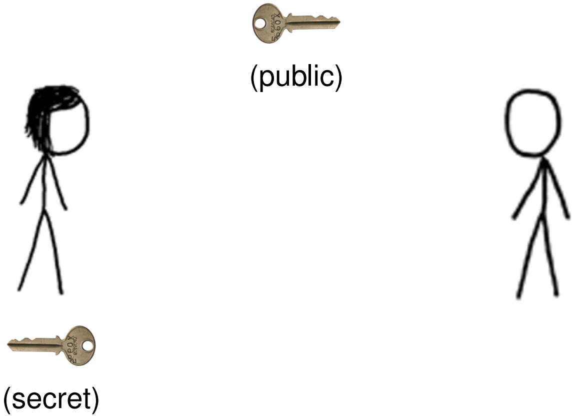
How to Use a Short Basis: Trapdoors for Hard Lattices and New Cryptographic Constructions

Chris Peikert
SRI

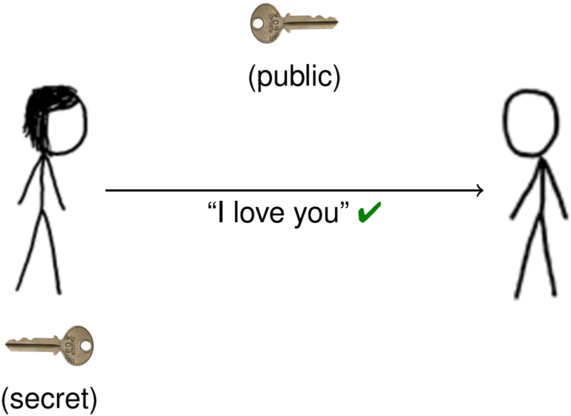
Work with Craig Gentry and Vinod Vaikuntanathan

Digital Signatures

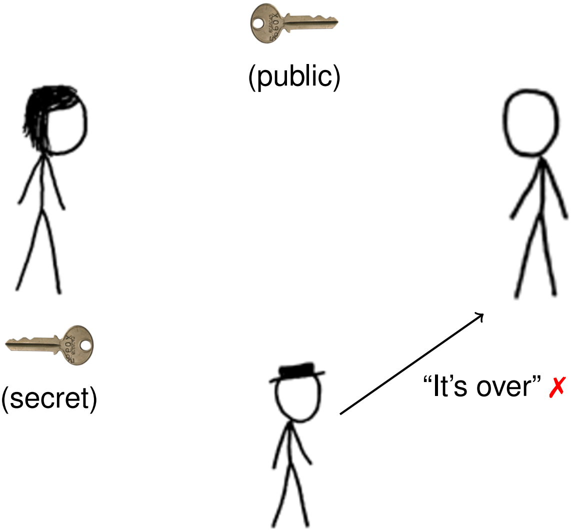
Digital Signatures



Digital Signatures



Digital Signatures

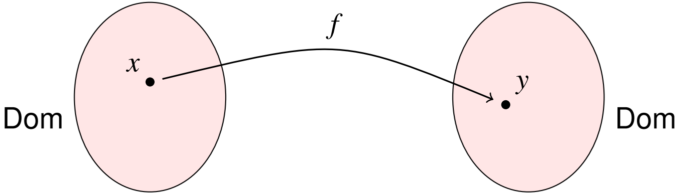


Trapdoor Permutations [DiffieHellman76]

- Public function f, secret “trapdoor” f^{-1}

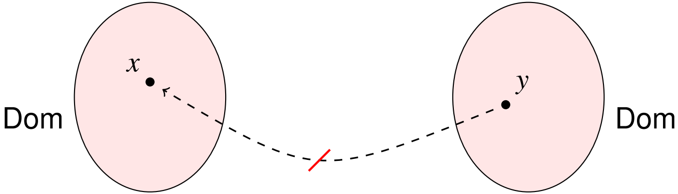
Trapdoor Permutations [DiffieHellman76]

- Public function f, secret "trapdoor" f^{-1}



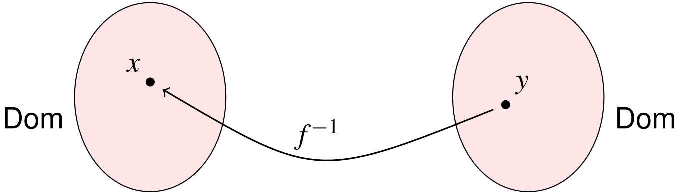
Trapdoor Permutations [DiffieHellman76]

- Public function f, secret "trapdoor" f^{-1}



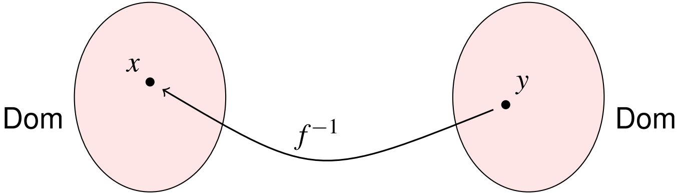
Trapdoor Permutations [DiffieHellman76]

- Public function f, secret "trapdoor" f^{-1}



Trapdoor Permutations [DiffieHellman76]

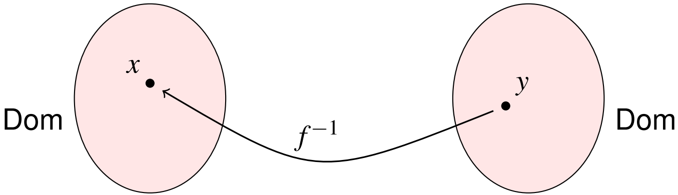
- Public function f, secret "trapdoor" f^{-1}



- Candidates: [RSA78,Rabin79,Paillier99]
  - ✔ "General assumption"
  - ✔ Applications: digital signatures, OT, NIZK, ...

