Fast Fourier Sampling and its Applications

Thomas Prest

PQShield

Lattices: From Theory to Practice

Motivation: GPV signatures over NTRU lattices

This talk: step ② of Sign

Keygen(1^λ)

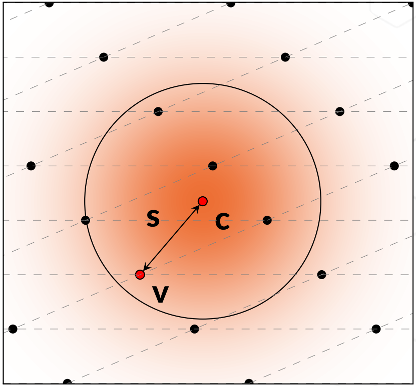
1. Gen. matrices $\mathbf{A}, \mathbf{B}$ s.t.:
   - $\mathbf{BA} = 0$
   - $\mathbf{B}$ has small coefficients
2. $pk := \mathbf{A}$, $sk := \mathbf{B}$

Sign$(msg, sk = \mathbf{B})$

1. Compute $\mathbf{c}$ such that $\mathbf{cA} = H(msg)$
2. $\mathbf{v} \leftarrow$ vector in $\mathcal{L}(\mathbf{B})$, close to $\mathbf{c}$
3. $sig := \mathbf{s} = (\mathbf{c} - \mathbf{v})$

$Verify(msg, pk = \mathbf{B}, sig = \mathbf{s})$

Check ($\mathbf{s}$ short) & $(\mathbf{sA} = H(msg))$



Motivation: GPV signatures over NTRU lattices

Thomas Pornin's talk: step ① of Keygen

Keygen(1^λ)

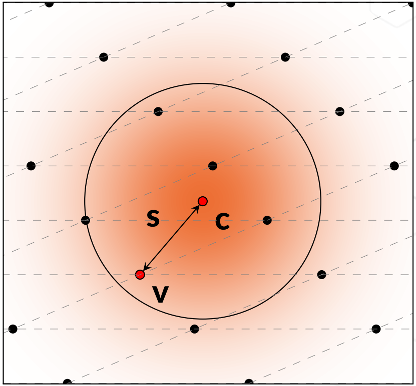
1. Gen. matrices $\mathbf{A}, \mathbf{B}$ s.t.:
   - $\mathbf{BA} = 0$
   - $\mathbf{B}$ has small coefficients
2. $pk := \mathbf{A}, sk := \mathbf{B}$

Sign$(msg, sk = \mathbf{B})$

1. Compute $\mathbf{c}$ such that $\mathbf{cA} = H(msg)$
2. $\mathbf{v} \leftarrow$ vector in $\mathcal{L}(\mathbf{B})$, close to $\mathbf{c}$
3. $sig := \mathbf{s} = (\mathbf{c} - \mathbf{v})$

Verify$(msg, pk = \mathbf{B}, sig = \mathbf{s})$

Check ($\mathbf{s}$ short) & $(\mathbf{sA} = H(msg))$



Motivation: GPV signatures over NTRU lattices

Alexandre's talk: the whole enchilada

Keygen(1^λ)

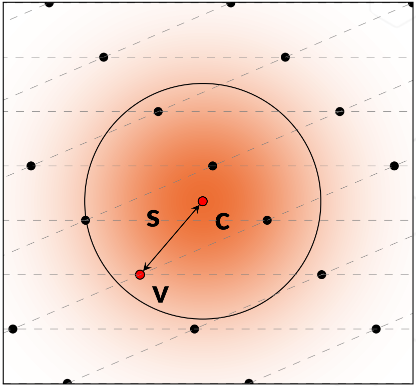
1. Gen. matrices $\mathbf{A}, \mathbf{B}$ s.t.:
   - $\mathbf{BA} = 0$
   - $\mathbf{B}$ has small coefficients
2. $pk := \mathbf{A}$, $sk := \mathbf{B}$

Sign$(msg, sk = \mathbf{B})$

1. Compute $\mathbf{c}$ such that $\mathbf{cA} = H(msg)$
2. $\mathbf{v} \leftarrow$ vector in $\mathcal{L}(\mathbf{B})$, close to $\mathbf{c}$
3. $sig := \mathbf{s} = (\mathbf{c} - \mathbf{v})$

$Verify(msg, pk = \mathbf{B}, sig = \mathbf{s})$

Check ($\mathbf{s}$ short) & ($\mathbf{sA} = H(msg)$)



Focus of this talk:

Given **B** *and* **c**, *how do we efficiently (and securely) compute* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *close to* **v**?

