Implementation of Lattice Trapdoors on Modules and Applications

Pauline Bert, Gautier Eberhart, Lucas Prabel, Adeline Roux-Langlois, and Mohamed Sabt
July 9, 2021

Univ Rennes, CNRS, IRISA

Contributions of the paper

- Development of efficient **Gaussian preimage sampling** techniques on **module lattices**.
- Applications to signatures and **identity-based encryption**.
- A **public and open-source implementation** without any external library dependencies.

Gaussian preimage sampling on module lattices

Using trapdoors to build signature schemes ([GPV08])

Idea

Public key	Matrix $A \in \mathbb{Z}_q^{n \times m}$ defining $\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m \mid Ax = 0 \ mod\ q\}$.
Secret key	Short basis $T \in \mathbb{Z}^{m \times m}$ of this lattice (T is the trapdoor for A).

$\longrightarrow$ **Signature :**

Message $M \in \{0,1\}^*$ —hash→ $u = H(M) \in \mathbb{Z}_q^n$ —Gaussian Preimage sampling using T→ x small such that $Ax = u \ mod\ q$

$\longrightarrow$ **Verification :**

- **Accept** if $Ax = u \ mod\ q$ and x small.
- **Reject** otherwise.

Module Gadget Trapdoor of [MP12]

Rings $\mathcal{R} = \mathbb{Z}[X]/\langle X^n + 1\rangle$ and $\mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$.

TrapGen algorithm outputs 2 matrices

$$A = [\, A' \mid HG - A'T \,] \in \mathcal{R}^{d \times m} \text{ and } T \in \mathcal{R}^{2d \times dk}$$

such that

$$A \begin{bmatrix} T \\ \hline I_{dk} \end{bmatrix} = HG.$$

- $G = I_d \otimes \mathbf{g}^T \in \mathcal{R}^{d \times dk}$ where $\mathbf{g}^T = \begin{bmatrix} 1 & b & b^2 & \cdots & b^{k-1} \end{bmatrix}$ with $k = \lceil \log_b q \rceil$.
- $H \in \mathcal{R}_q^{d \times d}$ an invertible matrix, called the tag.
- $T \leftarrow D_{\mathcal{R}^{2d \times dk}, \sigma}$.
- $A' \leftarrow [\, I_d \mid \hat{A} \,]$ where $\hat{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{d \times d})$.

$\longrightarrow$ Computing a small Gaussian vector $\mathbf{x} \in \mathcal{R}^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{u} \bmod q$ for a given $\mathbf{u} \in \mathcal{R}^d$.

First step : Module G-Sampling

- Sample $\mathbf{z} \leftarrow D_{\Lambda_q^{\mathbf{v}}(\mathbf{G}),\alpha}$ by ndk calls to the scalar sampler of [GM18] with $\mathbf{v} = \mathbf{H}^{-1}\mathbf{u}$.
- $\mathbf{z}$ verifies $\mathbf{G}\mathbf{z} = \mathbf{v} \bmod q$.
- Compute $\mathbf{x} = \begin{bmatrix} \mathbf{T} \\ \mathbf{I} \end{bmatrix}\mathbf{z}$.

$\longrightarrow$ We have $\mathbf{A}\mathbf{x} = \mathbf{A}\begin{bmatrix} \mathbf{T} \\ \mathbf{I} \end{bmatrix}\mathbf{z} = \mathbf{H}\mathbf{G}\mathbf{z} = \mathbf{H}\mathbf{v} = \mathbf{u} \bmod q$.

Problem

The distribution of $\mathbf{x}$ leaks information about the trapdoor $\mathbf{T}$:

$$\mathbf{\Sigma}_{\mathbf{x}} = \alpha^2 \begin{bmatrix} \mathbf{T} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{T}^T & \mathbf{I} \end{bmatrix}.$$

$\longrightarrow$ Computing a small Gaussian vector $\boldsymbol{x} \in \mathcal{R}^m$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{u}$ *mod q* for a given $\boldsymbol{u} \in \mathcal{R}^d$.