Trapdoor Permutations [DiffieHellman76]

- Public function f, secret "trapdoor" f^{-1}



- Candidates: [RSA78,Rabin79,Paillier99]
  - ✔ "General assumption"
  - ✔ Applications: digital signatures, OT, NIZK, ...
- All rely on hardness of factoring
  - ✗ Complex: 2048-bit exponentiation
  - ✗ Lack of diversity
  - ✗ Broken by quantum algorithms [Shor]

Lattice-Based Cryptography

What's To Like

- Simple & efficient: linear ops, small integers
- Resist subexp & quantum attacks (so far)
- Security from worst-case hardness [Ajtai,. . .]

Lattice-Based Cryptography

What's To Like

- Simple & efficient: linear ops, small integers
- Resist subexp & quantum attacks (so far)
- Security from worst-case hardness [Ajtai,. . .]

What's Known

1. One-way & collision-resistant functions [Ajtai,. . . ,MicciancioRegev]
2. Public-key encryption [AjtaiDwork,Regev]
3. Recent developments [LyubMicc,PeikWat,. . .]

Lattice-Based Cryptography

What's To Like

- Simple & efficient: linear ops, small integers
- Resist subexp & quantum attacks (so far)
- Security from worst-case hardness [Ajtai,...]

What's Known

1. One-way & collision-resistant functions [Ajtai,...,MicciancioRegev]
2. Public-key encryption [AjtaiDwork,Regev]
3. Recent developments [LyubMicc,PeikWat,...]

What's Missing

- Everything else!

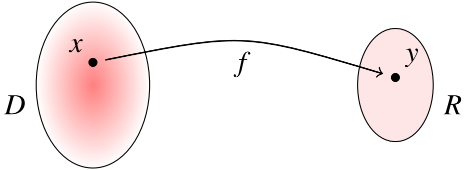
  Practical signatures, protocols, "advanced" crypto, ...

Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions

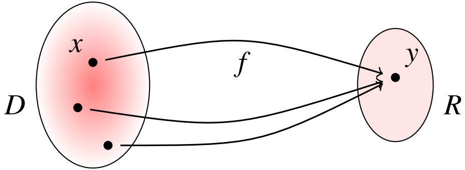
Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions



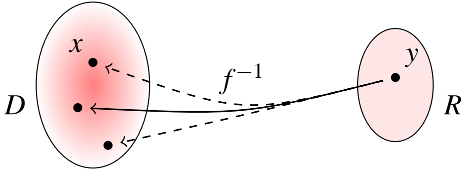
Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions



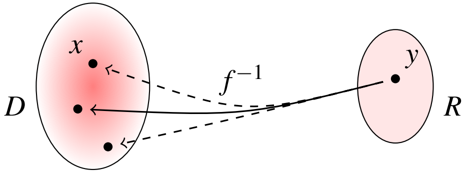
Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions



Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions

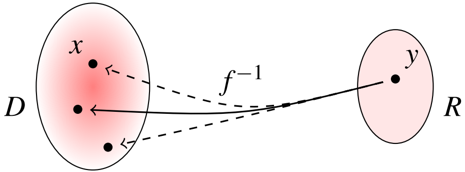


- Generate (x, y) in two equivalent ways:

$$D \longrightarrow x \xrightarrow{f} y \quad \Big| \quad x \xleftarrow{f^{-1}} y \longleftarrow R$$

Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions



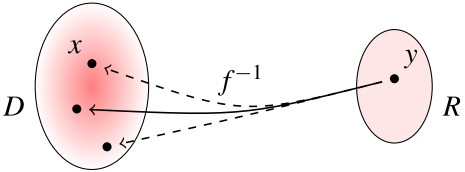
- Generate (x, y) in two equivalent ways:

$$D \longrightarrow x \xrightarrow{f} y \quad \Big| \quad x \xleftarrow{f^{-1}} y \longleftarrow R$$

- "As good as" trapdoor permutations in many applications

Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions



   - Generate (x, y) in two equivalent ways:

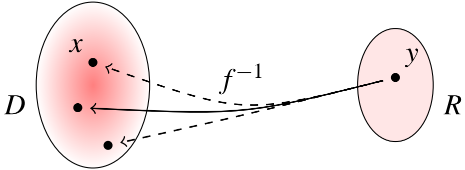
$$D \longrightarrow x \xrightarrow{f} y \qquad \Big| \qquad x \xleftarrow{f^{-1}} y \longleftarrow R$$

   - "As good as" trapdoor permutations in many applications

2. "Hash and sign" signatures: FDH etc.

Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions



- Generate (x, y) in two equivalent ways:

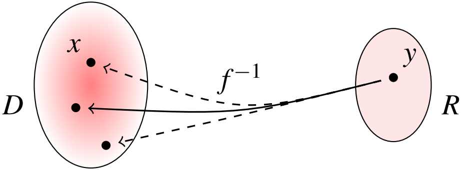
$$D \longrightarrow x \xrightarrow{f} y \qquad \Big| \qquad x \xleftarrow{f^{-1}} y \longleftarrow R$$

- "As good as" trapdoor permutations in many applications

2. "Hash and sign" signatures: FDH etc.
3. Identity-based encryption, OT [PVW], NCE [CDMW], NISZK [PV], ...

