This is a corrected version of Algorithmic Number Theory – Proceedings of ANTS-IV
(July 3–7, 2000, Leiden, Netherlands)
W. Bosma (Ed.), vol. 1838 of Lecture Notes in Computer Science, pages ???–???

Lattice Reduction in Cryptology: An Update

Phong Q. Nguyen and Jacques Stern

École Normale Supérieure
Département d'Informatique
45 rue d'Ulm, 75005 Paris, France
{Phong.Nguyen,Jacques.Stern}@ens.fr
http://www.di.ens.fr/~{pnguyen,stern}/

Abstract. Lattices are regular arrangements of points in space, whose study appeared in the 19th century in both number theory and crystallography. The goal of lattice reduction is to find useful representations of lattices. A major breakthrough in that field occurred twenty years ago, with the appearance of Lovász's reduction algorithm, also known as LLL or L^3. Lattice reduction algorithms have since proved invaluable in many areas of mathematics and computer science, especially in algorithmic number theory and cryptology. In this paper, we survey some applications of lattices to cryptology. We focus on recent developments of lattice reduction both in cryptography and cryptanalysis, which followed seminal works of Ajtai and Coppersmith.

1 Introduction

Lattices are discrete subgroups of $\mathbb{R}^n$. A lattice has infinitely many $\mathbb{Z}$-bases, but some are more useful than others. The goal of *lattice reduction* is to find interesting lattice bases, such as bases consisting of reasonably short and almost orthogonal vectors. From the mathematical point of view, the history of lattice reduction goes back to the reduction theory of quadratic forms developed by Lagrange [71], Gauss [44], Hermite [55], Korkine and Zolotarev [67, 68], among others, and to Minkowski's geometry of numbers [85]. With the advent of algorithmic number theory, the subject had a revival around 1980 with Lenstra's celebrated work on integer programming (see [74]), which was, among others, based on a novel but non-polynomial time[1] lattice reduction technique. That algorithm inspired Lovász to develop a polynomial-time algorithm that computes a so-called *reduced* basis of a lattice. It reached a final form in the seminal paper [73] where Lenstra, Lenstra and Lovász applied it to factor rational polynomials in polynomial time (back then, a famous problem), from which the name LLL comes. Further refinements of the LLL algorithm were later proposed, notably by Schnorr [101, 102].

Those algorithms have proved invaluable in many areas of mathematics and computer science (see [75, 64, 109, 52, 30, 69]). In particular, their relevance to

[1] The technique is however polynomial-time for fixed dimension, which was enough in [74].

cryptology was immediately understood, and they were used to break schemes based on the knapsack problem (see [99, 23]), which were early alternatives to the RSA cryptosystem [100]. The success of reduction algorithms at breaking various cryptographic schemes over the past twenty years (see [61]) have arguably established lattice reduction techniques as the most popular tool in public-key cryptanalysis. As a matter of fact, applications of lattices to cryptology have been mainly negative. Interestingly, it was noticed in many cryptanalytic experiments that LLL, as well as other lattice reduction algorithms, behave much more nicely than what was expected from the worst-case proved bounds. This led to a common belief among cryptographers, that lattice reduction is an easy problem, at least in practice.

That belief has recently been challenged by some exciting progress on the complexity of lattice problems, which originated in large part in two seminal papers written by Ajtai in 1996 and in 1997 respectively. Prior to 1996, little was known on the complexity of lattice problems. In his 1996 paper [3], Ajtai discovered a fascinating connection between the worst-case complexity and the average-case complexity of some well-known lattice problems. Such a connection is not known to hold for any other problem in NP believed to be outside P. In his 1997 paper [4], building on previous work by Adleman [2], Ajtai further proved the NP-hardness (under randomized reductions) of the most famous lattice problem, the shortest vector problem (SVP). The NP-hardness of SVP has been a long standing open problem. Ajtai's breakthroughs initiated a series of new results on the complexity of lattice problems, which are nicely surveyed by Cai [24, 25].

Those complexity results opened the door to positive applications in cryptology. Indeed, several cryptographic schemes based on the hardness of lattice problems were proposed shortly after Ajtai's discoveries (see [5, 49, 56, 26, 83, 41]). Some have been broken, while others seem to resist state-of-the-art attacks, for now. Those schemes attracted interest for at least two reasons: on the one hand, there are very few public-key cryptosystems based on problems different from integer factorization or the discrete logarithm problem, and on the other hand, some of those schemes offered encryption/decryption rates asymptotically higher than classical schemes. Besides, one of those schemes, by Ajtai and Dwork [5], enjoyed a surprising security proof based on worst-case (instead of average-case) hardness assumptions.

Independently of those developments, there has been renewed cryptographic interest in lattice reduction, following a beautiful work by Coppersmith [32] in 1996. Coppersmith showed, by means of lattice reduction, how to solve rigorously certain problems, apparently non-linear, related to the question of finding small roots of low-degree polynomial equations. In particular, this has led to surprising attacks on the celebrated RSA [100] cryptosystem in special settings such as low public or private exponent. Coppersmith's results differ from "traditional" applications of lattice reduction in cryptanalysis, where the underlying problem is already linear, and the attack often heuristic by requiring (at least) that current lattice reduction algorithms behave ideally, as opposed to what is theoretically guaranteed. The use of lattice reduction techniques to solve polynomial equations goes back to the eighties [54, 110]. The first result of that kind, the broadcast attack on low-exponent RSA due to Håstad [54], can be viewed as a weaker version of Coppersmith's theorem on univariate modular polynomial equations.

The rest of the paper is organized as follows. In Section 2, we give basic definitions and results on lattices and their algorithmic problems. In Section 3, we survey an old topic of lattice reduction in cryptology, the well-known subset sum or knapsack problem. Subsequent sections cover more recent applications. In Section 4, we discuss lattice-based cryptography, somehow a revival for knapsack-based cryptography. In Section 5, we review the only positive application known of the LLL algorithm in cryptology, related to the hidden number problem. In Section 6, we discuss developments on the problem of finding small roots of polynomial equations, inspired by Coppersmith's discoveries in 1996. In Section 7, we survey the surprising links between lattice reduction, the RSA cryptosystem, and integer factorization.

2 Lattice problems

2.1 Definitions

Recall that a *lattice* is a discrete (additive) subgroup of $\mathbb{R}^n$. In particular, any subgroup of $\mathbb{Z}^n$ is a lattice, and such lattices are called *integer lattices*. An equivalent definition is that a lattice consists of all integral linear combinations of a set of linearly independent vectors, that is,

$$L = \left\{ \sum_{i=1}^{d} n_i \mathbf{b}_i \mid n_i \in \mathbb{Z} \right\},$$

where the $\mathbf{b}_i$'s are linearly independent over $\mathbb{R}$. Such a set of vectors $\mathbf{b}_i$'s is called a lattice *basis*. All the bases have the same number $\dim(L)$ of elements, called the *dimension* (or *rank*) of the lattice.

There are infinitely many lattice bases. Any two bases are related to each other by some unimodular matrix (integral matrix of determinant ± 1), and therefore all the bases share the same Gram determinant $\det_{1 \le i,j \le d} \langle \mathbf{b}_i, \mathbf{b}_j \rangle$. The *volume* $\mathrm{vol}(L)$ (or *determinant*) of the lattice is by definition the square root of that Gram determinant, thus corresponding to the d-dimensional volume of the parallelepiped spanned by the $\mathbf{b}_i$'s. In the important case of full-dimensional lattices where $\dim(L) = n$, the volume is equal to the absolute value of the determinant of any lattice basis (hence the name determinant). If the lattice is further an integer lattice, then the volume is also equal to the index $[\mathbb{Z}^n : L]$ of L in $\mathbb{Z}^n$.

Since a lattice is discrete, it has a shortest non-zero vector: the Euclidean norm of such a vector is called the lattice *first minimum*, denoted by $\lambda_1(L)$ or $\|L\|$. Of course, one can use other norms as well : we will use $\|L\|_\infty$ to denote the first minimum for the infinity norm. More generally, for all $1 \leq i \leq \dim(L)$, Minkowski's i-th *minimum* $\lambda_i(L)$ is defined as the minimum of $\max_{1 \leq j \leq i} \|\mathbf{v}_j\|$ over all i linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_i \in L$. It will be convenient to define the *lattice gap* as the ratio $\lambda_2(L)/\lambda_1(L)$ between the first two minima.

Minkowski's Convex Body Theorem guarantees the existence of short vectors in lattices: a careful application shows that any d-dimensional lattice L satisfies $\|L\|_\infty \leq \mathrm{vol}(L)^{1/d}$, which is obviously the best possible bound. It follows that $\lambda_1(L) \leq \sqrt{d}\mathrm{vol}(L)^{1/d}$, which is not optimal, but shows that the value $\lambda_1(L)/\mathrm{vol}(L)^{1/d}$ is bounded when L runs over all d-dimensional lattices. The supremum of $\lambda_1(L)^2/\mathrm{vol}(L)^{2/d}$ is denoted by γ_d, and called Hermite's constant[2] of dimension d, because Hermite was the first to establish its existence in the language of quadratic forms. The best asymptotic bounds known for Hermite's constant are the following ones (see [84, Chapter II] for the lower bound, and [31, Chapter 9] for the upper bound):

$$\frac{d}{2\pi e} + \frac{\log(\pi d)}{2\pi e} + o(1) \leq \gamma_d \leq \frac{1.744d}{2\pi e}(1 + o(1)).$$

Minkowski proved more generally:

Theorem 1 (Minkowski). *For all d-dimensional lattice L and all $r \leq d$:*

$$\prod_{i=1}^{r} \lambda_i(L) \leq \sqrt{\gamma_d^r}\mathrm{vol}(L)^{r/d}.$$

More information on lattice theory can be found in numerous textbooks, such as [53, 108, 76].

2.2 Algorithmic problems

In the rest of this section, we assume implicitly that lattices are rational lattices (lattices in $\mathbb{Q}^n$), and d will denote the lattice dimension.

The most famous lattice problem is the *shortest vector problem* (SVP), which was apparently first stated by Dirichlet in 1842: given a basis of a lattice L, find $\mathbf{v} \in L$ such that $\|\mathbf{v}\| = \lambda_1(L)$. SVP_∞ will denote the analogue for the infinity norm. One defines approximate short vector problems by asking a non-zero $\mathbf{v} \in L$ with norm bounded by some approximation factor: $\|\mathbf{v}\| \leq f(d)\lambda_1(L)$.

The *closest vector problem* (CVP), also called the *nearest lattice point problem*, is a non-homogeneous version of the shortest vector problem: given a lattice basis and a vector $\mathbf{v} \in \mathbb{R}^n$, find a lattice vector minimizing the distance to $\mathbf{v}$. Again, one can define approximate versions.

Another problem is the *smallest basis problem* (SBP), which has many variants depending on the exact meaning of "smallest". The variant currently in vogue (see [3, 11]) is the following: find a lattice basis minimizing the maximum of the lengths of its elements. A more geometric variant asks instead to minimize the product of the lengths (see [52]).

