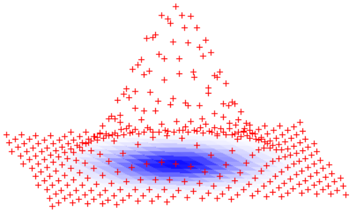
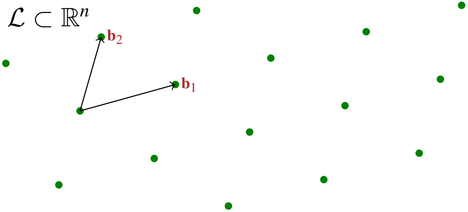
An Efficient and Parallel Gaussian Sampler for Lattices



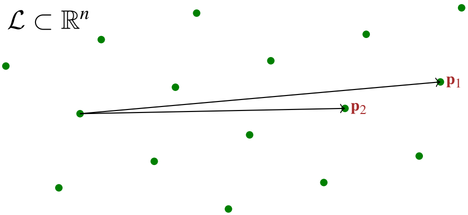
Chris Peikert
Georgia Tech

CRYPTO 2010

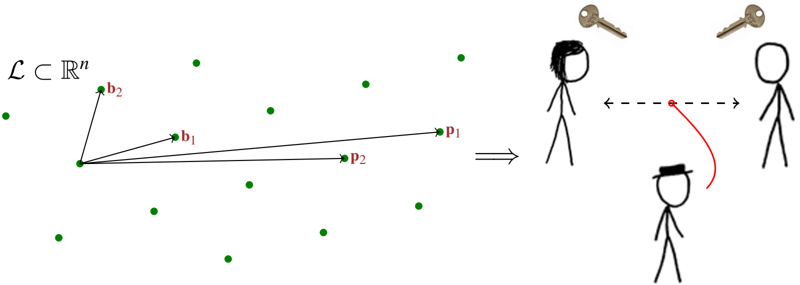
Lattice-Based Crypto



Lattice-Based Crypto

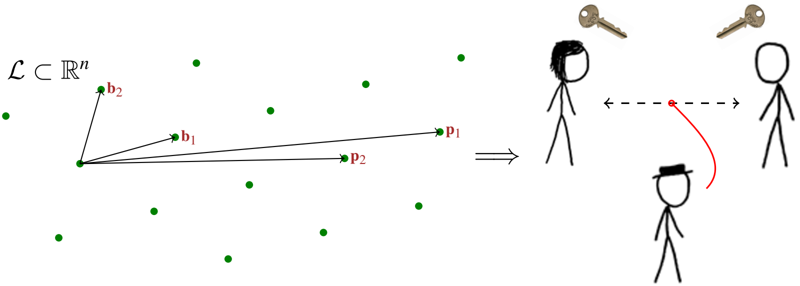


Lattice-Based Crypto



(Images courtesy xkcd.org)

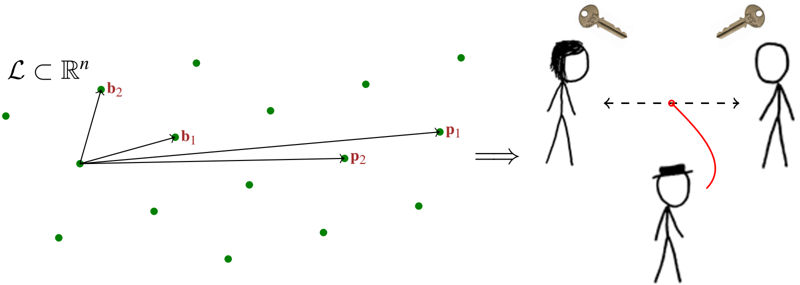
Lattice-Based Crypto



✔ Asymptotically efficient & highly parallelizable

(Images courtesy xkcd.org)

Lattice-Based Crypto

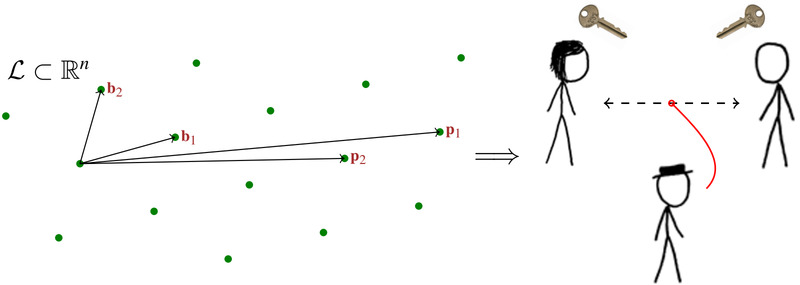


✔ Asymptotically efficient & highly parallelizable

✔ Worst-case assumptions (& quantum-resistant?) [Ajtai'96,. . .]

(Images courtesy xkcd.org)

Lattice-Based Crypto

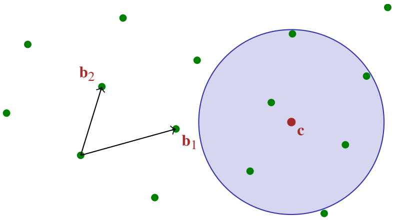


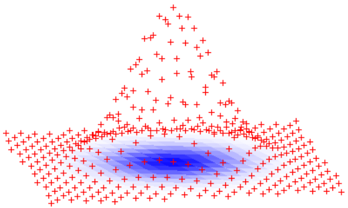
- ✔ Asymptotically efficient & highly parallelizable
- ✔ Worst-case assumptions (& quantum-resistant?) [Ajtai'96, . . .]
- ✔ Many rich applications:
  - ★ 'Hash-and-sign' signatures [GPV'08, CHKP'10, R'10, B'10]
  - ★ (Hierarchical) IBE [GPV'08, CHKP'10, ABB'10a, ABB'10b]
  - ★ Fully homomorphic encryption [G'09, SV'10, vDGHV'10]

