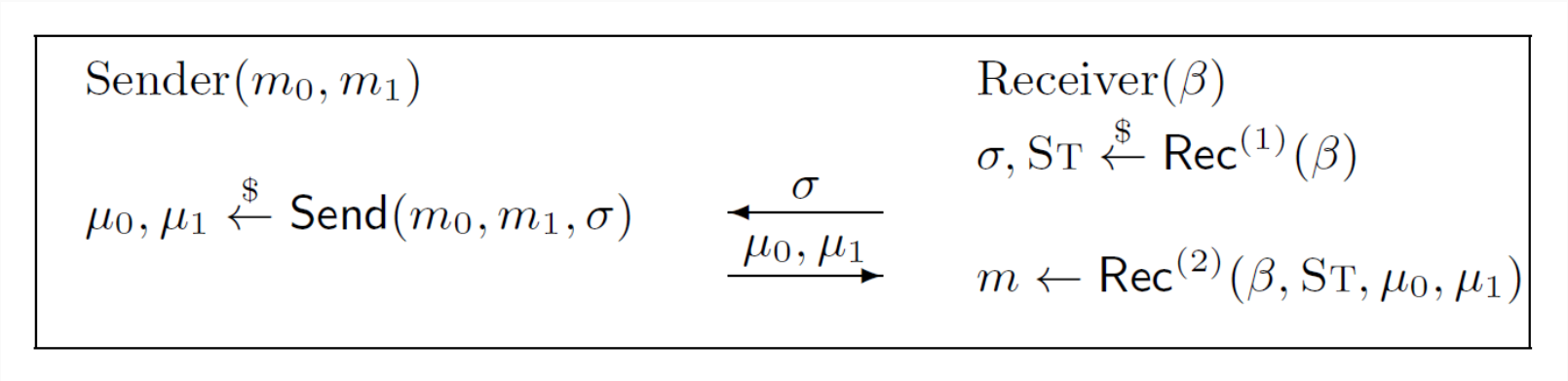
Simpler Statistically Sender Private Oblivious Transfer from Ideals of Cyclotomic Integers

Daniele Micciancio, Jessica Sorrell

Asiacrypt 2020

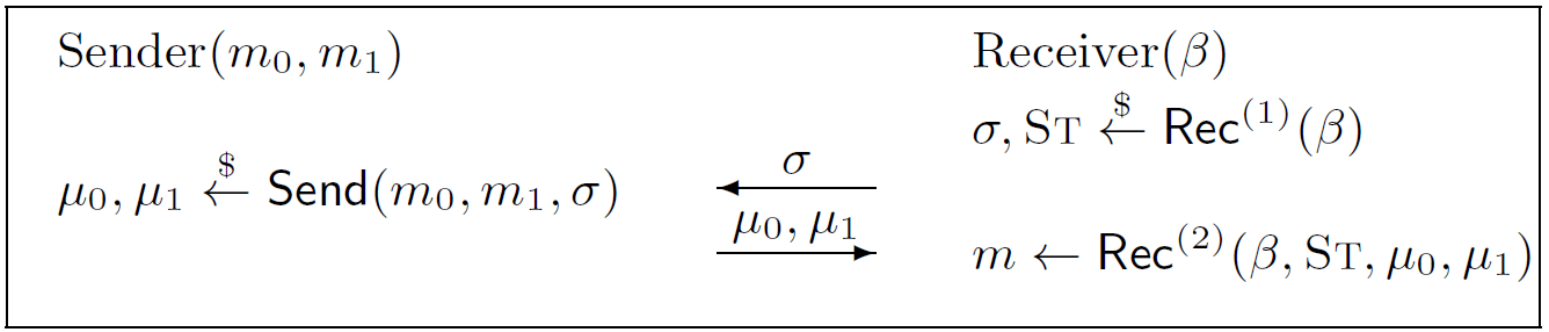
University of California, San Diego

Oblivious Transfer



- Oblivious Transfer (OT) [Rabin'81] is complete for secure multi-party computation (MPC) [Kilian'88]

Oblivious Transfer



- Oblivious Transfer (OT) [Rabin'81] is complete for secure multi-party computation (MPC) [Kilian'88]
- Allows a `Receiver` to receive one of two messages from a `Sender` without the `Sender` learning which message was sent, and without the `Receiver` learning anything about the other message.

2-Round OT Definitions

Correctness

An OT $= (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ protocol is *correct* if for any pair of messages m_0, m_1 and bit $\beta \in \{0,1\}$,

$$\Pr[\mathsf{Rec}^{(2)}(\mathsf{Send}(m_0, m_1, \mathsf{Rec}^{(1)}(\beta))) = m_\beta] \geq 1 - \epsilon$$

for some negligible function $\epsilon(n) = n^{-\omega(1)}$.