[2] For historical reasons, Hermite's constant refers to $\max \lambda_1(L)^2/\mathrm{vol}(L)^{2/d}$ and not $\max \lambda_1(L)/\mathrm{vol}(L)^{1/d}$.

2.3 Complexity results

We refer to Cai [24, 25] for an up-to-date survey of complexity results. Ajtai [4] recently proved that SVP is NP-hard under randomized reductions. Micciancio [82, 81] simplified and improved the result by showing that approximating SVP to within a factor $< \sqrt{2}$ is also NP-hard under randomized reductions. The NP-hardness of SVP under deterministic (Karp) reductions remains an open problem.

CVP seems to be a more difficult problem. Goldreich *et al.* [50] recently noticed that CVP cannot be easier than SVP: given an oracle that approximates CVP to within a factor $f(d)$, one can approximate SVP in polynomial time to within the same factor $f(d)$. Reciprocally, Kannan proved in [64] that any algorithm approximating SVP to within a non-decreasing function $f(d)$ can be used to approximate CVP to within $d^{3/2}f(d)^2$. CVP was shown to be NP-hard as early as in 1981 [40] (for a simplified proof, see [65]). Approximating CVP to within a quasi-polynomial factor $2^{\log^{1-\varepsilon} d}$ is NP-hard [6, 38].

However, NP-hardness results for SVP and CVP have limits. Goldreich and Goldwasser [46] showed that approximating SVP or CVP to within $\sqrt{d/O(\log d)}$ is not NP-hard, unless the polynomial-time hierarchy collapses.

Interestingly, SVP and CVP problems seem to be more difficult with the infinity norm. It was shown that SVP_∞ and CVP_∞ are NP-hard in 1981 [40]. In fact, approximating $\mathrm{SVP}_\infty/\mathrm{CVP}_\infty$ to within an almost-polynomial factor $d^{1/\log\log d}$ is NP-hard [37]. On the other hand, Goldreich and Goldwasser [46] showed that approximating $\mathrm{SVP}_\infty/\mathrm{CVP}_\infty$ to within $d/O(\log d)$ is not NP-hard, unless the polynomial-time hierarchy collapses.

We will not discuss Ajtai's worst-case/average-case equivalence [3, 27], which refers to special versions of SVP and SBP (see [24, 25, 11]) such as SVP when the lattice gap λ_2/λ_1 is at least polynomial in the dimension.

2.4 Algorithmic results

The main algorithmic results are surveyed in [75, 64, 109, 52, 30, 69, 24, 97]. No polynomial-time algorithm is known for approximating either SVP, CVP or SBP to within a polynomial factor in the dimension d. In fact, the existence of such algorithms is an important open problem. The best polynomial time algorithms achieve only slightly subexponential factors, and are based on the LLL algorithm [73], which can approximate SVP and SBP. However, it should be emphasized that these algorithms typically perform much better than is theoretically guaranteed, on instances of practical interest. Given as input any basis of a lattice L, LLL provably outputs in polynomial time a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ satisfying :

$$\|\mathbf{b}_1\| \le 2^{(d-1)/4}\mathrm{vol}(L)^{1/d}, \|\mathbf{b}_i\| \le 2^{(d-1)/2}\lambda_i(L) \text{ and } \prod_{i=1}^{d}\|\mathbf{b}_i\| \le 2^{\binom{d}{2}/2}\mathrm{vol}(L).$$

Thus, LLL can approximate SVP to within $2^{(d-1)/2}$. Schnorr[3] [101] improved the bound to $2^{O(d(\log\log d)^2/\log d)}$. In fact, he defined an LLL-based family of algorithms [101] (named BKZ for blockwise Korkine-Zolotarev) whose performances depend on a parameter called the blocksize. These algorithms use some kind of exhaustive search exponential in the blocksize. So far, the best reduction algorithms in practice are variants [104, 105] of those BKZ-algorithms, which apply a heuristic to reduce exhaustive search. But little is known on the average-case (and even worst-case) complexity of reduction algorithms.

Babai's nearest plane algorithm [7] uses LLL to approximate CVP to within $2^{d/2}$, in polynomial time (see also [66]). Using Schnorr's algorithm [101], this can be improved to $2^{O(d(\log\log d)^2/\log d)}$, due to Kannan's link between CVP and SVP (see previous section). In practice however, the best strategy seems to be the *embedding method* (see [49, 90]), which uses the previous algorithms for SVP and a simple heuristic reduction from CVP to SVP. Namely, given a lattice basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ and a vector $\mathbf{v} \in \mathbb{R}^n$, the embedding method builds the $(d+1)$-dimensional lattice (in $\mathbb{R}^{n+1}$) spanned by the row vectors $(\mathbf{b}_i, 0)$ and $(\mathbf{v}, 1)$. It is hoped[4] that a shortest vector of that lattice is of the form $(\mathbf{v} - \mathbf{u}, 1)$ where $\mathbf{u}$ is a closest vector to $\mathbf{v}$, in the original lattice . Depending on the lattice, one should choose a coefficient different than 1 in $(\mathbf{v}, 1)$.

For exact SVP or CVP, the best algorithms known (in theory) are Kannan's super-exponential algorithms [63, 65], with running time $2^{O(d\log d)}$.

3 Knapsacks

Cryptology and lattices share a long history with the *knapsack* (also called *subset sum*) problem, a well-known NP-hard problem considered by Karp: given a set $\{a_1, a_2, \ldots, a_n\}$ of positive integers and a sum $s = \sum_{i=1}^n x_i a_i$, where $x_i \in \{0, 1\}$, recover the x_i's.

In 1978, Merkle and Hellman[80] invented one of the first public-key cryptosystems, by converting some easy knapsacks into what they believed were hard knapsacks. It was basically the unique alternative to RSA until 1982, when Shamir [106] proposed an attack against the simplest version of the Merkle-Hellman scheme. Shamir used Lenstra's integer programming algorithm [74] but, the same year, Adleman [1] showed how to use LLL instead, making experiments much easier. Brickell [21, 22] later extended the attacks to the more general "iterated" Merkle-Hellman scheme, and showed that Merkle-Hellman was insecure for all realistic parameters. The cryptanalysis of Merkle-Hellman schemes was the first application of lattice reduction in cryptology.

Despite the failure of Merkle-Hellman cryptosystems, researchers continued ~~to search for~~ knapsack cryptosystems because such systems are very easy to

[3] Schnorr's result is usually cited in the literature as an approximation algorithm to within $(1+\varepsilon)^n$ for any constant $\varepsilon > 0$. However, Goldreich and Håstad noticed about a year ago that one can choose some $\varepsilon = o(1)$ and still have polynomial running time, for instance using the blocksize $k = \log d/\log\log d$ in [101].

[4] Note that there exist simple counter-examples (see for instance [81]).

implement and can attain very high encryption/decryption rates. But basically, all knapsack cryptosystems have been broken (for a survey, see [99]), either by specific (often lattice-based) attacks or by the low-density attacks. The last significant candidate to survive was the Chor-Rivest cryptosystem [29], broken by Vaudenay [112] in 1997 with algebraic (not lattice) methods.

3.1 Low-density attacks

We only mention some of the links between lattices and knapsacks. Note that Ajtai's original proof [4] for the NP-hardness (under randomized reductions) of SVP used a connection between the subset sum problem and SVP.

The knapsack *density* is defined as $d = n/\max_{1\leq i\leq n}\log_2 a_i$. The low-density attacks establish a reduction from the subset sum problem to the lattice shortest vector problem. The first low-density attack used the n-dimensional lattice $L(a_1,\ldots,a_n,s)$ in $\mathbb{Z}^{n+1}$ formed by the vectors $(y_1,\ldots,y_{n+1})$ such that $y_1a_1+\cdots+y_na_n = y_{n+1}s$. Such a lattice can easily be built in polynomial time from the a_i's and s. It was proved by Lagarias and Odlyzko [70] that if $d \leq 0.6463\ldots$, the target vector $(x_1,\ldots,x_n,1)$ was the shortest vector of $L(a_1,\ldots,a_n,s)$ with high probability over the choice of the a_i's. The proof relies on bounds [77] on the number of integer points in n-dimensional balls. Thus, if one has access to an SVP-oracle, one can solve most subset sum problems of density $d \leq 0.6463\ldots$. Coster *et al.* [34] later improved the connection between SVP and the knapsack problem. By using a simple variant of $L(a_1,\ldots,a_n,s)$, they showed that if $d \leq 0.9408\ldots$, the knapsack problem can be reduced to a lattice shortest vector problem (in dimension n) with high probability. In a different context (polynomial interpolation in the presence of noise), another example of attack based on provable reduction to SVP appeared recently in [10].

In the light of recent results on the complexity of SVP, those reductions from knapsack to SVP may seem useless. Indeed, the NP-hardness of SVP under randomized reductions suggests that there is no polynomial-time algorithm that solves SVP. However, it turns out that in practice, one can hope that standard lattice reduction algorithms behave like SVP-oracles, up to reasonably high dimensions. Experiments carried out in [70, 104, 105] show the effectiveness of such approach for solving low-density subset sums, up to n about the range of 100–200. It does not prove nor disprove that one can solve, in theory or in practice, low-density knapsacks with n over several hundreds. But it was sufficient to show that knapsack cryptography was impractical: indeed, the keysize of knapsack schemes grows in general at least quadratically with n, so that high values of n (as required by lattice attacks) are not practical.

One might wonder whether those reductions can lead to provable polynomial-time algorithms for certain subset sums. Recall that LLL is an SVP-oracle when the lattice gap is exponential in the lattice dimension. For lattices used in knapsack reductions, the gap increases as the knapsack density decreases, however the gap can be proved to be large enough only in extremely low density (see [42, 43]). Hence, lattice methods to solve the subset sum problem are very heuristic. And lattice attacks against knapsack cryptosystems are somehow even more heuristic, because the reductions from knapsack to SVP assume some (natural) property on the distribution of the weights a_i's, which is in general not satisfied by knapsacks arising from cryptosystems.

3.2 The orthogonal lattice

Recently, Nguyen and Stern proposed in [91] a natural generalization of the Lagarias-Odlyzko [70] lattices. More precisely, they defined for any integer lattice L in $\mathbb{Z}^n$, the *orthogonal lattice* $L^\perp$ as the set of integer vectors orthogonal to L, that is, the set of $\mathbf{x} \in \mathbb{Z}^n$ such that the dot product $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ for all $\mathbf{y} \in L$. Note that the lattice $L^\perp$ has dimension $n - \dim(L)$, and can be computed in polynomial time from L (see [30]). Interestingly, the links between duality and orthogonality (see Martinet's book [76, pages 34–35]) enable to prove that the volume of $L^\perp$ is equal to the volume of the intersection $\bar{L}$ of $\mathbb{Z}^n$ with the linear span of L. Thus, if a lattice in $\mathbb{Z}^n$ is low-dimensional, its orthogonal lattice is high-dimensional with a volume at most equal: the successive minima of the orthogonal lattice are likely to be much shorter than the ones of the original lattice. That property of orthogonal lattices has led to effective (though heuristic) lattice-based attacks on various cryptographic schemes [91, 93, 94, 92, 95]. We refer to [96, 97] for more information. In particular, it was used in [95] to solve the *hidden subset sum problem* (used in [20]) in low density. The hidden subset sum problem was apparently a non-linear version of the subset sum problem: given M and n in $\mathbb{N}$, and $b_1, \ldots, b_m \in \mathbb{Z}_M$, find $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}_M$ such that each b_i is some subset sum modulo M of $\alpha_1, \ldots, \alpha_n$.

