Lattice-based Cryptography

Oded Regev[*]

Tel Aviv University, Israel

Abstract. We describe some of the recent progress on lattice-based cryptography, starting from the seminal work of Ajtai, and ending with some recent constructions of very efficient cryptographic schemes.

1 Introduction

In this survey, we describe some of the recent progress on lattice-based cryptography. What is a lattice? It is a set of points in n-dimensional space with a periodic structure, such as the one illustrated in Figure 1. More formally, given n-linearly independent vectors $v_1, \ldots, v_n \in \mathbb{R}^n$, the lattice generated by them is the set of vectors

$$L(v_1, \ldots, v_n) := \left\{ \sum_{i=1}^{n} \alpha_i v_i \;\middle|\; \alpha_i \in \mathbb{Z} \right\}.$$

The vectors $v_1, \ldots, v_n$ are known as a *basis* of the lattice.

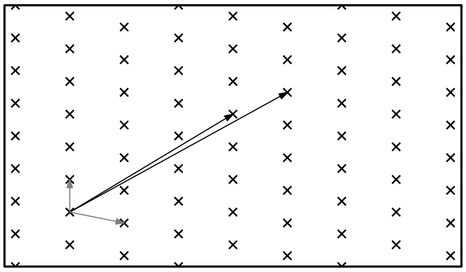

Fig. 1. A lattice in $\mathbb{R}^2$ and two of its bases

Historically, lattices were investigated since the late 18th century by mathematicians such as Lagrange, Gauss, and later Minkowski. More recently, lattices have become an active topic of research in computer science. They are used as an algorithmic tool to solve a wide variety of problems (e.g., [1–3]); they have

[*] Supported by an Alon Fellowship, by the Binational Science Foundation, by the Israel Science Foundation, and by the EU Integrated Project QAP.

found many applications in cryptanalysis (e.g., [4, 5]); and they have some unique properties from a computational complexity point of view (e.g., [6–8]). In this survey we will focus on their positive applications in cryptography, i.e., on the construction of cryptographic primitives whose security relies on the hardness of certain lattice problems.

Our starting point is Ajtai's seminal result from 1996 [9]. His surprising discovery was that lattices, which up to that point were used only as tools in cryptanalysis, can actually be used to *construct* cryptographic primitives. His work sparked a great interest in understanding the complexity of lattice problems and their relation to cryptography.

Ajtai's discovery was surprising for another reason: the security of his cryptographic primitive is based on the *worst-case hardness* of lattice problems. What this means is that if one succeeds in breaking the primitive, even with some small probability, then one can also solve *any* instance of a certain lattice problem. This remarkable property is what makes lattice-based cryptographic constructions so attractive. In contrast, virtually all other cryptographic constructions are based on some *average-case* assumption. For example, in cryptographic constructions based on factoring, the assumption is that it is hard to factor numbers chosen from a certain distribution. But how should we choose this distribution? Obviously, we should not use numbers with small factors (such as even numbers), but perhaps there are other numbers that we should avoid? In cryptographic constructions based on worst-case hardness, such questions do not even arise.

There are several other reasons for our interest in lattice-based cryptography. One is that the computations involved are very simple and often require only modular addition. This can be advantageous in certain practical scenarios when encryption is performed by a low-cost device. Another reason is that we currently do not have too many alternatives to traditional number-theoretic-based cryptography such as RSA. Such alternatives will be needed in case an efficient algorithm for factoring integers is ever found. In fact, efficient *quantum* algorithms for factoring integers and computing discrete logarithms already exist [10]. Although large-scale quantum computers are not expected to exist for at least a decade, this fact should already be regarded as a warning. There are currently no known quantum algorithms for lattice problems.

Our choice of topics for this survey is clearly biased by the author's personal taste and familiarity. One notable topic that we will not discuss here are cryptographic constructions following the design of Goldreich, Goldwasser, and Halevi [11]. In particular, this includes the highly efficient constructions developed by the company NTRU [12, 13].

