y_1^m e^{-\pi y_1^2} \leq e^{-\pi(\frac{y_1}{2})^2}$$

or equivalently that

$$m \log y_1 \leq \frac{3}{4}\pi y_1^2$$

Since $y_1 > m^{2/3}$, we have

$$\frac{3}{4}\pi y_1^2 = \frac{3}{4}\pi y_1^{\frac{1}{2}} y_1^{\frac{3}{2}} > \frac{3}{4}\pi y_1^{\frac{1}{2}} m > m \log y_1$$

This proves equation (A.20) and thus the lemma. □

The next lemma is a generalization and closely follows the outline of Lemma 4.2 of [MR07]. The main difference is the technique for bounding the function $\widehat{g_m}$, which was done in Lemmas A.4 and A.5.

Lemma A.6. *For any n-dimensional lattice Λ, point $\boldsymbol{c} \in \mathbb{R}^n$, unit vector $\boldsymbol{u}$, positive real $s > 2\eta_\epsilon(\Lambda)$, and all positive integers m,*

$$\left|Exp_{x \sim D_{\Lambda,s,\mathbf{c}}}\left[\langle \boldsymbol{x} - \boldsymbol{c}, \boldsymbol{u}\rangle^m\right]\right| \leq \begin{cases} s^m \left(\frac{\frac{m!}{(m/2)!(4\pi)^{m/2}} + m^{2m}\epsilon}{1-\epsilon}\right) & \text{if } m \text{ is even} \\ s^m \left(\frac{m^{2m}\epsilon}{1-\epsilon}\right) & \text{if } m \text{ is odd} \end{cases}$$

Proof. For any positive real $s > 0$, define $\Lambda' = \Lambda/s$, $\mathbf{c}' = \mathbf{c}/s$. Notice that, for any $\mathbf{x}$,

$$\Pr\{D_{\Lambda,s,\mathbf{c}} = s\mathbf{x}\} = \frac{\rho_{s,\mathbf{c}}(s\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)} = \frac{\rho_{\mathbf{c}'}(\mathbf{x})}{\rho_{\mathbf{c}'}(\Lambda')} = \Pr\{D_{\Lambda',\mathbf{c}'} = \mathbf{x}\},$$

i.e., the distribution $D_{\Lambda,s,\mathbf{c}}$ is equal to $D_{\Lambda',\mathbf{c}'}$ scaled by a factor of s. Therefore, it is enough to prove the lemma for $s = 1$. The general case follows by scaling the lattice by a factor s.

In the rest of the proof, we assume $s = 1$. We want to estimate the quantity $\text{Exp}_{\mathbf{x} \sim D_{\Lambda,\mathbf{c}}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u}\rangle^m]$. Without loss of generality, assume that $\mathbf{u}$ is the vector $(1, 0, \ldots, 0)$ We will show the lemma true for $s = 1$ and the general case will follow by scaling the lattice by a factor s.

Notice that

$$\underset{\mathbf{x} \sim D_{\Lambda,\mathbf{c}}}{\text{Exp}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u}\rangle^m] = \underset{\mathbf{x} \sim D_{\Lambda,\mathbf{c}}}{\text{Exp}}[(x_1 - c_1)^m] = \frac{g_j(\Lambda)}{\rho_{\mathbf{c}}(\Lambda)}.$$

Applying Poisson's summation formula (Lemma A.1) to the numerator and denominator, the above fraction can be rewritten as

$$\underset{\mathbf{x} \sim D_{\Lambda,\mathbf{c}}}{\text{Exp}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u}\rangle^m] = \frac{\det(\Lambda^*) \cdot \widehat{g_m}(\Lambda^*)}{\det(\Lambda^*) \cdot \widehat{\rho_{\mathbf{c}}}(\Lambda^*)} = \frac{\widehat{g_m}(\Lambda^*)}{\widehat{\rho_{\mathbf{c}}}(\Lambda^*)}. \tag{A.21}$$