Two parts:

1. Fast Fourier orthogonalization [DP16]
   - Purely algorithmic/algebraic
2. From FFO to fast Fourier sampling [Pre17, PFH^{+}17]
   - Statistical arguments (Rényi divergence)

Fast Fourier Orthogonalization

Gram-Schmidt orth. (GSO):

Given $\mathbf{B} \in \mathbb{R}^{n\times m}$ full-rank, compute:

$$\mathbf{B} = \mathbf{L} \times \tilde{\mathbf{B}} \qquad (1)$$

where:

- $\mathbf{L}$ is lower triangular with 1's on its diagonal
- $\tilde{\mathbf{B}}$ has orthogonal rows

Can be done in time $O(mn^2)$

LDL decomposition:

Given $\mathbf{G} \in \mathbb{C}^{n\times n}$ self-adjoint (i.e. $\mathbf{G}^* = \mathbf{G}$), compute:

$$\mathbf{G} = \mathbf{L} \times \tilde{\mathbf{D}} \times \mathbf{L}^* \qquad (2)$$

where:

- $\mathbf{L}$ is lower triangular with 1's on its diagonal
- $\mathbf{D}$ is diagonal

Can be done in time $O(n^3)$

Gram-Schmidt orth. (GSO):

Given $\mathbf{B} \in \mathbb{R}^{n \times m}$ full-rank, compute:

$$\mathbf{B} = \mathbf{L} \times \tilde{\mathbf{B}} \qquad (1)$$

where:

- $\mathbf{L}$ is lower triangular with 1's on its diagonal
- $\tilde{\mathbf{B}}$ has orthogonal rows

Can be done in time $O(mn^2)$

LDL decomposition:

Given $\mathbf{G} \in \mathbb{C}^{n \times n}$ self-adjoint (i.e. $\mathbf{G}^* = \mathbf{G}$), compute:

$$\mathbf{G} = \mathbf{L} \times \tilde{\mathbf{D}} \times \mathbf{L}^* \qquad (2)$$

where:

- $\mathbf{L}$ is lower triangular with 1's on its diagonal
- $\mathbf{D}$ is diagonal

Can be done in time $O(n^3)$

Fun fact 1: When $\mathbf{G} = \mathbf{B} \times \mathbf{B}^*$, the GSO and LDL are equivalent.

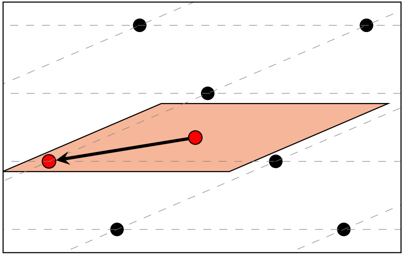
Fun fact 2: The GSO and LDL generalize to rings/fields of the form $\mathbb{Q}[x]/(\phi)$ with adequate definitions of adjoint/inner product.

How to compute efficiently a close vector:

RoundOff(B, c)

1. $\mathbf{t} \leftarrow \mathbf{c} \cdot \mathbf{B}^{-1}$
2. For $j \in \{n, \ldots, 1\}$:
   1. $z_j \leftarrow \lceil t_j \rfloor$
3. Return $\mathbf{v} := \mathbf{z} \cdot \mathbf{B}$

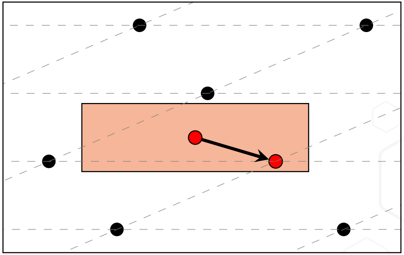
$\Downarrow$



NearestPlane(B, L, c)

1. $\mathbf{t} \leftarrow \mathbf{c} \cdot \mathbf{B}^{-1}$
2. For $j \in \{n, \ldots, 1\}$:
   1. $z_j \leftarrow \lceil t_j + \sum_{i>j} (t_1 - z_i) L_{i,j} \rfloor$
3. Return $\mathbf{v} := \mathbf{z} \cdot \mathbf{B}$

$\Downarrow$



It is common to take matrices/vectors with coefficients in $R = \mathbb{Z}_q[x]/(\phi)$, where ϕ can be:

1. A convolution polynomial $x^n - 1$
2. A cyclotomic polynomial, e.g. $x^n + 1$ for n a power-of-two
3. Another polynomial, e.g. $x^p - x - 1$ as in NTRU Prime

The techniques we describe provide speed-ups for subsets of 1 and 2 (tower of rings), but not 3.