For other surveys on the topic, see, e.g., [14, 15] and also the lecture notes [16, 17]. Another useful resource is the book by Micciancio and Goldwasser [18], which also contains a wealth of information on the computational complexity aspects of lattice problems.

The rest of this survey is organized as follows. In Section 2 we define the shortest vector problem and state some known results. In Section 3 we describe the known constructions of hash functions, starting from Ajtai's work [9]. Then, in Section 4 we describe the known constructions of public key cryptosystems. The only technical part of this survey is Section 5, where we outline the construction of a lattice-based collision resistant hash function together with its security proof. We end with some open questions in Section 6.

2 Lattice Problems

The main computational problem associated with lattices is the shortest vector problem (SVP). In SVP, given a lattice basis, we are supposed to output a shortest nonzero vector in the lattice. In fact, we will be mostly interested in the approximation variant of this problem, where our goal is to output a nonzero lattice vector whose norm is greater than that of the shortest nonzero lattice vector by at most some approximation factor γ. There are other interesting lattice problems (such as SIVP), and roughly speaking, the goal in most of them is to find short vectors under some appropriate definition of 'short'. We will encounter one such problem in Section 5. The behavior of these problems is often very similar to that of SVP, so for simplicity we do not discuss them in detail here (see [18] for more details).

Part of the difficulty of SVP comes from the fact that a lattice has many different bases and that typically, the given lattice basis contains very long vectors, much longer than the shortest nonzero vector. In fact, the well-known polynomial time algorithm of Lenstra, Lenstra, and Lovász (LLL) [1] from 1982 achieves an approximation factor of $2^{O(n)}$ where n is the dimension of the lattice. As bad as this might seem, this algorithm is surprisingly useful, with applications ranging from factoring polynomials over the rational numbers, integer programming, and many applications in cryptanalysis (such as attacks on knapsack based cryptographic systems and special cases of RSA). In 1987, Schnorr presented an improved algorithm obtaining an approximation factor that is slightly subexponential, namely $2^{O(n(\log\log n)^2/\log n)}$. This was recently improved to $2^{O(n\log\log n/\log n)}$ [19]. We should also mention that if one insists on an exact solution to SVP (or even just an approximation to within $\mathrm{poly}(n)$ factors), the best algorithm has a running time of $2^{O(n)}$ [19].

Given the above results, one might expect SVP to be NP-hard to approximate to within very large factors. However, the best known result only shows that approximating SVP to within factors $2^{(\log n)^{\frac{1}{2}-\varepsilon}}$ is NP-hard (under randomized quasi-polynomial time reductions) [8]. Moreover, SVP is not believed to be NP-hard to approximate to within factors above $\sqrt{n/\log n}$ [20, 6, 7], since for such approximation factors it lies in classes such as $\mathsf{NP} \cap \mathsf{coNP}$.

On the practical side, it is difficult to say what is the dimension n beyond which solving SVP becomes infeasible with today's computing power. A reasonable guess would be that taking n to be several hundreds makes the problem extremely difficult.

To conclude, the problem of approximating SVP to within polynomial factors n^c for $c \geq \frac{1}{2}$ seems to be very difficult (best algorithm runs in exponential time), however it is not believed to be NP-hard.

3 Hash Functions

As mentioned above, the first lattice-based cryptographic construction with worst-case security guarantees was presented in the seminal work of Ajtai [9]. More precisely, Ajtai presented a family of one-way functions whose security is based on the worst-case hardness of n^c-approximate SVP for some constant $c > 0$. In other words, he showed that being able to invert a function chosen from this family with non-negligible probability implies the ability to solve *any* instance of n^c-approximate SVP. Shortly after, Goldreich et al. [21] improved on Ajtai's result by constructing a stronger cryptographic primitive known as a family of *collision resistant hash functions*. Much of the subsequent work concentrated on decreasing the constant c (thereby improving the security assumption) [22–24]. In the most recent work, the constant is essentially $c = 1$ [24]. The hash function in all these constructions is essentially the modular subset sum function. We will see an example of such a construction in Section 5 below.