Results: New Lattice-Based Crypto

1. Preimage sampleable trapdoor functions



- Generate (x, y) in two equivalent ways:

$$D \longrightarrow x \xrightarrow{f} y \qquad \Bigg| \qquad x \xleftarrow{f^{-1}} y \longleftarrow R$$

- "As good as" trapdoor permutations in many applications

2. "Hash and sign" signatures: FDH etc.
3. Identity-based encryption, OT [PVW], NCE [CDMW], NISZK [PV], . . .

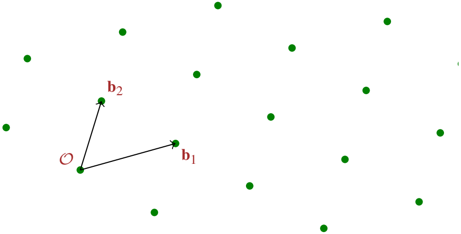
New Algorithmic Tool

- "Oblivious decoder" on lattices

Lattices

A lattice $\mathcal{L} \subset \mathbb{R}^n$ having basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is:

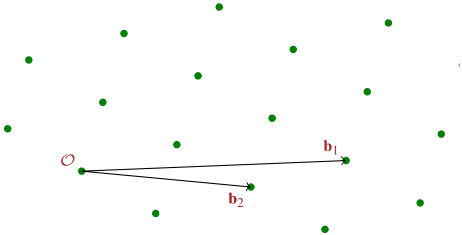
$$\mathcal{L} \quad = \quad \sum_{i=1}^{n} (\mathbb{Z} \cdot \mathbf{b}_i)$$



Lattices

A lattice $\mathcal{L} \subset \mathbb{R}^n$ having basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is:

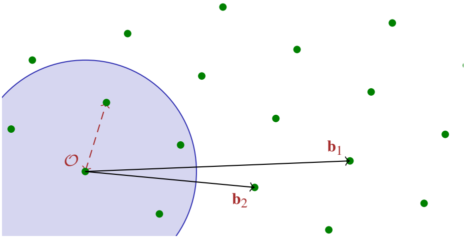
$$\mathcal{L} = \sum_{i=1}^{n} (\mathbb{Z} \cdot \mathbf{b}_i)$$



Lattices

A lattice $\mathcal{L} \subset \mathbb{R}^n$ having basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is:

$$\mathcal{L} = \sum_{i=1}^{n} (\mathbb{Z} \cdot \mathbf{b}_i)$$



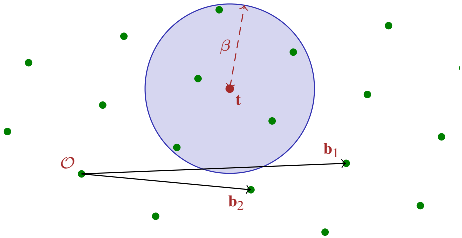
Shortest Vector Problem (SVP$_\gamma$)

- Given $\mathbf{B}$, find (nonzero) $\mathbf{v} \in \mathcal{L}$ within γ factor of shortest.

Lattices

A lattice $\mathcal{L} \subset \mathbb{R}^n$ having basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is:

$$\mathcal{L} = \sum_{i=1}^{n} (\mathbb{Z} \cdot \mathbf{b}_i)$$



Shortest Vector Problem (SVP$_\gamma$)

- Given $\mathbf{B}$, find (nonzero) $\mathbf{v} \in \mathcal{L}$ within γ factor of shortest.

Absolute Distance Decoding (ADD$_\beta$)

- Given $\mathbf{B}$ and target $\mathbf{t} \in \mathbb{R}^n$, find some $\mathbf{v} \in \mathcal{L}$ within distance β.

Complexity of Lattice Problems

SVP$_\gamma$ in the Worst Case

$\gamma =$	$O(1)$	$\text{poly}(n)$	2^n
	NP-hard	2^n time	$\text{poly}(n)$ time
	[Ajt,Mic,Kho]	[AKS]	[LLL,Sch]

Complexity of Lattice Problems

SVP_γ in the Worst Case

$\gamma =$	$O(1)$	$\mathrm{poly}(n)$	2^n
	NP-hard	2^n time	$\mathrm{poly}(n)$ time
	[Ajt,Mic,Kho]	[AKS]	[LLL,Sch]

Average-Case

- [Ajtai96,...,MicciancioRegev04]:

SVP_γ random lattice as hard as $\mathsf{SVP}_{\gamma \cdot n}$ every lattice

Complexity of Lattice Problems

SVP$_\gamma$ in the Worst Case

$\gamma =$	$O(1)$	poly(n)	2^n
	NP-hard	2^n time	poly(n) time
	[Ajt,Mic,Kho]	[AKS]	[LLL,Sch]

Average-Case

- [Ajtai96,. . . ,MicciancioRegev04]:

  ADD$_\beta$ random lattice as hard as SVP$_{\beta \cdot n}$ every lattice

- Decoding hard on average, too

Complexity of Lattice Problems

SVP_γ in the Worst Case

$\gamma =$	$O(1)$	$\mathrm{poly}(n)$	2^n
	NP-hard	2^n time	$\mathrm{poly}(n)$ time
	[Ajt,Mic,Kho]	[AKS]	[LLL,Sch]

Average-Case

- [Ajtai96,...,MicciancioRegev04]:

