Creating Cryptographic Challenges Using Multi-Party Computation: The LWE Challenge

Johannes Buchmann, Niklas Büscher[1], Florian Göpfert[1], Stefan Katzenbeisser[1], Juliane Krämer[1], Daniele Micciancio[1], Sander Siim[3], Christine van Vredendaal[4], and Michael Walter[2]

[1] Technische Universität Darmstadt, Germany
{buchmann@cdc,buescher@seceng,fgoepfert@cdc,katzenbeisser@seceng,
jkraemer@cdc}.informatik.tu-darmstadt.de
[2] University of California, San Diego, USA
{daniele, miwalter}@eng.ucsd.edu
[3] University of Tartu, Estonia & Cybernetica AS, Estonia
sander.siim@cyber.ee
[4] Technische Universiteit Eindhoven, Netherlands
c.v.vredendaal@tue.nl

Abstract. Practical hardness results are necessary to select parameters for cryptographic schemes. Cryptographic challenges proved to be useful for determining the practical hardness of computational problems that are used to build public-key cryptography. However, several of these problems have the drawback that it is not known how to create a challenge for them without knowing the solutions. Hence, for these problems the creators of the challenges are excluded from participating.

In this work, we present a method to create cryptographic challenges without excluding anyone from participating. This method is based on secure multi-party computation (MPC). We demonstrate that the MPC-based approach is indeed feasible by using it to build a challenge for the learning with errors (LWE) problem. The LWE problem is one of the most important problems in lattice-based cryptography. The security of many cryptographic schemes that have been proposed in the last decade is directly based on it. We identify parameters for LWE instances that provide the appropriate hardness level for a challenge while representing instances used to instantiate encryption schemes as close as possible. The LWE challenge is designed to determine the practical hardness of LWE, to gain an overview of the best known LWE solvers, and to motivate additional research effort in this direction.

Keywords lattices, learning with errors, LWE, secure multi-party computation, MPC

1 Introduction

The security of many cryptographic schemes is based on the hardness of a computational problem, such as the discrete logarithm problem. In order to be able to select parameters for these schemes that guarantee a chosen level of security, the hardness of such computational problems has to be estimated experimentally. Such experiments require a great deal of work since all possible attack algorithms have to be taken into account and implemented optimally. A method that proved to be useful to address this issue is to publish cryptographic challenges. Cryptographic challenges have been built for the factorization problem [3], for the elliptic curve discrete logarithm problem [1], for several lattice problems [20, 29], for the NTRU cryptosystem [2], and for multivariate cryptography [32]. Each challenge consists of a set of problem instances of varying hardness. Typically, a challenge is built as follows: the creator of the challenge generates an instance of the computational problem, and the challenge consists in computing a solution. However, for many computational problems it is hard to create a problem instance without knowing the solution. For a cryptographic challenge targeting such a problem, this means that the research community would have to trust the creator of the challenge to keep the secret and not to reveal any information about the solution to anyone. More important — the creator himself would not be able to participate in the challenge. Real examples of such problems and challenges, respectively, are the factorization problem and the multivariate quadratic problem [32].

In this work, we show that it is possible to create all cryptographic challenges without relying on the trustworthiness of a single party. We achieve this by using secure multi-party computation (MPC). MPC is a a subfield of cryptography that allows several parties to jointly compute a function without revealing anything about the inputs of the other parties. Despite big theoretical progress in recent years, real-world applications for MPC are still surprisingly rare. The main reason for this is the rather significant computational overhead that comes with most MPC solutions. We show that MPC is in fact efficient enough to securely create cryptographic challenges. To this end, we create a new cryptographic challenge: the LWE challenge[1][2].

The learning with errors (LWE) problem is an important problem in lattice-based cryptography. On the one hand, it is of high theoretic interest since it allows the construction of many highly sophisticated primitives like fully-homomorphic encryption [19] and group signatures [25]. On the other hand, many practical, more basic primitives like public-key encryption [26] or signature schemes [23] base their security on the hardness of LWE. In order to select concrete parameters for the practical schemes, it is important to understand the concrete hardness of the underlying LWE instances.

In order to allow for a meaningful conclusion about the hardness of the schemes, care must be taken to provide useful problem instances in a cryp-

[1] `https://www.latticechallenge.org/lwe_challenge`

[2] We are currently in the process of creating the instances. At the moment, the page only contains “dummy instances” that were created on a single machine and will be replaced as soon as the MPC generation is finished. Consequently, the page is at the moment password-protected (login name: reviewer, password: lwe_challenge) and will be made public after acceptance.

tographic challenge. For LWE, which is instantiated by several parameters, we carefully determined the parameters that determine the hardness of the instance, and considered their relation to each other for several practical cryptographic schemes. Afterwards, we investigated the capability of the best known attacks on LWE. That way, we identified LWE instances satisfying two properties at the same time:

1. the relation of the parameters to each other is as in the investigated cryptographic schemes, and
2. they lie on the border between barely breakable and unbreakable instances.

Thus, the future development of the challenge allows to draw conclusions about the advance of the analyses of LWE and the concrete complexity of LWE.

Our Techniques Instead of only by a single person, each instance of the challenge is built jointly by three parties[3]. We show that none of these parties (and no one else) has more information about the secret than what is actually given in the challenge, assuming

1. honesty of the majority of participants,
2. an MPC protocol that provides information-theoretic security in the semi-honest adversarial model,
3. collision resistance of a standard cryptographic hash function, and
4. security of a standard pseudorandom generator.

Information-theoretic security of the MPC protocol is needed to ensure that no creating party can gain any advantage by breaking a possibly ‘weaker’ cryptographic primitive used in the MPC protocol. We show that for cryptographic challenges it is sufficient to rely on MPC protocols that are secure against semi-honest (passive) adversaries (instead of using a significantly more expensive MPC protocol that is secure in the malicious adversarial model), by verifying honest behaviour once a challenge is solved.