We remark that all these constructions are based on the worst-case hardness of a problem not believed to be NP-hard. Although it seems unlikely, it is not entirely impossible that further improvements in these constructions would lead us to approximation factors of the form n^c for c strictly below $\frac{1}{2}$. That would mean that we managed to base the security on the worst-case hardness of a problem that might be NP-hard.

The constructions described above are not too efficient. For instance, $\tilde{O}(n^2)$ bits are necessary in order to specify a function in the family, where n is the dimension of the lattice underlying the security and the $\tilde{O}$ hides poly-logarithmic factors (in other words, the key size is $\tilde{O}(n^2)$ bits). So if, for instance, we choose n to be several hundreds, we might need roughly a megabyte just to specify the hash function. Recently, an improved construction was presented by Micciancio [25]. He gives a family of one-way functions where only $\tilde{O}(n)$ bits are needed to specify a function in the family. Its security is based on the worst-case hardness of lattice problems on a restricted set of lattices known as cyclic lattices. Since no better algorithms are known for this family, it is reasonable to assume that solving lattice problems on these lattices is as hard as the general case. Finally, in more recent work [26, 27], the hash function of [25] was modified, preserving the efficiency and achieving the stronger security property of collision resistance.

4 Public-key Cryptography

Following Ajtai's discovery of lattice-based hash functions, Ajtai and Dwork [28] constructed a *public-key cryptosystem* whose security is based on the worst-case hardness of a lattice problem. Several improvements were given in subsequent works [29, 30]. Unlike the case of hash functions, the security of these cryptosystems is based on the worst-case hardness of a special case of SVP known as unique-SVP. Here, we are given a lattice whose shortest nonzero vector is shorter by some factor γ than all other nonparallel lattice vectors, and our goal is to find a shortest nonzero lattice vector. The hardness of this problem is not understood as well as that of SVP, and it is a very interesting open question whether one can base public-key cryptosystems on the (worst-case) hardness of SVP.

As is often the case in lattice-based cryptography, the cryptosystems themselves have a remarkably simple description (most of the work is in establishing their security). For example, let us describe the cryptosystem from [30]. Let N be some large integer. The private key is simply an integer h chosen randomly in the range $[\sqrt{N}, 2\sqrt{N})$. The public key consists of $m = O(\log N)$ numbers $a_1, \ldots, a_m$ in $\{0, 1, \ldots, N-1\}$ that are 'close' to integer multiples of N/h (notice that h doesn't necessarily divide N). We also include in the public key an index $i_0 \in [m]$ such that a_{i_0} is close to an *odd* multiple of N/h. We encrypt one bit at a time. An encryption of the bit 0 is the sum of a random subset of $\{a_1, \ldots, a_m\}$ reduced modulo N. An encryption of the bit 1 is done in the same way except we add $\lfloor a_{i_0}/2 \rfloor$ to the result before reducing modulo N. On receiving an encrypted word w, we consider its remainder on division by N/h. If it is small, we decrypt 0 and otherwise we decrypt 1. To establish the correctness of the decryption procedure, notice that since $a_1, \ldots, a_m$ are all close to integer multiples of N/h, any sum of a subset of them is also close to a multiple of N/h and hence encryptions of 0 are decrypted correctly. Similarly, since $\lfloor a_{i_0}/2 \rfloor$ is far from a multiple of N/h, encryptions of 1 are also far from multiples of N/h and hence we again decrypt correctly. The proof of security is more difficult and we omit it here (but see Section 5 for a related proof).