  ADD_β random lattice as hard as $\mathrm{SVP}_{\beta\cdot n}$ every lattice

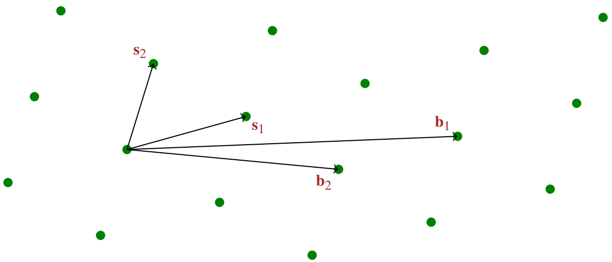
- Decoding hard on average, too

Bottom Line

- On random lattices, SVP_γ and ADD_β seem exponentially hard

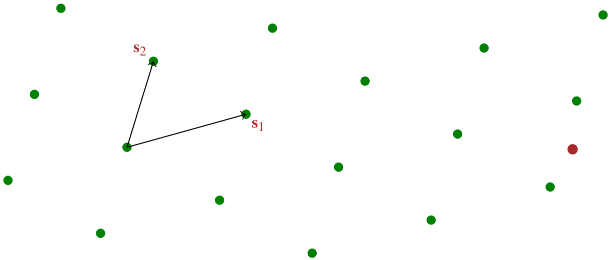
GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$



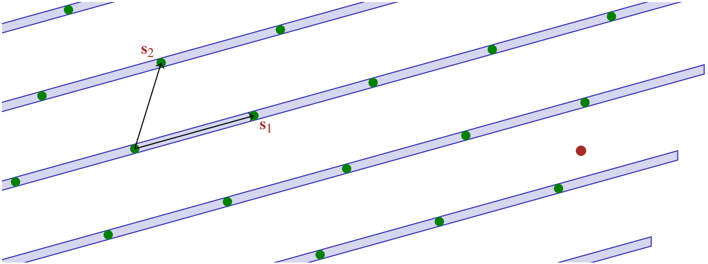
GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]



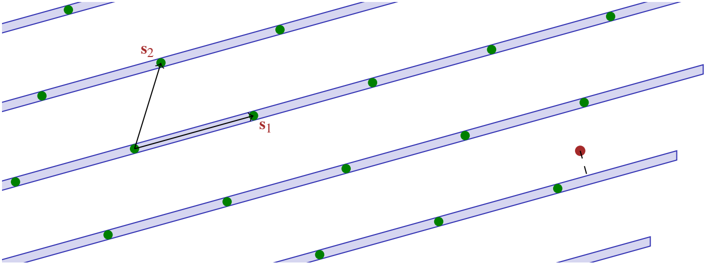
GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]



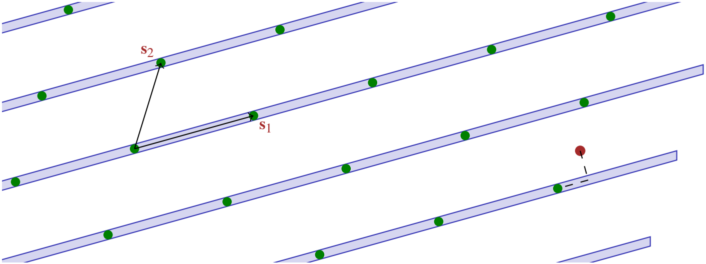
GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]



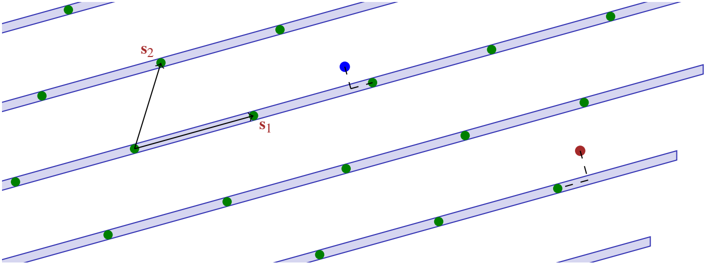
GGH Signatures [GoldreichGoldwasserHalevi96]

- “Hard” (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with “nearest-plane” algorithm [Babai]



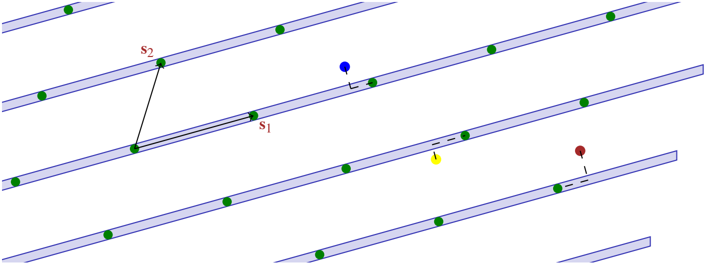
GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]



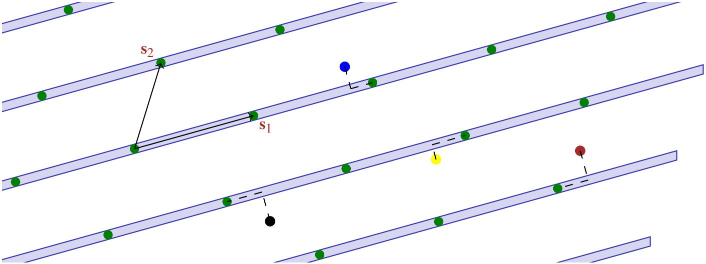
GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]



