CSC 2419: Lattice-based Cryptography **Fall 2025**

Lecture 2: Learning with Errors and Public-key Encryption

Instructor: Akshayaram Srinivasan *Scribe: Somnath Bhattacharjee*

Date: 2025-09-15

2.1 Recap

Last lecture we have been introduced with *Short Integer Solution (SIS)* and saw that the average-case hardness of SIS implies existence of *Collision Resistant Hash Function (CRHF)*. It is known that CRHF gives *One way functions (OWF)*, and OWF implies lots other important cryptographic features such as Pseudorandom Generators, Pseudorandom Functions, Digital Signatures etc.

Today we will see another computationally hard problem, namely *Learning with errors (LWE)* and construction of a *Public Key Encryption (PKE) scheme* based on the average-case hardness of LWE problem

2.2 Learning With Errors (LWE)

Definition 2.1 ***(B-Bounded distribution)*** *Let $B \in \mathbb{R}_{\geq 0}$. A distribution over integers χ is called B-*Bounded *if* $\Pr_{e \leftarrow \chi}[|e| \leq B] = 1$

We now define the LWE problem, namely the search variant $\mathsf{Search\text{-}LWE}_{n,m,q,B}$ parameterized by $n, m, q, B \in \mathbb{N}$

Definition 2.2 ($\mathsf{Search\text{-}LWE}_{n,m,q,B}$) *Let $n, m \in \mathbb{N}$ be the dimension, $q \in \mathbb{N}$ the modulus, and $B \in \mathbb{N}$ be the bound on the error. The search LWE problem is defined as follows:*

- ***Given:*** $(\mathbf{A}, \mathbf{b}^T := \mathbf{s}^T\mathbf{A} + \mathbf{e}^T)$ *where*
  1. $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$
  2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ *(The "secret")*
  3. $\mathbf{e} \leftarrow \chi^m$ *(The "error")*
- ***Find:*** $\mathbf{s}$ *such that* $||\mathbf{b}^T - \mathbf{s}^T\mathbf{A}|| \leq B$

Remark 2.3 *We highlight some remarks regarding the LWE problem.*

- ***Why error?*** *If we don't consider the error vector $\mathbf{e}$ as part of the LWE sample, we observe that an adversary can use Gaussian elimination to find $\mathbf{s}$ from input $\mathbf{b}^T = \mathbf{s}^T\mathbf{A}$. Therefore, the error is necessary for the LWE problem to be hard.*

- ***Uniqueness of solution and parameter paradigm.*** *Since* $\mathbf{A}$ *is a random matrix,* $\mathbf{s}^T\mathbf{A}$ *is a random code word as a random matrix has full rank with very high probability[?]. Since* $\mathbf{s}^T\mathbf{A} \in \mathbb{Z}_q^m$ *and* $\mathbf{s}^T \in \mathbb{Z}_q^n$*, for a sufficiently large* m *and* q*, the code space* $\mathcal{C} \gg \mathbb{Z}_q^n$*. Since the code is random, this results in the codes being "well" distributed such that the hamming distance of the codeword is large enough. In this paradigm, for* B *much smaller than the hamming distance, we have a unique solution with high probability. This paradigm of parameters are interesting and find many cryptographic applications. We refer to* q/B *as the modulus-to-noise ratio of the distributions, and study the LWE problem in the paradigm where modulus-to-noise ratio is sub-exponential i.e.* $q/B \leq 2^{n^\epsilon}$*.*

We also consider the *decision* variant of the Search-LWE problem as $\mathsf{Decision\text{-}LWE}_{n,m,q,B}$.