The Fourier transform $\widehat{\rho_{\mathbf{c}}}$ is easily computed using Equation A.1: $\widehat{\rho_{\mathbf{c}}}(\mathbf{y}) = \rho(\mathbf{y})e^{-2\pi i \langle \mathbf{y}, \mathbf{c}\rangle}$. In particular, $\widehat{\rho_{\mathbf{c}}}(\mathbf{0}) = 1$, $|\widehat{\rho_{\mathbf{c}}}(\mathbf{y})| = \rho(\mathbf{y})$, and

$$|\widehat{\rho_{\mathbf{c}}}(\Lambda^*)| = \left|1 + \sum_{\mathbf{y} \in \Lambda^* \setminus \{\mathbf{0}\}} \widehat{\rho_{\mathbf{c}}}(\mathbf{y})\right| \geq 1 - \rho(\Lambda^* \setminus \{\mathbf{0}\}). \tag{A.22}$$

Thus, we get the equation

$$Exp_{x \sim D_{\Lambda,s,\mathbf{c}}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u}\rangle^m] = \frac{\widehat{g_m}(\Lambda^*)}{\widehat{\rho_{\mathbf{c}}}(\Lambda^*)} \leq \frac{\widehat{g_m}(\Lambda^*)}{1 - \epsilon} = \frac{\sum_{\mathbf{y} \in \Lambda^*} \widehat{g_m}(\mathbf{y})}{1 - \epsilon} \tag{A.23}$$

$$= \frac{\widehat{g_m}(\mathbf{0}) + \sum_{\mathbf{y} \in \Lambda^* \setminus \{\mathbf{0}\}} \widehat{g_m}(\mathbf{y})}{1 - \epsilon} \tag{A.24}$$

Now we apply Lemma A.5 to get

$$\begin{aligned}\left|Exp_{x\sim D_{\Lambda,s,\mathbf{c}}}\left[\langle \mathbf{x}-\mathbf{c},\mathbf{u}\rangle^m\right]\right| &\leq \frac{|\widehat{g_m}(\mathbf{0})| + \sum\limits_{\mathbf{y}\in\Lambda^*\setminus\{\mathbf{0}\}} m^{2m}\rho_2(\mathbf{y})}{1-\epsilon} \\ &= \frac{|\widehat{g_m}(\mathbf{0})| + m^{2m}\rho_2(\Lambda^*\setminus\{\mathbf{0}\})}{1-\epsilon}\end{aligned}$$

which gives us the claim in the lemma. □

Proof of Lemma 2.25

Proof. For simplicity, assume that $\lfloor \log n \rfloor$ is an even integer. Then by Lemma A.6 we have

$$\left|Exp_{x\sim D_{\Lambda,s,\mathbf{c}}}\left[\langle \mathbf{x}-\mathbf{c},\mathbf{u}\rangle^{\lfloor \log n\rfloor}\right]\right| \leq s^{\lfloor \log n\rfloor}\left(\frac{\frac{(\log n)!}{((\log n)/2)!(4\pi)^{(\log n)/2}} + (\log n)^{2\log n}\epsilon}{1-\epsilon}\right) \tag{A.25}$$

$$\leq 2s^{\lfloor \log n\rfloor}(\log n)^{\frac{\log n}{2}} \tag{A.26}$$

Using the above equation, we obtain

$$\begin{aligned}Pr_{x\sim D_{\Lambda,s,\mathbf{c}}}[|\langle \mathbf{x}-\mathbf{c},\mathbf{u}\rangle| \geq s\log n] &= Pr_{x\sim D_{\Lambda,s,\mathbf{c}}}\left[\langle \mathbf{x}-\mathbf{c},\mathbf{u}\rangle^{\lfloor \log n\rfloor} \geq (s\log n)^{\lfloor \log n\rfloor}\right] \\ &\leq \frac{\left|Exp_{x\sim D_{\Lambda,s,\mathbf{c}}}\left[\langle \mathbf{x}-\mathbf{c},\mathbf{u}\rangle^{\lfloor \log n\rfloor}\right]\right|}{(s\log n)^{\lfloor \log n\rfloor}} \\ &\leq \frac{2s^{\lfloor \log n\rfloor}(\log n)^{\frac{\log n}{2}}}{(s\log n)^{\lfloor \log n\rfloor}} \\ &\leq n^{\frac{-\log\log n}{3}} = n^{-\omega(1)}\end{aligned}$$