Lemma (simplified)

Let $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} A & B \\ B^T & D \end{smallmatrix}\right] \in \mathbb{R}^{(r+s)\times(r+s)}$ and :

- $\boldsymbol{x}_1 \leftarrow D_{\mathbb{Z}^s, \sqrt{\boldsymbol{D}}, \boldsymbol{c}_1}$;
- $\boldsymbol{x}_0 \leftarrow D_{\mathbb{Z}^r, \sqrt{\boldsymbol{\Sigma}/\boldsymbol{D}}, \boldsymbol{c}_0 + \boldsymbol{B}\boldsymbol{D}^{-1}(\boldsymbol{x}_1 - \boldsymbol{c}_1)}$.

This process outputs a vector $\boldsymbol{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1) \in \mathbb{Z}^{r+s}$ whose distribution is statistically indistinguishable from $D_{\mathbb{Z}^{r+s}, \sqrt{\boldsymbol{\Sigma}}, \boldsymbol{c}}$.

Second step : Perturbation Sampling

- Sample $\boldsymbol{p} \leftarrow D_{\mathcal{R}^m, \sqrt{\boldsymbol{\Sigma}_p}}$.
- $\boldsymbol{p}$ has convariance matix $\boldsymbol{\Sigma}_p = \zeta^2 \boldsymbol{I} - \alpha^2 \left[\begin{smallmatrix} \boldsymbol{T} \\ \boldsymbol{I} \end{smallmatrix}\right] \left[\begin{smallmatrix} \boldsymbol{T}^T & \boldsymbol{I} \end{smallmatrix}\right]$.

$\longrightarrow$ Particular structure of $\boldsymbol{\Sigma}_p = \left[\begin{array}{c|c} A & -\alpha^2 \boldsymbol{T} \\ \hline -\alpha^2 \boldsymbol{T}^T & (\zeta^2 - \alpha^2)\boldsymbol{I} \end{array}\right]$ + using the Lemma iteratively.

$\longrightarrow$ Computing a small Gaussian vector $x \in \mathcal{R}^m$ such that $Ax = u \bmod q$ for a given $u \in \mathcal{R}^d$.

Preimage Sampling Algorithm

1. Sample $p \leftarrow D_{\mathcal{R}^m, \sqrt{\Sigma_p}}$ (**Perturbation Sampling**).
2. Compute $v = H^{-1}(u - Ap)$.
3. Sample $z \leftarrow D_{\Lambda_q^v(G), \alpha}$ (**G-Sampling**).
4. Return $x = p + \begin{bmatrix} T \\ I \end{bmatrix} z$.

- x lies in the desired coset.
- The covariance matrix of x is $\Sigma = \underbrace{\Sigma_p}_{\text{perturbation covariance matrix}} + \underbrace{\alpha^2 \begin{bmatrix} T \\ I \end{bmatrix} \begin{bmatrix} T^T & I \end{bmatrix}}_{\text{covariance matrix of } \begin{bmatrix} T \\ I \end{bmatrix} z} = \zeta^2 I.$

Dual-Regev encryption scheme

Dual-Regev encryption scheme [GPV08]

Alice

$pk = (A, u = Ax_{Bob} \text{ mod } q)$

$s \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$

$e, e' \leftarrow$ Gaussian error vectors

$c_0 = A^T s + e$

$c_1 = u^T s + e' + M \cdot \lfloor q/2 \rfloor$

(c_0, c_1) →

Bob

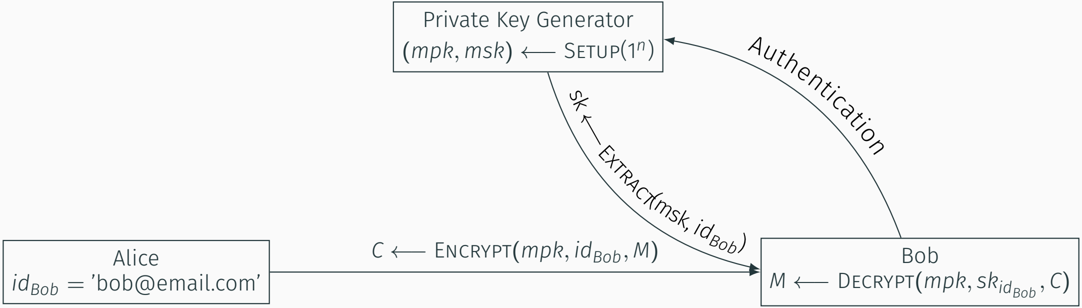
$pk = (A, u = Ax_{Bob} \text{ mod } q)$

$sk = x_{Bob} \leftarrow D_{\mathbb{Z}^m, \zeta}$

$$c_1 - x^T c_0 = \underbrace{e' - x^T e}_{small} + M \cdot \lfloor q/2 \rfloor$$