(Images courtesy xkcd.org)

Gaussian Sampling on Lattices

- Given 'good' basis $\mathbf{B}$ and center $\mathbf{c}$, sample discrete Gaussian on $\mathcal{L}$

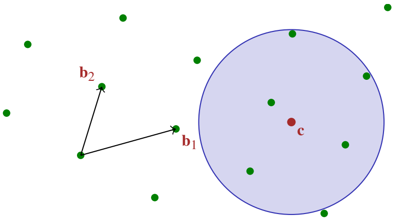


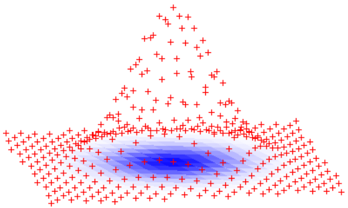


[B'93,R'03,AR'04,MR'04,...]

Gaussian Sampling on Lattices

- Given 'good' basis $\mathbf{B}$ and center $\mathbf{c}$, sample discrete Gaussian on $\mathcal{L}$
  - ⋆ 'Zero-knowledge' operation: leaks no information about $\mathbf{B}$ [GPV'08]

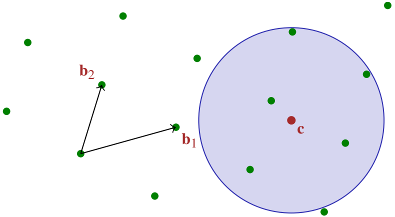


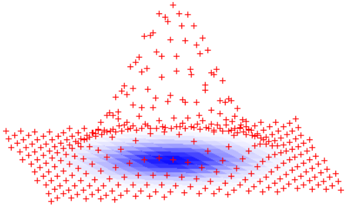


[B'93,R'03,AR'04,MR'04,. . .]

Gaussian Sampling on Lattices

- Given 'good' basis $\mathbf{B}$ and center $\mathbf{c}$, sample discrete Gaussian on $\mathcal{L}$
  - ⋆ 'Zero-knowledge' operation: leaks <u>no information</u> about $\mathbf{B}$ [GPV'08]





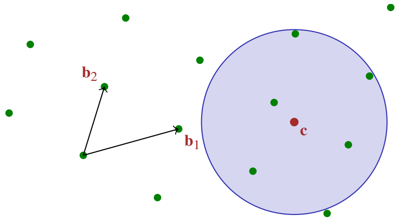
[B'93,R'03,AR'04,MR'04,. . .]

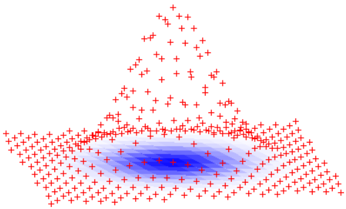
Crypto Applications

- 'Answering queries:' signing, (H)IBE key extraction, (NI)ZK

Gaussian Sampling on Lattices

- Given 'good' basis $\mathbf{B}$ and center $\mathbf{c}$, sample discrete Gaussian on $\mathcal{L}$
  - ⋆ 'Zero-knowledge' operation: leaks <u>no information</u> about $\mathbf{B}$ [GPV'08]





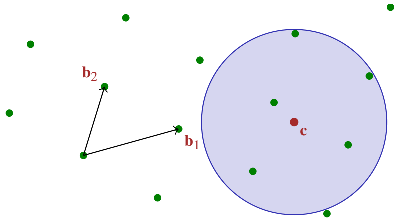
[B'93,R'03,AR'04,MR'04,. . .]

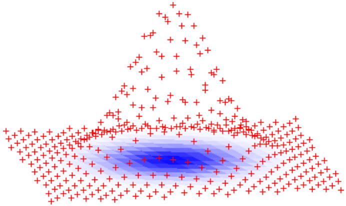
Crypto Applications

- 'Answering queries:' signing, (H)IBE key extraction, (NI)ZK
- Worst-case / average-case reductions [GPV'08,P'09,LPR'10,G'10]

Gaussian Sampling on Lattices

- Given 'good' basis $\mathbf{B}$ and center $\mathbf{c}$, sample discrete Gaussian on $\mathcal{L}$
  - ⋆ 'Zero-knowledge' operation: leaks no information about $\mathbf{B}$ [GPV'08]





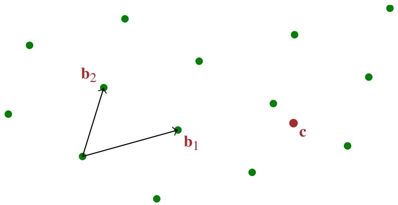
[B'93,R'03,AR'04,MR'04,...]