The aforementioned lattice-based cryptosystems are unfortunately quite inefficient. It turns out that when we base the security on lattices of dimension n, the size of the public key is $\tilde{O}(n^4)$ and each encrypted bit gets blown up to $\tilde{O}(n^2)$ bits. So if, for instance, we choose n to be several hundreds, the public key size is on the order of several gigabytes, which clearly makes the cryptosystem impractical.

Two recent works by Ajtai [31] and by the author [32] have tried to remedy this. Both works present cryptosystems whose public key scales like $\tilde{O}(n^2)$ (or even $\tilde{O}(n)$ if one can set up a pre-agreed random string of length $\tilde{O}(n^2)$) and each encrypted bit gets blown up to $\tilde{O}(n)$ bits. Combined with a very simple encryption process (involving only modular additions), this makes these two cryptosystems a good competitor for certain applications.

However, the security of these two cryptosystems is not as strong as that of other lattice-based cryptosystems. The security of Ajtai's cryptosystem [31] is based on a problem by Dirichlet, which is not directly related to any standard lattice problem. Moreover, his system has no worst-case hardness as the ones previously mentioned. However, his system, as well as many details in its proof of security, have the flavor of a lattice-based cryptosystem, and it might be that one day its security will be established based on the worst-case hardness of lattice problems.

The second cryptosystem [32] is based on the worst-case *quantum* hardness of the SVP. What this means is that breaking the cryptosystem implies an efficient quantum algorithm for approximating SVP. This security guarantee is incomparable to the one by Ajtai and Dwork: On one hand, it is stronger as it is based on the general SVP and not the special case of unique-SVP. On the other hand, it is weaker as it only implies a *quantum* algorithm for a lattice problem. Since no quantum algorithm is known to outperform classical algorithms for lattice problems, it is not unreasonable to conjecture that lattice problems are hard even quantumly. Moreover, it is possible that a more clever proof of security could establish the same worst-case hardness under a classical assumption. Finally, let us emphasize that the cryptosystem itself is entirely classical, and is in fact somewhat similar to the one of [30] described above.

5 An Outline of a Construction

In this section, we outline a construction of a lattice-based family of collision resistant hash functions. We will follow a simplified description of the construction in [24], without worrying too much about the exact security guarantee achieved.[1]

At the heart of the proof is the realization that by adding a sufficient amount of Gaussian noise to a lattice, one arrives at a distribution that is extremely close to uniform. An example of this effect is shown in Figure 2. This technique first appeared in [30], and is based on the work of Banaszczyk [33]. Let us denote by $\eta = \eta(L)$ the least amount of Gaussian noise required in order to obtain a distribution whose statistical distance to uniform is negligible (where by 'amount' we mean the standard deviation in each coordinate). This lattice parameter was analyzed in [24] where it was shown that it is 'relatively short' in the sense that finding nonzero lattice vectors of length at most $\mathrm{poly}(n)\eta$ is a 'hard' lattice problem as it automatically implies a solution to other, more standard, lattice problems such as an approximation to SVP to within polynomial factors.[2]

Before going on, we need to explain what exactly we mean by 'adding Gaussian noise to a lattice'. One way to rigorously define this is to consider the uniform distribution on all lattice points inside some large cube and then add Gaussian noise to this distribution. While this approach works just fine, it leads to some unnecessary technical complications due to the need to deal with the edges of the cube. Instead, we choose to take a mathematically cleaner approach (although it might be confusing at first): we work with the quotient $\mathbb{R}^n/L$. More explicitly, we define a function $h : \mathbb{R}^n \to [0,1)^n$ as follows. Given any $x \in \mathbb{R}^n$, write it as a linear combination of the lattice basis vectors $x = \sum_{i=1}^n \beta_i v_i$, and define $h(x) = (\beta_1, \ldots, \beta_n) \bmod 1$. So for instance, all points in L are mapped to $(0, \ldots, 0)$. Then the statement about the Gaussian noise above can be formally

[1] A more careful analysis of the construction described below shows that its security can be based on the worst-case hardness of $\tilde{O}(n^{1.5})$-approximate SIVP, which implies a security based on $\tilde{O}(n^{2.5})$-approximate SVP using standard reductions. In order to obtain the best known factor of $\tilde{O}(n)$, one needs to use an iterative procedure.

