NEW TRENDS IN LATTICE-BASED CRYPTOGRAPHY

Adela Mihăţă and Emil Simion

Abstract. Lattice-based cryptography has undergone rapid development in recent years and is very attractive due to the low asymptotic complexity and potential resistence to quantum computing attacks. This paper is intended to be a survey of the lattice problems which underlay many recent cryptographic schems, offering also some information on the computational complexity aspects of lattice problems and their use in cryptography.

2000 *Mathematics Subject Classification*: 94A60, 03G10.

1. Introduction

A lattice is the set of integer linear combinations of n linearly independent vectors $\mathbf{b_1}, \ldots, \mathbf{b_n}$. Despite their aprarent simplicity, lattices hide a rich combinatorial structure which has attracted the attention of great mathematicians over the last two centuries. As a result, lattices have found many applications in mathematics and computer science, ranging from number theory to combinatorial optimization and cryptography.

A lattice may have several bases, some of which are better than the others. The goal of lattice reduction algorithms is to find good bases, i.e bases which short enough vectors which are almost orthogonal.

Traditionally, in cryptography, lattices have been used mostly as an algorithmic tool for cryptanalysis. Since the development of the LLL basis reduction algorithm of Lenstra, Lenstra and Lov*á*sz [11] in the early 80's, lattices have been used to attack a wide range of public key cryptosystems (knapsack based cryptographic systems and special cases of RSA). It seems [17] that the basis reduction algorithms which were used succesfully in breaking various cryptographic schems, have transformed the lattice reduction tehniques in one of the most popular tools in public-key cryptanalysis.

In the late 90's, the computational complexity of lattice problems attracted renewed attention, stimulated by Ajtai's surprising discovery [1] of a connection between the worst-case and average case complexity of certain lattice approximation problems. He suggested a completely different way of using lattices in cryptography by showing how to use computationally intractable lattice problems to *build* cryptosystems which are impossible to break. Namely, design cryptographic functions that are as hard to break as it is to solve a computationally hard lattice problem. Cryptography requires problems that are hard to solve *on the average*, so that when a cryptographic key is chosen at random, the corresponding function is hard to break with high probability. Ajtai's discovery shows that hard-on-average problems can be obtained from the waeker assumption that lattice problems are intractable in the *worst case.* To date, all known cryptographis functions rely on average-case complexity assumptions. This is why lattice problems are exceptional: they provide provably secure cryptographic functions from worst-case complexity assumptions.

The discovery of Ajtai attracted interest within the theoretical cryptography and computational complexity and stimulated substantial research efforts in this area. Following the discovery of Ajtai, research has progressed in the folowing directions:

- Determining weaker and weaker worst-case assumptions on the complexity of lattice problems that still allow to construct average-case hardness;
- Improving the efficiency of lattice based functions both in terms of key size and computation time;
- Building more complex cryptographic primitives than simple one-way functions, like public key encryption schemes.

Besides Ajtai's discovery, the highly importance use of lattices in cryptography also relays in some other non-negligeable reasons:

- Lattice related constructions and cryptographic functions are efficient and easy to implement, typically involving only simple arithmetic operations on small numbers; this can be advantageous in certain practical scenarios when encryption is performed by a low-cost device;
- Currently, we 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. Efficient quantum algorithms for factoring integers and computing discrete logarithms already exist [21]. Although large-scale quantum computers are not expected to exist for at least a decade, this fact should already be regarded as a warning.

2. Mathematical background on lattices

Definition 1. Let $B = \{b_1, \ldots, b_n\}$ be linearly independent vector in $\mathbf{R}^m$. The lattice generated by B is the set

$$\mathcal{L}(B) = \{\sum_{i=1}^{n} x_i \cdot b_i | x_i \in \mathbf{Z}\}.$$

The set of vectors B is called a *basis* for the lattice. All the bases have the same number dim(L) of elements, called the *dimension* of the lattice. Any lattice admits multiple bases, and this fact is at the heart of many cryptographic applications. Some of the bases are "better" than the others. Usually, we are interested in bases with shorter vectors which are almost ortogonal. In the lattice from the below picture, there is a lattice with two bases, one of which is better than the other. In this paper, we will use the Euclidean norm $||x|| = \sqrt{\sum_i x_i^2}$, but all problems can be defined with respect to any norm.