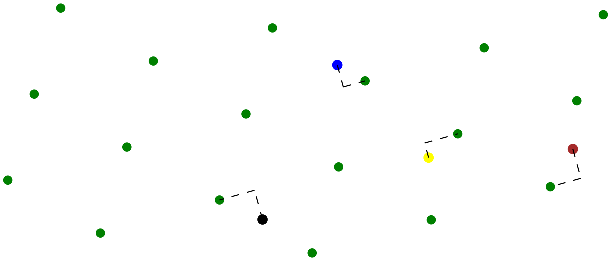
GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]



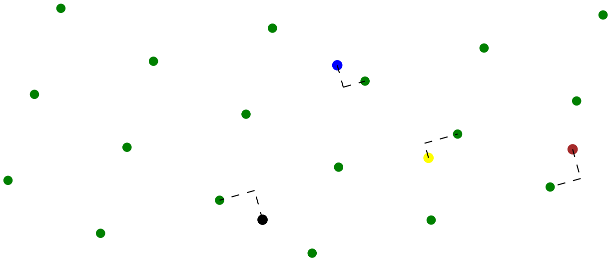
GGH Signatures [GoldreichGoldwasserHalevi96]

- “Hard” (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with “nearest-plane” algorithm [Babai]



GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]

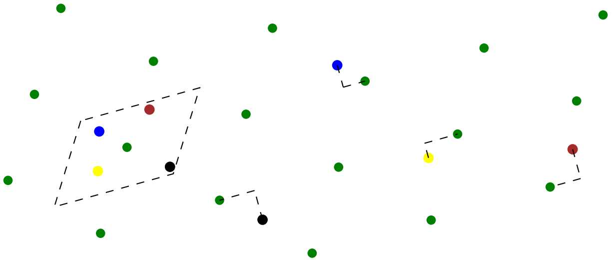


Issues

1. Generating short & hard bases together
   - Ad-hoc, no worst-case hardness

GGH Signatures [GoldreichGoldwasserHalevi96]

- "Hard" (public) verification basis $\mathbf{B}$, short (secret) signing basis $\mathbf{S}$
- Sign with "nearest-plane" algorithm [Babai]

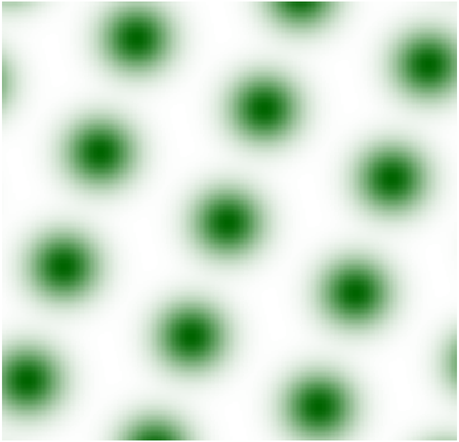


Issues

1. Generating short & hard bases together
   - Ad-hoc, no worst-case hardness
2. Secret key leakage
   - Total break after several signatures [NguyenRegev]

Gaussians and Lattices

Gaussians and Lattices



Gaussians and Lattices

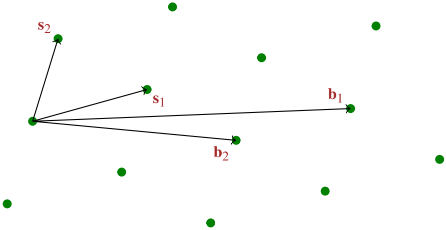
Gaussians and Lattices

"Uniform" in $\mathbb{R}^n$ when std dev $\geq$ shortest basis

[Regev,MicciancioRegev]

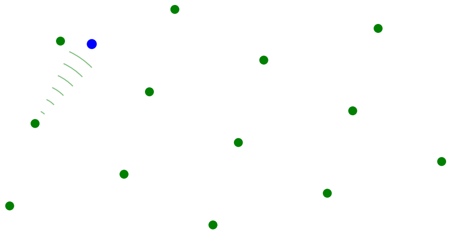
Our Trapdoor Function

- "Hard" public basis $\mathbf{B}$, short secret basis $\mathbf{S}$ [Ajtai99,AP08]



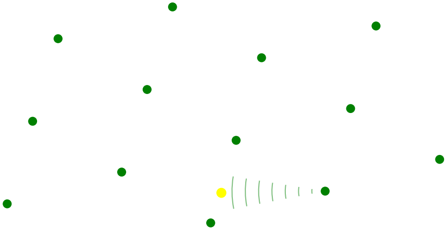
Our Trapdoor Function

- "Hard" public basis $\mathbf{B}$, short secret basis $\mathbf{S}$ [Ajtai99,AP08]
- Input $\mathbf{v} \in \mathcal{L}$, error $\mathbf{e}$



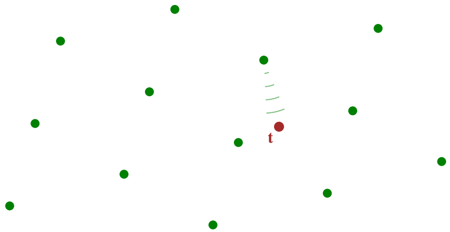
Our Trapdoor Function

- "Hard" public basis $\mathbf{B}$, short secret basis $\mathbf{S}$ [Ajtai99,AP08]
- Input $\mathbf{v} \in \mathcal{L}$, error $\mathbf{e}$