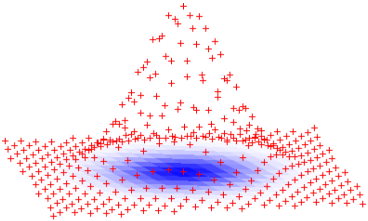
Crypto Applications

- 'Answering queries:' signing, (H)IBE key extraction, (NI)ZK
- Worst-case / average-case reductions [GPV'08,P'09,LPR'10,G'10]
- Narrower Gaussian ⇒ smaller keys ⇒ more efficient schemes

The GPV Sampling Algorithm

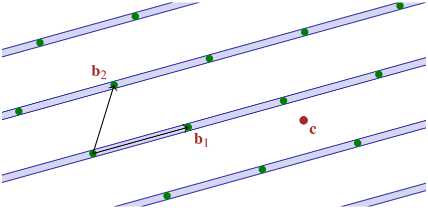
- 'Nearest-plane' algorithm w/ randomized rounding [Babai'86,Klein'00]

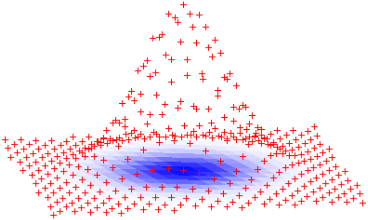




The GPV Sampling Algorithm

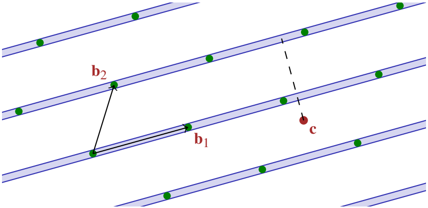
- ‘Nearest-plane’ algorithm w/ randomized rounding [Babai’86,Klein’00]

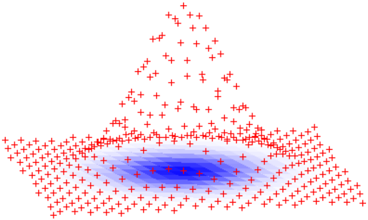




The GPV Sampling Algorithm

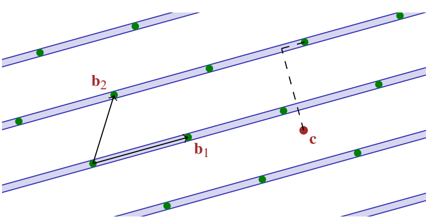
- 'Nearest-plane' algorithm w/ randomized rounding [Babai'86,Klein'00]

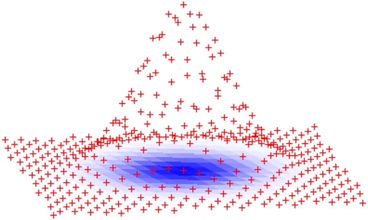




The GPV Sampling Algorithm

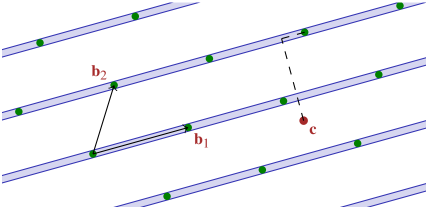
- ‘Nearest-plane’ algorithm w/ randomized rounding [Babai’86,Klein’00]

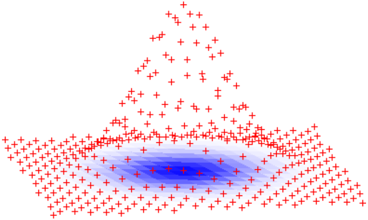




The GPV Sampling Algorithm

- 'Nearest-plane' algorithm w/ randomized rounding [Babai'86,Klein'00]

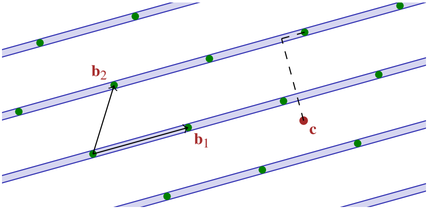


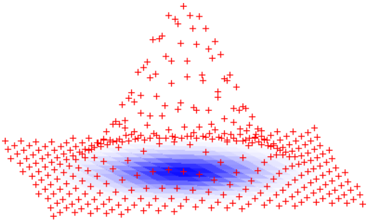


Good News, and Bad News...

The GPV Sampling Algorithm

- 'Nearest-plane' algorithm w/ randomized rounding [Babai'86,Klein'00]



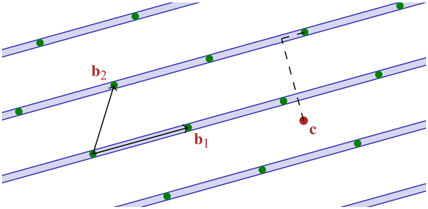


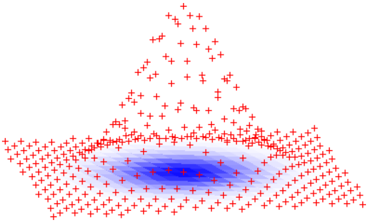
Good News, and Bad News...

✔ Narrow: width $\approx \max\|\widetilde{\mathbf{b}_i}\|$ = max dist between adjacent 'planes'

The GPV Sampling Algorithm

- 'Nearest-plane' algorithm w/ randomized rounding [Babai'86,Klein'00]





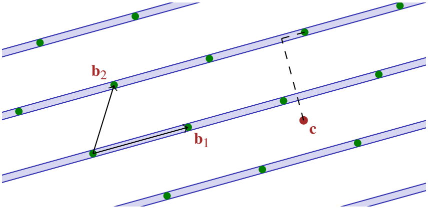
Good News, and Bad News. . .

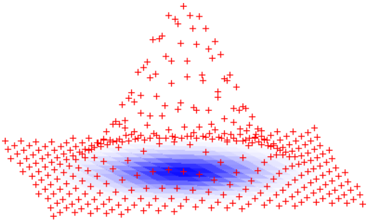
✔ Narrow: width $\approx \max \|\tilde{\mathbf{b}}_i\|$ = max dist between adjacent 'planes'

✗ Not efficient: time = $\Omega(n^3)$, high-precision real arithmetic

The GPV Sampling Algorithm

- 'Nearest-plane' algorithm w/ randomized rounding [Babai'86,Klein'00]





Good News, and Bad News...