The definition of lattice can be also stated another way: a lattice is a discrete additive subgroup of $\mathbf{R}^m$. Therefore, not every subgroup of $\mathbf{R}^m$ is a lattice. The simplest example of lattice is the set of all n-dimensional vectors with integer entries.

The *minimum distance* of a lattice $\mathcal{L}$ denoted λ_1, is the minimum distance between any two distinct lattice points, and equals the length of the shortest nonzero lattice vector:

$$\lambda_1 = min\{||x - y|| : x \neq y \in \mathcal{L}\} = min\{||x|| : x \in \mathcal{L}\backslash\{0\}\}.$$

The main computational problems associated with lattices are:

- Shortest Vector Problem (SVP): Find the shortest nonzero vector in $\mathcal{L}$, i.e. find $0 \neq v \in \mathcal{L}$ such that $||v||$ is minimized;

- Closest Vector Problem (CVP): Given a vector w which is not in L, find the vector $v \in L$ closest to w, i.e., find $v \in L$ such that $||v - w||$ is minimized;

- Shortest Independent Vector Problem (SIVP): Find n linearly independent vectors with maximum length smaller than the maximum length of any base of the lattice, i.e. find $v_1, \cdots, v_n$ so that $max||v_i|| \leq max_B||b_i||$ for any base $B = \{b_1, \cdots, b_n\}$ of the lattice.

In lattice-based cryptography, one usually considers the approximation variant of these problems, which are denoted by an additional subscript γ indicating the approximation factor. For instance, in SVP_γ the goal is to find a vector whose norm is at most γ times larger than that of the shortest nonzero vector.

These problems appear to be very difficult as the dimension n becomes large. Solutions, or even partial solutions to these problems also turn out to have surprisingly many applications in a number of different fields. The CVP is known to be NP-hard and SVP is NP-hard under a certain "randomized reduction" hypothesis.

Also, SVP is NP-hard when the norm or distance used is the l^{∞} norm. In practice, a CVP can often be reduced to a SVP and is thought of as being "a little bit harder" than SVP. Reduction of CVP to SVP is used in [9] to prove that SVP is hard in Ajtai's probabilistic sense. In a real world scenario, a cryptosystem based on an NP-hard or NP-complete problem may use a particular subclass of that problem to achieve efficiency. It is possible that this subclass of problems could be easier to solve than the general problem.

3. Special classes of lattices and problems

Part of the difficulty of SVP comes from the fact that a lattice has many different bases and that usually, the given lattice basis contains very long vectors, much longer than the shortest nonzero vector. The well-known polynomial time algorithm of Lenstra, Lenstra, and Lov*á*sz (LLL) from 1982 achieves an approximation factor of $2^{O(n)}$ where n is the dimension of the lattice, while running in polynomial time. The algorithm works by applying succesive elementary transformations to the input basis in order to make its vectors shorter and more orthogonal. Currently, the best polynomial algorithm for lattice reduction achieves only a small improvement of the approximation factor to $2^{O(n \log\log n/\log n)}$. We should also mention that for an exact solution to SVP (or even just an approximation to within poly(n) factors), the best algorithm has a running time of $2^{O(n)}$.

In fact, there are two main techniques for lattice problems. The first, known as *lattice reduction*, started with the LLL algorithm mentioned above. A second technique is the *enumeration technique* which, in its simplest form, implies exhausitve search for the best integer combination of the basis vectors. The running time of this algorithm highly depends on the quality of the input basis. That's why this algorithm is applied to a basis only after it has been reduced, but not directly on the given basis. In a very recent work [4] the fundamental enumeration algorithms are exponentially speeded up, both in theory and practice, using a method called *extreme pruning.*

The complexity of lattice problems in the range of polynomial approximation factors is of particular interest. For example, the seminal work of Ajtai is based on the hardness of approximation in this region.