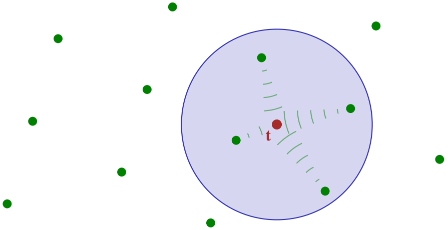
Our Trapdoor Function

- "Hard" public basis $\mathbf{B}$, short secret basis $\mathbf{S}$ [Ajtai99,AP08]
- Input $\mathbf{v} \in \mathcal{L}$, error $\mathbf{e}$
- Uniform output $\mathbf{t}$



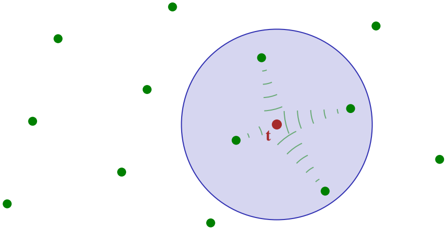
Our Trapdoor Function

- "Hard" public basis $\mathbf{B}$, short secret basis $\mathbf{S}$ [Ajtai99,AP08]
- Input $\mathbf{v} \in \mathcal{L}$, error $\mathbf{e}$
- Uniform output $\mathbf{t}$

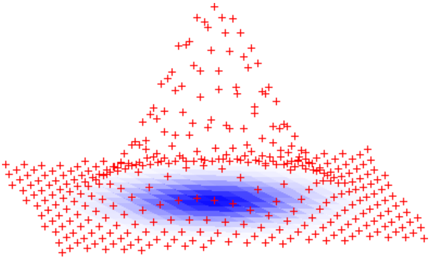


Our Trapdoor Function

- "Hard" public basis $\mathbf{B}$, short secret basis $\mathbf{S}$ [Ajtai99,AP08]
- Input $\mathbf{v} \in \mathcal{L}$, error $\mathbf{e}$
- Uniform output $\mathbf{t}$



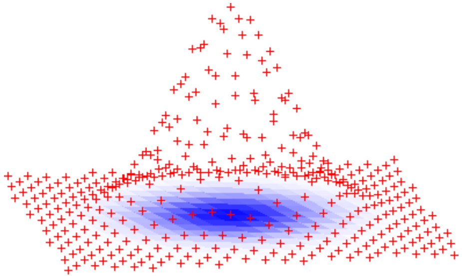
- Conditional distribution is "discrete Gaussian" $D_{\mathcal{L},\mathbf{t}}$



Analysis tool in [Ban,AR,Reg,MR,Pei,. . .]

Inverting: Gaussian Sampler / Decoder

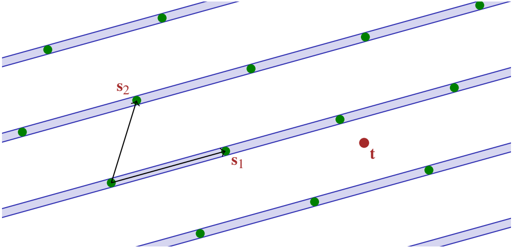
- Given basis $\mathbf{S}$, samples $D_{\mathcal{L},\mathbf{t}}$ for any std dev $\geq \max\|\mathbf{s}_i\|$
  - Leaks nothing about $\mathbf{S}$!



Inverting: Gaussian Sampler / Decoder

- Given basis $\mathbf{S}$, samples $D_{\mathcal{L},\mathbf{t}}$ for any std dev $\geq \max\|\mathbf{s}_i\|$
  - Leaks nothing about $\mathbf{S}$!

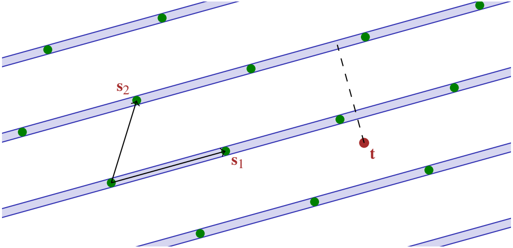
- Randomized nearest-plane [Babai,Klein]



Inverting: Gaussian Sampler / Decoder

- Given basis $\mathbf{S}$, samples $D_{\mathcal{L},\mathbf{t}}$ for any std dev $\geq \max\|\mathbf{s}_i\|$
  - Leaks nothing about $\mathbf{S}$!

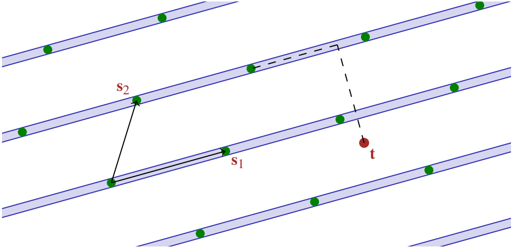
- Randomized nearest-plane [Babai,Klein]



Inverting: Gaussian Sampler / Decoder

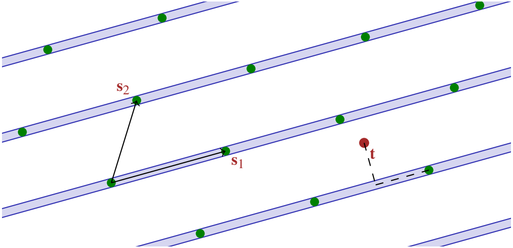
- Given basis $\mathbf{S}$, samples $D_{\mathcal{L},\mathbf{t}}$ for any std dev $\geq \max\|\mathbf{s}_i\|$
  - Leaks nothing about $\mathbf{S}$!