[2] To be precise, we need slightly more than just finding vectors of length at most $\mathrm{poly}(n)\eta$; we need to be able to find n linearly independent vectors of this length. As it turns out, by repeatedly calling the procedure described below, one can obtain such vectors.

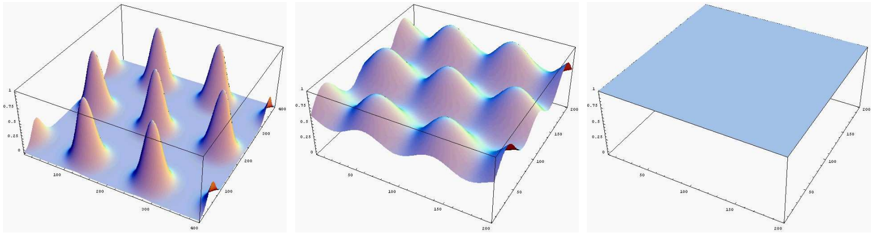

Fig. 2. A lattice with different amounts of Gaussian noise

stated as follows: if we sample a point x from a Gaussian distribution in $\mathbb{R}^n$ centered around 0 of standard deviation η in each coordinate, then the statistical distance between the distribution of $h(x)$ and the uniform distribution on $[0,1)^n$ is negligible.

We now turn to the construction. Our family of hash functions is the modular subset-sum function over $\mathbb{Z}_q^n$, as defined next. Fix $q = 2^{2n}$ and $m = 4n^2$. For each $a_1, \ldots, a_m \in \mathbb{Z}_q^n$, the family contains the function $f_{a_1,\ldots,a_m} : \{0,1\}^m \to \{0,1\}^{n \log q}$ given by

$$f_{a_1,\ldots,a_m}(b_1,\ldots,b_m) = \sum_{i=1}^{m} b_i a_i \bmod q.$$

Notice that with our choice of parameters, $m > n \log q$ so collisions are guaranteed to exist. Clearly, these functions are easy to compute. Our goal is therefore to show that they are collision resistant. We establish this by proving that if there exists a polynomial-time algorithm CollisionFind that given $a_1, \ldots, a_m$ chosen uniformly from $\mathbb{Z}_q^n$, finds with some non-negligible probability $b_1, \ldots, b_m \in \{-1,0,1\}$, not all zero, such that $\sum_{i=1}^{m} b_i a_i = (0,\ldots,0) \pmod{q}$, then there is a polynomial-time algorithm that finds vectors of length at most $\mathrm{poly}(n)\eta$ in any given lattice L (which, as mentioned before, implies a solution to approximate SVP).

Our first observation is that from CollisionFind we can easily construct another algorithm, call it CollisionFind$'$, that performs the following task: given elements $a_1, \ldots, a_m$ chosen uniformly from $[0,1)^n$, it finds with some non-negligible probability $b_1, \ldots, b_m \in \{-1,0,1\}$, not all zero, such that $\sum_{i=1}^{m} b_i a_i \in [-\frac{m}{q}, \frac{m}{q}]^n \pmod{1}$. In other words, it finds a $\{-1,0,1\}$ combination of $a_1, \ldots, a_m$ that is extremely close to $(0,\ldots,0)$ modulo 1. To see this, observe that CollisionFind$'$ can simply apply CollisionFind to $\lfloor q a_1 \rfloor, \ldots, \lfloor q a_m \rfloor$.

Our goal now is to show that using CollisionFind$'$ we can find vectors of length at most $\mathrm{poly}(n)\eta$ in any given lattice L. So let L be some lattice given by its basis $v_1, \ldots, v_n$. Our first step is to apply the LLL algorithm to $v_1, \ldots, v_n$. This makes sure that $v_1, \ldots, v_n$ are not 'unreasonably' long: namely, none of these vectors is longer than $2^n \eta$.