We sketch the solution of [95] to give a flavour of cryptanalyses based on orthogonal lattices. We first restate the hidden subset sum problem in terms of vectors. We are given an integer M, and a vector $\mathbf{b} = (b_1, \ldots, b_m) \in \mathbb{Z}^m$ with entries in $[0..M-1]$ such that there exist integers $\alpha_1, \ldots, \alpha_n \in [0..M-1]$, and vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}^m$ with entries in $\{0, 1\}$ satisfying:

$$\mathbf{b} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n \pmod{M}.$$

We want to determine the α_i's. There exists a vector $\mathbf{k} \in \mathbb{Z}^m$ such that:

$$\mathbf{b} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n + M\mathbf{k}.$$

Notice that if $\mathbf{u}$ in $\mathbb{Z}^n$ is orthogonal to $\mathbf{b}$, then $\mathbf{p_u} = (\langle \mathbf{u}, \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{u}, \mathbf{x}_n \rangle, \langle \mathbf{u}, \mathbf{k} \rangle)$ is orthogonal to the vector $\mathbf{v}_\alpha = (\alpha_1, \ldots, \alpha_n, M)$. But $\mathbf{v}_\alpha$ is independent of m, and so is the n-dimensional lattice $\mathbf{v}_\alpha^\perp$. On the other hand, as m grows for a fixed M, most of the vectors of any reduced basis of the $(m-1)$-dimensional lattice $\mathbf{b}^\perp$ should get shorter and shorter, because they should have norm close to $\mathrm{vol}(\mathbf{b}^\perp)^{1/(m-1)} \leq \mathrm{vol}(\mathbf{b})^{1/(m-1)} = \|\mathbf{b}\|^{1/(m-1)} \approx (M\sqrt{m})^{1/(m-1)}$. For such vectors $\mathbf{u}$, the corresponding vectors $\mathbf{p}_u$ also get shorter and shorter. But if $\mathbf{p}_u$ gets smaller than $\lambda_1(\mathbf{v}_\alpha^\perp)$ (which is independent of m), then it is actually zero, that is, $\mathbf{u}$ is orthogonal to all the $\mathbf{x}_j$'s and $\mathbf{k}$. Note that one expects $\lambda_1(\mathbf{v}_\alpha^\perp)$ to be of the order of $\|\mathbf{v}_\alpha\|^{1/n} \approx (M\sqrt{n})^{1/n}$.

This suggests that if $(\mathbf{u}_1, \ldots, \mathbf{u}_{m-1})$ is a sufficiently reduced basis of $\mathbf{b}^\perp$, then the first $m-(n+1)$ vectors $\mathbf{u}_1, \ldots, \mathbf{u}_{m-(n+1)}$ should heuristically be orthogonal to all the $\mathbf{x}_j$'s and $\mathbf{k}$. One cannot expect that more than $m-(n+1)$ vectors are orthogonal because the lattice L_x spanned by the $\mathbf{x}_j$'s and $\mathbf{k}$ is likely to have dimension $(n+1)$. From the previous discussion, one can hope that the heuristic condition is satisfied when the density $n/\log(M)$ is very small (so that $\lambda_1(\mathbf{v}_\alpha^\perp)$ is not too small), and m is sufficiently large. And if the heuristic condition is satisfied, the lattice $\bar{L}_x$ is disclosed, because it is then equal to the orthogonal lattice $(\mathbf{u}_1, \ldots, \mathbf{u}_{m-(n+1)})^\perp$. Once $\bar{L}_x$ is known, it is not difficult to recover (heuristically) the vectors $\mathbf{x}_j$'s by lattice reduction, because they are very short vectors. One eventually determines the coefficients α_j's from a linear modular system. The method is quite heuristic, but it works in practice for small parameters in low density (see [95] for more details).

4 Lattice-based cryptography

We review state-of-the-art results on the main lattice-based cryptosystems. To keep the presentation simple, descriptions of the schemes are intuitive, referring to the original papers for more details. Only one of these schemes (the GGH cryptosystem [49]) explicitly works with lattices.

4.1 The Ajtai-Dwork cryptosystem

Description. The Ajtai-Dwork cryptosystem [5] (AD) works in $\mathbb{R}^n$, with some finite precision depending on n. Its security is based on a variant of SVP.

The private key is a uniformly chosen vector u in the n-dimensional unit ball. One then defines a distribution $\mathcal{H}_u$ of points $\mathbf{a}$ in a large n-dimensional cube such that the dot product $\langle \mathbf{a}, \mathbf{u} \rangle$ is very close to $\mathbb{Z}$.

The public key is obtained by picking $\mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{v}_1, \ldots, \mathbf{v}_m$ (where $m = n^3$) independently at random from the distribution $\mathcal{H}_u$, subject to the constraint that the parallelepiped w spanned by the $\mathbf{w}_i$'s is not flat. Thus, the public key consists of a polynomial number of points close to a collection of parallel affine hyperplanes, which is kept secret.

The scheme is mainly of theoretical purpose, as encryption is bit-by-bit. To encrypt a '0', one randomly selects $b_1, \ldots, b_m$ in $\{0, 1\}$, and reduces $\sum_{i=1}^m b_i \mathbf{v}_i$ modulo the parallelepiped w. The vector obtained is the ciphertext. The ciphertext of '1' is just a randomly chosen vector in the parallelepiped w. To decrypt a ciphertext $\mathbf{x}$ with the private key $\mathbf{u}$, one computes $\tau = \langle \mathbf{x}, \mathbf{u} \rangle$. If τ is sufficiently close to $\mathbb{Z}$, then $\mathbf{x}$ is decrypted as '0', and otherwise as '1'. Thus, an encryption of '0' will always be decrypted as '0', and an encryption of '1' has a small probability to be decrypted as '0'. These decryption errors can be removed (see [48]).

Security. The Ajtai-Dwork [5] cryptosystem received wide attention due to a surprising security proof based on worst-case assumptions. Indeed, it was shown that any probabilistic algorithm distinguishing encryptions of a '0' from encryptions of a '1' with some polynomial advantage can be used to solve SVP in any n-dimensional lattice with gap λ_2/λ_1 larger than n^8. There is a converse, due to Nguyen and Stern [93]: one can decrypt in polynomial time with high probability, provided an oracle that approximates SVP to within $n^{0.5-\varepsilon}$, or one that approximates CVP to within $n^{1.33}$. It follows that the problem of decrypting ciphertexts is unlikely to be NP-hard, due to the result of Goldreich-Goldwasser [46].

Nguyen and Stern [93] further presented a heuristic attack to recover the secret key. Experiments suggest that the attack is likely to succeed up to at least $n = 32$. For such parameters, the system is already impractical, as the public key requires 20 megabytes and the ciphertext for each bit has bit-length 6144. This shows that unless major improvements[5] are found, the Ajtai-Dwork cryptosystem is only of theoretical importance.

Cryptanalysis overview. At this point, the reader might wonder how lattices come into play, since the description of AD does not involve lattices. Any ciphertext of '0' is a sum of $\mathbf{v}_i$'s minus some integer linear combination of the $\mathbf{w}_i$'s. Since the parallelepiped spanned by the $\mathbf{w}_i$'s is not too flat, the coefficients of the linear combination are relatively small. On the other hand, any linear combination of the $\mathbf{v}_i$'s and the $\mathbf{w}_i$'s with small coefficients is close to the hidden hyperplanes. This enables to build a particular lattice of dimension $n + m$ such that any ciphertext of '0' is in some sense close to the lattice, and reciprocally, any point sufficiently close to the lattice gives rise to a ciphertext of '0'. Thus, one can decrypt ciphertexts provided an oracle that approximates CVP sufficiently well. The analogous version for SVP uses related ideas, but is technically more complicated. For more details, see [93].

The attack to recover the secret key can be described quite easily. One knows that each $\langle \mathbf{v}_i, \mathbf{u} \rangle$ is close to some unknown integer V_i. It can be shown that any sufficiently short linear combination of the $\mathbf{v}_i$'s give information on the V_i's. More precisely, if $\sum_i \lambda_i \mathbf{v}_i$ is sufficiently short and the λ_i's are sufficiently small, then $\sum_i \lambda_i V_i = 0$ (because it is a too small integer). Note that the V_i's are disclosed if enough such equations are found. And each V_i gives an approximate linear equation satisfied by the coefficients of the secret key $\mathbf{u}$. Thus, one can compute a sufficiently good approximation of $\mathbf{u}$ from the V_i's. To find the V_i's, we produce many short combinations $\sum_i \lambda_i \mathbf{v}_i$ with small λ_i's, using lattice reduction. Heuristic arguments can justify that there exist enough such combinations. Experiments showed that the assumption was reasonable in practice.

4.2 The Goldreich-Goldwasser-Halevi cryptosystem

The Goldreich-Goldwasser-Halevi cryptosystem [49] (GGH) can be viewed as a lattice-analog to the McEliece [78] cryptosystem based on algebraic coding theory. In both schemes, a ciphertext is the addition of a random noise vector

[5] A variant of AD with less message expansion was proposed in [26], however without any security proof. It mixes AD with a knapsack.

to a vector corresponding to the plaintext. The public key and the private key are two representations of the same object (a lattice for GGH, a linear code for McEliece). The private key has a particular structure allowing to cancel noise vectors up to a certain bound. However, the domains in which all these operations take place are quite different.

Description. The GGH scheme works in $\mathbb{Z}^n$. The private key is a non-singular $n \times n$ integral matrix R, with very short row vectors[6] (entries polynomial in n). The lattice L is the full-dimensional lattice in $\mathbb{Z}^n$ spanned by the rows of R. The basis R is then transformed to a non-reduced basis B, which will be public. In the original scheme, B is the multiplication of R by sufficiently many small unimodular matrices. Computing a basis as "good" as the private basis R, given only the non-reduced basis B, means approximating SBP.

The message space is a "large enough" cube in $\mathbb{Z}^n$. A message $\mathbf{m} \in \mathbb{Z}^n$ is encrypted into $\mathbf{c} = \mathbf{m}B + \mathbf{e}$ where $\mathbf{e}$ is an error vector uniformly chosen from $\{-\sigma, \sigma\}^n$, where σ is a security parameter. A ciphertext $\mathbf{c}$ is decrypted as $\lfloor \mathbf{c}R^{-1} \rceil RB^{-1}$ (note: this is Babai's round method [7] to solve CVP). But an eavesdropper is left with the CVP-instance defined by $\mathbf{c}$ and B. The private basis R is generated in such a way that the decryption process succeeds with high probability. The larger σ is, the harder the CVP-instances are expected to be. But σ must be small for the decryption process to succeed.