We focus on $\phi = x^n + 1$ with $n = 2^\kappa$, and note $\underline{\mathbb{K}_n = \mathbb{Q}[x]/(x^n + 1)}$.

It is common to take matrices/vectors with coefficients in $R = \mathbb{Z}_q[x]/(\phi)$, where ϕ can be:

1. A convolution polynomial $x^n - 1$
2. A cyclotomic polynomial, e.g. $x^n + 1$ for n a power-of-two
3. Another polynomial, e.g. $x^p - x - 1$ as in NTRU Prime

The techniques we describe provide speed-ups for subsets of 1 and 2 (tower of rings), but not 3.

We focus on $\phi = x^n + 1$ with $n = 2^\kappa$, and note $\underline{\mathbb{K}_n = \mathbb{Q}[x]/(x^n + 1)}$.

Our goal: provide a faster NearestPlane algorithm over towers of rings:

1. Compact representation of the orthogonalization
2. How to use this compact representation

Generalize to module lattices (with base ring a tower of rings).

If no obvious way to exploit the ring structure, one can map everything to $\mathbb{Z}$ (or $\mathbb{Q}$). For example, this ring endomorphism

$$\begin{array}{rccl} T: & \mathbb{K}_4 & \to & \mathbb{K}_4 \\ & \mathbf{g}(x) & \mapsto & (a + bx + cx^2 + dx^3) \cdot \mathbf{g}(x) \end{array}$$

can be interpreted as the endomorphism of $\mathbb{Q}^4$ with this associated matrix over the canonical basis $\{1, x, x^2, x^3\}$:

$$\begin{array}{cccc} 1 & x & x^2 & x^3 \end{array}$$

$$\begin{pmatrix} a & b & c & d \\ -d & a & b & c \\ -c & -d & a & b \\ -b & -c & -d & a \end{pmatrix} \begin{array}{c} 1 \\ x \\ x^2 \\ x^3 \end{array}$$

Problem: the power basis is not adequate for GSO/LDL!

$$\underbrace{\begin{bmatrix} a & b & c & d \\ -d & a & b & c \\ -c & -d & a & b \\ -b & -c & -d & a \end{bmatrix}}_{\mathbf{B}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ * & 1 & 0 & 0 \\ * & * & 1 & 0 \\ * & * & * & 1 \end{bmatrix}}_{\mathbf{L}} \times \underbrace{\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}}_{\tilde{\mathbf{B}}}$$

Consequence: not obvious that the ring structure provide a gain $\tilde{O}(n)$:

- in storage (storing **L**)
- in computation (using **L** in NearestPlane())

Lets find a better representation!

Observation: Representing T in the basis $\{1, x^2, x, x^3\}$ instead of $\{1, x, x^2, x^3\}$ gives:

$$\left[\begin{array}{cc|cc} a & c & b & d \\ -c & a & -d & b \\ \hline -d & b & a & c \\ -b & -d & -c & a \end{array}\right] \tag{3}$$

More formally: If we write $\boldsymbol{f} \in \mathbb{K}_n$ in the $\mathbb{K}_{n/2}$-basis $\{1, x\}$:

$$\boldsymbol{f}(x) = \boldsymbol{f}_0(x^2) + x \cdot \boldsymbol{f}_1(x^2) \tag{4}$$

with $\boldsymbol{f}_0, \boldsymbol{f}_1 \in \mathbb{K}_{n/2}$, the transformation matrix of $T : \boldsymbol{g} \in \mathbb{K}_n \mapsto \boldsymbol{f} \cdot \boldsymbol{g}$ is:

$$\left[\begin{array}{c|c} \boldsymbol{f}_0 & \boldsymbol{f}_1 \\ \hline x \cdot \boldsymbol{f}_1 & \boldsymbol{f}_0 \end{array}\right] \tag{5}$$

Note: This change of basis is a ring morphism that is also an isometry!