3.1. Lattices with special algebraic structure

The work of Micciancio [15] opened the door to the use of cyclic lattices as a new source of hardness assumptions, and motivates their study from a computational perspective.

Definition 2. For any vector $\mathbf{x} = \{\mathbf{x_1}, \cdots, \mathbf{x_n}\}$ define the cyclic rotation $rot(\mathbf{x}) = \{x_n, x_1, \cdots, x_{n-1}\}$. A lattice $\mathcal{L}(B)$ is *cyclic* if it is closed under the rotation operation, i.e. if $\mathbf{x} \in \mathcal{L}(B)$ implies $rot(\mathbf{x}) \in \mathcal{L}(B)$.

Very little is known about the computational complexity of lattice problems on cyclic lattices. From an algorithmic point of view, it is not clear how to exploit the cyclic structure of the lattice in lattice basis reduction algorithms. There are some papers [14] that show how the solution of certain lattice problems can be speeded up by a factor n when the lattice is cyclic of dimension n.

It is conjectured that approximation problems on cyclic lattices are computationally hard, at least in the worst case and for small polynomial approximation factors. The definition above of cyclic lattices is analogous to the definition of cyclic codes, one of the most useful and widely studied classes of codes in coding theory. Still, no polynomial time algorithm is known for many computational problems on cyclic codes (or lattices).

The first result which implies the use of cyclic lattices is proved by Micciancio [15], by showing that solving the generalized compact knapsack problem on the average is as hard as solving certain worst-case problems for cyclic lattices. This result yielded very efficient one-way functions whose security was based on worst-case hardness assumptions, functions that later were modified to be collision resistant under mostly the same assumptions but not only on cyclic lattices, but also on ideal lattices, a class of lattices that includes cyclic lattices as a special case.

Definition 3. An *ideal lattice* is an integer lattice $\mathcal{L}(B) \subseteq \mathbf{Z}^n$ such that $\mathcal{L}(B)) = \{g \bmod f | g \in I\}$ for some monic polynomial f of degree n and ideal $I \subseteq \mathbf{Z}[x]/f$.

It turns out that f should be irreducible. In other words, a lattice corresponding to an ideal means that the vector $(a_0, ..., a_{n-1})$ is in the lattice, if and only if the polynomial $a_0 + a_1x + ... + a_{n-1}x^{n-1}$ is in the ideal. Despite the added structure of these algebraic lattices, the best algorithms to solve the shortest vector problem are the same ones as for arbitrary lattices.

In 2008, Lyubashevski and Micciancio [12] give a direct construction of digital signatures based on the complexity of approximating the shortest vector in ideal lattices. The construction is provably secure based on the worst-case hardness of approximating the shortest vector in such lattices within a polynomial factor, and it is also efficient: the time complexity of the signing and verification algorithms, as well as key and signature size is almost linear (up to poly-logarithmic factors) in the dimension n of the underlying lattice. It seems that their construction is optimal performance/security trade-off since there is no sub-exponential (in n) time algorithm to solve lattice problems in the worst case, even when restricted to ideal lattices.

In 2009, Gentry [5] describes his innovative method for constructing fully homomorphic encryption schemes, the first credible solution to a very long-standing (30 years) major problem in cryptography and theoretical computer science at large. It is interesting that he uses ideal lattices which seem to be very well suited for the construction of his public key decryption scheme.

Another class of lattices with special algebraic structure is the class of convolution modular lattices. The ring-based cryptosystem NTRU [10] uses lattices that are similar to ideal lattices. Its security rests on the difficulty of solving CVP in these lattices. Even if the cryptosystem has no known security proof, it has resisted attacks. This is perhaps due to the inherent hardness of ring-based cryptographic constructions.