Identity-based encryption

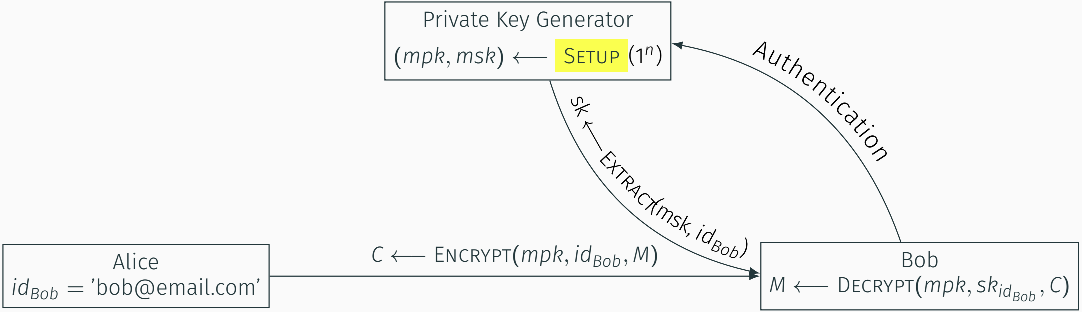
Identity-based encryption



IBE algorithms

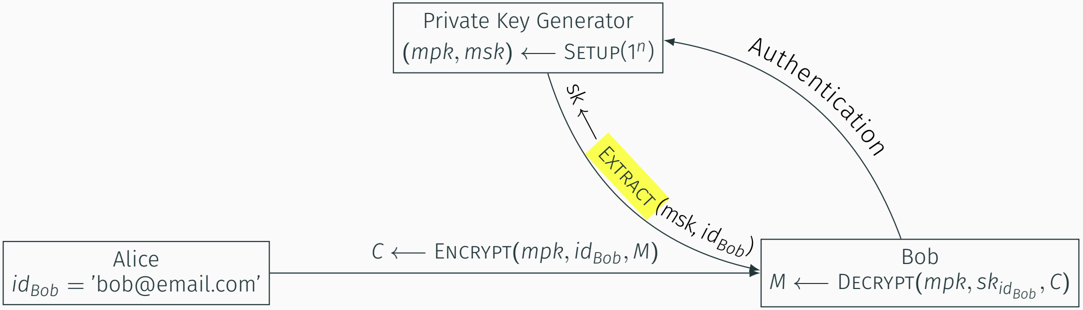
- $\textsc{Setup}(1^n) \longrightarrow (mpk, msk)$.
- $\textsc{Extract}(1^n, mpk, msk, id) \longrightarrow sk_{id}$.
- $\textsc{Encrypt}(1^n, mpk, id, M) \longrightarrow C$.
- $\textsc{Decrypt}(1^n, sk_{id}, C) \longrightarrow (M, Error)$.