Fun fact 3: Distinct morphisms allow various levels of granularity.

$$
\overbrace{\begin{bmatrix}
a & b & c & d & 0 & -d & -c & -b \\
b & a & b & c & d & 0 & -d & -c \\
c & b & a & b & c & d & 0 & -d \\
d & c & b & a & b & c & d & 0 \\
0 & d & c & b & a & b & c & d \\
-d & 0 & d & c & b & a & b & c \\
-c & -d & 0 & d & c & b & a & b \\
-b & -c & -d & 0 & d & c & b & a
\end{bmatrix}}^{\{1, x, \ldots, x^7\}}
\Rightarrow
\overbrace{\left[\begin{array}{cccc|cccc}
a & c & 0 & -c & b & d & -d & -b \\
c & a & c & 0 & b & b & d & -d \\
0 & c & a & c & d & b & b & d \\
-c & 0 & c & a & -d & d & b & b \\
\hline
b & b & d & -d & a & c & 0 & -c \\
d & b & b & d & c & a & c & 0 \\
-d & d & b & b & 0 & c & a & c \\
-b & -d & d & b & -c & 0 & c & a
\end{array}\right]}^{\{1, x^2, x^4, x^6 | x, x^3, x^5, x^7\}}
$$

Fun fact 3: Distinct morphisms allow various levels of granularity.

$$\overbrace{\begin{bmatrix} a & b & c & d & 0 & -d & -c & -b \\ b & a & b & c & d & 0 & -d & -c \\ c & b & a & b & c & d & 0 & -d \\ d & c & b & a & b & c & d & 0 \\ 0 & d & c & b & a & b & c & d \\ -d & 0 & d & c & b & a & b & c \\ -c & -d & 0 & d & c & b & a & b \\ -b & -c & -d & 0 & d & c & b & a \end{bmatrix}}^{\{1, x, \ldots, x^7\}} \Rightarrow \overbrace{\left[\begin{array}{cc|cc|cc|cc} a & 0 & c & -c & b & -d & d & -b \\ 0 & a & c & c & d & b & b & d \\ \hline c & c & a & 0 & b & d & b & -d \\ -c & c & 0 & a & -d & b & d & b \\ \hline b & d & b & -d & a & 0 & c & -c \\ -d & b & d & b & 0 & a & c & c \\ \hline d & b & b & d & c & c & a & 0 \\ -b & d & -d & b & -c & c & 0 & a \end{array}\right]}^{\{1, x^4 \mid x^2, x^6 \mid x, x^5 \mid x^3, x^7\}}$$

Fun fact 3: Distinct morphisms allow various levels of granularity.

$$\overbrace{\begin{bmatrix} a & b & c & d & 0 & -d & -c & -b \\ b & a & b & c & d & 0 & -d & -c \\ c & b & a & b & c & d & 0 & -d \\ d & c & b & a & b & c & d & 0 \\ 0 & d & c & b & a & b & c & d \\ -d & 0 & d & c & b & a & b & c \\ -c & -d & 0 & d & c & b & a & b \\ -b & -c & -d & 0 & d & c & b & a \end{bmatrix}}^{\{1, x, \ldots, x^7\}} \Rightarrow \overbrace{\left[\begin{array}{cc|cc|cc|cc} a & 0 & c & -c & b & -d & d & -b \\ 0 & a & c & c & d & b & b & d \\ \hline c & c & a & 0 & b & d & b & -d \\ -c & c & 0 & a & -d & b & d & b \\ \hline b & d & b & -d & a & 0 & c & -c \\ -d & b & d & b & 0 & a & c & c \\ \hline d & b & b & d & c & c & a & 0 \\ -b & d & -d & b & -c & c & 0 & a \end{array}\right]}^{\{1, x^4 \mid x^2, x^6 \mid x, x^5 \mid x^3, x^7\}}$$

So what's the point? We combine the facts 1 to 3:

1. $GSO \Leftrightarrow LDL$
2. We can generalize the GSO/LDL to rings like $\mathbb{K}_n$
3. $\mathbb{K}_n \cong (\mathbb{K}_{n'})^{n/n'}$ via an isomorphism that is also an isometry.