The convolution modular lattice L_h associated to the vector $\mathbf{h}$ and modulus q is the 2N dimensional lattice with basis given by the rows of the matrix:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{pmatrix}.$$

Another way to describe L_h is the set of vectors:

$$L_h = \{(\mathbf{a}, \mathbf{b}) \in \mathbf{Z}^{2N} | \mathbf{a} * \mathbf{h} \equiv \mathbf{b} \bmod q\}.$$

3.2. Learning With Errors problem

In recent years, the Learning with Errors (LWE) problem, introduced in [**?**], has turned out to be an a amazingly flexible basis for cryptographic constructions. It is very famous because of being as hard as worst-case lattice problems, hence rendering all cryptographic constructions based on it secure under the assumption that worst-case lattice problems are hard.

LWE. The LWE problem asks to recover a secret $\mathbf{s} \in \mathbf{Z}_q^n$, given a sequence of "approximate" random linear equations on $\mathbf{s}$. For instance, the input might be similar to [**?**]:

$$\begin{cases} 14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 \text{ (mod17)} \\ 13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 \text{ (mod17)} \\ 6s_1 + 10s_2 + 13s_3 + 1s_4 \approx 3 \text{ (mod17)} \\ 10s_1 + 4s_2 + 12s_3 + 16s_4 \approx 12 \text{ (mod17)} \\ 9s_1 + 5s_2 + 9s_3 + 6s_4 \approx 9 \text{ (mod17)} \\ 3s_1 + 6s_2 + 4s_3 + 5s_4 \approx 16 \text{ (mod17)} \\ 6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 \text{ (mod17)} \end{cases}$$

where each equation is correct up to some small additive error, our goal being to recover $\mathbf{s}$.

It can be easly seen that without the error, the problem of finding $\mathbf{s}$ would be easy: using about n equations, we can recover $\mathbf{s}$ in polynomial time with the Gauss elimination algorithm. The introduction of the error makes the problem significantly more difficult. The Gaussian elimination algorithm takes linear combination of n equations, amplifying the error to unmanageable levels.

Formally, the problem can be defined in the following manner: Fix a size parameter $n \geq 1$, a modulus $q \geq 2$, and an "error" probability distribution χ on $\mathbf{Z}_q$. Let $A_{s,\chi}$ on $\mathbf{Z}_q^n \times \mathbf{Z}_q$ be the probability distribution obtained by choosing a vector $\mathbf{a}$ uniformly at random, choosing $e \in \mathbf{Z}_q$ according to χ and outputting $(\mathbf{a}, \mathbf{a} * \mathbf{s} + e)$ where additions are made in $\mathbf{Z}_q$.We say that an algorithm solves LWE with modulus q and error distribution χif, for any $\mathbf{s} \in \mathbf{Z}_q^n$, given an arbitrary number of independent samples from $A_{s,\chi}$ it outputs s (with high probability). The special case $q = 2$ corresponds to the well-known *learning parity with noise* (LPN) problem.

Choosing parameters. In all applications, the error distribution χ is chosen to be a normal distribution rounded to the nearest integer (and reduced modulo q) of standard deviation αq where $\alpha > 0$ is typically taken to be 1/poly(n). The modulus q is typically taken to be polynomial in n. The number of equations seems to be, for most purposes, insignificant. For instance, the known hardness results are essentially independent of it. This can be partly explained by the fact that from a given fixed polynomial number of equations, one can generate an arbitrary number of additional equations that are almost as good as new, with only a slight worsening in the error distribution, property which was shown in [**?**].

Blum, Kalai, and Wasserman [**?**] provided the first subexponential algorithm for this problem. Their algorithm requires only $2^{O(n/\log n)}$ equations/time and is currently the best known algorithm for the problem. It is based on a clever idea that allows to find a small set S of equations (say, O($\sqrt{n}$)) among $2^{O(n/\log n)}$ equations, such that $\sum_S a_i$ is, say, (1, 0, $\cdots$, 0) which gives a guess for the first bit of $\mathbf{s}$ that is correct with probability that can be improved to a higher probability by repeating the whole procedure only $2^{O(\sqrt{n})}$ times. Their idea was later shown to have other important applications, such as the first $2^{O(n)}$- time algorithm for solving the shortest vector problem.