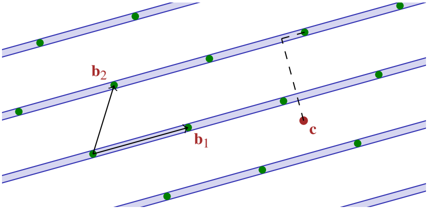
✔ Narrow: width $\approx \max \|\widetilde{\mathbf{b}_i}\|$ = max dist between adjacent 'planes'

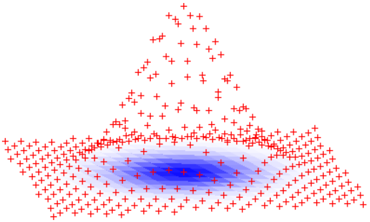
✗ Not efficient: time = $\Omega(n^3)$, high-precision real arithmetic

✗ Inherently sequential: n adaptive iterations

The GPV Sampling Algorithm

- 'Nearest-plane' algorithm w/ randomized rounding [Babai'86,Klein'00]





Good News, and Bad News...

- ✔ Narrow: width $\approx \max \|\widetilde{\mathbf{b}_i}\|$ = max dist between adjacent 'planes'
- ✗ Not efficient: time = $\Omega(n^3)$, high-precision real arithmetic
- ✗ Inherently sequential: n adaptive iterations
- ✗ No efficiency improvement for ring-based crypto [NTRU'98,M'02,...]

Our Contributions

1. A new Gaussian sampling algorithm for lattices.

Our Contributions

1. A new Gaussian sampling algorithm for lattices.

Key Features

- ⋆ Simple & efficient: $\approx 4n^2$ online adds and mults, modulo a small integer

Our Contributions

1. A new Gaussian sampling algorithm for lattices.

Key Features

- ⋆ Simple & efficient: $\approx 4n^2$ online adds and mults, modulo a small integer

  Even better: $\tilde{O}(n)$ time for ring-based schemes!

Our Contributions

1. A new Gaussian sampling algorithm for lattices.

Key Features

- ⋆ Simple & efficient: $\approx 4n^2$ online adds and mults, modulo a small integer

  Even better: $\tilde{O}(n)$ time for ring-based schemes!

- ⋆ Fully parallelizable: n^2/P operations on each of $P \leq n^2$ processors

Our Contributions

1. A new Gaussian sampling algorithm for lattices.

Key Features

- ⋆ Simple & efficient: $\approx 4n^2$ online adds and mults, modulo a small integer

  Even better: $\tilde{O}(n)$ time for ring-based schemes!

- ⋆ Fully parallelizable: n^2/P operations on each of $P \leq n^2$ processors

- ⋆ High quality: for crypto lattices, same* Gaussian width as GPV

Our Contributions

1. A new Gaussian sampling algorithm for lattices.

Key Features

- ⋆ Simple & efficient: $\approx 4n^2$ online adds and mults, modulo a small integer

  Even better: $\tilde{O}(n)$ time for ring-based schemes!

- ⋆ Fully parallelizable: n^2/P operations on each of $P \leq n^2$ processors

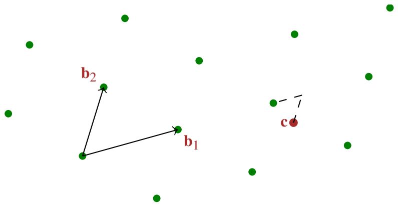
- ⋆ High quality: for crypto lattices, same* Gaussian width as GPV

2. A general 'convolution theorem' for discrete Gaussians.

   Other applications: LWE error distribution,
   bi-deniable encryption [OP'10], . . .

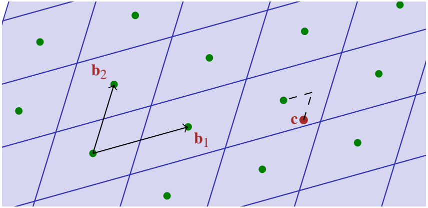
A First Attempt

- [Babai'86] 'simple rounding:' $\mathbf{c} \mapsto \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil$. (Fast & Parallel!)



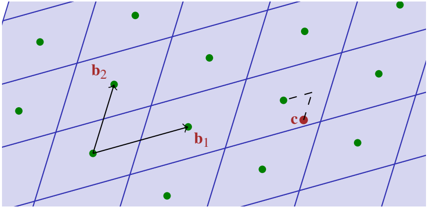
A First Attempt

- [Babai'86] 'simple rounding:' $\mathbf{c} \mapsto \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil$. (Fast & Parallel!)



A First Attempt

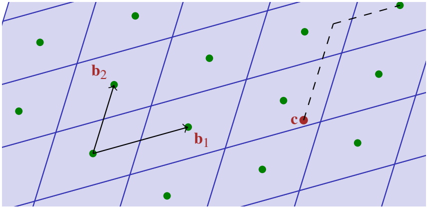
- [Babai'86] 'simple rounding:' $\mathbf{c} \mapsto \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil$. (Fast & Parallel!)
- Deterministic rounding is insecure [NguyenRegev'06] ...



A First Attempt

- [Babai'86] 'simple rounding:' $\mathbf{c} \mapsto \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil_{\$}$. (Fast & Parallel!)
- Deterministic rounding is <u>insecure</u> [NguyenRegev'06] ...

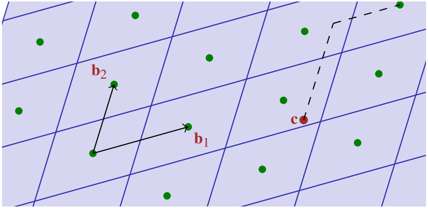
  ...but what about randomized rounding?