2-Round OT Definitions

Correctness

An OT $= (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ protocol is *correct* if for any pair of messages m_0, m_1 and bit $\beta \in \{0,1\}$,

$$\Pr[\mathsf{Rec}^{(2)}(\mathsf{Send}(m_0, m_1, \mathsf{Rec}^{(1)}(\beta))) = m_\beta] \geq 1 - \epsilon$$

for some negligible function $\epsilon(n) = n^{-\omega(1)}$.

Statistical sender privacy (SSP)

An OT $= (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ protocol is *statistically sender private* if for any receiver message σ, there exists a bit b, such that for any pair of messages (m_0, m_1) the two distributions

$$\{\mathsf{Send}(m_0, m_1, \sigma)\} \approx_\Delta \{\mathsf{Send}(m_b, m_b, \sigma)\}$$

are statistically close.

2-Round OT Definitions

Computational receiver privacy

An OT $= (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ protocol is *computationally receiver private* if the two distributions

$$\mathsf{Rec}^{(1)}(1) \approx_c \mathsf{Rec}^{(1)}(0)$$

are computationally indistinguishable.

Statistical sender privacy (SSP)

An OT $= (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ protocol is *statistically sender private* if for any receiver message σ, there exists a bit b, such that for any pair of messages (m_0, m_1) the two distributions

$$\{\mathsf{Send}(m_0, m_1, \sigma)\} \approx_\Delta \{\mathsf{Send}(m_b, m_b, \sigma)\}$$

are statistically close.

Prior Work

- SSP OT from Decisional Diffie Hellman [Naor-Pinkas'01], Quadratic Residuosity [Halevi-Kalai'12]

Prior Work

- SSP OT from Decisional Diffie Hellman [Naor-Pinkas'01], Quadratic Residuosity [Halevi-Kalai'12]
- [Peikert-Vaikuntanathan-Waters'08] gave a universally composable (UC-secure) lattice-based OT protocol

Prior Work

- SSP OT from Decisional Diffie Hellman [Naor-Pinkas'01], Quadratic Residuosity [Halevi-Kalai'12]
- [Peikert-Vaikuntanathan-Waters'08] gave a universally composable (UC-secure) lattice-based OT protocol
- [Brakerski-Döttling'18] gave first SSP OT protocol from lattice assumptions

Prior Work

- SSP OT from Decisional Diffie Hellman [Naor-Pinkas'01], Quadratic Residuosity [Halevi-Kalai'12]
- [Peikert-Vaikuntanathan-Waters'08] gave a universally composable (UC-secure) lattice-based OT protocol
- [Brakerski-Döttling'18] gave first SSP OT protocol from lattice assumptions
- SSP OT from compressible fully-homomorphic encryption [Gentry-Halevi'19] [Brakerski-Döttling-Garg-Malavolta'19] [Badrinarayanan-Garg-Ishai-Sahai-Wadia'17]

Prior Work

- SSP OT from Decisional Diffie Hellman [Naor-Pinkas'01], Quadratic Residuosity [Halevi-Kalai'12]
- [Peikert-Vaikuntanathan-Waters'08] gave a universally composable (UC-secure) lattice-based OT protocol
- [Brakerski-Döttling'18] gave first SSP OT protocol from lattice assumptions
- SSP OT from compressible fully-homomorphic encryption [Gentry-Halevi'19] [Brakerski-Döttling-Garg-Malavolta'19] [Badrinarayanan-Garg-Ishai-Sahai-Wadia'17]

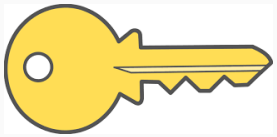
Scheme	Modulus q	Receiver Comm. (bits)	Sender Comm. (bits)	Overall Rate	Operations
[BD18]	$\Theta(n^3 \log^{2.5} n \cdot \gamma(n))$	$\Theta(n^2 \log^2 n)$	$\Theta(n \log^2 n)$	$\Theta(1/n \log^2 n)$	$\Theta(n^\omega)$
[DGI+19]	$\boldsymbol{\Theta(n^{2.5})}$	$\Theta(n^2 \log^2 n)$	$\boldsymbol{\Theta(n \log n)}$	$\Theta(1/n \log^2 n)$	$\Theta(n^3 \log n)$
[GH19]	$\Omega(n^{17.5} \log^{10} n)$	$\Theta(n^2 \log^2 n)$	$\Theta(n^2 \log n)$	$\Theta(1/n \log^2 n)$	$\Omega(n^{1+\omega})$
[BDGM19]	$\Theta(n^{2.5} \log^2 n)$	$\Theta(n^2 \log^2 n)$	$\boldsymbol{\Theta(n \log n)}$	$\Theta(1/n \log^2 n)$	$\Omega(n^3 \log^2 n)$
This work	$\Theta(n^4 \gamma^6(n))$	$\boldsymbol{\Theta(n \log n)}$	$\boldsymbol{\Theta(n \log n)}$	$\boldsymbol{\Theta(1/\log n)}$	$\boldsymbol{\Theta(n \log n)}$