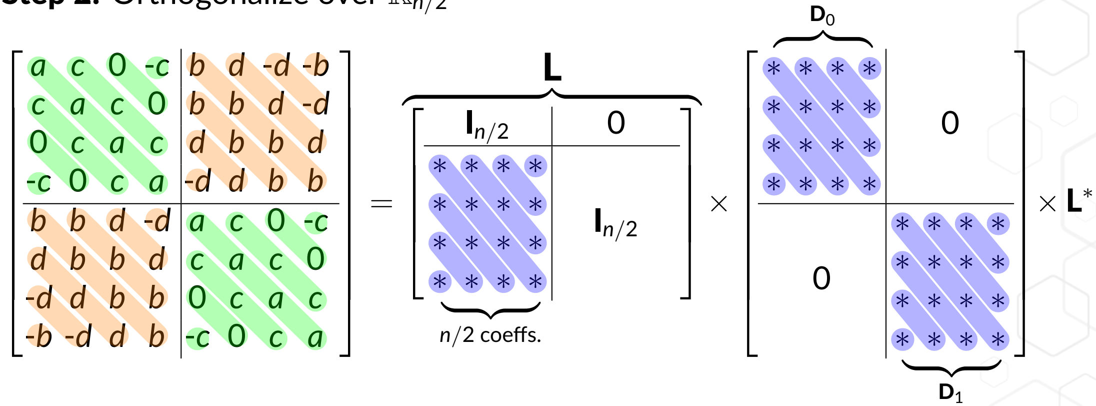
Suppose we have a nega-circulant Gram matrix.

Step 1: “break” the matrix

$$\begin{bmatrix} \mathrm{a} & \mathrm{b} & \mathrm{c} & \mathrm{d} & 0 & -\mathrm{d} & -\mathrm{c} & -\mathrm{b} \\ \mathrm{b} & \mathrm{a} & \mathrm{b} & \mathrm{c} & \mathrm{d} & 0 & -\mathrm{d} & -\mathrm{c} \\ \mathrm{c} & \mathrm{b} & \mathrm{a} & \mathrm{b} & \mathrm{c} & \mathrm{d} & 0 & -\mathrm{d} \\ \mathrm{d} & \mathrm{c} & \mathrm{b} & \mathrm{a} & \mathrm{b} & \mathrm{c} & \mathrm{d} & 0 \\ 0 & \mathrm{d} & \mathrm{c} & \mathrm{b} & \mathrm{a} & \mathrm{b} & \mathrm{c} & \mathrm{d} \\ -\mathrm{d} & 0 & \mathrm{d} & \mathrm{c} & \mathrm{b} & \mathrm{a} & \mathrm{b} & \mathrm{c} \\ -\mathrm{c} & -\mathrm{d} & 0 & \mathrm{d} & \mathrm{c} & \mathrm{b} & \mathrm{a} & \mathrm{b} \\ -\mathrm{b} & -\mathrm{c} & -\mathrm{d} & 0 & \mathrm{d} & \mathrm{c} & \mathrm{b} & \mathrm{a} \end{bmatrix} \Rightarrow \left[\begin{array}{cccc|cccc} a & c & 0 & -c & b & d & -d & -b \\ c & a & c & 0 & b & b & d & -d \\ 0 & c & a & c & d & b & b & d \\ -c & 0 & c & a & -d & d & b & b \\ \hline b & b & d & -d & a & c & 0 & -c \\ d & b & b & d & c & a & c & 0 \\ -d & d & b & b & 0 & c & a & c \\ -b & -d & d & b & -c & 0 & c & a \end{array}\right]$$

Suppose we have a nega-circulant Gram matrix.

Step 1: "break" the matrix
Step 2: Orthogonalize over $\mathbb{K}_{n/2}$



Step 3: Store non-trivial coeffs of **L** and recurse on $\mathbf{D}_0, \mathbf{D}_1$.

Complexity: $O(n \log n)$ in storage and computation (always stay in FFT).