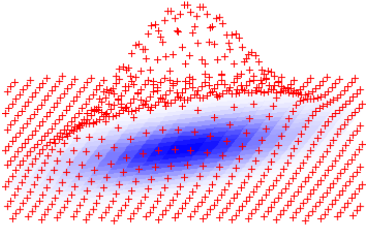


A First Attempt

- [Babai'86] 'simple rounding:' $\mathbf{c} \mapsto \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil_{\$}$. (Fast & Parallel!)
- Deterministic rounding is <u>insecure</u> [NguyenRegev'06] . . .

  . . . but what about randomized rounding?

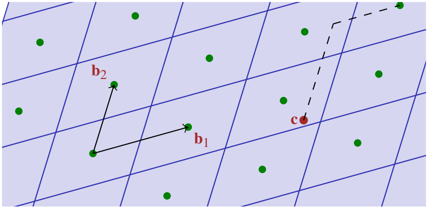


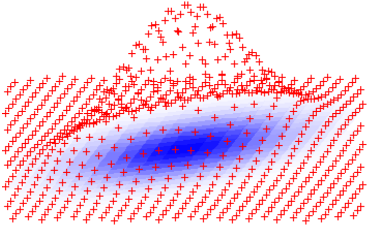


A First Attempt

- [Babai'86] 'simple rounding:' $\mathbf{c} \mapsto \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil_{\$}$. (Fast & Parallel!)
- Deterministic rounding is insecure [NguyenRegev'06] ...

  ...but what about randomized rounding?





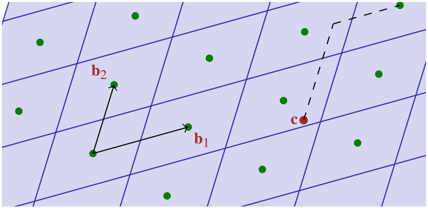
- Non-spherical distribution: has covariance

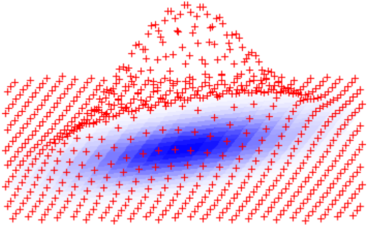
$$\Sigma := \operatorname*{Exp}_{\mathbf{x}} \left[\mathbf{x} \cdot \mathbf{x}^t \right] \approx \mathbf{B} \cdot \mathbf{B}^t.$$

A First Attempt

- [Babai'86] 'simple rounding:' $\mathbf{c} \mapsto \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil_{\$}$. (Fast & Parallel!)
- Deterministic rounding is insecure [NguyenRegev'06] ...

  ...but what about randomized rounding?





- Non-spherical distribution: has covariance

$$\Sigma := \operatorname*{Exp}_{\mathbf{x}} \left[\mathbf{x} \cdot \mathbf{x}^t \right] \approx \mathbf{B} \cdot \mathbf{B}^t.$$

  Covariance can be measured — and it leaks $\mathbf{B}$! (up to rotation)

Inspiration: Some Facts About Gaussians

1. Continuous Gaussian $\Longleftrightarrow$ positive definite covariance matrix Σ.

   (pos def: $\mathbf{u}^t \, \Sigma \, \mathbf{u} > 0$ for all unit $\mathbf{u}$.)

Inspiration: Some Facts About Gaussians

1. Continuous Gaussian $\iff$ positive definite covariance matrix Σ.

   (pos def: $\mathbf{u}^t \, \Sigma \, \mathbf{u} > 0$ for all unit $\mathbf{u}$.)

   Spherical Gaussian $\iff$ covariance $s^2 \, \mathbf{I}$.

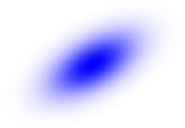
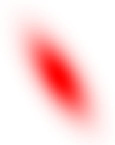
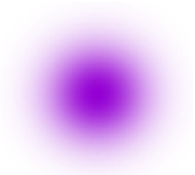
Inspiration: Some Facts About Gaussians

1. Continuous Gaussian $\iff$ positive definite covariance matrix Σ.

   (pos def: $\mathbf{u}^t \, \Sigma \, \mathbf{u} > 0$ for all unit $\mathbf{u}$.)

   Spherical Gaussian $\iff$ covariance $s^2 \, \mathbf{I}$.

2. Convolution of Gaussians:

 $+$  $=$ 

$$\Sigma_1 \quad + \quad \Sigma_2 \quad = \quad \Sigma = s^2 \, \mathbf{I}$$

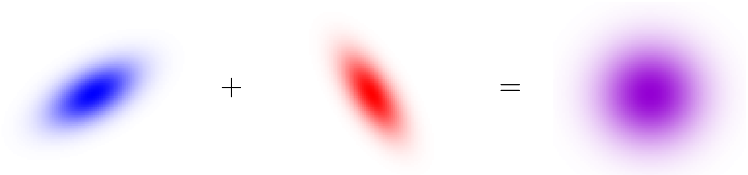
Inspiration: Some Facts About Gaussians

1. Continuous Gaussian $\Longleftrightarrow$ positive definite covariance matrix Σ.

   (pos def: $\mathbf{u}^t\,\Sigma\,\mathbf{u} > 0$ for all unit $\mathbf{u}$.)

   Spherical Gaussian $\Longleftrightarrow$ covariance $s^2\,\mathbf{I}$.