Improvements. In the original scheme, the public matrix B is the multiplication of the secret matrix by sufficiently many unimodular matrices. This means that without appropriate precaution, the public matrix can be as large as $O(n^3 \log n)$ bits.[7] Micciancio [83] therefore suggested to define instead B as the Hermite normal form (HNF) of R. Recall that the HNF of an integer square matrix R in row notation is the unique lower triangular matrix with coefficients in $\mathbb{N}$ such that: the rows span the same lattice as R, and any entry below the diagonal is strictly less than the diagonal entry in its column. Here, one can see that the HNF of R is $O(n^2 \log n)$ bits, which is much better but still big. When using the HNF, one should encode messages into the error vector $\mathbf{e}$ instead of a lattice point, because the HNF is unbalanced. The ciphertext is defined as the reduction of $\mathbf{e}$ modulo the HNF, and hence uses less than $O(n \log n)$ bits. One can easily prove that the new scheme (which is now deterministic) cannot be less secure than the original GGH scheme (see [83]).

Security. GGH has no proven worst-case/average-case property, but it is much more efficient than AD. Specifically, for security parameter n, key-size and encryption time can be $O(n^2 \log n)$ for GGH (McEliece is slightly better though),

[6] A different construction for R based on tensor product was proposed in [41], but seems to worsen the decryption process.

[7] Since the determinant has $O(n \log n)$ bits, one can always make the matrix smaller than $O(n^3 \log n)$ bits.

vs. at least $O(n^4)$ for AD. For RSA and El-Gamal systems, key size is $O(n)$ and computation time is $O(n^3)$. The authors of GGH argued that the increase in size of the keys was more than compensated by the decrease in computation time. To bring confidence in their scheme, they published on the Internet a series of five numerical challenges [47], in dimensions 200, 250, 300, 350 and 400. In each of these challenges, a public key and a ciphertext were given, and the challenge was to recover the plaintext.

The GGH scheme is now considered broken, at least in its original form, due to an attack recently developed by Nguyen [90]. As an application, using small computing power and Shoup's NTL library [107], Nguyen was able to solve all the GGH challenges, except the last one in dimension 400. But already in dimension 400, GGH is not very practical: in the 400-challenge, the public key takes 1.8 Mbytes without HNF or 124 Kbytes using the HNF.[8]

Nguyen's attack used two "qualitatively different" weaknesses of GGH. The first one is inherent to the GGH construction: the error vectors used in the encryption process are always much shorter[9] than lattice vectors. This makes CVP-instances arising from GGH easier than general CVP-instances. The second weakness is the particular form of the error vectors in the encryption process. Recall that $\mathbf{c} = \mathbf{m}B + \mathbf{e}$ where $\mathbf{e} \in \{\pm\sigma\}^n$. The form of $\mathbf{e}$ was apparently chosen to maximize the Euclidean norm under requirements on the infinity norm. However, by looking at that equation modulo some well-chosen integer (such as σ or even better, 2σ), it is possible to derive information on the message $\mathbf{m}$, which in turn leads to a simplification of the original closest vector problem, by shortening the error vector $\mathbf{e}$. The simplified closest vector problem happens to be within reach (in practice) of current lattice reduction algorithms, thanks to the embedding strategy that heuristically reduces CVP to SVP. We refer to [90] for more information.

It is easy to fix the second weakness by selecting the entries of the error vector $\mathbf{e}$ at random in $[-\sigma \cdots +\sigma]$ instead of $\{\pm\sigma\}$. However, one can argue that the resulting GGH system would still be impractical, even using [83]. Indeed, Nguyen's experiments [90] showed that SVP could be solved in practice up to dimensions as high as 350, for (certain) lattices with gap as small as 10. To be competitive, the new GGH system would require the hardness (in lower dimensions due to the size of the public key, even using [83]) of SVP for certain lattices of only slightly smaller gap, which means a rather smaller improvement in terms of reduction. Note also that those experiments do not support the practical hardness of Ajtai's variant of SVP in which the gap is polynomial in the lattice dimension. Besides, it is not clear how to make decryption efficient without a huge secret key (Babai's rounding requires the storage of R^{-1} or a good approximation, which could be in [49] over 1 Mbytes in dimension 400).

[8] The challenges do not use the HNF, as they were proposed before [83]. Note that 124 Kbytes is about twice as large as McEliece for the recommended parameters.

[9] In all GGH-like constructions known, the error vector is always at least twice as short. The situation is even worse in [41].

4.3 The NTRU cryptosystem

Description. The NTRU cryptosystem [56], proposed by Hoffstein, Pipher and Silverman, works in the ring $R = \mathbb{Z}[X]/(X^N - 1)$. An element $F \in R$ is seen as a polynomial or a row vector: $F = \sum_{i=0}^{N-1} F_i x^i = [F_0, F_1, \ldots, F_{N-1}]$. To select keys, one uses the set $\mathcal{L}(d_1, d_2)$ of polynomials $F \in R$ such that d_1 coefficients are equal to 1, d_2 coeffients are equal to -1, and the rest are zero. There are two small coprime moduli $p < q$: a possible choice is $q = 128$ and $p = 3$. There are also three integer parameters d_f, d_g and d_ϕ quite a bit smaller than N (which is around a few hundreds).

The private keys are $f \in \mathcal{L}(d_f, d_f - 1)$ and $g \in \mathcal{L}(d_g, d_g)$. With high probability, f is invertible mod q. The public key $h \in R$ is defined as $h = g/f \mod q$. A message $m \in \{-(p-1)/2 \cdots + (p-1)/2\}^N$ is encrypted into: $e = (p\phi * h + m)$ mod q, where ϕ is randomly chosen in $\mathcal{L}(d_\phi, d_\phi)$. The user can decrypt thanks to the congruence $e * f \equiv p\phi * g + m * f \pmod q$, where the reduction is centered (one takes the smallest residue in absolute value). Since ϕ, f, g and m all have small coefficients and many zeroes (except possibly m), that congruence is likely to be a polynomial equality over $\mathbb{Z}$. By further reducing $e * f$ modulo p, one thus recovers $m * f \mod q$, hence m.

Security. The best attack known against NTRU is based on lattice reduction. The simplest lattice-based attack can be described as follows. Coppersmith and Shamir [33] noticed that the target vector $f \| g \in \mathbb{Z}^{2N}$ (the symbol $\|$ denotes vector concatenation) belongs to the following natural lattice:

$$L_{CS} = \{F \| G \in \mathbb{Z}^{2N} \mid F \equiv h * G \mod q \text{ where } F, G \in R\}.$$

It is not difficult to see that L_{CS} is a full-dimensional lattice in $\mathbb{Z}^{2N}$, with volume q^N. The volume suggests that the target vector is a shortest vector of L_{CS} (but with small gap), so that a SVP-oracle should heuristically output the private keys f and g. However, based on numerous experiments with Shoup's NTL library [107], the authors of NTRU claimed in [56] that all such attacks are exponential in N, so that even reasonable choices of N ensure sufficient security. Note that the keysize of NTRU is only $O(N \log q)$, which makes NTRU the leading candidate among knapsack-based and lattice-based cryptosystems, and allows high lattice dimensions. It seems that better attacks or better lattice reduction algorithms are required in order to break NTRU. To date, none of the numerical challenges proposed in [56] has been solved. However, cryptographic concerns have been expressed about the lack of security proofs for NTRU: there is no known result proving that NTRU or variants of its encryption scheme satisfy standard security requirements (such as semantic security or non-malleability,[10] see [79]), assuming the hardness of a sufficiently precise problem. Besides, there exist simple chosen ciphertext attacks [60] that can recover the secret key, so that appropriate padding is necessary.

[10] NTRU without padding cannot be semantically secure since $e(1) \equiv m(1) \pmod q$ as polynomials. And it is easily malleable using multiplications by X of polynomials (circular shifts).

5 The hidden number problem

5.1 Hardness of Diffie-Hellman bits

There is only one example known in which the LLL algorithm plays a positive role in cryptology. In [18], Boneh and Venkatesan used LLL to solve the *hidden number problem*, which enables to prove the hardness of the most significant bits of secret keys in Diffie-Hellman and related schemes in prime fields. Recall the Diffie-Hellman key exchange protocol [36]: Alice and Bob fix a finite cyclic G and a generator g. They respectively pick random $a, b \in [1, |G|]$ and exchange g^a and g^b. The secret key is g^{ab}. Proving the security of the protocol under "reasonable" assumptions has been a challenging problem in cryptography (see [12]). Computing the most significant bits of g^{ab} is as hard as computing g^{ab} itself, in the case of prime fields:

Theorem 2 (Boneh-Venkatesan). *Let q be an n-bit prime and g be a generator of $\mathbb{Z}_q^*$. Let $\varepsilon > 0$ be fixed, and set $\ell = \ell(n) = \lceil \varepsilon\sqrt{n} \rceil$. Suppose there exists an expected polynomial time (in n) algorithm $\mathcal{A}$, that on input q, g, g^a and g^b, outputs the ℓ most significant bits of g^{ab}. Then there is also an expected polynomial time algorithm that on input q, g, g^a, g^b and the factorization of $q-1$, computes all of g^{ab}.*

The above result is slightly different[11] from [18]. The same result holds for the least significant bits. For a more general statement when g is not necessarily a generator, and the factorization of $q-1$ is unknown, see [51]. No such results are known for other groups (there is some kind of analogous result [113] for finite fields though).

The proof goes as follows. We are given some g^a and g^b, and want to compute g^{ab}. We repeatedly pick a random r until g^{a+r} is a generator of $\mathbb{Z}_q^*$ (thanks to the factorization of $q-1$). For each r, the probability of success is $\phi(q-1)/(q-1) \geq 1/\log\log q$. Next, we apply $\mathcal{A}$ to the points g^{a+r} and g^{b+t} for many random values of t, so that we learn the most significant bits of $g^{(a+r)b}g^{(a+r)t}$, where $g^{(a+r)t}$ is a random element of $\mathbb{Z}_q^*$ since g^{a+r} is a generator. Note that one can easily recover g^{ab} from $\alpha = g^{(a+r)b}$. The problem becomes the *hidden number problem* (HNP): given $t_1, \ldots, t_d$ chosen uniformly and independently at random in $\mathbb{Z}_q^*$, and $\mathrm{MSB}_\ell(\alpha t_i \bmod q)$ for all i, recover $\alpha \in \mathbb{Z}_q$. Here, $\mathrm{MSB}_\ell(x)$ for $x \in \mathbb{Z}_q$ denotes any integer z satisfying $|x - z| < q/2^{\ell+1}$.

To achieve the proof, Boneh and Venkatesan presented a simple solution to HNP when ℓ is not too small, by reducing HNP to a lattice closest vector problem. We sketch this solution in the next section. One can try to prove the hardness of Diffie-Hellman bits for different groups with the same method. Curiously, for the important case of elliptic curve groups, no efficient solution is known for the corresponding hidden number problem, except when one uses projective coordinates to represent elliptic curve points.