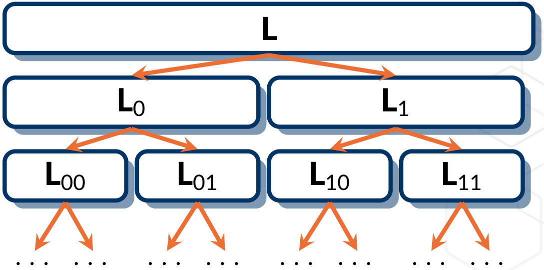
FFNearestPlane$(\mathbf{T}, \boldsymbol{t})$ – informal

1. If base field is $\mathbb{Q}$, compute $\boldsymbol{z} \leftarrow \text{NearestPlane}(\mathbf{I}, \mathbf{L}_{\text{leaf}} = \mathbf{T}.\text{value}, \boldsymbol{t})$
2. Else, split $\boldsymbol{t}$ in $(\boldsymbol{t}_0, \boldsymbol{t}_1)$
   1. $\boldsymbol{z}_1 \leftarrow \text{FFNearestPlane}(\mathbf{T}_{\text{rightchild}}, \boldsymbol{t}_1)$
   2. $\bar{\boldsymbol{t}}_0 \leftarrow \boldsymbol{t}_0 + (\boldsymbol{t}_1 - \boldsymbol{z}_1) \cdot \mathbf{L}$ [with $\mathbf{L} = \mathbf{T}.\text{value}$]
   3. $\boldsymbol{z}_0 \leftarrow \text{FFNearestPlane}(\mathbf{T}_{\text{leftchild}}, \bar{\boldsymbol{t}}_0)$

   Return $\boldsymbol{z}$

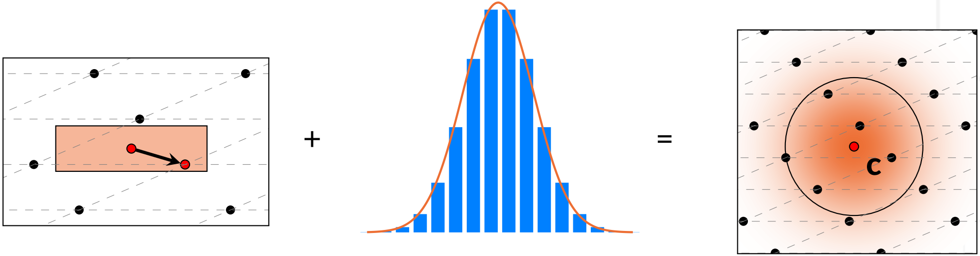
Orthogonalization data can be stored in a tree **T**:

- Computing **T** on-the-fly reduces storage cost to $O(n)$ [PFH+17, GM18, OSHG19, Por19]
- By tweaking ②, $\boldsymbol{t}$ and $\boldsymbol{z}$ can share the same buffer (no $\bar{\boldsymbol{t}}_0$)



Fast Fourier Sampling

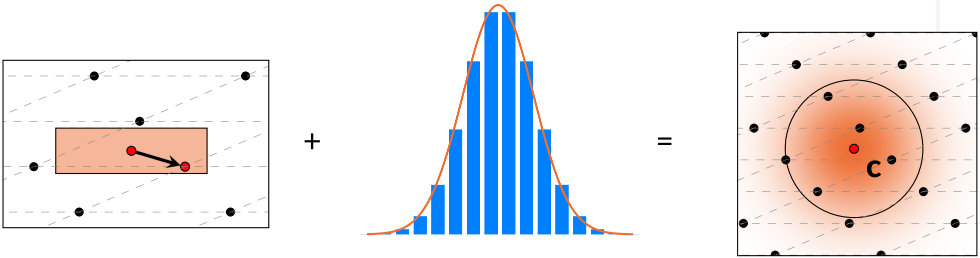
To make (fast Fourier) nearest plane secure, combine it with Gaussians:



Applications:

- Signatures (Falcon [PFH+17])
- (H)IBE (ETSI proposal LATTE)
- Ring signatures (Raptor [LAZ19])
- Group signatures [dLS18], etc.

To make (fast Fourier) nearest plane secure, combine it with Gaussians:



Applications:

- Signatures (Falcon [PFH+17])
- (H)IBE (ETSI proposal LATTE)
- Ring signatures (Raptor [LAZ19])
- Group signatures [dLS18], etc.

Practical questions:

- How large should be the Gaussian?
- What about the floating-point precision?

We address both questions w/ a Rényi divergence analysis [BLL+15, Pre17].