Contribution The contribution of this work is threefold:

1. we introduce a method based on multi-party computation that allows to create cryptographic challenges whithout relying on the trustworthiness of a single party or excluding anyone from participating,
2. we show that this method is practical by creating a challenge for the LWE problem that will stimulate further research on LWE solvers, and
3. to evaluate experimentally the hardness of LWE, we choose parameters for instances that make the results of the LWE challenge useful for further use: the relation between the parameters is as suggested for LWE-based encryption schemes and at the same time the parameters are at the limits of current LWE solvers.

[3] The affiliations of the parties wil be published after acceptance.

Organization In the next section, we provide the basics of multi-party computation and introduce the LWE problem. In Section 3, we describe the protocol that we used to build the instances for the LWE challenge and that will be used to check the submitted solutions. In Section 4, we describe the LWE challenge in more detail: we explain how the parameters of the instances are chosen and why their solutions are unique. In Section 5, we give more details about the web page and show how to participate.

2 Background

In this section, we provide the basics of secure multi-party computation and give some background information on the LWE problem.

2.1 Secure Multi-Party Computation

In MPC, parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ want to securely compute a function f on their joint secret inputs $x_1, \ldots, x_n$ to receive $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$, without any $\mathcal{P}_i$ learning anything about the inputs or outputs of other parties besides what can be deduced from x_i and y_i.

The two main security models for MPC are the *semi-honest (passive)* and *malicious (active)* model. The semi-honest model provides security against an adversary that exactly follows the protocol description, but tries to extract information about other parties' inputs or outputs from all messages he sees during the protocol run. The malicious model considers a stronger adversary that may arbitrarily deviate from the protocol description to learn secret information or affect the outcome of the computation.

2.2 Learning with Errors

Throughout this paper, we denote vectors with bold, lowercase letters (e.g., $\mathbf{s} \in \mathbb{Z}_q^n$), and matrices with bold, capital letters (e.g., $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$). Abusing notation, we identify $\mathbb{Z}_q$ with the set $[-\frac{q}{2}, \frac{q}{2}[\cap \mathbb{Z}$. This leads directly to a natural definition of the norm of a vector in $\mathbb{Z}_q^m$.

An instance of the learning with errors problem for natural numbers n, m, q, and an error distribution χ on $\mathbb{Z}_q$ is created as follows: first, a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, a uniformly random vector $\mathbf{s} \in \mathbb{Z}_q^n$, and an error vector $\mathbf{e} \leftarrow \chi^m$ are sampled. Then, $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$ is calculated. The LWE problem is to recover $\mathbf{s}$, given $\mathbf{A}$ and $\mathbf{b}$. [4]

Typically, χ is a discrete (or discretized) Gaussian distribution over the integers with parameter $\sigma \in \mathbb{R}^+$. It is defined as the distribution D_σ that samples every integer $x \in \mathbb{Z}$ with probability proportional to $\exp(-1/2 \cdot |x|^2/\sigma^2)$. Similarly,

[4] In the original definition of LWE, an attacker has access to arbitrarily many LWE samples. The LWE challenge, however, is designed to cover practical applications of LWE. There, the number of samples is usually limited.

the discrete Gaussian distribution over $\mathbb{Z}_q$ ($D_{\mathbb{Z}_q,\sigma}$) is defined as the distribution that samples every $x \in \mathbb{Z}_q$ with probability proportional to $\sum_{y\in\mathbb{Z}} \exp(-1/2 \cdot |x' + yq|^2/\sigma^2)$, where $x' \in \mathbb{Z}$ is an arbitrary representative of $x \in \mathbb{Z}_q$. Note that the sum is well-defined (i.e., it is independent of the choice of x') and its value is finite.

In his original work on LWE [31], Regev used the relative error rate $\alpha := \sigma/q$ to define the Gaussian distribution. In fact, it seems to be more natural to define the hardness of LWE depending on α. The main reason for this is modulus switching [18], a technique that allows to change the modulus q. In this process, α remains constant except for a rather small factor (see Section 4.1).

2.3 Attacks on LWE

In the following, we shortly sketch the most important attacks on the LWE problem. Knowing the attacks is crucial to determine the instances that are provided by the challenge.

Decoding attack For any LWE instance $(\mathbf{A}, \mathbf{b})$ with $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, the vector $\mathbf{b}$ is a linear combination of the columns of $\mathbf{A}$, disturbed by the error vector $\mathbf{e}$. Consider the lattice

$$\Lambda_q(\mathbf{A}) := \{\mathbf{w} \in \mathbb{Z}^m \mid \exists \mathbf{v} \in \mathbb{Z}_q^n : \mathbf{A}\mathbf{v} = \mathbf{w} \mod q\}.$$

Since the error vector is Gaussian distributed, there is a lattice vector $\mathbf{w}$ such that the distance between $\mathbf{b}$ and $\mathbf{w}$ is bounded by $\sqrt{m} \cdot 2 \cdot \sigma$ with overwhelming probability (see Equation (1) in Section 4.2). Hence, the LWE problem can be seen as a bounded distance decoding problem (BDD) in the lattice $\Lambda_q(\mathbf{A})$. The most standard approach for solving BDD is Babai's nearest plane algorithm [10] and its improvements by Lindner and Peikert [26] and Liu and Nguyen [27]. The idea in all those approaches is to enumerate all lattice vectors in a certain search-rectangle centered around the target vector, hoping that the lattice vector $\mathbf{A}\mathbf{s} \bmod q$ is among them.

Reduction to SVP A different lattice-based approach is to construct a lattice that contains the error vector $\mathbf{e}$. This can be done by adding $\mathbf{b}$ to the lattice $\Lambda_q(\mathbf{A})$ [6], but other approaches are possible as well [11,13]. Since $\mathbf{e}$ is short, it is typically the shortest vector in the lattice and can be found with basis reduction.