There are several reasons to believe the LWE problem is hard:

1. The first reason is that best known algorithms for LWE run in exponential time (and even quantum algorithms don't seem to do better);

2. The second reason is that LWE is a generalization of LPN, which, is an extensively studied problem in learning theory and widely believed to be hard (it is worth noting that LPN can be seen as a decoding from random linear bynary codes hence, any progress concerning LPN would be important also in the area of coding theory);

3. Third, because LWE is known to be hard based on certain assumptions regarding the worst-case quantum hardness of standard lattice problems such as GAPSVP (the decision version of the shortest vector problem) and SIVP (the shortest independent vectors problem) [**?**].

The LWE problem can be reduced to many, apparently easier, problems. These reductions are one of the main reason the LWE problem finds so many applications in cryptography. Among these reductions, a very famous one is a search to decision reduction, which tries to distinguish random linear equations, which have been perturbed by a small amount of noise, from truly uniform ones.

The LWE problem has a "dual" problem known as the SIS problem (which stands for Small Integer Solution). The SIS problem may be seen as a variant of subset-sum over a particular additive group. In more detail, let $n \geq 1$ be an integer dimension and $q \geq 2$ be an integer modulus; the problem is, given polynomially many random and independent $\mathbf{a_i} \in \mathbf{Z}_q^n$,to find a "small" integer combination of them that sums to $\mathbf{0} \in \mathbf{Z}_q^n$.

In recent years, a multitude of cryptographic schemes have been proposed around the SIS and LWE problems. As a search problem (without unique solution), SIS has been the foundation for one-way [**?**] and collision-resistant hash functions [**?**], digital signatures [**?**]. The LWE problem has proved to be amazingly good for encryption schemes, serving as the basis for secure public-key encryption under both chosen-plaintext and chosen-ciphertext attacks, identity-based encryption and many other applications.

One real problem of schemes based on the SIS and LWE problems is that they tend not to be efficient enough for practical applications. Even the simplest primitives, such as one-way functions, have key sizes at least quadratic in the primary security parameter, which needs to be quite high (several hundreds) for sufficient security against the best known attacks.

An important approach for avoiding the lack of efficiency is to use lattices that have some special algebraic structure. That's what Lyubashevsky, Peikert and Regev [?] did when adapted the LWE problem in the ring setting, called Ring-LWE. In LWE, getting just one extra random-looking number requires n random numbers. It is wishful to get n random numbers and produce $O(n)$ pseudo-random numbers in "one shot". This can be done by defining the multiplication in a polynomial ring $\mathbf{Z}_q[x]/(x^n+1)$. Mathematically speaking, the Ring-LWE replaces the group $\mathbf{Z}_q^n$ with the ring $\mathbf{Z}_q[x]/x^n + 1$.

The SIS problem can also be defined in a polynomial ring. Just like the hardness results for standard SIS, the result for Ring-SIS concerning hardness shows that solving the Ring-SIS problem implies a solution to worst-case instances of lattice problems. However, the worst-case lattice problems are restricted to the family of ideal lattices.

Obtaining analogous hardness results for Ring-LWE turned out to be quite non-trivial, and was only achieved very recently. There is a first result [?] which is based on a quantum reduction from Ring-SIS to Ring-LWE. The second one [?], follows the outline of the original LWE hardness proof of [?] and is independent of the number of samples.

The range of cryptographic applications of the LWE problem has by now become very wide. The security of these constructions is based on the difficulty of the LWE problem and the others derived from it, like Ring-LWE, SIS, Ring-SIS. While in LWE, the dimensions of the keys were quadratic in the primary security parameter, in Ring-LWE, the dimensions decrease to linear size.

The cryptography system introduced by Regev [?] is perhaps the most efficient lattice-based cryptosystem to date supported by a theoretical proof of security. Some very significant improvements to the cryptosystem in efficiency were given by Peikert et al. [?]

4. Lattice based cryptographic constructions

In 1996 - 1997, several cryptosystems were introduced, having as underlying hard problem was SVP or CVP in a lattice L of dimension n. This were: Ajtai-Dwork (ECCC report 1997), GGH presented at Crypto 1997 and NTRU, presented at the rump session of Crypto 1996.