- Randomized nearest-plane [Babai,Klein]



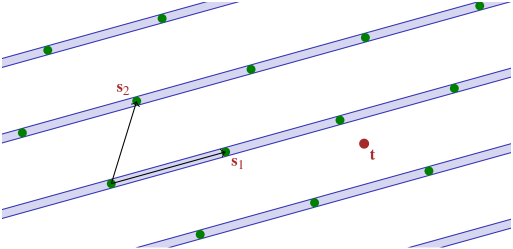
Inverting: Gaussian Sampler / Decoder

- Given basis $\mathbf{S}$, samples $D_{\mathcal{L},\mathbf{t}}$ for any std dev $\geq \max\|\mathbf{s}_i\|$
  - Leaks nothing about $\mathbf{S}$!
- Randomized nearest-plane [Babai,Klein]



Inverting: Gaussian Sampler / Decoder

- Given basis $\mathbf{S}$, samples $D_{\mathcal{L},\mathbf{t}}$ for any std dev $\geq \max\|\mathbf{s}_i\|$
  - Leaks nothing about $\mathbf{S}$!
- Randomized nearest-plane [Babai,Klein]

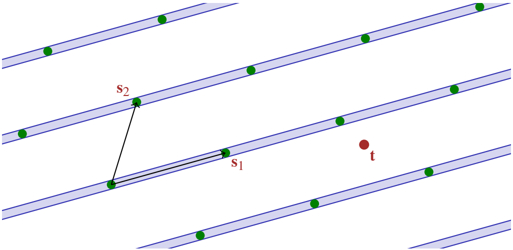


[Klein]: std dev $\leq \min\|\tilde{\mathbf{s}}_i\| \Rightarrow$ solves CVP variant

Inverting: Gaussian Sampler / Decoder

- Given basis $\mathbf{S}$, samples $D_{\mathcal{L},\mathbf{t}}$ for any std dev $\geq \max\|\mathbf{s}_i\|$
  - Leaks nothing about $\mathbf{S}$!

- Randomized nearest-plane [Babai,Klein]



[Klein]: std dev $\leq \min\|\tilde{\mathbf{s}}_i\| \Rightarrow$ solves CVP variant

[This work]: std dev $\geq \max\|\tilde{\mathbf{s}}_i\| \Rightarrow$ samples $D_{\mathcal{L},\mathbf{t}}$ exactly*

Identity-Based Encryption

- Proposed by [Shamir84]:

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID "Alice" or "Bob" or ...

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID "Alice" or "Bob" or ...
  - With msk: extract sk_{Alice} or sk_{Bob} or ...

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID "Alice" or "Bob" or ...
  - With msk: extract sk_{Alice} or sk_{Bob} or ...
- [BonehFranklin01]: bilinear pairings

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID "Alice" or "Bob" or ...
  - With msk: extract sk_{Alice} or sk_{Bob} or ...
- [BonehFranklin01]: bilinear pairings
- [Cocks01]: quadratic residuosity (mod $N = pq$)

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID “Alice” or “Bob” or ...
  - With msk: extract sk_{Alice} or sk_{Bob} or ...
- [BonehFranklin01]: bilinear pairings
- [Cocks01]: quadratic residuosity (mod $N = pq$)

	Lattice-based	QR-based [Cocks,BGH]
mpk	random lattice	random $N = p \cdot q$

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID “Alice” or “Bob” or ...
  - With msk: extract sk_{Alice} or sk_{Bob} or ...
- [BonehFranklin01]: bilinear pairings
- [Cocks01]: quadratic residuosity (mod $N = pq$)

	Lattice-based	QR-based [Cocks,BGH]
mpk	random lattice	random $N = p \cdot q$
msk	trapdoor basis	trapdoor p, q

Identity-Based Encryption

- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID “Alice” or “Bob” or ...
  - With msk: extract sk_{Alice} or sk_{Bob} or ...
- [BonehFranklin01]: bilinear pairings
- [Cocks01]: quadratic residuosity (mod $N = pq$)

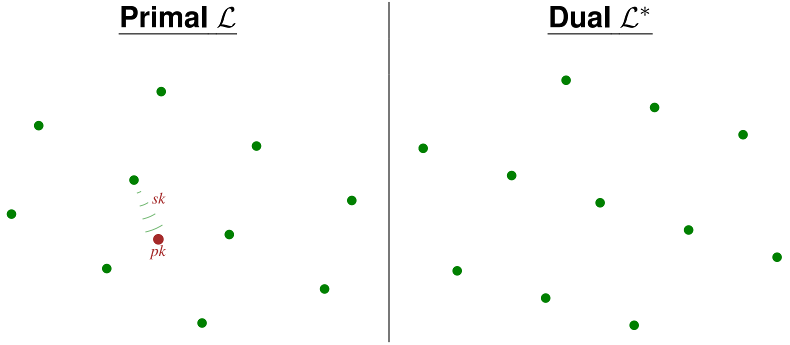
	Lattice-based	QR-based [Cocks,BGH]
mpk	random lattice	random $N = p \cdot q$
msk	trapdoor basis	trapdoor p, q
Hash(ID)	uniform $\mathbf{y} \in \mathbb{R}^n$	uniform $y \in QR_N$