BKW Another approach to solve LWE is based on the work from Blum, Kalai, and Wasserman [14], which was originally developed to solve the learning parity with noise problem. The main idea is to combine few LWE samples to get a new sample that depends only on a small part of the secret $\mathbf{s}$. This part can then be recovered by brute-forcing all possibilities [5], or by advanced techniques like the multidimensional Fourier transform [22].

Arora-Ge Arora and Ge [8] introduced a new method to solve LWE by reducing it to a set of noise-free non-linear equations. These equations are then solved by linearization techniques. Recently, Albrecht et al. [4] showed how to apply Gröbner basis techniques to solve the equation system. The main advantage of the latter approach is that it requires significantly less LWE samples.

3 Creating LWE Instances with MPC

In this section, we first describe a general approach to create LWE challenges with the help of MPC without leaking any information that could lead to a computational advantage for the creating parties. However, due to the lack of an efficient MPC protocol that is secure in the malicious adversarial model, we then present a novel approach that only requires a semi-honest MPC protocol but still guarantees to detect malicious behavior in a verification step. Finally, we present technical insights into the actual implementation.

3.1 Secure Challenge Creation

Correctness and privacy are the basic properties that any secure MPC protocol fulfills. Privacy ensures that the parties running the protocol do not learn anything about the other parties' inputs or intermediate results except for the information that they can derive from the output of the functionality and their own input.

Thus, to securely generate an LWE challenge, the following functionality needs to be implemented for use in an MPC protocol: as public inputs the parties provide the challenge parameters, as private inputs the parties provide randomness. Then, the matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and the vectors $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \chi^m$ have to be sampled securely before the actual challenge vector $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$ can be computed. The intermediate values $\mathbf{s}$ and $\mathbf{e}$ are never visible to any computing party. At the end of the protocol, the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are revealed to the parties.

3.2 Efficient Creation through Verification

Given any MPC protocol that is information theoretically secure against malicious adversaries, LWE challenges can be created following the approach described in the previous section. However, achieving security in the malicious adversary model is significantly more costly for all currently known practical MPC protocols than achieving security in the semi-honest model alone. Even worse and to the best of our knowledge, none of the existing available MPC frameworks that provide security against malicious adversaries are capable of computing larger LWE challenges under reasonable time constraints. To overcome this situation, we show how to extend a three-party protocol[5] that is secure in the semi-honest

[5] Three parties is the most efficient case, but more parties are also imaginable.

model with an efficient verification step to check the correct behavior of all parties in hindsight. This approach does not protect against malicious behavior, i.e., the generating parties can still manipulate the challenge creation. However, by guaranteeing that malicious behavior will be detected, the participating parties risk loosing their reputation when being caught cheating. That way, none of the participating parties has an interest in manipulating the challenge instances.

The security model that we apply in this work is similar to the covert adversarial model by Aumann and Lindell [9]. However, instead of requiring to detect malicious behavior during the protocol execution with some probability $\epsilon > 0$, we extend the semi-honest model by requiring to detect malicious behavior only after the computation, but then with a probability of $\epsilon = 1$. Our protocol is secure, i.e., cheating is detected, if the adversary corrupts up to one out of three parties.

In the following paragraphs, we show that by using a self-synchronizing three-party MPC protocol, we detect any deviation from an honest behavior. A protocol is called self-synchronizing if the adversary cannot force (semi-)honest participants to start a new communication round until all other participants have completed the previous round. The core idea of our approach is to commit to all randomness used for the MPC protocol before the protocol starts. A fixed randomness guarantees a deterministic communication (content and order of messages) between all parties. Giving the same randomness to a protocol simulator, a transcript between three honest parties can be simulated. By comparing this honest transcript with the transcript of the actual protocol execution, any deviation from an honest behavior is inevitably detected.

Two protocols are required for the verified creation of LWE challenges. Protocol 1 (CREATE_CHALLENGE) is run by the parties that generate the challenges. Protocol 2 (CHECK_CHALLENGE) can be run by anyone to verify that the challenges have been created in an honest manner.

CREATE_CHALLENGE extends the used MPC protocol to allow a later verification. In a first step, every party uniformly samples two seeds $\mathbf{y}'_i$, $\mathbf{y}''_i$ (Line 3) that have a length κ, which denotes the bit length of the used randomness. In a practical implementation κ is the length of the seed used for a secure PRNG Then all parties commit to their seeds (Line 4), to ensure that these have been chosen independently from the other parties. Afterwards, the seed $\mathbf{y}''_i$ is published and sent to the other parties. The final seed $\mathbf{y}_i$, used as the actual source of randomness in the MPC protocol, is computed by xor-ing all public seeds $\mathbf{y}''_1$, $\mathbf{y}''_2$ and $\mathbf{y}''_3$ with the private seed $\mathbf{y}'_i$ (Line 9). This guarantees a private and uniformly distributed seed for every party, if at least one party behaves honestly. Consequently, the protocol cannot be influenced through a cleverly chosen seed. After the successful execution of the MPC protocol (Line 10), the parties publish the generated instances and publish hashes of the communication with the other parties (Line 11).