Definition 2.4 ($\mathsf{Decision\text{-}LWE}_{n,m,q,B}$) *Let* $n, m \in \mathbb{N}$ *be the dimension,* $q \in \mathbb{N}$ *the modulus, and* $B \in \mathbb{N}$ *be the bound on the error. The decision LWE problem is defined as follows:*

- ***Given:*** *Distribution* $\mathcal{D}_b$ *for a random coin* $b \xleftarrow{\$} \{0,1\}$ *where the distributions are defined as follows:*

$\underline{\mathcal{D}_0}$	$\underline{\mathcal{D}_1}$
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$	$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$
$\mathbf{e} \xleftarrow{\$} \chi^m$	$\mathbf{b}^T \xleftarrow{\$} \mathbb{Z}_q^m$
$\mathbf{s}^T \xleftarrow{\$} \mathbb{Z}_q^n$	**Output** $(\mathbf{A}, \mathbf{b}^T)$
Output $(\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T)$	

- ***Find:*** *the distribution from which the instance* $(\mathbf{A}, \mathbf{b}^T)$ *is sampled from i.e.* b*.*

Definition 2.5 (Average-case hardness of Decision-LWE) *For parameters* $n, m, q, B = \mathsf{poly}(\lambda)$ *with* $\lambda \in \mathbb{N}$ *as the security parameter,* $\mathsf{Decision\text{-}LWE}_{n,m,q,B}$ *is hard if for all Probabilistic Polytime (PPT) adversary* $\mathcal{A}$ *,* $\exists$ *a negligible function* $\mathsf{negl}(\cdot)$ *such that* $\forall \lambda \in \mathbb{N}$,

$$|\Pr_{(\mathbf{A},\mathbf{b}^T)\leftarrow\mathcal{D}_0}[\mathcal{A}(\mathbf{A}, \mathbf{b}^T) = 1] - \Pr_{(\mathbf{A},\mathbf{b}^T)\leftarrow\mathcal{D}_1}[\mathcal{A}(\mathbf{A}, \mathbf{b}^T) = 1]| \leq \mathsf{negl}(\mathrm{sec}$$

Average-case hardness of decision-LWE. For specific parameters, spefically when $mB \ll q$, the average-case hardness of $dlwe_{n,m,q,B}$ reduces to the average-case hardness of $\mathsf{SIS}_{n,m,q}$ problem. This result is highlighted in Theorem 2.6. Note that this reductions results in the blowup of the "error" by a factor of m. In more general parameter paradigms, there has been a long line of work [Reg05, Pei09, BLP$^+$13] which have proved the hardness of Decision-LWE problem assuming the hardness of $\alpha - \mathsf{GapSVP}$ problem. We highlight this reslut in Theorem 2.8. Please refer to the survey by Peikert [Pei16] for more details.

Theorem 2.6 $\mathsf{Decision\text{-}LWE}_{n,m,q,B}$ *is atleast as hard as* $\mathsf{SIS}_{n,m,q}$ *given that* $B \ll q$*.*

Proof: We briefly sketch the reduction, where an adversary for the LWE problem $\mathcal{A}_{\mathsf{LWE}}$ invokes the adversary for the SIS problem $\mathcal{A}_{\mathsf{SIS}}$. $\mathcal{A}_{\mathsf{LWE}}$, given an instance $(\mathbf{A}, \mathbf{b}^T)$ computes $\mathbf{e}^T \leftarrow \mathcal{A}_{\mathsf{SIS}}(\mathbf{A})$ such that $||\mathbf{e}||_\infty = 1$[1]. Subsequently, the adversary computes $< \mathbf{b}^T, \mathbf{e} >$. If $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}_1^T$, then $< \mathbf{b}^T, \mathbf{e} >= \mathbf{s}^T\mathbf{A}\mathbf{e} + < \mathbf{e}_1^T, \mathbf{e} >=< \mathbf{e}_1^T, \mathbf{e} >$, such that $||\mathbf{b}^T||_\infty \leq mB$ i.e a low-norm vector. On the other hand, if $\mathbf{b}^T \xleftarrow{\$} \mathbb{Z}_q^m$, then $< \mathbf{b}^T, \mathbf{e} >$ is uniformly random. ■

[1]The proof works for a more general variant of the SIS problem, where the solution has a low norm.