[11] Due to an error in the proof of [18] spotted by [51].

5.2 Solving the hidden number problem by lattice reduction

Consider an HNP-instance: let $t_1, \ldots, t_d$ be chosen uniformly and independently at random in $\mathbb{Z}_q^*$, and $a_i = \mathrm{MSB}_\ell(\alpha t_i \bmod q)$ where $\alpha \in \mathbb{Z}_q$ is hidden. Clearly, the vector $\mathbf{t} = (t_1\alpha \bmod q, \ldots t_d\alpha \bmod q, \alpha/2^{\ell+1})$ belongs to the $(d+1)$-dimensional lattice $L = L(q, \ell, t_1, \ldots, t_d)$ spanned by the rows of the following matrix:

$$\begin{pmatrix} q & 0 & \cdots & 0 & 0 \\ 0 & q & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \ldots & 0 & q & 0 \\ t_1 & \ldots & \ldots & t_d & 1/2^{\ell+1} \end{pmatrix}$$

The vector $\mathbf{a} = (a_1, \ldots, a_d, 0)$ is very close to L, because it is very close to $\mathbf{t}$. Indeed, $\|\mathbf{t} - \mathbf{a}\| \le q\sqrt{d+1}/2^{\ell+1}$. It is not difficult to show that any lattice point sufficiently close to $\mathbf{a}$ discloses the hidden number α (see [18, Theorem 5] or [98]):

Lemma 3 (Uniqueness). *Set $d = 2\lceil\sqrt{\log q}\rceil$ and $\mu = \frac{1}{2}\sqrt{\log q} + 3$. Let α be in $\mathbb{Z}_q^*$. Choose integers $t_1, \ldots, t_d$ uniformly and independently at random in $\mathbb{Z}_q^*$. Let $\mathbf{a} = (a_1, \ldots, a_d, 0)$ be such that $|(\alpha t_i \bmod q) - a_i| < q/2^\mu$. Then with probability at least $\frac{1}{2}$, all $\mathbf{u} \in L$ with $\|\mathbf{u} - \mathbf{a}\| < \frac{q}{2^\mu}$ are of the form:*

$$\mathbf{u} = (t_1\beta \bmod q, \ldots t_d\beta \bmod q, \beta/2^{\ell+1}) \text{ where } \alpha \equiv \beta \pmod{q}.$$

Since $\mathbf{a}$ is close enough to L, Babai's nearest plane CVP approximation algorithm [7] yields a lattice point sufficiently close to $\mathbf{a}$, which leads to:

Theorem 4 (Boneh-Venkatesan). *Let α be in $\mathbb{Z}_q^*$. Let $\mathcal{O}$ be a function defined by $\mathcal{O}(t) = \mathrm{MSB}_\ell(\alpha t \bmod q)$ with $\ell = \lceil\sqrt{\log q}\rceil + \lceil\log\log q\rceil$. There exists a deterministic polynomial time algorithm $\mathcal{A}$ which, on input $t_1, \ldots, t_d, \mathcal{O}(t_1), \ldots, \mathcal{O}(t_d)$, outputs α with probability at least $1/2$ over $t_1, \ldots, t_d$ chosen uniformly and independently at random from $\mathbb{Z}_q^*$, where $d = 2\lceil\sqrt{\log q}\rceil$.*

Thus, the hidden number problem can be solved using $\ell = \sqrt{\log q} + \log\log q$ bits. Using Schnorr's improved lattice reduction algorithms, this can be asymptotically improved to $\varepsilon\sqrt{\log q}$ for any fixed $\varepsilon > 0$. One may also replace the bound $\frac{1}{2}$ by $\frac{1}{2\sqrt{n}}$ and reduce the number of bits required by $\log\log q$. Then, the expected run time goes up by a factor $\sqrt{\log q}$. One can alternately run $\sqrt{\log q}$ copies of the algorithm in parallel. Theorem 2 is a simple consequence.

5.3 Lattice attacks on DSA

Interestingly, the previous solution of the hidden number problem also has a dark side: it leads to a simple attack against the Digital Signature Algorithm [88, 79] (DSA) in special settings (see [59, 98]). Recall that the DSA uses a public element $g \in \mathbb{Z}_p$ of order q, a 160-bit prime dividing $p-1$ where p is a large prime (at least

512 bits). The signer has a secret key $\alpha \in \mathbb{Z}_q^*$ and a public key $\beta = g^\alpha \bmod p$. The DSA signature of a message m is $(r, s) \in \mathbb{Z}_q^2$ where $r = (g^k \bmod p) \bmod q$, $s = k^{-1}(h(m) + \alpha r) \bmod q$, h is SHA-1 hash function and k is a random element in $\mathbb{Z}_q^*$ chosen at each signature.

It is well-known that the secret key α can easily be recovered if the random nonce k is disclosed, or if k is produced by a cryptographically weak pseudo-random generator such as Knuth's linear congruential generator with known parameters [8][12] and a few signatures are available. Recently, Howgrave-Graham and Smart [59] noticed that Babai's nearest plane algorithm could heuristically recover α, provided that sufficiently many signatures and sufficiently many bits of the corresponding nonces k are known. This is not surprising, because the underlying problem is in fact very close to the hidden number problem.

Indeed, assume that for d signatures (r_i, s_i) of messages m_i, the ℓ least significant bits of the random nonce k_i are known to the attacker: one knows $a_i < 2^\ell$ such that $k_i - a_i$ is of the form $2^\ell b_i$. Then $\alpha r_i \equiv s_i(a_i + b_i 2^\ell) - h(m_i) \pmod q$, which can be rewritten as: $\alpha r_i 2^{-\ell} s_i^{-1} \equiv (a_i - s_i^{-1} h(m_i)) \cdot 2^{-\ell} + b_i \pmod q$. Letting $t_i = r_i 2^{-\ell} s_i^{-1} \bmod q$, one sees that $\mathrm{MSB}_\ell(\alpha t_i \bmod q)$ is known. Recovering the secret key α is therefore a slightly different hidden number problem in which the t_i's are not assumed to be independent and uniformly distributed over $\mathbb{Z}_q$, but are of the form $r_i 2^{-\ell} s_i^{-1}$ where the underlying k_i's are independent and uniformly distributed over $\mathbb{Z}_q^*$. In other words, HNP is an idealized version of the problem of breaking DSA (or related signature schemes) when the ℓ least significant bits (or more generally, ℓ consecutive bits) of the random nonce k are known for many signatures. It follows that Theorem 4 does not directly imply a provable attack on DSA in such settings.

But an attacker can ignore the difference between the distribution of $r_i 2^{-\ell} s_i^{-1}$ and the uniform distribution, and simply identify the DSA problem to HNP. Since lattice reduction algorithms can behave much better than theoretically expected, one can even hope to solve CVP exactly, yielding better bounds to Theorem 4. It is straightforward to extend Theorem 4 to the case where a CVP-oracle is available, by going through the proof of Lemma 3. For the case of a 160-bit prime q as in DSA, one obtains that HNP can be solved using respectively $\ell = 2$ bits and $d = 160$, or $\ell = 6$ bits and $d = 100$ respectively, when an oracle for CVP_∞ or CVP is available (see [98]). In fact, the bounds are even better in practice. It turns out that using standard lattice reduction algorithms implemented in Shoup's NTL library [107], one can often solve HNP for a 160-bit prime q using $\ell = 3$ bits and $d = 100$ (see [98]).

6 Finding small roots of low-degree polynomial equations

We survey an important application of lattice reduction found in 1996 by Coppersmith [32], and its developments. These results illustrate the power of linearization combined with lattice reduction.

[12] Note that even in the simple case where the parameters of the linear congruential generator are hidden, the attack of [8] does not apply.

6.1 Univariate modular equations

The general problem of solving univariate polynomial equations modulo some integer N of unknown factorization seems to be hard. Indeed, notice that for some polynomials, it is equivalent to the knowledge of the factorization of N. And the particular case of extracting e-th roots modulo N is the problem of decrypting ciphertexts in the RSA cryptosystem, for an eavesdropper. Curiously, Coppersmith [32] showed using LLL that the special problem of finding small roots is easy:

Theorem 5 (Coppersmith). *Let P be a monic polynomial of degree δ in one variable modulo an integer N of unknown factorization. Then one can find in time polynomial in $(\log N, \delta)$ all integers x_0 such that $P(x_0) \equiv 0 \pmod{N}$ and $|x_0| \leq N^{1/\delta}$.*

Related (but weaker) results appeared in the eighties [54, 110].[13] We sketch a proof of Theorem 5, as presented by Howgrave-Graham [57], who simplified Coppersmith's original proof (see also [62]). Coppersmith's method reduces the problem of finding small modular roots to the (easy) problem of solving polynomial equations over $\mathbb{Z}$. More precisely, it applies lattice reduction to find an integral polynomial equation satisfied by all small modular roots of P. The intuition is to linearize all the equations of the form $x^i P(x)^j \equiv 0 \pmod{N^j}$ for appropriate integral values of i and j. Such equations are satisfied by any solution of $P(x) \equiv 0 \pmod{N}$. Small solutions x_0 give rise to unusually short solutions to the resulting linear system. To transform modular equations into integer equations, the following elementary lemma[14] is used, with the notation $\|r(x)\| = \sqrt{\sum a_i^2}$ for any polynomial $r(x) = \sum a_i x^i \in \mathbb{Z}[x]$:

Lemma 6. *Let $r(x) \in \mathbb{Z}[x]$ be a polynomial of degree n and let X be a positive integer. Suppose $\|r(xX)\| < N^h/\sqrt{n}$. If $r(x_0) \equiv 0 \pmod{N^h}$ with $|x_0| < X$, then $r(x_0) = 0$ holds over the integers.*

Now the trick is to, given a parameter h, consider the $n = (h+1)\delta$ polynomials $q_{u,v}(x) = N^{h-v} x^u P(x)^v$, where $0 \leq u \leq \delta - 1$ and $0 \leq v \leq h$. Notice that any root x_0 of $P(x)$ modulo N is a root modulo N^h of $q_{u,v}(x)$, and therefore, of any integer linear combination $r(x)$ of the $q_{u,v}(x)$'s. If such a combination $r(x)$ further satisfies $\|r(xX)\| < N^h/\sqrt{n}$, then by Lemma 6, solving the equation $r(x) = 0$ over $\mathbb{Z}$ yields all roots of $P(x)$ modulo N less than X in absolute value. This suggests to look for a short vector in the lattice corresponding to the $q_{u,v}(xX)$'s. More precisely, define the $n \times n$ matrix M whose i-th row consists of the coefficients of $q_{u,v}(xX)$, starting by the low-degree terms, where $v = \lfloor (i-1)/\delta \rfloor$ and $u = (i-1) - \delta v$. Notice that M is lower triangular, and a simple calculation leads to $\det(M) = X^{n(n-1)/2} N^{nh/2}$. We apply an LLL-reduction to

[13] Håstad [54] presented his result in terms of system of low-degree modular equations, but he actually studies the same problem, and his approach achieves the weaker bound $N^{2/(\delta(\delta+1))}$.