Identity-based encryption



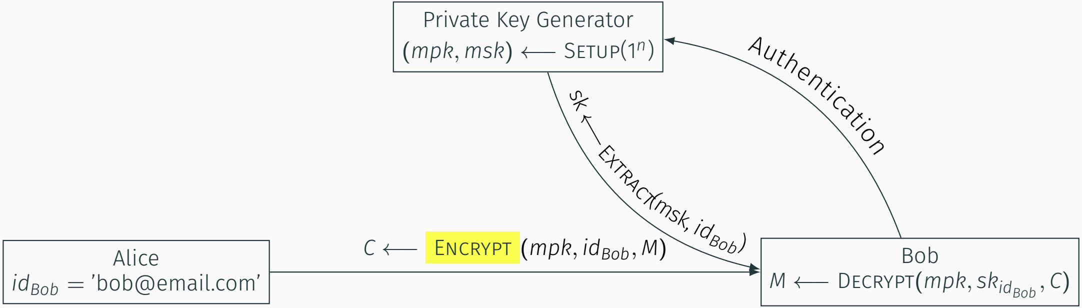
IBE algorithms

- $\textsc{Setup}(1^n) \longrightarrow (mpk, msk)$.
- $\textsc{Extract}(1^n, mpk, msk, id) \longrightarrow sk_{id}$.
- $\textsc{Encrypt}(1^n, mpk, id, M) \longrightarrow C$.
- $\textsc{Decrypt}(1^n, sk_{id}, C) \longrightarrow (M, Error)$.



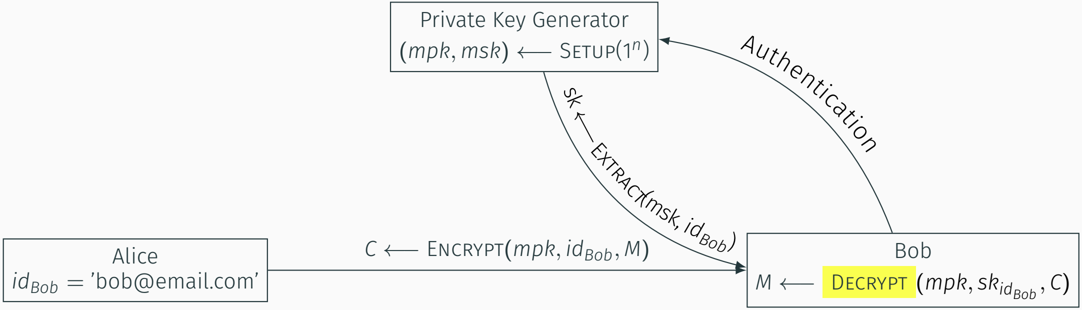
IBE algorithms

- $\textsc{Setup}(1^n) \longrightarrow (mpk, msk)$.
- $\textsc{Extract}(1^n, mpk, msk, id) \longrightarrow sk_{id}$.
- $\textsc{Encrypt}(1^n, mpk, id, M) \longrightarrow C$.
- $\textsc{Decrypt}(1^n, sk_{id}, C) \longrightarrow (M, Error)$.



IBE algorithms

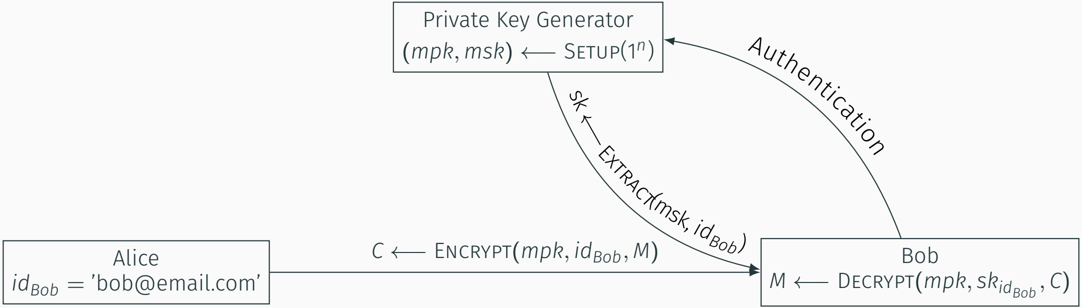
- $\textsc{Setup}(1^n) \longrightarrow (mpk, msk)$.
- $\textsc{Extract}(1^n, mpk, msk, id) \longrightarrow sk_{id}$.
- Encrypt $(1^n, mpk, id, M) \longrightarrow C$.
- $\textsc{Decrypt}(1^n, sk_{id}, C) \longrightarrow (M, Error)$.