where the first inequality follows by Markov's inequality. □

Appendix A is, in part, a reprint, of the paper "Generalized Compact Knapsacks Are Collision Resistant" co-authored with Daniele Micciancio and appearing in the proceedings of ICALP 2006. The dissertation author was the primary investigator and author of this paper.

Bibliography

[AABN02] M. Abdalla, J.H. An, M. Bellare, and C. Namprempre, *From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security*, EUROCRYPT, 2002, pp. 418–433.

[AD97] M. Ajtai and C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence*, STOC, 1997, pp. 284–293.

[Ajt96] M. Ajtai, *Generating hard instances of lattice problems*, STOC, 1996, pp. 99–108.

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar, *A sieve algorithm for the shortest lattice vector problem*, STOC, 2001, pp. 601–610.

[AR05] D. Aharonov and O. Regev, *Lattice problems in NP $\cap$ coNP*, Journal of the ACM **52** (2005), no. 5, 749–765.

[BC92] J. Bos and D. Chaum, *Provably unforgeable signatures*, CRYPTO, 1992, pp. 1–14.

[BCJ$^+$05] E. Biham, R. Chen, A. Joux, P. Carribault, W. Jalby, and C. Lemuet, *Collisions of SHA-0 and reduced SHA-1*, EUROCRYPT, 2005.

[BM84] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudo-random bits*, SIAM J. Comput. **13** (1984), no. 4, 850–864.

[BM96a] D. Bleichenbacher and U. Maurer, *On the efficiency of one-time digital signatures*, ASIACRYPT, 1996, pp. 145–158.

[BM96b] ______, *Optimal tree-based one-time digital signature schemes*, STACS, 1996, pp. 363–374.

[BMG07] B. Barak and M. Mahmoody-Ghidary, *Lower bounds on signatures from symmetric primitives*, FOCS, 2007, pp. 680–688.

[BN06] M. Bellare and G. Neven, *Multi-signatures in the plain public-key model and a general forking lemma*, ACM Conference on Computer and Communications Security, 2006, pp. 390–399.

[BN07] J. Blömer and S. Naewe, *Sampling methods for shortest vectors, closest vectors and successive minima*, ICALP, 2007, pp. 65–77.

[BR93] M. Bellare and P. Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, ACM Conference on Computer and Communications Security, 1993, pp. 62–73.

[BS07] M. Bellare and S. Shoup, *Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles*, Public Key Cryptography, 2007, pp. 201–216.

[CGH04] R. Canetti, O. Goldreich, and S. Halevi, *The random oracle methodology, revisited*, J. ACM **51** (2004), no. 4, 557–594.

[Coh96] H. Cohen, *A course in computational algebraic number theory*, Springer, 1996.

[CR88] B. Chor and R. L. Rivest, *A knapsack type public-key cryptosystem based on arithmetic in finite fields*, IEEE Trans. Inform. Theory **34** (1988), no. 5, 901–909.

[Dam] I. Damgard, *A design principle for hash functions*, CRYPTO '89, pp. 416–427.

[DH76] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **IT-22** (1976), no. 6, 644–654.

[Din02] I. Dinur, *Approximating SVP_∞ to within almost-polynomial factors is NP-hard.*, Theor. Comput. Sci. **285** (2002), no. 1, 55–71.

[Ebe02] W. Ebeling, *Lattices and codes*, Friedr. Vieweg & Sohn., 2002.