Remark 2.7 Decision-LWE *problem and* SIS *problem are equivalent (given quantum reductions).*

Theorem 2.8 (([Reg05, Pei09, BLP+13], simplified)) *If* $\alpha -$ GapSVP *is hard in the worst case then* Decision-LWE *is hard in average case for some* $\alpha = O(\mathsf{poly}(n)\frac{q}{B})$

Remark 2.9 *The hardness of* $\alpha -$ GapSVP *is well studied for* $\alpha = 2^{n^\epsilon}$*, which translates to the modulus-to-noise ratio* $q/B \leq 2^{n^\epsilon}$*, which makes this parameter interesting for cryptographic applications.*

2.2.1 Search to Decision for LWE

We have already seen some results concerning the hardness of *decision-LWE* problem. We now look at the hardness of *search-LWE*, which we will see can be based on the hardness of *decision-LWE*.

Theorem 2.10 Search-LWE$_{n,m,q,B}$ *is atleast as hard as* Decision-LWE$_{n,m,q,B}$

Proof Idea: We define an adversary $\mathcal{A}_{\textsf{Search-LWE}}$, which, when given access to adversary for the decision LWE problem $\mathcal{A}_{\textsf{Decision-LWE}}$, can solve the Search-LWE problem in polynomial time. Given the instance $(\mathbf{A}, \mathbf{b}^T)$ to $\mathcal{A}_{\textsf{Search-LWE}}$, the adversary initialises $\mathbf{s}^T := \mathbf{0}$ where $\mathbf{s}^T \in \mathbb{Z}_q^n$ and computes the following: for each $i \in [n]$,

- For each $g \in \mathbb{Z}_q$:
  1. Sample $\mathbf{c}_i^T \xleftarrow{\$} \mathbb{Z}_q^m$
  2. Set $\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{c}_i^T \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{Z}_q^{n\times m}$ $\quad$ (i^{th} row is c_i)
  3. Compute $b \leftarrow \mathcal{A}_{\textsf{Decision-LWE}}(\mathbf{A}+\mathbf{B}, \mathbf{b}^T + g\cdot \mathbf{c}_i^T)$. Set $\mathbf{s}_i = g$ if $b = 0$

Clearly the above algorithm works in $\mathsf{poly}(n)$ time. We will give the intuition of the correctness:

If $\mathcal{A}_{\textsf{Decision-LWE}}(\mathbf{A}+\mathbf{B}, \mathbf{b}^T + g\cdot \mathbf{c}_i^T) \to 0$, then

$$\mathbf{b}^T + g\cdot \mathbf{c}_i^T = \mathbf{s}^T(\mathbf{A}+\mathbf{B}) + \mathbf{e}_1^T$$

Also, from the properties of SIS problem,

$$\begin{aligned} \mathbf{b}^T + g\cdot \mathbf{c}_i^T =& \mathbf{s}^T\mathbf{A} + \mathbf{e}^T + g\cdot \mathbf{c}_i^T \\ \implies g\cdot \mathbf{c}_i^T =& \mathbf{s}^T\mathbf{B} + (\mathbf{e}_1^T - \mathbf{e}^T) \\ =& \mathbf{s}^T\mathbf{B} + \mathbf{e}_2^T \end{aligned}$$

where $||\mathbf{e}_2^T||_\infty \leq ||\mathbf{e}_1^T||_\infty + ||\mathbf{e}||_\infty \leq 2B$

Note that $\mathbf{s}^T\mathbf{B}$ is a random codeword with a large hamming distance. Therefore, for error with low norm B, the above equation has a unique solution i.e. $g = \mathbf{s}_i$.