[14] A similar lemma is used in [54]: the bound eventually obtained in [54] is weaker because only $h = 1$ is considered. Note also the resemblance with [73, Prop. 2.7].

the full-dimensional lattice spanned by the rows of M. The first vector of the reduced basis corresponds to a polynomial of the form $r(xX)$, and has Euclidean norm $\|r(xX)\|$. The theoretical bounds of the LLL algorithm ensure that:

$$\|r(xX)\| \leq 2^{(n-1)/4} \det(M)^{1/n} = 2^{(n-1)/4} X^{(n-1)/2} N^{h/2}.$$

Recall that we need $\|r(xX)\| \leq N^h/\sqrt{n}$ to apply the lemma. Hence, for a given h, the method is guaranteed to find modular roots up to X if:

$$X \leq \frac{1}{\sqrt{2}} N^{h/(n-1)} n^{-1/(n-1)}.$$

The limit of the upper bound, when h grows to ∞, is $\frac{1}{\sqrt{2}} N^{1/\delta}$. Theorem 5 follows from an appropriate choice of h. This result is practical (see [35, 58] for experimental results) and has many applications. It can be used to attack RSA encryption when a very low public exponent is used (see [13] for a survey). Boneh *et al.* [17] applied it to factor efficiently numbers of the form $N = p^r q$ for large r. Boneh [14] used a variant to find smooth numbers in short interval. See also [10] for an application to Chinese remaindering in the presence of noise.

Remarks. Theorem 5 is trivial if P is monic. Note also that one cannot hope to improve the (natural) bound $N^{1/\delta}$ for all polynomials and all moduli N. Indeed, for the polynomial $P(x) = x^\delta$ and $N = p^\delta$ where p is prime, the roots of P mod N are the multiples of p. Thus, one cannot hope to find all the small roots (slightly) beyond $N^{1/\delta} = p$, because there are too many of them. This suggests that even a SVP-oracle (instead of LLL) should not help Theorem 5 in general, as evidenced by the value of the lattice volume (the fudge factor $2^{(n-1)/4}$ yielded by LLL is negligible compared to $\det(M)^{1/n}$). It was recently noticed in [10] that if one only looks for the smallest root mod N, an SVP-oracle can improve the bound $N^{1/\delta}$ for very particular moduli (namely, squarefree N of known factorization, without too small factors). Note that in such cases, finding modular roots can still be difficult, because the number of modular roots can be exponential in the number of prime factors of N.

6.2 Multivariate modular equations

Interestingly, Theorem 5 can heuristically extend to multivariate polynomial modular equations. Assume for instance that one would like to find all small roots of $P(x, y) \equiv 0 \pmod{N}$, where $P(x, y)$ has total degree δ and has at least one monic monomial $x^\alpha y^{\delta-\alpha}$ of maximal total degree. If one could obtain two algebraically independent integral polynomial equations satisfied by all sufficiently small modular roots (x, y), then one could compute (by resultant) a univariate integral polynomial equation satisfied by x, and hence find efficiently all small (x, y). To find such equations, one can use an analogue of lemma 6 to bivariate polynomials, with the (natural) notation $\|r(x, y)\| = \sqrt{\sum_{i,j} a_{i,j}^2}$ for $r(x, y) = \sum_{i,j} a_{i,j} x^i y^j$:

Lemma 7. *Let $r(x,y) \in \mathbb{Z}[x,y]$ be a sum of at most w monomials. Assume $\|r(xX,yY)\| < N^h/\sqrt{w}$ for some $X,Y \geq 0$. If $r(x_0,y_0) \equiv 0 \pmod{N^h}$ with $|x_0| < X$ and $|y_0| < Y$, then $r(x_0,y_0) = 0$ holds over the integers.*

By analogy, one chooses a parameter h and select $r(x,y)$ as a linear combination of the polynomials $q_{u_1,u_2,v}(x,y) = N^{h-v}x^{u_1}y^{u_2}P(x,y)^v$, where $u_1+u_2+\delta v \leq h\delta$ and $u_1,u_2,v \geq 0$ with $u_1 < \alpha$ or $u_2 < \delta-\alpha$. Such polynomials have total degree less than $h\delta$, and therefore are linear combinations of the $n = (h\delta+1)(h\delta+2)/2$ monic monomials of total degree $\leq \delta h$. Due to the condition $u_1 < \alpha$ or $u_2 < \delta-\alpha$, such polynomials are in bijective correspondence with the n monic monomials (associate to $q_{u_1,u_2,v}(x,y)$ the monomial $x^{u_1+v\alpha}y^{u_2+v(\delta-\alpha)}$). One can represent the polynomials as n-dimensional vectors in such a way that the $n \times n$ matrix consisting of the $q_{u_1,u_2,v}(xX,yY)$'s (for some ordering) is lower triangular with coefficients $N^{h-v}X^{u_1+v\delta}Y^{u_2+v(\delta-\alpha)}$ on the diagonal.

Now consider the first two vectors $r_1(xX,yY)$ and $r_2(xX,yY)$ of an LLL-reduced basis of the lattice spanned by the rows of that matrix. Since any root (x_0,y_0) of $P(x,y)$ modulo N is a root of $q_{u_1,u_2,v}(x,y)$ modulo N^h, we need $\|r_1(xX,yY)\|$ and $\|r_2(xX,yY)\|$ to be less than $N^h/\sqrt{n}$ to apply Lemma 7. A (tedious) computation of the triangular matrix determinant enables to prove that $r_1(x,y)$ and $r_2(x,y)$ satisfy that bound when $XY < N^{1/\delta-\varepsilon}$ and h is sufficiently large (see [62]). Thus, one obtains two integer polynomial bivariate equations satisfied by all small modular roots of $P(x,y)$.

The problem is that, although such polynomial equations are linearly independent as vectors, they might be algebraically dependent, making the method heuristic. This heuristic assumption is unusual: many lattice-based attacks are heuristic in the sense that they require traditional lattice reduction algorithms to behave as SVP-oracles. An important open problem is to find sufficient conditions to make Coppersmith's method provable for bivariate (or multivariate) equations. Note that the method cannot work all the time. For instance, the polynomial $x-y$ has clearly too many roots over $\mathbb{Z}^2$ and hence too many roots mod any N (see [32] for more general counterexamples).

Such a result may enable to prove several attacks which are for now, only heuristic. Indeed, there are applications to the security of the RSA encryption scheme when a very low public exponent or a low private exponent is used (see [13] for a survey), and related schemes such as the KMOV cryptosystem (see [9]). In particular, the experimental evidence of [15, 9] shows that the method is very effective in practice for certain polynomials.

Remarks. In the case of univariate polynomials, there was basically no choice over the polynomials $q_{u,v}(x) = N^{h-1-v}x^uP(x)^v$ used to generate the appropriate univariate integer polynomial equation satisfied by all small modular roots. There is much more freedom with bivariate modular equations. Indeed, in the description above, we selected the indices of the polynomials $q_{u_1,u_2,v}(x,y)$ in such a way that they corresponded to all the monomials of total degree $\leq h\delta$, which form a triangle in $\mathbb{Z}^2$ when a monomial x^iy^j is represented by the point (i,j). This corresponds to the general case where a polynomial may have several monomials of maximal total degree. However, depending on the shape of the polynomial $P(x,y)$ and the bounds X and Y, other regions of (u_1,u_2,v) might lead to better bounds.

Assume for instance $P(x,y)$ is of the form $x^{\delta_x}y^{\delta_y}$ plus a linear combination of x^iy^j's where $i \leq \delta_x$, $j \leq \delta_y$ and $i+j < \delta_x+\delta_y$. Intuitively, it is better to select the (u_1,u_2,v)'s to cover the rectangle of sides $h\delta_x$ and $h\delta_y$ instead of the previous triangle, by picking all $q_{u_1,u_2,v}(x,y)$ such that $u_1+v\delta_x \leq h\delta_x$ and $u_2+v\delta_y \leq h\delta_y$, with $u_1 < \delta_x$ or $u_2 < \delta_y$. One can show that the polynomials $r_1(x,y)$ and $r_2(x,y)$ obtained from the first two vectors of an LLL-reduced basis of the appropriate lattice satisfy Lemma 7, provided that h is sufficiently large, and the bounds satisfy $X^{\delta_x}Y^{\delta_y} \leq N^{2/3-\varepsilon}$. Boneh and Durfee [15] applied similar and other tricks to a polynomial of the form $P(x,y) = xy+ax+b$. This allowed better bounds than the generic bound, leading to improved attacks on RSA with low secret exponent.

6.3 Multivariate integer equations

The general problem of solving multivariate polynomial equations over $\mathbb{Z}$ is also hard, as integer factorization is a special case. Coppersmith [32] showed that a similar[15] lattice-based approach can be used to find small roots of bivariate polynomial equations over $\mathbb{Z}$:

Theorem 8 (Coppersmith). *Let $P(x,y)$ be a polynomial in two variables over $\mathbb{Z}$, of maximum degree δ in each variable separately, and assume the coefficients of f are relatively prime as a set. Let X,Y be bounds on the desired solutions x_0,y_0. Define $\hat{P}(x,y) = P(Xx,Yy)$ and let D be the absolute value of the largest coefficient of $\hat{P}$. If $XY < D^{2/(3\delta)}$, then in time polynomial in $(\log D,\delta)$, we can find all integer pairs (x_0,y_0) such that $P(x_0,y_0)=0$, $|x_0| < X$ and $|y_0| < Y$.*

Again, the method extends heuristically to more than two variables, and there can be improved bounds depending on the shape[16] of the polynomial (see [32]). Theorem 8 was introduced to factor in polynomial time an RSA–modulus[17] $N = pq$ provided that half of the (either least or most significant) bits of either p or q are known (see [32, 14, 16]). This was sufficient to break an ID-based RSA encryption scheme proposed by Vanstone and Zuccherato [111]. Boneh *et al.* [16] provide another application, for recovering the RSA secret key when a large fraction of the bits of the secret exponent is known. Curiously, none of the applications cited above happen to be "true" applications of Theorem 8. It was later realized in [58, 17] that those results could alternatively be obtained from a (simple) variant of the univariate modular case (Theorem 5).

[15] However current proofs are somehow more technical than for Theorem 5. A simplification analogue to what has been obtained for Theorem 5 would be useful.

[16] The coefficient $2/3$ is natural from the remarks at the end of the previous section for the bivariate modular case. If we had assumed P to have total degree δ, the bound would be $XY < D^{1\delta}$.

[17] p and q are assumed to have similar size.

7 Lattices and RSA

Section 6 suggests to clarify the links existing between lattice reduction and RSA [100], the most famous public-key cryptosystem. We refer to [79] for an exposition of RSA, and to [13] for a survey of attacks on RSA encryption. Recall that in RSA, one selects two prime numbers p and q of approximately the same size. The number $N = pq$ is public. One selects an integer d coprime with $\phi(N) = (p-1)(q-1)$. The integer d is the private key, and is called the RSA *secret exponent*. The *public exponent* is the inverse e of d modulo $\phi(N)$.