We now arrive at the main part of the procedure. We first choose m vectors $x_1, \ldots, x_m$ independently from the Gaussian distribution in $\mathbb{R}^n$ centered around

0 of standard deviation η in each coordinate. (To be precise, we don't know η, but we can obtain a good enough estimate by trying a polynomial number of values.) Next, we compute $a_i = h(x_i)$ for $i = 1, \dots, m$. By the discussion above, we know that each a_i is distributed essentially uniformly on $[0, 1)^n$. We can therefore apply $\textsc{CollisionFind}'$ to $a_1, \dots, a_m$ and obtain with non-negligible probability $b_1, \dots, b_m \in \{-1, 0, 1\}$ such that $\sum_{i=1}^m b_i a_i \in [-\frac{m}{q}, \frac{m}{q}]^n \pmod 1$. Now consider the vector $y = \sum_{i=1}^m b_i x_i$. On one hand, this is a short vector, as it is the sum of at most m vectors of length roughly $\sqrt{n}\eta$ each. On the other hand, by the linearity of h, we have that $h(y) \in [-\frac{m}{q}, \frac{m}{q}]^n \pmod 1$. What this means is that y is extremely close to a lattice vector. Indeed, write $y = \sum \beta_i v_i$ for some reals $\beta_1, \dots, \beta_m$. Then we have that each β_i is within $\pm\frac{m}{q}$ of an integer. Consider now the *lattice vector* $y' = \sum \lfloor \beta_i \rceil v_i$ obtained by rounding each β_i to the nearest integer. Then the distance between y and y' is

$$\|y - y'\| \leq \sum_{i=1}^n \frac{m}{q} \|v_i\| \leq \frac{mn}{q} 2^n \eta \ll \eta$$

and in particular we found a lattice vector y' of length at most $\mathrm{poly}(n)\eta$. The procedure can now output y', which is a short lattice vector.

So are we done? Well, not completely: we still have to show that y' is *nonzero* (with some non-negligible probability). The proof of this requires some effort, so we just give the main idea. Recall that we define y' as a (rounding of a) $\{-1, 0, 1\}$ combination of $x_1, \dots, x_m$ obtained by calling $\textsc{CollisionFind}'$ with $a_1, \dots, a_m$. The difficulty in proving that $y' \neq 0$ is that we have no control over $\textsc{CollisionFind}'$, and in particular it might act in some 'malicious' way, trying to set the $b_1, \dots, b_m$ so that y' ends up being the zero vector. To solve this issue, one can prove that the a_i do not contain enough information about the x_i. In other words, conditioned on any fixed values to the a_i, the x_i still have enough uncertainty in them to guarantee that no matter what $\textsc{CollisionFind}'$ outputs, y' is nonzero with very high probability.

To conclude, we have seen that by a single call to the collision finder, one can find in *any* given lattice, a nonzero vector of length at most $(m\sqrt{n} + 1)\eta = O(n^{2.5}\eta)$ with some non-negligible probability. Obviously, by repeating this a polynomial number of times, we can obtain such a vector with very high probability. The essence of the proof, and what makes possible the connection between the average-case collision finding problem and the worst-case lattice problem, is the realization that all lattices look the same after adding a small amount of noise — they turn into a uniform distribution.

6 Open Questions

- **Cryptanalysis:** Attacks on lattice-based cryptosystems, such as the one by Nguyen and Stern [34], seem to be limited to low dimensions (a few tens). Due to the greatly improved efficiency of the new cryptosystems in [31, 32], using much higher dimensions has now become possible. It would be very interesting to see attempts to attack these new cryptographic constructions.