To prove the validity of the reduction, we also need to show that when $g \neq \mathbf{s}_i$, then the distribution received by $\mathcal{A}_{\textsf{Decision-LWE}}$ is uniformly random. To see this, observe that

$$\mathbf{b}^T + g \cdot \mathbf{c}_i^T = \mathbf{s}^T \underbrace{(\mathbf{A}+\mathbf{B})}_{\text{uniform}} + \mathbf{e}^T + \underbrace{(g - \mathbf{s}_i)\mathbf{c}_i^T}_{\text{independent from}\mathbf{A}+\mathbf{B}}$$

Now note the the distribution over $\mathbf{A}+\mathbf{B}$ is uniform. Furthermore note that the distribution over c_i is independent of $(\mathbf{A}+\mathbf{B})$ (since $\mathbf{A}$ is uniform). So $\mathbf{b}^T + g \cdot \mathbf{c}_i^T$ is also uniformly distributed and independent of $\mathbf{A}+\mathbf{B}$. Note that this reduction works for the case when $\mathcal{A}_{\textsf{Decision-LWE}}$ is a perfect distinguisher. Please refer to [Reg05, Section 4] for the case where $\mathcal{A}_{\textsf{Decision-LWE}}$ is an imperfect distinguisher. ■

2.2.2 Normal form LWE

Up until now, we have looked at search and decision variants of LWE where the secret $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. It would be interesting to study the problem in a different paradigm, namely where the secret is sampled from a different distribution, as this allows for cryptographic primitives with hardness based on a wider class of hardness assumptions. In this section, we study the **Normal Form** of LWE, where the secret $\mathbf{s}^T$ is sampled from the same low-norm distribution χ^n as the error.

Definition 2.11 (Normal-form Decision-LWE$_{n,m,q,B}$) *Let $n, m \in \mathbb{N}$ be the dimension, $q \in \mathbb{N}$ the modulus, and $B \in \mathbb{N}$ be the bound on the error. The Normal-Form LWE problem is defined as follows:*

- ***Given:*** *Distribution $\mathcal{D}_b$ for a random coin $b \xleftarrow{\$} \{0,1\}$ where the distributions are defined as follows:*

$$\underline{\mathcal{D}_\mathbf{0}}$$
$$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$$
$$\mathbf{e} \xleftarrow{\$} \chi^m$$
$$\mathbf{s}^T \xleftarrow{\$} \chi^n$$
$$\textbf{Output } (\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T)$$

$$\underline{\mathcal{D}_\mathbf{1}}$$
$$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$$
$$\mathbf{b}^T \xleftarrow{\$} \mathbb{Z}_q^m$$
$$\textbf{Output } (\mathbf{A}, \mathbf{b}^T)$$

- ***Find:*** *the distribution from which the instance $(\mathbf{A}, \mathbf{b}^T)$ is sampled from i.e. b.*

We now study the hardness of *Normal-form* Decision-LWE i.e. Normal-LWE. Notably, the hardness of Normal-LWE can be based on the hardness of standard *dlwe*. This result is stated in Theorem 2.12

Theorem 2.12 *Hardness of* Decision-LWE *implies hardness of* Decision-LWE *with normal form*[2]

Proof: Let $\mathcal{A}_{\textsf{Decision-LWE}}$ be a PPT adversary for the Decision-LWE problem. We define an adversary $\mathcal{A}_{\textsf{Normal-LWE}}$ for the Normal-LWE problem with access to $\mathcal{A}_{\textsf{Decision-LWE}}$. On input $(\mathbf{A}, \mathbf{b}^T)$, the adversary computes the following:

1. Let $A = [\underbrace{A_1}_{\mathbb{Z}_q^{n\times n}} || \underbrace{A_2}_{\mathbb{Z}_q^{n\times(m-n)}}]$ (Note A_1 is invertible with high probability as it is uniformly random)

[2] One can extend this proof for search version

2. Let $\mathbf{b}^T = (\underbrace{\mathbf{b}_1^T}_{\chi^n} || \underbrace{\mathbf{b}_2^T}_{\chi^{m-n}})$

3. Set $\tilde{\mathbf{A}} = -\mathbf{A}_1^{-1}\mathbf{A}_2$, $\tilde{\mathbf{b}}^T = \mathbf{b}_1^T\tilde{\mathbf{A}} - \mathbf{b}_2^T$