Once a correct solution $\mathbf{s}$ has been submitted, the general public can verify the challenge creation process with the help of CHECK_CHALLENGE. The generating parties are asked to reveal the seeds that they used for the challenge creation.

```
Protocol 1: CREATE_CHALLENGE
   Input   : · LWE parameters $m, n, q, \sigma$
             · security parameter $\kappa$
   Output  : · an LWE instance $\mathbf{A}, \mathbf{b} \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$
             · commitment to private seeds $h_{y'_i}, h_{y''_i}$, seed $\mathbf{y}''_i$
             · hashed comm. transcripts with other parties: $h_{\mathrm{com},i,j}$
 1 begin
 2     every party $i \in \{1, 2, 3\}$:
 3         sample random seeds $\mathbf{y}'_i, \mathbf{y}''_i \xleftarrow{\$} \{0,1\}^k$
 4         publish commitments $h_{y'_i} = \mathrm{commit}(\mathbf{y}'_i)$ and $h_{y''_i} = \mathrm{commit}(\mathbf{y}''_i)$
 5         wait until all other parties published their commitments
 6         publish $\mathbf{y}''_i$
 7         wait until all other parties published their seeds
 8         verify seeds of other parties, abort if exists $j$: $h_{y''_j} \neq \mathrm{commit}(\mathbf{y}''_j)$
 9         compute seed $\mathbf{y}_i = \mathbf{y}'_i \oplus \mathbf{y}''_1 \oplus \mathbf{y}''_2 \oplus \mathbf{y}''_3$
10         jointly run MPC protocol seeded with $\mathbf{y}_i$
11         publish $\mathbf{A}, \mathbf{b}$, hashes of the communication with other parties: $h_{\mathrm{com},i,j}$
12 end
```

Keeping these a secret rules the challenge as invalid. After verifying the seeds with the published commitments (Line 2), the MPC protocol can be simulated (Line 4). Abnormal behavior is detected by comparing the hashes of the original transcripts with the transcripts of the simulated protocol (Line 6).

To actively manipulate an MPC protocol, modified messages have to be sent by a malicious party, which will be reflected in the hashed transcript observed by either of the honest parties. Consequently, the two protocols guarantee that malicious behavior of a single corrupted party will be caught.

3.3 Implementation

Our implementation uses the information-theoretically secure MPC protocol suite of the Sharemind platform [17]. Sharemind has been used in many prototype applications and real-world MPC deployments, processing people's personal data in a privacy-preserving manner [15,24]. As such, the framework's code base has been audited and seen extensive optimization and testing, which leads to an efficient solution and significantly reduces the chance of implementation failures. Sharemind's additive secret sharing based MPC protocol suite provides information-theoretical security in the semi-honest model when a majority of participating parties are honest. The primitive protocols are provably universally composable and can therefore be combined to build larger computations securely [16].

Deployment and implementation details The LWE instances are jointly created by three parties, whose names will be published after acceptance. Each party

```
Protocol 2: CHECK_CHALLENGE
Input   : · LWE instance $\mathbf{A} \in \mathbb{Z}_q^{m \times n}, \mathbf{b} \in \mathbb{Z}_q^m$
          · hashes of transcript with other parties $h_{\mathrm{com,i,j}}$
          · seeds $\mathbf{y}'_i, \mathbf{y}''_i$ and their commitments $h_{y'_i}, h_{y''_i}$,
Output  : · TRUE (if the challenge was created correctly) or abort
begin
    verify the commitments $h_{y'_j}$, $h_{y''_j}$ of every party $j \in \{1, 2, 3\}$
    compute the seed $\mathbf{y}_j = \mathbf{y}'_j \oplus \mathbf{y}''_1 \oplus \mathbf{y}''_2 \oplus \mathbf{y}''_3$ of every party $j \in \{1, 2, 3\}$
    simulate MPC protocol on seeds $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$
    compute hashes of transcripts: $h_{\mathrm{com',i,j}}$
    pairwise verify the transcripts' hashes, abort if exists $i, j$: $h_{\mathrm{com,i,j}} \neq h_{\mathrm{com',i,j}}$
    return TRUE;
end
```

runs the protocol on its own hardware. The protocol is highly interactive and demands all parties to be online during the challenge generation. The generation is centrally initiated and controlled by one of the parties. The challenges, commitments and hashes of the transcripts are published by each party on the challenge web site for universal verification. Being public, all these are visible to all parties to validate their correctness at any point in time. The MPC code (SecreC) and client application code (C++ and Python) will be made available for public review on the challenge website. For the verification algorithm, we extended the protocol software by functionalities to export the seeds used for the PRNG and all sent and received messages. Moreover, our implementation also allows to induce a seed to simulate the MPC protocol for verification purposes. Thus, the verification can be done efficiently and without the development of a protocol simulator by locally running three application servers on the same machine.

Runtime and Communication Costs Although highly optimized, the generation of LWE instances is a time-consuming computational task. We observed creation times up to 10 hours for the largest LWE instances, whereas the smallest ones can be created in a few minutes. Even though, having the same computational workload, the verification step is executed in a fraction of that time, when run on a single machine. This is because communication is the dominating cost factor, especially when operating in an intercontinental setup with high latency and varying bandwidth. Detailed timing results will be given in the final paper.

4 LWE Instances Provided by the LWE Challenge

In order to allow meaningful conclusions about the hardness of LWE, the hardness of the provided problem instances is crucial. On the one hand, if the provided instances are too easy, the attacks that are easier to implement would dominate the hall of fame, and not the attacks with the best runtime. On the other hand, if the instances are too hard, none of them would get broken, and the challenge would fail to provide useful information about the practical hardness of LWE.

In this section, we first explain how we chose the parameters of the instances, i.e., n, m, q, and α. Then, we show that every instance has (with high probability) a unique solution.

4.1 Choice of Parameters

LWE is parametrized by several different parameters. On the one hand, this is a big advantage: it allows to generate instances that are crafted specifically for a certain application, which leads to more efficient schemes. An example for this are the different moduli used in signature schemes: while there are examples of secure schemes with a modulus length of about 14 bits (see [23]), other techniques require the modulus to be about 30 bits long or even longer (see [11]). On the other hand, the flexibility of LWE makes estimating its hardness much more complicated, since there is not one best algorithm for all instances [7]. This makes the selection of the right instances provided by the challenge a non-trivial task.

Fortunately, both theoretical and experimental results show that the hardness of LWE mainly depends on the secret dimension n, and the error parameter α (see [7,12,18,30]). The number of samples m and the modulus q appear to play a minor role. In the following, we explain our parameter choices in detail.