IBE algorithms

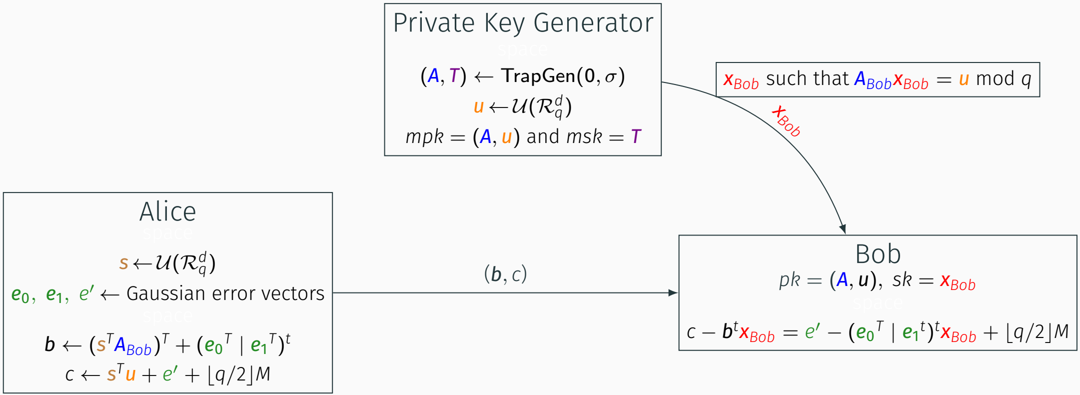
- $\text{SETUP}(1^n) \longrightarrow (mpk, msk)$.
- $\text{EXTRACT}(1^n, mpk, msk, id) \longrightarrow sk_{id}$.
- $\text{ENCRYPT}(1^n, mpk, id, M) \longrightarrow C$.
- $\text{DECRYPT}(1^n, sk_{id}, C) \longrightarrow (M, Error)$.

Identity-based encryption

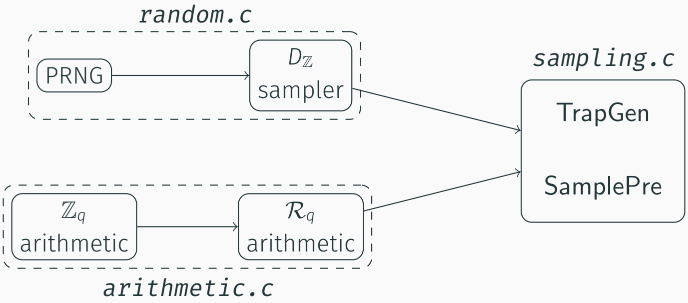


History

- 1984 IBE concept introduced by Shamir.
- 2001 First IBE constructions by Boneh and Franklin (bilinear maps) and Cocks (quadratic residue assumptions).
- 2008 First lattice based IBE, by Gentry, Peikert, and Vaikuntanathan ([GPV08]).
- 2010 Efficient lattice based IBE secure in the standard model ([ABB10]).
- 2014 Efficient IBE over NTRU lattices ([DLP14]).

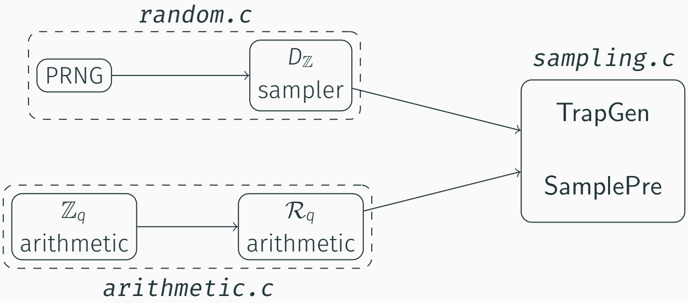

Private Key Generator
$(A, T) \leftarrow \mathsf{TrapGen}(0, \sigma)$
$u \leftarrow \mathcal{U}(\mathcal{R}_q^d)$
$mpk = (A, u)$ and $msk = T$
x_{Bob} such that $A_{Bob} x_{Bob} = u \bmod q$
x_{Bob}
Alice
$s \leftarrow \mathcal{U}(\mathcal{R}_q^d)$
$e_0,\ e_1,\ e'$ ← Gaussian error vectors
$b \leftarrow (s^T A_{Bob})^T + ({e_0}^T \mid {e_1}^T)^t$
$c \leftarrow s^T u + e' + \lfloor q/2 \rfloor M$
(b, c)
Bob
$pk = (A, u),\ sk = x_{Bob}$
$c - b^t x_{Bob} = e' - ({e_0}^T \mid {e_1}^t)^t x_{Bob} + \lfloor q/2 \rfloor M$