2. Convolution of Gaussians:



$$\Sigma_1 \quad + \quad \Sigma_2 \quad = \quad \Sigma = s^2\,\mathbf{I}$$

3. Given Σ_1, how small can s be? For $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$,

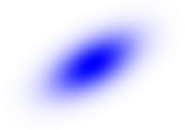
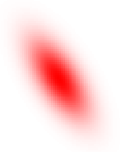
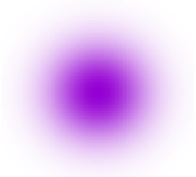
Inspiration: Some Facts About Gaussians

1. Continuous Gaussian $\iff$ positive definite covariance matrix Σ.

   (pos def: $\mathbf{u}^t\,\Sigma\,\mathbf{u} > 0$ for all unit $\mathbf{u}$.)

   Spherical Gaussian $\iff$ covariance $s^2\,\mathbf{I}$.

2. Convolution of Gaussians:

 $+$  $=$ 

$$\Sigma_1 \quad + \quad \Sigma_2 \quad = \quad \Sigma = s^2\,\mathbf{I}$$

3. Given Σ_1, how small can s be? For $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$,

$$\mathbf{u}^t\,\Sigma_2\,\mathbf{u} \;=\; s^2 - \mathbf{u}^t\,\Sigma_1\,\mathbf{u} > 0 \quad \iff \quad \boxed{s^2 > \max \lambda_i(\Sigma_1)}$$

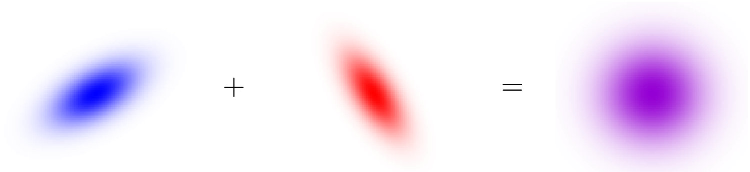
Inspiration: Some Facts About Gaussians

1. Continuous Gaussian $\iff$ positive definite covariance matrix Σ.

   (pos def: $\mathbf{u}^t \, \Sigma \, \mathbf{u} > 0$ for all unit $\mathbf{u}$.)

   Spherical Gaussian $\iff$ covariance $s^2 \, \mathbf{I}$.

2. Convolution of Gaussians:



$$\Sigma_1 \quad + \quad \Sigma_2 \quad = \quad \Sigma = s^2 \, \mathbf{I}$$

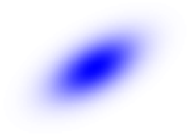
3. Given Σ_1, how small can s be? For $\Sigma_2 := s^2 \, \mathbf{I} - \Sigma_1$,

$$\mathbf{u}^t \, \Sigma_2 \, \mathbf{u} \; = \; s^2 - \mathbf{u}^t \, \Sigma_1 \, \mathbf{u} > 0 \quad \iff \quad \boxed{s^2 > \max \lambda_i(\Sigma_1)}$$

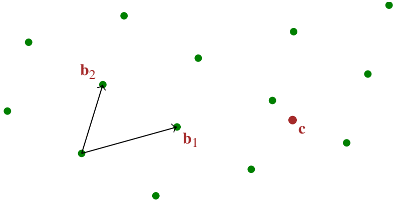
   When $\Sigma_1 = \mathbf{B} \, \mathbf{B}^t$, any $\boxed{s > s_1(\mathbf{B}) := \text{max singular val of } \mathbf{B}.}$

Our New Sampling Algorithm

- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,

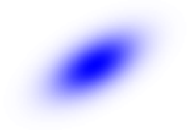


$$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$$

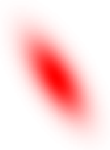


Our New Sampling Algorithm

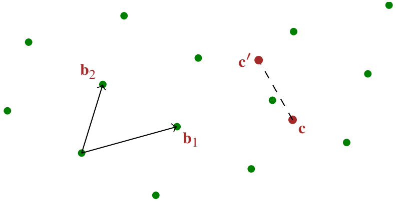
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$

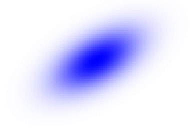


Σ_2

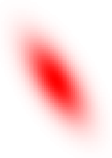


Our New Sampling Algorithm

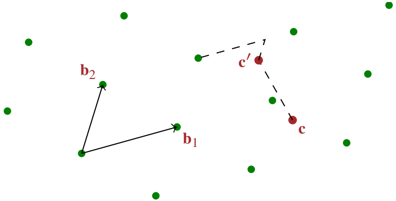
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_{\$}$



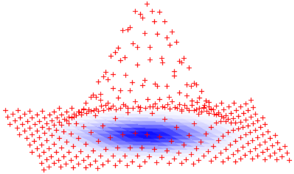
$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



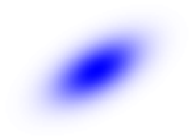
Σ_2



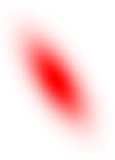
Our New Sampling Algorithm



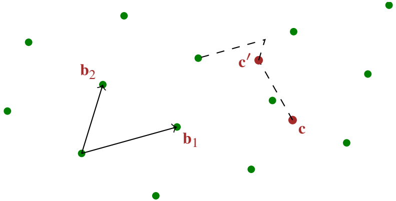
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2 \mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_{\$}$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



Σ_2

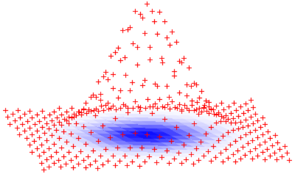


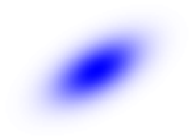
'Convolution' Theorem

Algorithm generates the discrete, spherical Gaussian over $\mathcal{L}$.