[EGM96] S. Even, O. Goldreich, and S. Micali, *On-line/off-line digital signatures*, J. Cryptology **9** (1996), no. 1, 35–67.

[FFS88] U. Feige, A. Fiat, and A. Shamir, *Zero-knowledge proofs of identity.*, J. Cryptology **1** (1988), no. 2, 77–94.

[FS86] A. Fiat and A. Shamir, *How to prove yourself: Practical solutions to identification and signature problems.*, CRYPTO, 1986, pp. 186–194.

[FS90] U. Feige and A. Shamir, *Witness indistinguishable and witness hiding protocols*, STOC, 1990, pp. 416–426.

[GG00] O. Goldreich and S. Goldwasser, *On the limits of nonapproximability of lattice problems*, J. Comput. Syst. Sci. **60** (2000), no. 3.

[GGH96] O. Goldreich, S. Goldwasser, and S. Halevi, *Collision-free hashing from lattice problems*, Electronic Colloquium on Computational Complexity (ECCC) **3** (1996), no. 42.

[GGH97] ______, *Public-key cryptosystems from lattice reduction problems*, CRYPTO, 1997, pp. 112–131.

[GMR88] S. Goldwasser, S. Micali, and R. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM J. Comput. **17** (1988), no. 2, 281–308.

[GN08a] N. Gama and P. Q. Nguyen, *Finding shortest vectors with Mordell's inequaity*, STOC, 2008.

[GN08b] ______, *Predicting lattice reduction*, EUROCRYPT, 2008.

[GPS06] M. Girault, G. Poupard, and J. Stern, *On the fly authentication and signature schemes based on groups of unknown order*, J. Cryptology **19** (2006), no. 4, 463–487.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan, *Trapdoors for hard lattices, and new cryptographic constructions*, STOC, 2008, (To appear.).

[GQ88] L. Guillou and J.J. Quisquater, *A "paradoxical" indentity-based signature scheme resulting from zero-knowledge.*, CRYPTO, 1988, pp. 216–231.

[HHGP$^+$03] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte, *Ntrusign: Digital signatures using the NTRU lattice*, CT-RSA, 2003, pp. 122–140.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman, *NTRU: A ring-based public key cryptosystem.*, ANTS, 1998, pp. 267–288.

[JG94] A. Joux and L. Granboulan, *A practical attack against knapsack based hash functions*, EUROCRYPT'94, 1994, pp. 58–66.

[LLL82] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovasz, *Factoring polynomials with rational coefficients*, Mathematische Annalen (1982), no. 261, 513–534.

[LM06] V. Lyubashevsky and D. Micciancio, *Generalized compact knapsacks are collision resistant.*, ICALP (2), 2006, pp. 144–155.

[LM08] ______, *Asymptotically efficient lattice-based digital signatures*, TCC, 2008, pp. 37–54.

[LMPR08] V. Lyubashevsky, D. Micciancio, C. Peikert, and R.Rosen, *SWIFFT: a modest proposal for FFT hashing.*, FSE, 2008.

[Lyu08] V. Lyubashevsky, *Lattice-based identification schemes secure under active attacks*, Public Key Cryptography, 2008, pp. 162–179.

[Mer87] R. Merkle, *A digital signature based on a conventional encryption function*, CRYPTO, 1987, pp. 369–378.

[Mer89] ______, *A certified digital signature*, CRYPTO, 1989, pp. 218–238.

[MG02] D. Micciancio and S. Goldwasser, *Complexity of lattice problems: A cryptographic perspective*, Kluwer Academic Publishers, 2002.

[MH78] R.C. Merkle and M.E. Hellman, *Hiding information and signatures in trapdoor knapsacks*, IEEE Transactions on Information Theory **IT-24** (1978), 525–530.

[Mic07] D. Micciancio, *Generalized compact knapsacks, cyclic lattices, and efficient one-way functions*, Computational Complexity **16** (2007), no. 4, 365–411, (Preliminary version in FOCS 2002).