- **Improved cryptosystems:** As we have seen in Section 4, the situation with lattice-based cryptosystems is not entirely satisfactory: The original construction of Ajtai and Dwork, as well as some of the follow-up work, are based on the hardness of the unique-SVP and are moreover quite inefficient. Two recent attempts [31, 32] give much more efficient constructions, but with less-than-optimal security guarantees. Other constructions, such as the one by NTRU [12], are extremely efficient but have no provable security. A very interesting open question is to obtain efficient lattice-based cryptosystems based on the worst-case hardness of unique-SVP (or preferably SVP). Another interesting direction is whether specific families of lattices, such as cyclic lattices, can be used to obtain more efficient constructions.
- **Comparison with number theoretic cryptography:** Can one factor integers or compute discrete logarithms using an oracle that solves, say, $\sqrt{n}$-approximate SVP? Such a result would prove that lattice-based cryptosystems are superior to traditional number-theoretic-based ones (see [35, 36] for related work).
- **Reverse reductions:** Is the security of lattice-based cryptographic constructions *equivalent* to the hardness of lattice-problems? More concretely, assuming we have an oracle that solves (say) $\sqrt{n}$-approximate SVP, can we break lattice-based cryptography? A result along these lines is known for the Ajtai-Dwork cryptosystem [34], but it is still open if the same can be shown for newer cryptosystems such as the ones in [31, 32].
- **Signature schemes:** Lattices have been successfully used in constructing hash functions and public key cryptosystems. Can one also construct signature schemes with worst-case hardness guarantees and similar efficiency? See [37] for some related work.
- **Security against chosen-ciphertext attacks:** The Ajtai-Dwork cryptosystem, as well as all subsequent work, are not secure against chosen-ciphertext attacks. Indeed, it is not too difficult to see that one can extract the private key given access to the decryption oracle. In practice, there are known methods to deal with this issue. It would be interesting to find an (efficient) solution with a rigorous proof of security in the standard model (for related work, see, e.g., [38]).
- **Applications in Learning Theory:** The cryptosystems of [30, 32] were recently used by Klivans and Sherstov to obtain cryptographic hardness results for problems in learning theory [39]. It would be interesting to extend this line of research.

Acknowledgements I am grateful to Ishay Haviv, Julia Kempe, Daniele Micciancio, and Phong Nguyen for many helpful comments.

References

1. Lenstra, A.K., Lenstra, Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4) (1982) 515–534

2. Lenstra, Jr., H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4) (1983) 538–548
3. Babai, L.: On Lovasz' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1) (1986) 1–13 Preliminary version in STACS 1985.
4. Lagarias, J.C., Odlyzko, A.M.: Solving low-density subset sum problems. J. Assoc. Comput. Mach. **32**(1) (1985) 229–246
5. Coppersmith, D.: Finding small solutions to small degree polynomials. Lecture Notes in Computer Science **2146** (2001) 20–31
6. Goldreich, O., Goldwasser, S.: On the limits of nonapproximability of lattice problems. Journal of Computer and System Sciences **60**(3) (2000) 540–563 Preliminary version in STOC 1998.
7. Aharonov, D., Regev, O.: Lattice problems in NP intersect coNP. Journal of the ACM **52**(5) (2005) 749–765 Preliminary version in FOCS 2004.
8. Khot, S.: Hardness of approximating the shortest vector problem in lattices. In: Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS). (2004) 126–135
9. Ajtai, M.: Generating hard instances of lattice problems. In: Proc. 28th ACM Symp. on Theory of Computing. (1996) 99–108 Available from ECCC at `http://www.uni-trier.de/eccc/`.
10. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. on Computing **26**(5) (1997) 1484–1509
11. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Advances in cryptology. Volume 1294 of Lecture Notes in Comput. Sci. Springer (1997) 112–131
12. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Algorithmic number theory (ANTS). Volume 1423 of Lecture Notes in Comput. Sci. Springer (1998) 267–288
13. Hoffstein, J., Graham, N.A.H., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSIGN: Digital signatures using the NTRU lattice. In: Proc. of CT-RSA. Volume 2612 of Lecture Notes in Comput. Sci., Springer-Verlag (2003) 122–140
14. Nguyen, P.Q., Stern, J.: The two faces of lattices in cryptology. In Silverman, J.H., ed.: Cryptography and Lattices, International Conference (CaLC 2001). Number 2146 in Lecture Notes in Computer Science (2001) 146–180
15. Kumar, R., Sivakumar, D.: Complexity of SVP – a reader's digest. SIGACT News **32**(3) (2001) 40–52
16. Micciancio, D.: Lattices in cryptography and cryptanalysis (2002) Lecture notes of a course given in UC San Diego.
17. Regev, O.: Lattices in computer science (2004) Lecture notes of a course given in Tel Aviv University.
18. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptographic Perspective. Volume 671 of The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, Massachusetts (2002)
19. Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proc. 33rd ACM Symp. on Theory of Computing. (2001) 601–610
20. Lagarias, J.C., Lenstra, Jr., H.W., Schnorr, C.P.: Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. Combinatorica **10**(4) (1990) 333–348
21. Goldreich, O., Goldwasser, S., Halevi, S.: Collision-free hashing from lattice problems. Technical Report TR96-056, Electronic Colloquium on Computational Complexity (ECCC) (1996)