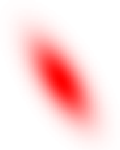
Our New Sampling Algorithm

- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$

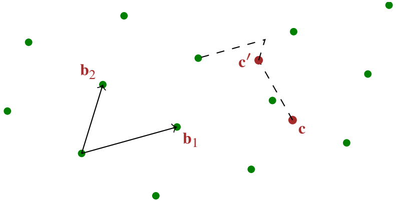




$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



Σ_2

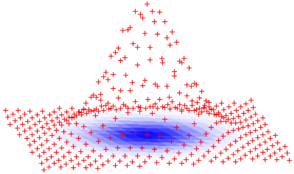


‘Convolution’ Theorem

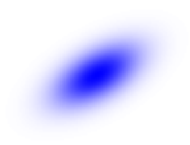
Algorithm generates the discrete, spherical Gaussian over $\mathcal{L}$.

(NB: not really a convolution, since step 2 depends on step 1.
Proof uses ‘smoothing parameter’ [MR’04] to reduce to an actual convolution.)

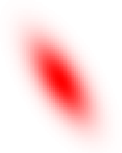
Our New Sampling Algorithm



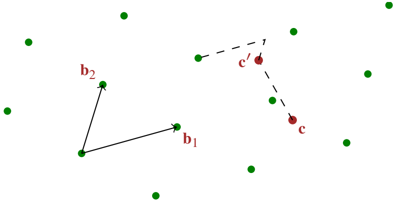
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



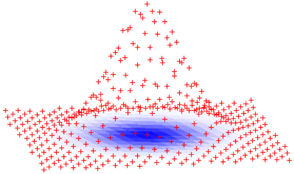
Σ_2



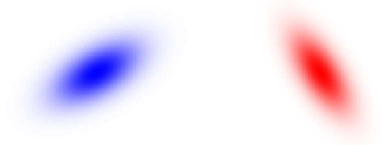
Optimizing for Crypto Applications

1. Precompute offline: Σ_2, $\mathbf{B}^{-1}$, perturbation(s)

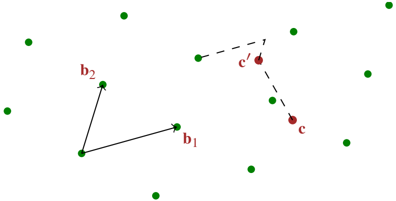
Our New Sampling Algorithm



- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2 \mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$



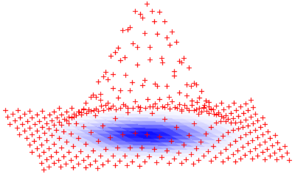
$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$ $\qquad$ Σ_2



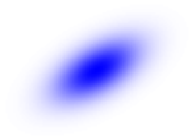
Optimizing for Crypto Applications

1. Precompute offline: Σ_2, $\mathbf{B}^{-1}$, perturbation(s)
2. Use integer perturbations and arithmetic

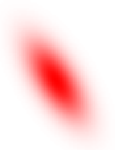
Our New Sampling Algorithm



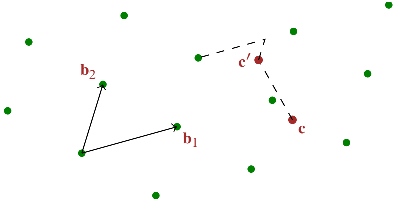
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



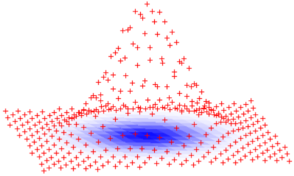
Σ_2



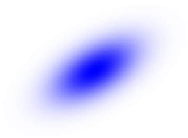
Optimizing for Crypto Applications

1. Precompute offline: Σ_2, $\mathbf{B}^{-1}$, perturbation(s)
2. Use integer perturbations and arithmetic
3. Exploit ‘q-ary’ lattices: mod q operations, offline rounding

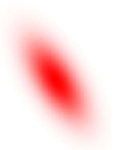
Our New Sampling Algorithm



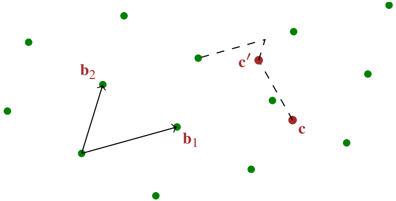
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



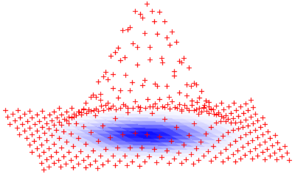
Σ_2



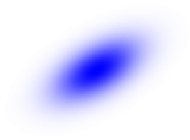
Optimizing for Crypto Applications

1. Precompute offline: Σ_2, $\mathbf{B}^{-1}$, perturbation(s)
2. Use integer perturbations and arithmetic
3. Exploit 'q-ary' lattices: mod q operations, offline rounding
4. Batch multi-sample using fast matrix mult

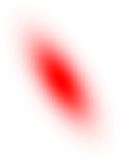
Our New Sampling Algorithm



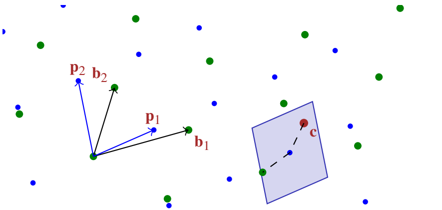
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



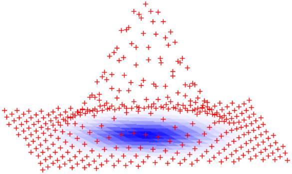
Σ_2



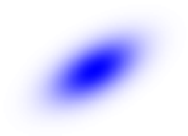
Some Perspective

- Resembles ‘perturbation’ heuristic of NTRUSign [HHG+’03]. But...