Known attacks on LWE can be roughly divided in two classes: lattice-based attacks (like the distinguishing attack [26], the decoding attack [10,26,27] or the embedding approach [6]) work with few samples, while other approaches (like BKW [5] or the Arora-Ge algorithm [8]) often require subexponentially many (or even more). In theory, this is not a big issue, since an attacker has access to arbitrarily many samples in the original definition of LWE. However, nearly all practical applications only provide a limited number of samples (e.g., [11,23,26]). Consequently, we consider modifying the latter attacks to run with less samples an important challenge. In fact, progress in this direction by lowering the required number of samples [22] or generating new samples [28] shows that it may be possible to overcome this problem.

To motivate further progress, the LWE challenge provides only $m = n^2$ samples per instance. This is enough to run all sample-efficient solvers, but should exclude the sample-consuming ones. Besides motivating further research for sample-efficient (and therefore realistic) attacks, this is leads to a more realistic picture about the limitations of current attacks in practice.

While being important for the correctness and the efficiency of many schemes, the modulus q appears to play a minor role for the hardness of LWE. Following the original proposal by Regev [30], the LWE challenge is restricted to instances with q being the smallest prime that is bigger than n^2. Prime numbers are the most frequently choice in practical schemes (e.g., [11,23,26,30]). Fortunately, it is not necessary to include other values for q (like powers of 2). The reason for this is a technique introduced by Brakerski and Vaikuntanathan [19] called modulus switching. It allows an attacker to transfer an LWE instance with modulus q to an LWE instance with an arbitrary different modulus q' with the same secret $\mathbf{s}$.

The next choice concerns the size of the error. In the literature, the standard deviation of the gaussian error is either given by the standard deviation σ, or by the relative error size $\alpha = \sigma/q$. Following Regev's original proposal, we select α as error size parameter instead of σ. This choice is supported by modulus switching: When switching the modulus q to q', the relative error rate α remains constant except for a small factor, which shows that the hardness of LWE depends on α rather than on σ.

The last choice concerns concrete values for n and α. In the first cryptographic application of LWE [30], Oded Regev proposed to to choose $\alpha = o(\frac{1}{\sqrt{n}\log(n)})$. For the instances provided by the challenge, the lattice dimension n ranges from 40 to 120, and the relative error size α ranges from 0.005 to 0.070. They are chosen such that they capture the proportion of n and α proposed by Regev (see Figure 1).

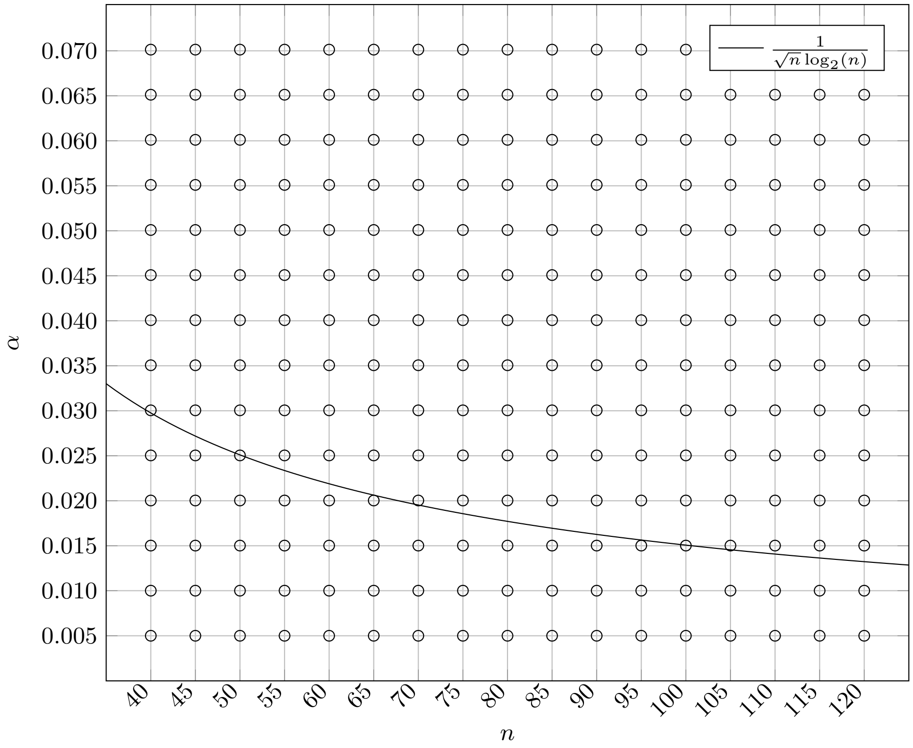


Fig. 1. Values for the error rate α and the dimension n as proposed by Regev (solid line) and provided by the LWE challenge (small circles)

At the same time, the hardness of the instances lies in a reasonable range, i.e., the easiest instances can be solved fairly easy using standard techniques, while the hardest challenges are likely to remain unsolved for at least several years. The hardness estimates are based on the simulator by Albrecht et al. [7], that estimates the runtime of the known attacks on a given LWE instance. On the one hand, the applicability of this simulator should be taken with a grain of salt, since it was crafted for LWE instances with higher hardness levels (like the instances proposed for cryptographic applications). On the other hand, it should at least give an idea of the hardness of the instances, and comparing the performance of the attacks in practice to the values predicted by theory is an interesting future work made possible by the challenge. Additionally, we ran attacks on LWE instances with the easiest parameter sets to confirm that they are breakable within a reasonable time. The challenge was build such that more instances can be included once our estimations prove wrong or better algorithms which significantly decrease the hardness of LWE are developed.