22. Cai, J.Y., Nerurkar, A.: An improved worst-case to average-case connection for lattice problems. In: Proc. 38th IEEE Symp. on Found. of Comp. Science. (1997) 468–477
23. Micciancio, D.: Improved cryptographic hash functions with worst-case/average-case connection. In: Proc. 34th ACM Symp. on Theory of Computing (STOC). (2002) 609–618
24. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In: Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS). (2004) 372–381
25. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. Computational Complexity (2006) To appear. Preliminary version in ECCC report TR04-095.
26. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: 33rd International Colloquium on Automata, Languages and Programming (ICALP). (2006)
27. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: 3rd Theory of Cryptography Conference (TCC). (2006) 145–166
28. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: Proc. 29th Annual IEEE Symp. on Foundations of Computer Science (FOCS). (1997) 284–293
29. Goldreich, O., Goldwasser, S., Halevi, S.: Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In: Advances in cryptology. Volume 1294 of Lecture Notes in Comput. Sci. Springer (1997) 105–111
30. Regev, O.: New lattice-based cryptographic constructions. Journal of the ACM **51**(6) (2004) 899–942 Preliminary version in STOC'03.
31. Ajtai, M.: Representing hard lattices with $O(n \log n)$ bits. In: Proc. 37th Annual ACM Symp. on Theory of Computing (STOC). (2005)
32. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proc. 37th ACM Symp. on Theory of Computing (STOC). (2005) 84–93
33. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen **296**(4) (1993) 625–635
34. Nguyen, P., Stern, J.: Cryptanalysis of the Ajtai-Dwork cryptosystem. In: Advances in cryptology (CRYPTO). Volume 1462 of Lecture Notes in Comput. Sci. Springer (1998) 223–242
35. Schnorr, C.P.: Factoring integers and computing discrete logarithms via Diophantine approximation. In Cai, J.Y., ed.: Advances in computational complexity. Volume 13 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. AMS (1993) 171–182 Preliminary version in Eurocrypt '91.
36. Adleman, L.M.: Factoring and lattice reduction (1995) Unpublished manuscript.
37. Micciancio, D., Vadhan, S.: Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In: Advances in cryptology (CRYPTO). Volume 2729 of Lecture Notes in Computer Science., Springer-Verlag (2003) 282–298
38. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: Advances in cryptology (EUROCRYPT). Volume 3027 of Lecture Notes in Comput. Sci. Springer (2004) 342–360
39. Klivans, A., Sherstov, A.: Cryptographic hardness results for learning intersections of halfspaces (2006) Available as ECCC report TR06-057.