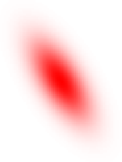
Our New Sampling Algorithm



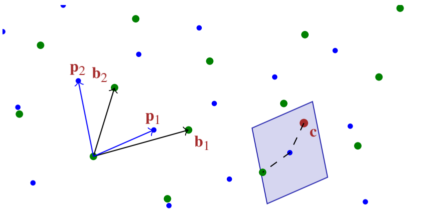
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



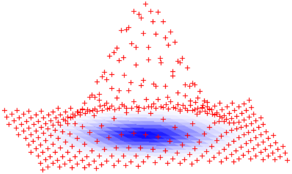
Σ_2



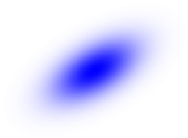
Some Perspective

- Resembles ‘perturbation’ heuristic of NTRUSign [HHG+’03]. But. . .
- NTRU perturbations are deterministic & inherently online. And. . .

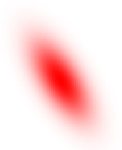
Our New Sampling Algorithm



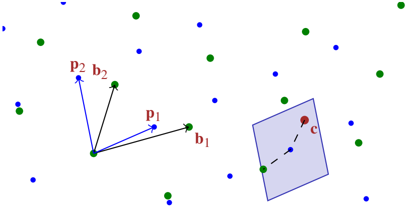
- Given basis $\mathbf{B}$, center $\mathbf{c}$, and $s > s_1(\mathbf{B})$,
  1. Perturb $\mathbf{c}$ with covariance $\Sigma_2 := s^2\,\mathbf{I} - \Sigma_1$
  2. Randomly round: return $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c}' \rceil_\$$



$\Sigma_1 = \mathbf{B}\,\mathbf{B}^t$



Σ_2



Some Perspective

- Resembles ‘perturbation’ heuristic of NTRUSign [HHG+’03]. But. . .
- NTRU perturbations are deterministic & inherently online. And. . .
- They may be insecure anyway [MPSW’10].

How Does the Quality Compare?

Narrower is Better!

- GPV: width $\approx \|\widetilde{\mathbf{B}}\| :=$ max Gram-Schmidt length of $\mathbf{B} \leq \max\|\mathbf{b}_i\|$
- New: width $\approx s_1(\mathbf{B}) :=$ max singular value of $\mathbf{B}$

How Does the Quality Compare?

Narrower is Better!

- GPV: width $\approx \|\widetilde{\mathbf{B}}\|$:= max Gram-Schmidt length of $\mathbf{B} \leq \max\|\mathbf{b}_i\|$
- New: width $\approx s_1(\mathbf{B})$:= max singular value of $\mathbf{B}$

Bad News, and Good News...

How Does the Quality Compare?

Narrower is Better!

- GPV: width $\approx \|\widetilde{\mathbf{B}}\|$:= max Gram-Schmidt length of $\mathbf{B} \leq \max\|\mathbf{b}_i\|$
- New: width $\approx s_1(\mathbf{B})$:= max singular value of $\mathbf{B}$

Bad News, and Good News...

✗ In general,

$$\|\widetilde{\mathbf{B}}\| \ \leq \ s_1(\mathbf{B}) \ \leq \ n \cdot \|\widetilde{\mathbf{B}}\|$$

(Both inequalities are tight.)

How Does the Quality Compare?

Narrower is Better!

- GPV: width $\approx \|\widetilde{\mathbf{B}}\|$:= max Gram-Schmidt length of $\mathbf{B} \leq \max\|\mathbf{b}_i\|$
- New: width $\approx s_1(\mathbf{B})$:= max singular value of $\mathbf{B}$

Bad News, and Good News...

✗ In general,

$$\|\widetilde{\mathbf{B}}\| \;\leq\; s_1(\mathbf{B}) \;\leq\; n \cdot \|\widetilde{\mathbf{B}}\|$$

(Both inequalities are tight.)

✔ We show: for random cryptographic bases [AP'09,CHKP'10],

$$\boxed{\|\widetilde{\mathbf{B}}\| \;\approx\; s_1(\mathbf{B})}$$

because bases are 'well-rounded.'

Epilogue

- In an upcoming work [MP'10], we tackle basis generation and Gaussian sampling jointly.

  $\Rightarrow$ Simple constructions, optimal constants, practical algorithms

Epilogue

- In an upcoming work [MP'10], we tackle basis generation and Gaussian sampling jointly.

  $\Rightarrow$ Simple constructions, optimal constants, practical algorithms

- Implementation: 1000s of samples / sec at moderate security.

  (Without batching or parallelism!)

Epilogue

- In an upcoming work [MP'10], we tackle basis generation and Gaussian sampling jointly.

  $\Rightarrow$ Simple constructions, optimal constants, practical algorithms

- Implementation: 1000s of samples / sec at moderate security.

  (Without batching or parallelism!)

  $\Rightarrow$ Essentially as fast as the public-key operation.

  $\Rightarrow$ Bottleneck: n^2 cost inherent to general lattices.

  Ring-based schemes will be much faster!

Epilogue

- In an upcoming work [MP'10], we tackle basis generation and Gaussian sampling jointly.

  ⇒ Simple constructions, optimal constants, practical algorithms

- Implementation: 1000s of samples / sec at moderate security.

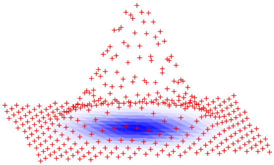
  (Without batching or parallelism!)

  ⇒ Essentially as fast as the public-key operation.

  ⇒ Bottleneck: n^2 cost inherent to general lattices.

  Ring-based schemes will be much faster!

- Stay tuned ...



Thanks!