4.2 Uniqueness and Correctness of Solutions

LWE is typically instantiated such that the solution is unique. This sounds surprising at first glance because for every $\mathbf{s} \in \mathbb{Z}_q^n$, there is an error vector $\mathbf{e} \in \mathbb{Z}_q^m$ such that $\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{b}$. Since there is no bound for values sampled according to a Gaussian distribution, each $\mathbf{s}$ could be the secret. However, for a typical instantiation of LWE, there is only one vector $\mathbf{s}$ that leads to a reasonable error $\mathbf{e}$, by which we mean that all other errors are much bigger and therefore only sampled with a negligible probability.

For the LWE challenge, uniqueness is a little bit easier to define. The challenge accepts a submission $\mathbf{s}$ if (and only if) the corresponding error $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{s}$ satisfies $\|\mathbf{e}\| \leq 2\sqrt{m}\sigma$ with $\sigma = \alpha q$. This is justified by the fact that Lemma 2.2 in [23] by Ducas et al. bounds the size of a Gaussian distributed vector as

$$\Pr[\|\mathbf{e}\| > 2\sqrt{m}\sigma; \mathbf{e} \xleftarrow{\$} D_{\mathbb{Z}^m,\sigma}] < 2\left(2\exp(-3/2)\right)^m < 2^{-m+1}. \tag{1}$$

Note that this probability is extremely small for our values of m ranging from 1600 to 14400. Consequently, correct solutions get accepted with overwhelming probability.

In the following, we show that all challenges have (with high probability) one unique solution. For an arbitrary lattice $\Lambda \subset \mathbb{Z}^m$, let $\lambda_1(\Lambda)$ be the norm of the shortest non-zero vector in Λ. To see why the solutions are unique, imagine two secret-error tuples satisfying

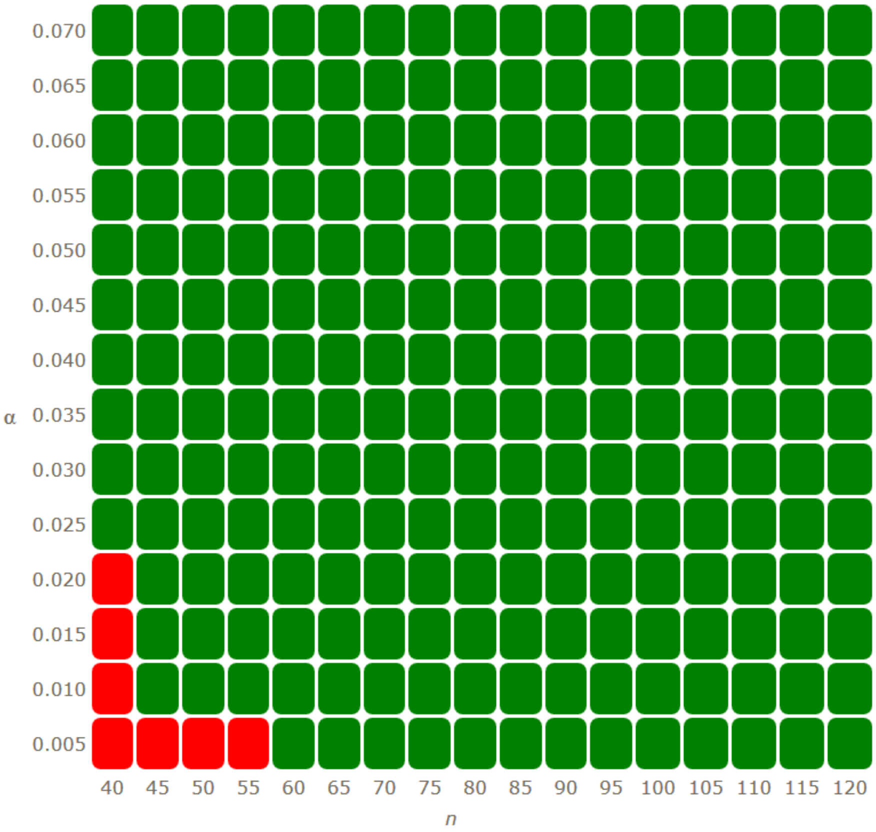
$$\mathbf{A}\mathbf{s}_1 + \mathbf{e}_1 = \mathbf{b} = \mathbf{A}\mathbf{s}_2 + \mathbf{e}_2 \mod q$$

and $\|\mathbf{e}_i\| \leq 2\sqrt{m}\sigma$. The triangle inequality immediately leads to

$$\|\mathbf{A}(\mathbf{s}_1 - \mathbf{s}_2)\| \leq 4\sqrt{m}\sigma,$$

which shows that the lattice

$$\Lambda_q(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m \mid \exists \mathbf{w} \in \mathbb{Z}^n : \mathbf{A}\mathbf{w} = \mathbf{v} \mod q\}$$



α \ n	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120
0.070	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.065	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.060	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.055	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.050	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.045	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.040	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.035	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.030	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.025	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.020	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.015	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.010	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
0.005	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Fig. 2. The LWE challenge table. The lattice dimension n is shown on the x-axis, the relative error size α is shown on the y-axis. Green points stand for unsolved instances, while red points indicate that the respective instance has already been solved. By clicking on one of the points, the respective challenge can be downloaded. By clicking on a green point, additionally a solution to the challenge can be submitted.

contains two points $\mathbf{v}_1, \mathbf{v}_2$ satisfying $\|\mathbf{v}_1 - \mathbf{v}_2\| \leq 4\sqrt{m}\sigma$. Consequently, the existence of two LWE solutions would imply

$$\lambda_1(\Lambda_q(\mathbf{A})) \leq 4\sqrt{m}\sigma. \tag{2}$$

The following Lemma bounds the probability that such an extraordinary short lattice vector exists.