Implementation



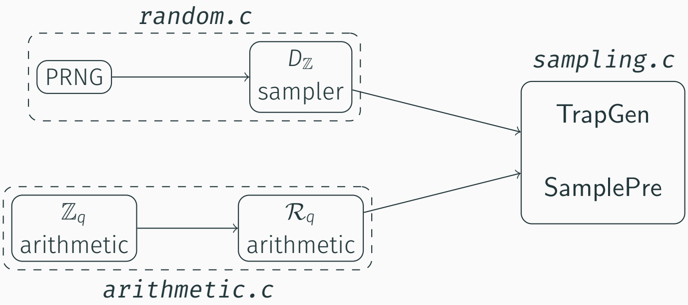
Modularity of the implementation

- C implementation **without any external library dependency**.



Modularity of the implementation

- C implementation **without any external library dependency**.
- Blocks can be **swapped out**.



Modularity of the implementation

- C implementation **without any external library dependency**.
- Blocks can be **swapped out**.
- Easy to modify the **arithmetic** on $\mathcal{R}_q$.

- **Partial NTT** to speed up polynomial arithmetic in $\mathcal{R}_q$.
- Representation of polynomials by their complex **CRT representation**.
- Efficient **low-degree FRD encoding** to map identities to matrices in $\mathcal{R}_q^{d\times d}$.

Table 1: Suggested parameter sets.

Parameter set	I	II	III	IV
nd	1024	1280	1536	2048
n	1024	256	512	2048
k	30	30	30	30
d	1	5	3	1
σ	7.00	5.55	6.15	6.85
α	48.34	54.35	60.50	67.40
ζ	83832	83290	112522	160778
BKZ blocksize b to break LWE	367	478	614	896
Classical security	107	139	179	262
Quantum security	97	126	163	237
BKZ blocksize b to break SIS	364	482	583	792
Classical security	106	140	170	231
Quantum security	96	127	154	210

Table 2: Timings of the different operations of our scheme: Setup, Extract, Encrpt, and Decrypt

Parameter Set	Setup	Extract	Encrypt	Decrypt
I	9.82 ms	16.54 ms	4.87 ms	0.99 ms
II	44.91 ms	18.09 ms	5.48 ms	1.04 ms

Table 3: Timings of the different operations for some IBE schemes.

Scheme	(λ, n)	Setup	Extract	Encrypt	Decrypt
BF-128	(128, –)	–	0.55 ms	7.51 ms	5.05 ms
DLP-14	$(80, 512)$	4.034 ms	3.8 ms	0.91 ms	0.62 ms

$\longrightarrow$ Less efficient but secure in the standard model and **without the NTRU assumption**.

$\longrightarrow$ Implementation of [BFR+18] **obsolete** + **limited security**.

Conclusion

Future problems

- Using **approximate sampling** techniques of [CGM19] to make the schemes faster and more compact.
- Adapting the schemes to achieve **adaptive security**.
- Using better **Integers Gaussian Samplers** to achieve better performance.

Thanks !

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, Lecture Notes in Computer Science. Springer, 2010.

[BFR+18] P. Bert, P.-A. Fouque, A. Roux-Langlois, and M. Sabt. Practical implementation of ring-sis/lwe based signature and IBE. In *PQCrypto*, Lecture Notes in Computer Science. Springer, 2018.

[CGM19] Y. Chen, N. Genise, and P. Mukherjee. Approximate trapdoors for lattices and smaller hash-and-sign signatures. In *ASIACRYPT (3)*, Lecture Notes in Computer Science. Springer, 2019.

[DLP14] L. Ducas, V. Lyubashevsky, and T. Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT (2)*, Lecture Notes in Computer Science. Springer, 2014.

[GM18] N. Genise and D. Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In *EUROCRYPT (1)*, Lecture Notes in Computer Science. Springer, 2018.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*. ACM, 2008.

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: simpler, tighter, faster, smaller. In Lecture Notes in Computer Science. Springer, 2012.