The public key sizes associated to these cryptosystems were: $O(n^4)$ for Ajtai-Dwork, $O(n^2)$ for GGH, and $O(n \log n)$for NTRU.

The system proposed by Ajtai and Dwork was particularly interesting because it was provably secure unless a worst case lattice problem could be solved in polynomial time. Anyway, large key size was an impediment for practice. Subsequently, Nguyen and Stern showed, in fact, that any efficient implementation of the Ajtai-Dwork system was insecure.

The GGH cryptosystem, proposed by Goldreich, Goldwasser, and Halevi in [?] is essentially a lattice analogue of the McEliece cryptosystem proposed twenty years earlier based on the hardness of decoding linear codes over finite fields. The basic idea is very simple and appealing. At a high level, the GGH cryptosystem works as follows: the private key is a "good" lattice basis B (a good basis is a basis consisting of short, almost orthogonal vectors); the public key H is a "bad" basis for the same lattice. The encryption process consists of adding a short noise vector **r** (somehow encoding the message to be encrypted) to a properly chosen lattice point v. The decryption problem corresponds to finding the lattice point v closest to the target ciphertext $c = (r \bmod H) = v + \mathbf{r}$, and the associated error vector $r = c - v$.

The security relies on the assumption that without knowledge of a special basis (that is, given only the worst possible basis H), solving the instances of the closest vector problem in $L(B) = L(H)$ is computationally hard. There are known attacks that break the cryptosystem in practice for moderately large values of the security parameter, and that can be avoided by making the security parameter bigger but this makes the cryptosystem impractical.

The NTRU cryptosystem, which was already mentioned above, was described at the rump session of Crypto '96 as a ring based public key system that could be translated into an SVP problem in a special class of lattices. The hard problem underlying the NTRU public key cryptosystem is that of finding a very short vector in a lattice of very high dimension.

The National Institute of Standards & Technology (NIST) accredited NTRU with being the most practical lattice-based cryptographic solution that can resist a quantum computing attack. As usually happens with many cryptographic algorithms , a simply modification allows both encryption system (NTRUEncrypt) and digital signatures (NTRUSign). NTRUEncrypt is standardized since 2009 by IEEE as IEEE Std 1363.1-2008.

In 2010, a very appreciated paper [?] introduces a new lattice-based cryptographic structure called a *a bonsai tree* which has applications in some important open problems in the area. The first regards an efficient "hash-and sign" signature scheme in the standard model (without use of random oracles, and this is to be appeciated because the problem of random oracles is very controversial in cryptography), which has as underlying hard problem the aboved mentioed SIS problem. The second application is the first hierarchial identity-based encryption(HIBE) scheme that does not rely on bilinear pairings, with the hard underlying problem, the LWE.

5. Conclusions and open problems

There is still a lot of work to do in lattice-based cryptography and more work is still needed to increase confidence and understanding in this area, and in order to support widespread use of lattice-based cryptography.

The LWE-based cryptosystem proposed by Regev is reasonably efficient and has a security proof based on a worst-case connection. Still, one might hope to considerably improve the efficiency, and in particular the public key size, by using structured lattices such as cyclic lattices.

The lattices with special algebraic structure, as cyclic and ideal lattices need to be studied in more detail. It seems that they may offer good support for efficiency, but too much is not known until now about them.

Can one factor integers or compute discrete logarithms using an oracle that solves, say, $\sqrt{n}$-approximate SVP? Such a result would be useful to prove that the security of lattice-based cryptosystems is superior to that of traditional number-theoretic-based cryptosystems.

References

[1] M. Ajtai, *Generating hard instances of lattice problems*, Proc. of 28th ACM Symp. on Theory of Computing, 1996, pp. 99−108.

[2] A. Blum, A. Kalai and H. Wasserman, *Noise-tolerant learning, the parity problem, and the statistical query model*, Journal of the ACM, 50(4), 2003, pp. 506−519.