The Rényi Divergence

Definition. For $\alpha \in (1, +\infty)$, the Rényi divergence between two distributions $\mathcal{P}, \mathcal{Q}$ is

$$R_\alpha(\mathcal{P} \| \mathcal{Q}) = \left(\sum_{x \in Supp(\mathcal{P})} \frac{\mathcal{P}(x)^\alpha}{\mathcal{Q}(x)^{\alpha-1}} \right)^{\frac{1}{\alpha-1}} \tag{6}$$

Motivation. Consider a scheme doing q queries to a distribution $\mathcal{D}_i$, note ϵ_i the prob. of an event breaking the scheme and $\epsilon_{Ideal} = 2^{-\lambda}$.

- With the statistical distance:

$$\epsilon_{Ideal} \geq \epsilon_{Real} - q\Delta_{SD}(\mathcal{D}_{Real}, \mathcal{D}_{Ideal}) \qquad \boxed{\text{Take } \Delta_{SD} \leq 2^{-\lambda}} \tag{7}$$

- With the Rényi divergence:

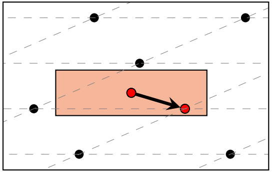
$$\epsilon_{Ideal} \geq \epsilon_{Real}^{\frac{\alpha}{\alpha-1}} / R_\alpha(\mathcal{D}_{Real} \| \mathcal{D}_{Ideal})^q \qquad \boxed{\text{Take } (\alpha \geq \lambda) \ \& \ (R_\alpha \leq 1 + 1/q)} \tag{8}$$

Use when: Search problem + moderate number of queries (e.g. $\leq 2^{64}$)

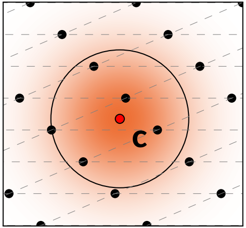
Picking the right σ

We combine FFNearestPlane with Gaussian rounding to (hopefully) obtain a discretized Gaussian of standard deviation σ.

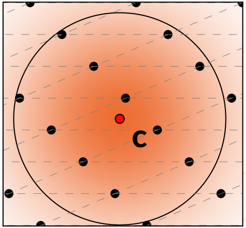
σ too small



The "right" σ

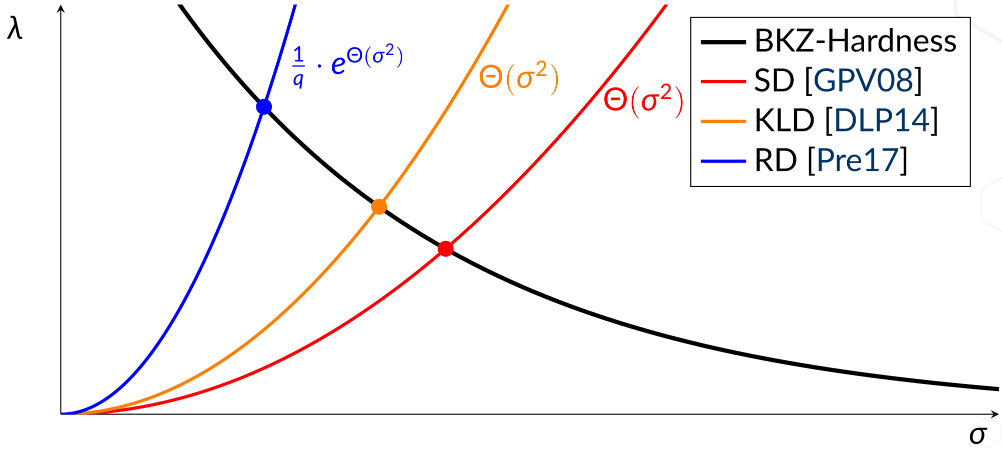


σ too big



1. σ too small $\Rightarrow$ vulnerable to learning attacks [NR06, DN12]
2. σ too large $\Rightarrow$ suboptimal for cryptography

Standard deviation analysis



For the example of Falcon and $q = 2^{64}$, we gain about 30 bits of security (compared to the SD).

Precision analysis

We note *Ideal* (resp. *Real*) the output of fast Fourier sampling with infinite (resp. finite) precision.

Statistical distance analysis: If the (absolute) precision loss is $|x - \bar{x}| < \delta$:

$$\Delta_{SD}(Real, Ideal) = \delta \cdot poly(n, \dots) \tag{9}$$

This entails a *bit* precision of $\lambda + polylog(n, \dots)$, unacceptable in practice.