[MR07] D. Micciancio and O. Regev, *Worst-case to average-case reductions based on Gaussian measures*, SIAM J. on Computing **37** (2007), no. 1, 267–302.

[MV03] D. Micciancio and S. Vadhan, *Statistical zero-knowledge proofs with efficient provers: Lattice problems and more*, CRYPTO, 2003, pp. 282–298.

[NR06] P.Q. Nguyen and O. Regev, *Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures*, EUROCRYPT, 2006, pp. 271–288.

[NY89] M. Naor and M. Yung, *Universal one-way hash functions and their cryptographic applications*, STOC, 1989, pp. 33–43.

[Oka92a] T. Okamoto, *Provably secure and practical identification schemes and corresponding signature schemes.*, CRYPTO, 1992, pp. 31–53.

[Oka92b] ______, *Provably secure and practical identification schemes and corresponding signature schemes*, CRYPTO, 1992, pp. 31–53.

[Pei] C. Peikert, Private Communication.

[Pei07] ______, *Limits on the hardness of lattice problems in ℓ_p norms*, IEEE Conference on Computational Complexity, 2007, pp. 333–346.

[Poi00] D. Pointcheval, *The composite discrete logarithm and secure authentication*, Public Key Cryptography, 2000, pp. 113–128.

[PR06] C. Peikert and A. Rosen, *Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices*, TCC, 2006.

[PR07] ______, *Lattices that admit logarithmic worst-case to average-case connection factors*, STOC, 2007.

[PS00] D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, J. Cryptology **13** (2000), no. 3, 361–396.

[PW08] C. Peikert and B. Waters, *Lossy trapdoor functions and their applications*, STOC, 2008.

[Reg03] O. Regev, *New lattice based cryptographic constructions*, STOC, 2003, pp. 407–416.

[Reg05] ______, *On lattices, learning with errors, random linear codes, and cryptography*, STOC, 2005.

[Rom90] J. Rompel, *One-way functions are necessary and sufficient for secure signatures*, STOC, 1990, pp. 387–394.

[Sch87] C. P. Schnorr, *A hierarchy of polynomial time basis reduction algorithms*, Theoretical Computer Science **53** (1987), 201–224.

[Sch91] C.P. Schnorr, *Efficient signature generation by smart cards.*, J. Cryptology **4** (1991), no. 3, 161–174.

[Sha84] A. Shamir, *A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem*, IEEE Transactions on Information Theory **IT-30** (1984), no. 5, 699–704.

[Sha89] A. Shamir, *An efficient identification scheme based on permuted kernels (extended abstract)*, CRYPTO, 1989, pp. 606–609.

[Sho97] P. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. **26** (1997), no. 5, 1484–1509.

[Sho99] V. Shoup, *On the security of a practical identification scheme.*, J. Cryptology **12** (1999), no. 4, 247–260.

[Ste96] Stern, *A new paradigm for public key identification*, IEEE Transactions on Information Theory **42** (1996).

[Szy04] M. Szydlo, *Merkle tree traversal in log space and time*, EUROCRYPT, 2004, pp. 541–554.

[van81] P. van Emde Boas, *Another NP-complete problem and the complexity of computing short vectors in a lattice.*, Tech. Report Technical Report 81-04, University of Amsterdam, http://turing.wins.uva.nl/ peter/, 1981.

[Vau01] S. Vaudenay, *Cryptanalysis of the Chor–Rivest cryptosystem*, Journal of Cryptology **14** (2001), no. 2, 87–100.

[WLF$^+$05] X. Wang, X. Lai, D. Feng, H. Chen, and X. Yu, *Cryptanalysis for hash functions MD4 and RIPEMD*, EUROCRYPT, 2005.

[WY05] X. Wang and H. Yu, *How to break MD5 and other hash functions*, EUROCRYPT, 2005.