4. Output $\mathcal{A}_{\textsf{Decision-LWE}}(\tilde{\mathbf{A}}, \tilde{\mathbf{b}}^T)$

Observe that if $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$ for a uniformly sampled secret $\mathbf{s}^T$, then

$$\begin{aligned}\tilde{\mathbf{b}}^T &= \mathbf{b}_1^T\tilde{\mathbf{A}} - \mathbf{b}_2^T \\ &= (\mathbf{s}^T\mathbf{A}_1 + \mathbf{e}_1^T)\tilde{\mathbf{A}} - \mathbf{s}^T\mathbf{A}_2 - \mathbf{e}_2^T \\ &= \underbrace{\mathbf{e}_1^T}_{\text{new secret}} \tilde{\mathbf{A}} + \underbrace{(-\mathbf{e}_2^T)}_{\text{new error}}\end{aligned}$$

Here, the new secret is sampled from the distribution χ^m. On the other hand, if $\mathbf{b}^T \xleftarrow{\$} \mathbb{Z}_q^m$, then $\mathbf{b}_2^T$ is uniformly random and independent of $\mathbf{b}_1^T\tilde{\mathbf{A}}$, resulting in a uniformly random $\tilde{\mathbf{b}}$. ■

2.3 Public-Key encryption

Suppose *Alice* want to send a message to *Bob*, but the communication can be seen by a third party *Charlie*. To hide the underlying message in the communication from *Charlie*, *Alice* can use a cryptographic primitive called *Public-Key Encryption(PKE)* to encrypt the message and send the encrypted message to *Bob* such that only a party having the corresponding secret key can decrypt and read the message. We define the primitive in Definition **??**

Definition 2.13 (Public Key Encryption(PKE)) *PKE consists of tuple of PPT algorithms* $(\textsf{KeyGen}, \textsf{Enc}, \textsf{Dec})$ *where:*

1. $\textsf{KeyGen}(1^\lambda) \to (\textsf{pk}, \textsf{sk})$*: The key generation algorithm outputs public-secret key tuple* $(\textsf{pk}, \textsf{sk})$

2. $\textsf{ct} := \textsf{Enc}(\textsf{pk}, \mu; r)$*: On input the public key* $\textsf{pk}$*, message* $\mu \in \{0, 1\}$*, and randomness* r*, the encryption algorithm returns a ciphertext* $\textsf{ct}$

3. $\textsf{Dec}(\textsf{sk}, \textsf{ct})$*: On input the secret key* $\textsf{sk}$ *and the ciphertext* $\textsf{ct}$*, the decryption algorithm returns a bit* $\mu' \in \{0, 1\}$.

The PKE scheme satisfies the following properties:

- ***Correctness:*** *For security parameter* $\lambda \in \mathbb{N}$*,* $(\textsf{pk}, \textsf{sk}) \leftarrow \textsf{KeyGen}(1^\lambda)$*, and for* $\mu \in \{0, 1\}$*, the following holds true:*

$$\Pr[\textsf{Dec}(\textsf{sk}, \textsf{Enc}(\textsf{pk}, \mu)) = \mu] = 1$$

- ***IND-CPA Security:*** *For security parameter* $\lambda \in \mathbb{N}$*,* $(\textsf{pk}, \textsf{sk}) \leftarrow \textsf{KeyGen}(1^\lambda)$*, and PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\textsf{negl}(\cdot)$ *such that*

$$|\Pr[\mathcal{A}(\textsf{pk}, \textsf{ct}_0) = 1] - \Pr[\mathcal{A}(\textsf{pk}, \textsf{ct}_1) = 1]| \leq 1/2 + \textsf{negl}(\lambda)$$

where $\textsf{ct}_b = \textsf{Enc}(\textsf{pk}, b; r)$*.In other words, no PPT adversary can distinguish between encryption of 0 and 1 with more than negligible probability.*

2.3.1 PKE from LWE

Now we will discuss the PKE (for one bit message) construction from LWE given in [Reg05]. Given n, m, q with $m > n \log q$ and $\mu \in \{0, 1\}$, we define the PKE scheme as follows:

1. $\mathsf{KeyGen}(1^\lambda)$: Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}^T \leftarrow \chi^m$, set $\mathsf{pk} = (\mathbf{A}, \mathbf{b}^T := \mathbf{s}^T\mathbf{A} + \mathbf{e}^T)$ and $\mathsf{sk} = \mathbf{s}$
2. $\mathsf{Enc}(\mathsf{pk}, \mu; \mathbf{r} \xleftarrow{\$} \{0, 1\}^m)$: Compute $\mathsf{ct} := (\mathbf{Ar} \mod q, < \mathbf{b}^T, \mathbf{r} > + \mu(q/2))$
3. $\mathsf{Dec}(\mathbf{s}, \mathsf{ct})$: Parse $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$ and compute $\mu' := \mathsf{ct}_2 - < \mathbf{s}^T, \mathsf{ct}_1 >$, output 0 if $|\mu| < \frac{q}{4}$ and 1 otherwise.

We now show that the above scheme is a valid public key encryption.

Correctness: Note that

$$\begin{aligned}\mu' :=& \mathsf{ct}_2 - \mathbf{s}^T\mathsf{ct}_1 \\ =& < \mathbf{b}^T, \mathbf{r}^T > + \mu(q/2) - \mathbf{s}^T\mathbf{Ar} \\ =& \mu(q/2) + \mathbf{e}^T\mathbf{r} \\ \leq& mB + \mu(q/2)\end{aligned}$$

So if $\mu = 0$, then $x \leq mB$, So if $B \leq q/4m$ we are done. If $\mu = 1$, then for the same value of B we are done.

Security: We want to show that the adversary cannot distinguish between ciphertexts for 1 and 0. In other words, we want to show that the following indistinguishability relation holds:

$$(\mathbf{A}, \mathbf{b}^T, (\mathbf{Ar} \mod q, < \mathbf{b}^T, \mathbf{r} >) \approx_c (\mathbf{A}, \mathbf{b}^T, (\mathbf{Ar} \mod q, < \mathbf{b}^T, \mathbf{r} > + q/2)$$

.

From the **hardness of LWE** we know $(\mathbf{A}, \mathbf{b}^T, \mathbf{Ar} \mod q, < \mathbf{b}^T, \mathbf{r} >) \approx_c (\mathbf{A}, \tilde{\mathbf{b}}^T, \mathbf{Ar} \mod q, < \tilde{\mathbf{b}}^T, \mathbf{r} >)$ where $\tilde{\mathbf{b}}^T \xleftarrow{\$} \mathbb{Z}_q^m$. If we consider the matrix $\tilde{\mathbf{A}} = \mathbf{A}||\mathbf{b}$, then the distribution is $(\tilde{\mathbf{A}}, \tilde{\mathbf{A}}\mathbf{r} \mod q)$ where $\tilde{\mathbf{A}}$ is uniformly random. Using the *Leftover Hash Lemma* (Lemma 2.14), we conclude that $(\tilde{\mathbf{A}}, \tilde{\mathbf{A}}\mathbf{r} \mod q) \approx_s (\tilde{\mathbf{A}}, \mathbf{u}^T)$ where $\mathbf{u}^T$ is a random vector. Similarly, it can be shown that $(\mathbf{A}, \mathbf{b}^T, (\mathbf{Ar} \mod q, < \mathbf{b}^T, \mathbf{r} > + q/2) \approx_c (\tilde{\mathbf{A}}, \mathbf{u}^T)$.

Lemma 2.14 (Leftover Hash Lemma(LHL)) *for* $m \geq 2n \log q$, $(A, Ar) \approx_s (A, u)$[3]

References

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 575–584, New York, NY, USA, 2013. Association for Computing Machinery.

[3] All the notations follows from the previous arguments. For detailed version of the lemma check: https://en.wikipedia.org/wiki/Leftover˙hash˙lemma

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 333–342, New York, NY, USA, 2009. Association for Computing Machinery.

[Pei16] Chris Peikert. 2016.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA, 2005. Association for Computing Machinery.