Lemma 1. *Let $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix. If $q^{1-\frac{n}{m}} \geq 1250$ then*

$$\Pr\left[\lambda_1(\Lambda_q(\boldsymbol{A})) \leq \frac{1}{5}\sqrt{m}q^{1-\frac{n}{m}}\right] \leq 0.9^{-m} + q^{n-m}.$$

Proof. The idea of the proof is to consider the probability of a random integer vector being in the lattice and take a union bound over all short vectors. Assume we have $\det(\Lambda_q(\mathbf{A})) = q^{m-n}$ (which happens with probability at least $1-q^{n-m}$). The probability of a random integer vector to be in the lattice is q^{n-m}. The tricky part is to bound the number of integer vectors inside the ball of radius $1/5\sqrt{m}q^{1-\frac{n}{m}}$. Note that using cubes of edge length 1 centered at each integer vector, one can tile the entire space. However, not all cubes centered at integer points inside the ball are completely enclosed by the ball. So we consider a second ball with radius larger than the first one by an addititive factor of $\frac{1}{2}\sqrt{m}$ to ensure that all cubes centered inside the original ball are entirely enclosed in the second ball. This allows us to bound the number of integer points inside the first ball by the volume of the second ball. Note that we can obtain the necessary extension of the radius by multiplying the radius of the first ball with 1.002 due to the condition in the lemma. So by the union bound and Stirling approximation of the Gamma function we have

$$\begin{aligned}
&\Pr\left[\lambda_1(\Lambda_q(\mathbf{A})) \leq \frac{1}{5}\sqrt{m}q^{1-\frac{n}{m}} \,|\, \det(\Lambda_q(\mathbf{A})) = q^{m-n}\right] \\
&\leq \frac{\pi^{m/2}}{\Gamma(\frac{m}{2}+1)}\left(0.2004\sqrt{m}q^{\frac{m-n}{m}}\right)^m q^{n-m} \\
&\leq \Gamma\left(\frac{m}{2}+1\right)^{-1}\left(0.2004^2\pi m\right)^{m/2} \\
&\leq \left(\frac{0.2004^2\pi m}{m/2+1}\right)^{m/2} \\
&\leq (\sqrt{2\pi e}0.2004)^m \\
&\leq 0.9^m.
\end{aligned}$$

□

Corollary 1. *Let $\boldsymbol{A} \in \mathbb{Z}_q^{m\times n}, \boldsymbol{b} \in \mathbb{Z}_q^m$ be an LWE instance with parameters $n, q, \alpha,$ and m. If $m = n^2$, $q^{1-\frac{n}{m}} \geq 1250$, and*

$$\alpha\sqrt[n]{q} < 1/20, \tag{3}$$

the probability that two different vectors $\boldsymbol{s}_1 \in \mathbb{Z}_q^n, \boldsymbol{s}_2 \in \mathbb{Z}_q^n$ satisfy

$$\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{s}_i\| \leq \sqrt{m}\alpha q$$

is bounded by $0.9^{-n^2} + q^{n-n^2}$.

Proof. Follows directly from Equation (2), Lemma 1, and the easy calculation

$$\begin{aligned}
& 4\sqrt{m}\sigma < 1/5\sqrt{m}\cdot q^{1-n/m} \\
\Leftrightarrow\quad & 20\alpha q < q^{1-1/n} \\
\Leftrightarrow\quad & \alpha\sqrt[n]{q} < 1/20.
\end{aligned}$$

While all proposed instances meet the condition in Lemma 1, not all satisfy Equation (3). However, this does not mean that the solutions of the other challenges are not unique. To the contrary, the Gaussian heuristic strongly indicates that all solutions are unique: it estimates the length of the shortest non-zero vector in the above lattice to be

$$\lambda_1(\Lambda_q(\mathbf{A})) \approx \frac{\Gamma(1+m/2)^{1/m}}{\sqrt{\pi}} q^{1-n/m}.$$

Consequently, for all our instances, two valid solutions would imply a lattice vector shorter than 0.7 times the Gaussian heuristic. However, the existence of such a short vector is very unlikely. This is, among others, confirmed by the results of the SVP challenge: despite big efforts by many researchers, no one was able to find a lattice vector shorter than 0.8 times the prediction of the Gaussian heuristic so far.

5 The LWE Challenge

In this section, we explain the challenge web page in more detail and show how one can participate in the LWE challenge.

5.1 How to Download Challenges

The LWE challenge website provides a challenge table that is shown in Figure 2. This table contains all available instances, ordered by lattice dimension n and relative error rate α. The green points of the challenge table stand for unsolved instances. By clicking on a green point, the respective instance can be downloaded.

On the day of the release of the LWE challenge website, all points of the challenge table will be green. Once an instance is solved, i.e., the correct solution has been submitted, the respective point turns red. Hence, while the challenge table provides the challenge instances, it also serves as a visual representation of the development of the LWE challenge and hence, of the LWE problem.

In addition to using the challenge table, LWE instances can also be downloaded directly in the download section (right column). After selecting n and α, a click on the download button leads directly to the file containing the corresponding challenge.

The LWE challenge website also provides some smaller instances at the download section. These toy challenges could for example be used to test the correctness of an LWE solver implementation.

Format of the Challenges The LWE challenges are provided in the following format: in the first three rows, the integers n, m, and q are listed. In the fourth row, the real α is found. It is written in US notation, i.e., with a period as decimal point. In the fifth row the vector $\mathbf{b}$ - which actually is a column vector - is given and finally in the sixth row, the matrix $\mathbf{A}$ starts. It consists of m rows which n entries each. A description of the format of the instances can also be found at the download section on the LWE challenge website.

5.2 How to Submit Solutions

The LWE challenge accepts a submitted solution $\mathbf{s}$ for a challenge $\mathbf{A}, \mathbf{b}$ with parameters n, m, α, and q if (and only if) $\|\mathbf{b} - \mathbf{A}\mathbf{s}\| \leq 2\sqrt{m}\alpha q$. Equation (1) shows that such an $\mathbf{s}$ gets accepted with overwhelming probability if $\mathbf{b}$ was actually created as $\mathbf{A}\mathbf{s} + \mathbf{e} \mod q$. On the other hand, as mentioned in Section 4, with very high probability all instances have a unique solution.