[3] D. Cash, D. Hofheinz, E. Klitz and C. Peikert, *Bonsai trees, or how to delegate a lattice basis*, EUROCRYPT 2010.

[4] N. Gama, P. Q. Nguyen and O. Regev, *Lattice enumeration using extreme pruning*, EUROCRYPT 2010.

[5] C. Gentry, *Fully homomorphic encryption using ideal lattices*, In: STOC, 2009, pp. 169−178.

[6] C. Gentry, C. Peikert and V. Vaikuntanathan, *Trapdoors for hard lattices and new cryptographic constructions*, Proc. 40th ACM Symp. on Theory of Computing (STOC), 2008, pp. 197−206.

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

[8] O. Goldreich, S. Goldwasser and S. Halevi, *Public-key cryptosystems from lattice reduction problems*, In Advances in cryptology, volume 1294 of Lecture Notes in Comput. Sci., Springer, 1997, pp. 112−131.

[9] O. Goldreich, D. Micciancio, S. Safra, J.P. Seifert, *Approximating shortest lattice vectors is not harder than approximating closest lattice vectors*, in Inform. Process. Lett. 71(2), 1999, pp. 55−61.

[10] J. Hoffstein , J. Pipher and J.H. Silverman, *NTRU: a ring-based public key cryptosystem*, In: Algorithmic number theory (ANTS), volume 1423 of Lecture Notes in Comput. Sci. Springe, 1998, pp. 267−288.

[11] A. K. Lenstra, H. W. Lenstra and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann., 261(4), 1982, pp. 515−534.

[12] V. Lyubashevsky, D. Micciancio , *Asymptotically efficient lattice-based digital signatures*, Canetti, R. (ed.) TCC 2008, vol. 4948, Springer, 2008, pp. 37−54.

[13] V. Lyubashevsky, C. Peikert and O. Regev, *On ideal lattices and learning with errors over rings*, Advances in Cryptology, EUROCRYPT 2010, Lecture Notes in Computer Science, 2010, Volume 6110/2010, pp. 1−23.

[14] A. May and J. H. Silverman. *Dimension reduction methods for convolution modular lattices.* In J. Silverman, editor, Cryptography and lattices conference−CaLC 2001, volume 2146 of Lecture Notes in Computer Science, Providence, RI, USA, Mar. 2001, Springer, pp. 110−125.

[15] D. Micciancio, *Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions*, Computational Complexity, Volume 16, Number 4, Springer, 2007, pp. 365−411.

[16] D. Micciancio, O. Regev, *Lattice-based cryptography*, Book chapter in Post-quantum Cryptography, D. J. Bernstein and J. Buchmann (eds.), Springer, 2008.

[17] P. Nguyen and J. Stern, *The two faces of lattices in cryptology*, In J. Silverman, editor, Cryptography and lattices conference CaLC 2001, volume 2146 of Lecture Notes in Computer Science, Springer, 2001, pp. 146−180.

[18] C. Peikert, V. Vaikuntanathan and B. Waters, *A framework for efficient and composable oblivious transfer*, In Advances in Cryptology (CRYPTO), LNCS. Springer, 2008.

[19] O. Regev, *On lattices, learning with errors, random linear codes and cryptography*, In Proc. 37th ACM Symp. on Theory of Computing, 2005, pp. 84−93.

[20] O. Regev, *The Learning with Errors Problem*, Invited survey in CCC 2010.

[21] P.W. Schor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. on Computing 26(5), 1997, pp. 1484−1509.

[22] D. Stehle, R. Steinfeld, K. Tanaka and K. Xagawa, *Efficient public key encryption based on ideal lattices*, In ASIACRYPT, 2009, pp. 617−635.

Adela Mihăiţă
Faculty of Mathematics and Computer Science, University of Bucharest,
Academiei Street, no.14, Romania.
email: *adela@fmi.unibuc.ro*

Emil Simion
"Simion Stoilow" Institute of Mathematics of the Romanian Academy (associate researcher)
P.O. Box 1-764, 014700 Bucharest, Romania.
email: *esimion@fmi.unibuc.ro*