7.1 Lattice attacks on RSA encryption

Small public exponent. When the public exponent e is very small, such as 3, one can apply Coppersmith's method (seen in the previous section) for univariate polynomials in various settings (see [13, 32, 35] for exact statements):

- An attacker can recover the plaintext of a given ciphertext, provided a large part of the plaintext is known.
- If a message is randomized before encryption, by simply padding random bits at a known place, an attacker can recover the message provided the amount of randomness is small.
- Håstad [54] attacks can be improved. An attacker can recover a message broadcasted (by RSA encryption and known affine transformation) to sufficiently many participants, each holding a different modulus N. This precisely happens if one sends a similar message with different known headers or time-stamps which are part of the encryption block.

None of the attacks recover the secret exponent d: they can only recover the plaintext. The attacks do not work if appropriate padding is used (see current standards and [79]), or if the public exponent is not too small. For instance, the popular choice $e = 65537$ is not threatened by these attacks.

Small private exponent. When $d \leq N^{0.25}$, an old result of Wiener [114] shows that one can easily recover the secret exponent d (and thus the factorization of N) from the continued fractions algorithm. Boneh and Durfee [15] recently improved the bound to $d \leq N^{0.292}$, by applying Coppersmith's technique to bivariate modular polynomials and improving the generic bound. Note that the attack is heuristic (see Section 6), but experiments showed that it works well in practice (no counterexample has ever been found). All those attacks on RSA with small private exponent also hold against the RSA signature scheme. A related result (using Coppersmith's technique for either bivariate integer or univariate modular polynomials) is an attack [16] to recover d when a large portion of the bits of d is known (see [13]).

7.2 Lattice attacks on RSA signature

The RSA cryptosystem is often used as a digital signature scheme. To prevent various attacks, one must apply a preprocessing scheme to the message, prior to signature. The recommended solution is to use hash functions and appropriate padding (see current standards and [79]). However, several alternative simple solutions not involving hashing have been proposed, and sometimes accepted as standards. Today, all such solutions have been broken (see [45]), some of them by lattice reduction techniques (see [86, 45]). Those lattice attacks are heuristic but work well in practice. They apply lattice reduction algorithms to find small solutions to (affine) linear systems, which leads to signature forgeries for certain proposed RSA signature schemes. Finding such small solutions is seen as a closest vector problem for some norm.

7.3 Factoring and lattice reduction

In the general case, the best attack against RSA encryption or signature is integer factorization. Note that to prove (or disprove) the equivalence between integer factorization and breaking RSA encryption remains an important open problem in cryptology (latest results [19] suggest that breaking RSA encryption may actually be easier). We already pointed out that in some special cases, lattice reduction leads to efficient factorization: when the factors are partially known [32], or when the number to factor has the form $p^r q$ with large r [17].

Schnorr [103] was the first to establish a link between integer factorization and lattice reduction, which was later extended by Adleman [2]. Schnorr [103] proposed a heuristic method to factor general numbers, using lattice reduction to approximate the closest vector problem in the infinity or the L_1 norm. Adleman [2] showed how to use the Euclidean norm instead, which is more suited to current lattice reduction algorithms. Those methods use the same underlying ideas as sieving algorithms (see [30]): to factor a number n, they try to find many congruences of smooth numbers to produce random square congruences of the form $x^2 \equiv y^2 \pmod{n}$, after a linear algebra step. Heuristic assumptions are needed to ensure the existence of appropriate congruences. The problem of finding such congruences is seen as a closest vector problem. Still, it should be noted that those methods are theoretical, since they are not adapted to currently known lattice reduction algorithms. To be useful, they would require very good lattice reduction for lattices of dimension over at least several thousands.

We close this review by mentioning that current versions of the Number Field Sieve (NFS) (see [72, 30]), the best algorithm known for factoring large integers, use lattice reduction. Indeed, LLL plays a crucial role in the last stage of NFS where one has to compute an algebraic square root of a huge algebraic number given as a product of hundreds of thousands of small ones. The best algorithm known to solve this problem is due to Montgomery (see [87, 89]). It has been used in all recent large factorizations, notably the record factorization [28] of a 512-bit RSA-number of 155 decimal digits proposed in the RSA challenges. There, LLL is applied many times in low dimension (less than 10) to find nice algebraic integers in integral ideals. But the overall running time of NFS is dominated by other stages, such as sieving and linear algebra.

8 Conclusions

Lovász's algorithm and other lattice basis reduction algorithms have proved invaluable in cryptology. They have become the most popular tool in public-key cryptanalysis. In particular, they play a crucial role in several attacks against the RSA cryptosystem. The past few years have seen new, sometimes provable, lattice-based methods for solving problems which were *a priori* not linear, and this definitely opens new fields of applications. Paradoxically, at the same time, a series of complexity results on lattice reduction has emerged, giving rise to another family of cryptographic schemes based on the hardness of lattice problems. The resulting cryptosystems have enjoyed different fates, but it is probably too early to tell whether or not secure and practical cryptography can be built using hardness of lattice problems. Indeed, several questions on lattices remain open. In particular, we still do not know whether or not it is easy to approximate the shortest vector problem up to some polynomial factor, or to find the shortest vector when the lattice gap is larger than some polynomial in the dimension. Besides, only very few lattice basis reduction algorithms are known, and their behaviour (both complexity and output quality) is still not well understood. And so far, there has not been any massive computer experiment in lattice reduction comparable to what has been done for integer factorization or the elliptic curve discrete logarithm problem. Twenty years of lattice reduction yielded surprising applications in cryptology. We hope the next twenty years will prove as exciting.

Acknowledgements. We thank Dan Boneh, Glenn Durfee, Arjen Lenstra, László Lovász, Daniele Micciancio, Igor Shparlinski and Joe Silverman for helpful discussions and comments.

References

1. L. M. Adleman. On breaking generalized knapsack publick key cryptosystems. In *Proc. of 15th STOC*, pages 402–412. ACM, 1983.
2. L. M. Adleman. Factoring and lattice reduction. Unpublished manuscript, 1995.
3. M. Ajtai. Generating hard instances of lattice problems. In *Proc. of 28th STOC*, pages 99–108. ACM, 1996. Available at [39] at TR96-007.
4. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proc. of 30th STOC*. ACM, 1998. Available at [39] as TR97-047.
5. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. of 29th STOC*, pages 284–293. ACM, 1997. Available at [39] at TR96-065.
6. S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.
7. L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.

8. M. Bellare, S. Goldwasser, and D. Micciancio. "Pseudo-random" number generation within cryptographic algorithms: The DSS case. In *Proc. of Crypto '97*, volume 1294 of *LNCS*. IACR, Springer-Verlag, 1997.
9. D. Bleichenbacher. On the security of the KMOV public key cryptosystem. In *Proc. of Crypto '97*, volume 1294 of *LNCS*. IACR, Springer-Verlag, 1997.
10. D. Bleichenbacher and P. Q. Nguyen. Noisy polynomial interpolation and noisy Chinese remaindering. In *Proc. of Eurocrypt'2000*, LNCS. IACR, Springer-Verlag, 2000.
11. J. Blömer and J.-P. Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *Proc. of 31st STOC*. ACM, 1999.
12. D. Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory – Proc. of ANTS-III*, volume 1423 of *LNCS*. Springer-Verlag, 1998.
13. D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999.
14. D. Boneh. Finding smooth integers in short intervals using CRT decoding. In *Proc. of 32nd STOC*. ACM, 2000.
15. D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $n^{0.292}$. In *Proc. of Eurocrypt '99*, volume 1592 of *LNCS*, pages 1–11. IACR, Springer-Verlag, 1999.
16. D. Boneh, G. Durfee, and Y. Frankel. An attack on RSA given a small fraction of the private key bits. In *Proc. of Asiacrypt '98*, volume 1514 of *LNCS*, pages 25–34. Springer-Verlag, 1998.
17. D. Boneh, G. Durfee, and N. Howgrave-Graham. Factoring $N = p^r q$ for large r. In *Proc. of Crypto '99*, volume 1666 of *LNCS*. IACR, Springer-Verlag, 1999.
18. D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in diffie-hellman and related schemes. In *Proc. of Crypto '96*, LNCS. IACR, Springer-Verlag, 1996.
19. D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In *Proc. of Eurocrypt '98*, volume 1233 of *LNCS*. IACR, Springer-Verlag, 1998.
20. V. Boyko, M. Peinado, and R. Venkatesan. Speeding up discrete log and factoring based schemes via precomputations. In *Proc. of Eurocrypt '98*, volume 1403 of *LNCS*, pages 221–235. IACR, Springer-Verlag, 1998.
21. E. F. Brickell. Solving low density knapsacks. In *Proc. of Crypto '83*. Plenum Press, 1984.
22. E. F. Brickell. Breaking iterated knapsacks. In *Proc. of Crypto '84*, volume 196 of *LNCS*. Springer-Verlag, 1985.
23. E. F. Brickell and A. M. Odlyzko. Cryptanalysis: A survey of recent results. In *Contemporary Cryptology*, pages 501–540. IEEE Press, 1991.
24. J.-Y. Cai. Some recent progress on the complexity of lattice problems. In *Proc. of FCRC*, 1999. Available at [39] as TR99-006.
25. J.-Y. Cai. The complexity of some lattice problems. In *Proc. of ANTS-IV*, LNCS. Springer-Verlag, 2000. In these proceedings.
26. J.-Y. Cai and T. W. Cusick. A lattice-based public-key cryptosystem. *Information and Computation*, 151:17–31, 1999.
27. J.-Y. Cai and A. P. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *Proc. of 38th FOCS*, pages 468–477. IEEE, 1997.
28. S. Cavallar, B. Dodson, A. K. Lenstra, W. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand,