Analogous to downloading instances, there are two possibilities for submitting a solution: first, the solution can be submitted by clicking on the respective green point in the challenge table on the website. This will lead to a submission form where the lattice dimension and the Gaussian parameter are already entered. Second, the solution can be submitted by following the submission link on the right side of the start page of the LWE challenge website. This link leads to a submission form where both the lattice dimension and the Gaussian parameter can be selected independently.

Note that for the solved instances, i.e., those represented with a red point in the challenge table, no solution can be submitted. Solutions for the toy instances can not be submitted either.

Hall of Fame At the bottom of the LWE challenge website's start page, we show the latest five successful submissions. We also provide a list with all successful submissions, i.e., a hall of fame. It can be found by clicking on a link below the latest submissions. Here, all correct solutions are listed in a chronological order, together with some meta information.

Once an instance is broken, there is no way to find a better solution. Therefore, old results will not be suppressed and hence, these early contributions will stay visible. This will prevent a picture similar to the one on the original lattice challenge, where only few names dominate the hall of fame since as long as the shortest vector is not found, better submissions can replace older solutions.

Acknowledgement. This work has been co-funded by the DFG as part of projects P1 and S5 within the CRC 1119 CROSSING.

References

1. Certicom ECC challenge. `https://www.certicom.com/images/pdfs/challenge-2009.pdf`. Accessed: 2015-12-21.
2. NTRU challenge. `https://www.securityinnovation.com/uploads/ntru-challenge-parameter-sets-and-public-keys-new.pdf`. Accessed: 2015-12-21.
3. The RSA factoring challenge. `http://www.emc.com/emc-plus/rsa-labs/historical/the-rsa-factoring-challenge.htm`. Accessed: 2015-12-18.
4. M. R. Albrecht, C. Cid, J. Faugère, R. Fitzpatrick, and L. Perret. Algebraic algorithms for LWE problems. *IACR Cryptology ePrint Archive*, 2014:1018, 2014.
5. M. R. Albrecht, C. Cid, J. Faugère, R. Fitzpatrick, and L. Perret. On the complexity of the BKW algorithm on LWE. *Des. Codes Cryptography*, 74(2):325–354, 2015.
6. M. R. Albrecht, R. Fitzpatrick, and F. Göpfert. On the efficacy of solving LWE by reduction to unique-svp. In H. Lee and D. Han, editors, *Information Security and Cryptology - ICISC 2013 - 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers*, volume 8565 of *Lecture Notes in Computer Science*, pages 293–310. Springer, 2013.
7. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015. `http://eprint.iacr.org/`.
8. S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.
9. Y. Aumann and Y. Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. *J. Cryptology*, 23(2):281–343, 2010.
10. L. Babai. On lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
11. S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. In J. Benaloh, editor, *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, volume 8366 of *Lecture Notes in Computer Science*, pages 28–47. Springer, 2014.
12. S. Bai and S. D. Galbraith. Lattice decoding attacks on binary LWE. In W. Susilo and Y. Mu, editors, *Information Security and Privacy - 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings*, volume 8544 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2014.
13. D. J. Bernstein, J. Buchmann, and E. Dahmen, editors. *Post-quantum cryptography.* Mathematics and Statistics. 2009.
14. A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003.
15. D. Bogdanov, M. Jõemets, S. Siim, and M. Vaht. How the estonian tax and customs board evaluated a tax fraud detection system based on secure multi-party computation. In *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*, volume 8975 of *LNCS*, pages 227–234. Springer, 2015.

16. D. Bogdanov, P. Laud, S. Laur, and P. Pullonen. From input private to universally composable secure multi-party computation primitives. In *IEEE 27th Computer Security Foundations Symposium, CSF 2014*, pages 184–198. IEEE, July 2014.
17. D. Bogdanov, S. Laur, and J. Willemson. Sharemind: A framework for fast privacy-preserving computations. In S. Jajodia and J. López, editors, *Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*, volume 5283 of *Lecture Notes in Computer Science*, pages 192–206. Springer, 2008.
18. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013.
19. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In R. Ostrovsky, editor, *FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE Computer Society, 2011.
20. J. A. Buchmann, R. Lindner, and M. Rückert. Explicit hard instances of the shortest vector problem. In J. A. Buchmann and J. Ding, editors, *Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings*, volume 5299 of *Lecture Notes in Computer Science*, pages 79–94. Springer, 2008.
21. C. Chekuri, K. Jansen, J. D. P. Rolim, and L. Trevisan, editors. *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, volume 3624 of *Lecture Notes in Computer Science*. Springer, 2005.
22. A. Duc, F. Tramèr, and S. Vaudenay. Better algorithms for LWE and LWR. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 173–202. Springer, 2015.
23. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2013.
24. L. Kamm. *Privacy-preserving statistical analysis using secure multi-party computation.* PhD thesis, University of Tartu, 2015.
25. A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *PKC 2014, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, pages 345–361, 2014.
26. R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In A. Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.
27. M. Liu and P. Q. Nguyen. Solving BDD by enumeration: An update. In E. Dawson, editor, *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013, San Francisco,CA, USA, February 25-March 1, 2013. Proceedings*, volume 7779 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2013.

28. V. Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In Chekuri et al. [21], pages 378–389.
29. T. Plantard and M. Schneider. Creating a challenge for ideal lattices. *IACR Cryptology ePrint Archive*, 2013:39, 2013.
30. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.
31. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, Sept. 2009.
32. T. Yasuda, X. Dahan, Y. Huang, T. Takagi, and K. Sakurai. MQ challenge: Hardness evaluation of solving multivariate quadratic problems. *IACR Cryptology ePrint Archive*, 2015:275, 2015.