Rényi divergence analysis: Under the same conditions:

$$R_\alpha(Real \| Ideal) = 1 + \alpha \cdot \delta^2 \cdot poly(n, \dots) \tag{10}$$

Combining that with:

$$R_\alpha(Real \| Ideal)^q \cdot \varepsilon_{Ideal} \geq \varepsilon_{Real}^{\alpha/(\alpha-1)} \tag{11}$$

gives a *bit* precision of $(\log_2 \lambda q)/2 + polylog(n, \dots)$ for a security loss $O(1)$.

	Statistical distance	**Rényi divergence**
Sec = f(std dev)	$\lambda = \Theta(\sigma^2)$	$\lambda = \frac{1}{q} \cdot e^{\Theta(\sigma^2)}$
Bit precision	$\lambda + polylog(\ldots)$	$\frac{\log_2 \lambda q}{2} + polylog(\ldots)$

Conclusion

Related works:

- *Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus* [GM18]
  - Applies similar ideas to the Micciancio-Peikert framework
- *Algebraic and Euclidean Lattices: Optimal Lattice Reduction and Beyond* [KEF19]
  - Applies similar ideas to LLL over tower rings

Related works:

- *Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus* [GM18]
  - Applies similar ideas to the Micciancio-Peikert framework
- *Algebraic and Euclidean Lattices: Optimal Lattice Reduction and Beyond* [KEF19]
  - Applies similar ideas to LLL over tower rings

Open questions:

1. Cryptanalytic applications beyond [KEF19]?
2. Getting rid of floating-point arithmetic?
   1. Micciancio-Peikert trapdoors?
   2. Iterating from [DGPY19]?
3. Masking?

Questions?

Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld.
Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance.
In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 3–24. Springer, Heidelberg, November / December 2015.

Léo Ducas, Steven Galbraith, Thomas Prest, and Yang Yu.
Integral matrix gram root and lattice Gaussian sampling without floats.
Cryptology ePrint Archive, Report 2019/320, 2019.
https://eprint.iacr.org/2019/320.

Léo Ducas, Vadim Lyubashevsky, and Thomas Prest.
Efficient identity-based encryption over NTRU lattices.
In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41. Springer, Heidelberg, December 2014.

Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler.

Lattice-based group signatures and zero-knowledge proofs of automorphism stability.
In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 574–591. ACM Press, October 2018.

Léo Ducas and Phong Q. Nguyen.
Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures.
In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 433–450. Springer, Heidelberg, December 2012.

Léo Ducas and Thomas Prest.
Fast fourier orthogonalization.
In Sergei A. Abramov, Eugene V. Zima, and Xiao-Shan Gao, editors, *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016*, pages 191–198. ACM, 2016.

Nicholas Genise and Daniele Micciancio.
Faster Gaussian sampling for trapdoor lattices with arbitrary modulus.

In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 174–203. Springer, Heidelberg, April / May 2018.

Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan.
Trapdoors for hard lattices and new cryptographic constructions.
In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

Paul Kirchner, Thomas Espitau, and Pierre-Alain Fouque.
Algebraic and euclidean lattices: Optimal lattice reduction and beyond.
Cryptology ePrint Archive, Report 2019/1436, 2019.
https://eprint.iacr.org/2019/1436.

Xingye Lu, Man Ho Au, and Zhenfei Zhang.
Raptor: A practical lattice-based (linkable) ring signature.
In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 110–130. Springer, Heidelberg, June 2019.

Phong Q. Nguyen and Oded Regev.

Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures.
In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 271–288. Springer, Heidelberg, May / June 2006.

Tobias Oder, Julian Speith, Kira Höltgen, and Tim Güneysu.
Towards practical microcontroller implementation of the signature scheme Falcon.
In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 65–80. Springer, Heidelberg, 2019.

Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang.
FALCON.
Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

Thomas Pornin.

New efficient, constant-time implementations of Falcon.
Cryptology ePrint Archive, Report 2019/893, 2019.
https://eprint.iacr.org/2019/893.

Thomas Prest.
Sharper bounds in lattice-based cryptography using the Rényi divergence.
In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 347–374. Springer, Heidelberg, December 2017.