F. Morain, A. Muffett, C. Putnam, C. Putnam, and P. Zimmermann. Factorization of 512-bit RSA key using the number field sieve. In *Proc. of Eurocrypt'2000*, LNCS. IACR, Springer-Verlag, 2000. Factorization announced in August, 1999.
29. B. Chor and R.L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inform. Theory*, 34, 1988.
30. H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer-Verlag, 1995. Second edition.
31. J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups.* Springer-Verlag, 1998. Third edition.
32. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. of Cryptology*, 10(4):233–260, 1997. Final version of two articles from Eurocrypt '96.
33. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *Proc. of Eurocrypt '97*, LNCS. IACR, Springer-Verlag, 1997.
34. M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.-P. Schnorr, and J. Stern. Improved low-density subset sum algorithms. *Comput. Complexity*, 2:111–128, 1992.
35. C. Coupé, P. Nguyen, and J. Stern. The effectiveness of lattice attacks against low-exponent RSA. In *Proc. of PKC'99*, volume 1431 of *LNCS*. Springer-Verlag, 1999.
36. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, Nov 1976.
37. I. Dinur. Approximating SVP_∞ to within almost-polynomial factors is NP-hard. Available at [39] at TR99-016.
38. I. Dinur, G. Kindler, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In *Proc. of 39th FOCS*, pages 99–109. IEEE, 1998. Available at [39] at TR98-048.
39. ECCC. `http://www.eccc.uni-trier.de/eccc/`. The Electronic Colloquium on Computational Complexity.
40. P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical report, Mathematische Instituut, University of Amsterdam, 1981. Report 81-04. Available at `http://turing.wins.uva.nl/~peter/`.
41. R. Fischlin and J.-P. Seifert. Tensor-based trapdoors for CVP and their application to public key cryptography. In *Cryptography and Coding*, volume 1746 of *LNCS*, pages 244–257. Springer-Verlag, 1999.
42. A. M. Frieze. On the lagarias-odlyzko algorithm for the subset sum problem. *SIAM J. Comput*, 15(2):536–539, 1986.
43. M. L. Furst and R. Kannan. Succinct certificates for almost all subset sum problems. *SIAM J. Comput*, 18(3):550–558, 1989.
44. C.F. Gauss. *Disquisitiones Arithmeticæ.* Leipzig, 1801.
45. M. Girault and J.-F. Misarsky. Cryptanalysis of countermeasures proposed for repairing ISO 9796–1. In *Proc. of Eurocrypt'2000*, LNCS. IACR, Springer-Verlag, 2000.
46. O. Goldreich and S. Goldwasser. On the limits of non-approximability of lattice problems. In *Proc. of 30th STOC.* ACM, 1998. Available at [39] as TR97-031.
47. O. Goldreich, S. Goldwasser, and S. Halevi. Challenges for the GGH cryptosystem. Available at `http://theory.lcs.mit.edu/~shaih/challenge.html`.
48. O. Goldreich, S. Goldwasser, and S. Halevi. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 105–111. IACR, Springer-Verlag, 1997. Available at [39] as TR97-018.

49. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 112–131. IACR, Springer-Verlag, 1997. Available at [39] as TR96-056.
50. O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. Available at [39] at TR99-002.
51. M. I. González Vasco and I. E. Shparlinski. On the security of Diffie-Hellman bits. In K.-Y. Lam, I. E. Shparlinski, H. Wang, and C. Xing, editors, *Proc. Workshop on Cryptography and Comp. Number Theory (CCNT'99)*. Birkhauser, 2000.
52. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1993.
53. M. Gruber and C. G. Lekkerkerker. *Geometry of Numbers*. North-Holland, 1987.
54. J. Håstad. Solving simultaneous modular equations of low degree. *SIAM J. Comput.*, 17(2):336–341, April 1988. Early version in Proc. of Crypto '85.
55. C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre. *J. Reine Angew. Math.*, 40:279–290, 1850. Also in the first volume of Hermite's complete works (Gauthier-Villars).
56. J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: a ring based public key cryptosystem. In *Proc. of ANTS III*, volume 1423 of *LNCS*, pages 267–288. Springer-Verlag, 1998. Additional information at `http://www.ntru.com`.
57. N. A. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and Coding*, volume 1355 of *LNCS*, pages 131–142. Springer-Verlag, 1997.
58. N. A. Howgrave-Graham. *Computational Mathematics Inspired by RSA*. PhD thesis, University of Bath, 1998.
59. N. A. Howgrave-Graham and N. P. Smart. Lattice attacks on digital signature schemes. Technical report, HP Labs, 1999. HPL-1999-90.
60. É. Jaulmes and A. Joux. A chosen ciphertext attack on NTRU. In *Proc. of Crypto '2000,LNCS*. IACR, Springer-Verlag, 2000. See also a related attack at the website of [56].
61. A. Joux and J. Stern. Lattice reduction: A toolbox for the cryptanalyst. *J. of Cryptology*, 11:161–185, 1998.
62. C. S. Jutla. On finding small solutions of modular multivariate polynomial equations. In *Proc. of Eurocrypt '98*, volume 1403 of *LNCS*, pages 158–170. IACR, Springer-Verlag, 1998.
63. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proc. of 15th STOC*, pages 193–206. ACM, 1983.
64. R. Kannan. Algorithmic geometry of numbers. *Annual review of computer science*, 2:231–267, 1987.
65. R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.
66. P. Klein. Finding the closest lattice vector when it's unusually close. In *Proc. of SODA '2000*. ACM–SIAM, 2000.
67. A. Korkine and G. Zolotareff. Sur les formes quadratiques positives ternaires. *Math. Ann.*, 5:581–583, 1872.
68. A. Korkine and G. Zolotareff. Sur les formes quadratiques. *Math. Ann.*, 6:336–389, 1873.
69. J. C. Lagarias. Point lattices. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, volume 1, chapter 19. Elsevier, 1995.
70. J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the Association for Computing Machinery*, January 1985.

71. L. Lagrange. Recherches d'arithmétique. *Nouv. Mém. Acad.*, 1773.
72. A. K. Lenstra and H. W. Lenstra, Jr. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.
73. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.
74. H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.
75. L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50. SIAM, 1986. CBMS-NSF Regional Conference Series in Applied Mathematics.
76. J. Martinet. *Les Réseaux Parfaits des Espaces Euclidiens*. Éditions Masson, 1996. English translation to appear at Springer-Verlag.
77. J. E. Mazo and A. M. Odlyzko. Lattice points in high-dimensional spheres. *Monatsh. Math.*, 110:47–61, 1990.
78. R.J. McEliece. A public-key cryptosystem based on algebraic number theory. Technical report, Jet Propulsion Laboratory, 1978. DSN Progress Report 42-44.
79. A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
80. R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, IT-24:525–530, September 1978.
81. D. Micciancio. *On the Hardness of the Shortest Vector Problem*. PhD thesis, Massachusetts Institute of Technology, 1998.
82. D. Micciancio. The shortest vector problem is NP-hard to approximate within some constant. In *Proc. of 39th FOCS*. IEEE, 1998. Available at [39] at TR98-016.
83. D. Micciancio. Lattice based cryptography: A global improvement. Technical report, Theory of Cryptography Library, 1999. Report 99-05.
84. J. Milnor and D. Husemoller. *Symmetric Bilinear Forms*. Springer-Verlag, 1973.
85. H. Minkowski. *Geometrie der Zahlen*. Teubner-Verlag, Leipzig, 1896.
86. J.-F. Misarsky. A multiplicative attack using LLL algorithm on RSA signatures with redundancy. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 221–234. IACR, Springer-Verlag, 1997.
87. P. L. Montgomery. Square roots of products of algebraic numbers. In Walter Gautschi, editor, *Mathematics of Computation 1943-1993: a Half-Century of Computational Mathematics*, Proc. of Symposia in Applied Mathematics, pages 567–571. American Mathematical Society, 1994.
88. National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard*, May 1994.
89. P. Nguyen. A Montgomery-like square root for the number field sieve. In *Proc. of ANTS-III*, volume 1423 of *LNCS*. Springer-Verlag, 1998.
90. P. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto '97. In *Proc. of Crypto '99*, volume 1666 of *LNCS*, pages 288–304. IACR, Springer-Verlag, 1999.
91. P. Nguyen and J. Stern. Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 198–212. IACR, Springer-Verlag, 1997.
92. P. Nguyen and J. Stern. Cryptanalysis of a fast public key cryptosystem presented at SAC '97. In *Selected Areas in Cryptography – Proc. of SAC '98*, volume 1556 of *LNCS*. Springer-Verlag, 1998.
93. P. Nguyen and J. Stern. Cryptanalysis of the Ajtai-Dwork cryptosystem. In *Proc. of Crypto '98*, volume 1462 of *LNCS*, pages 223–242. IACR, Springer-Verlag, 1998.

94. P. Nguyen and J. Stern. The Béguin-Quisquater server-aided RSA protocol from Crypto '95 is not secure. In *Proc. of Asiacrypt '98*, volume 1514 of *LNCS*, pages 372–379. Springer-Verlag, 1998.
95. P. Nguyen and J. Stern. The hardness of the hidden subset sum problem and its cryptographic implications. In *Proc. of Crypto '99*, volume 1666 of *LNCS*, pages 31–46. IACR, Springer-Verlag, 1999.
96. P. Nguyen and J. Stern. The orthogonal lattice: A new tool for the cryptanalyst. Manuscript submitted to J. of Cryptology, 2000.
97. P. Q. Nguyen. *La Géométrie des Nombres en Cryptologie*. PhD thesis, Université Paris 7, November 1999. Available at `http://www.di.ens.fr/~pnguyen/`.
98. P. Q. Nguyen. The dark side of the hidden number problem: Lattice attacks on DSA. In K.-Y. Lam, I. E. Shparlinski, H. Wang, and C. Xing, editors, *Proc. Workshop on Cryptography and Comp. Number Theory (CCNT'99)*. Birkhauser, 2000.
99. A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *Cryptology and Computational Number Theory*, volume 42 of *Proc. of Symposia in Applied Mathematics*, pages 75–88. A.M.S., 1990.
100. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. of the ACM*, 21(2):120–126, 1978.
101. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
102. C. P. Schnorr. A more efficient algorithm for lattice basis reduction. *J. of algorithms*, 9(1):47–62, 1988.
103. C. P. Schnorr. Factoring integers and computing discrete logarithms via diophantine approximation. In *Proc. of Eurocrypt '91*, volume 547 of *LNCS*, pages 171–181. IACR, Springer-Verlag, 1991.
104. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, 66:181–199, 1994.
105. C. P. Schnorr and H. H. Hörner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Proc. of Eurocrypt '95*, volume 921 of *LNCS*, pages 1–12. IACR, Springer-Verlag, 1995.
106. A. Shamir. A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. In *Proc. of 23rd FOCS*, pages 145–152. IEEE, 1982.
107. V. Shoup. Number Theory C++ Library (NTL) version 3.9. Available at `http://www.shoup.net/ntl/`.
108. C. L. Siegel. *Lectures on the Geometry of Numbers*. Springer-Verlag, 1989.
109. B. Vallée. La réduction des réseaux: autour de l'algorithme de Lenstra, Lenstra, Lovász. *RAIRO Inform. Théor. Appl.*, 23(3):345–376, 1989. English translation in *CWI Quaterly*, 3(2): 95–120, 1990.
110. B. Vallée, M. Girault, and P. Toffin. How to guess ℓ-th roots modulo n by reducing lattice bases. In *Proc. of AAEEC-6*, volume 357 of *LNCS*, pages 427–442. Springer-Verlag, 1988.
111. S. A. Vanstone and R. J. Zuccherato. Short RSA keys and their generation. *J. of Cryptology*, 8(2):101–114, 1995.
112. S. Vaudenay. Cryptanalysis of the Chor-Rivest cryptosystem. In *Proc. of Crypto '98*, volume 1462 of *LNCS*. Springer-Verlag, 1998. Appeared first at the "rump session" of Crypto '97.
113. E. R. Verheul. Certificates of recoverability with scalable recovery agent security. In *Proc. of PKC '2000*, LNCS. Springer-Verlag, 2000.

114. M. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inform. Theory*, 36(3):553–558, 1990.