Identity-Based Encryption

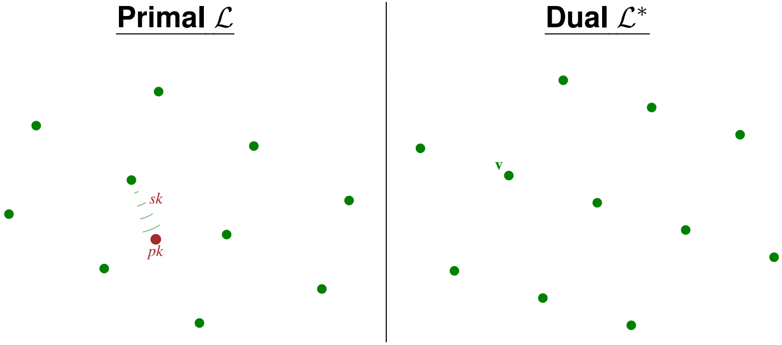
- Proposed by [Shamir84]:
  - Master keys mpk, msk
  - With mpk: encrypt to ID "Alice" or "Bob" or . . .
  - With msk: extract sk_{Alice} or sk_{Bob} or . . .
- [BonehFranklin01]: bilinear pairings
- [Cocks01]: quadratic residuosity (mod $N = pq$)

	Lattice-based	QR-based [Cocks,BGH]
mpk	random lattice	random $N = p \cdot q$
msk	trapdoor basis	trapdoor p, q
Hash(ID)	uniform $\mathbf{y} \in \mathbb{R}^n$	uniform $y \in QR_N$
sk_{ID}	random $\in f^{-1}(\mathbf{y})$	random $\sqrt{y}$

Cryptosystem with Master Trapdoor

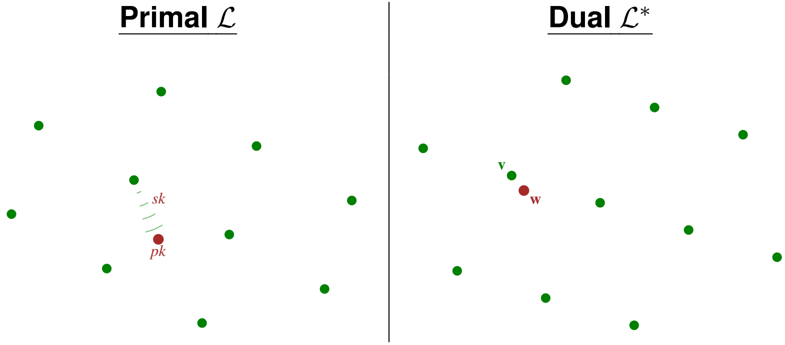


Cryptosystem with Master Trapdoor



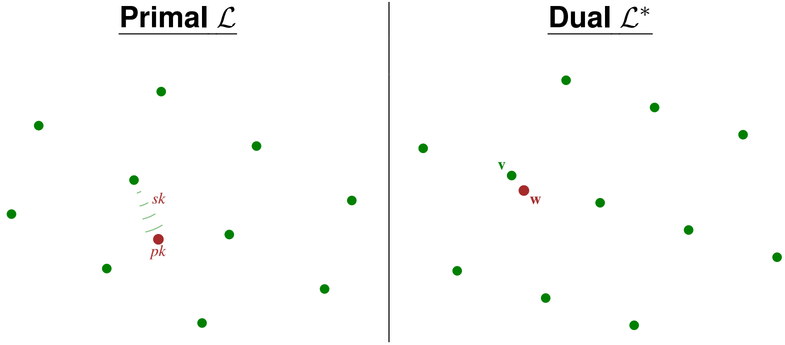
- For $\mathbf{v} \in \mathcal{L}^*$: $\langle \mathbf{v}, pk \rangle = \langle \mathbf{v}, sk \rangle \bmod 1$

Cryptosystem with Master Trapdoor



- For $\mathbf{v} \in \mathcal{L}^*$: $\langle \mathbf{v}, pk \rangle = \langle \mathbf{v}, sk \rangle \bmod 1$
- For $\mathbf{w} \approx \mathbf{v}$: $\langle \mathbf{v}, pk \rangle \approx \langle \mathbf{w}, sk \rangle \bmod 1$ “quasi”-agreement

Cryptosystem with Master Trapdoor



- For $\mathbf{v} \in \mathcal{L}^*$: $\langle \mathbf{v}, pk \rangle = \langle \mathbf{v}, sk \rangle \bmod 1$
- For $\mathbf{w} \approx \mathbf{v}$: $\langle \mathbf{v}, pk \rangle \approx \langle \mathbf{w}, sk \rangle \bmod 1$ "quasi"-agreement
- Security: decoding $\mathbf{w}$, a.k.a. "learning with errors"
  - Quantum worst-case connection [Regev]
  - Now: classical worst-case hardness [P]

Open Problems

1. Tighter sampling for random lattices ?

Open Problems

1. Tighter sampling for random lattices ?

2. Practical "plain model" signatures ?

Open Problems

1. Tighter sampling for random lattices ?

2. Practical “plain model” signatures ?

3. Relate factoring to lattice problems ?

Open Problems

1. Tighter sampling for random lattices ?
2. Practical “plain model” signatures ?
3. Relate factoring to lattice problems ?
4. “Essence” of quantum-immune crypto ?

Open Problems

1. Tighter sampling for random lattices ?
2. Practical “plain model” signatures ?
3. Relate factoring to lattice problems ?
4. “Essence” of quantum-immune crypto ?

Thanks!

(Artwork courtesy of xkcd.org)