SSP OT from Lossy Encryption [PVW'08, BD'18]

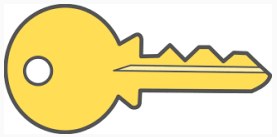
Following [PVW'08, BD'18] we construct SSP OT from "lossy encryption."

SSP OT from Lossy Encryption [PVW'08, BD'18]

Following [PVW'08, BD'18] we construct SSP OT from "lossy encryption."



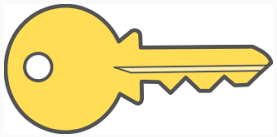
pk_0



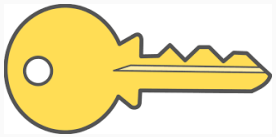
pk_1

SSP OT from Lossy Encryption [PVW'08, BD'18]

Following [PVW'08, BD'18] we construct SSP OT from "lossy encryption."



pk_0

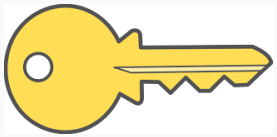


pk_1

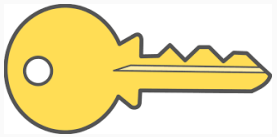
Message preserving

SSP OT from Lossy Encryption [PVW'08, BD'18]

Following [PVW'08, BD'18] we construct SSP OT from "lossy encryption."



pk_0



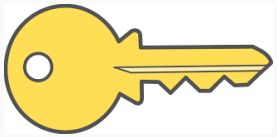
pk_1

Message preserving

Message lossy

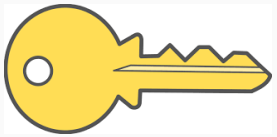
SSP OT from Lossy Encryption [PVW'08, BD'18]

Following [PVW'08, BD'18] we construct SSP OT from "lossy encryption."



pk_0

$\approx$



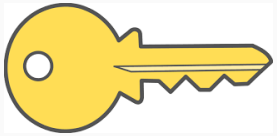
pk_1

Message preserving

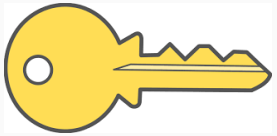
Message lossy

SSP OT from Lossy Encryption [PVW'08, BD'18]

Following [PVW'08, BD'18] we construct SSP OT from "lossy encryption."



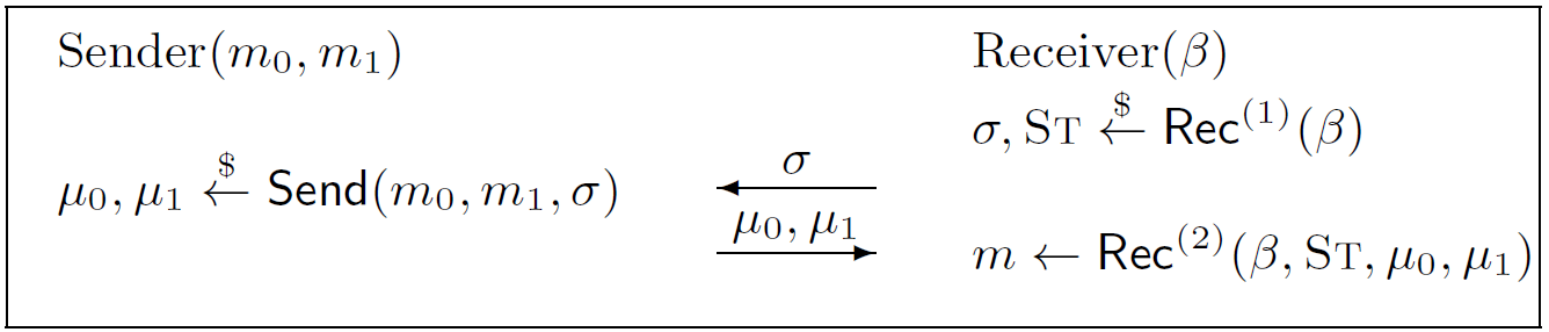
pk_0 $\approx$



pk_1

Message preserving

Message lossy



Lattice Defs

- A *lattice* is a discrete additive subgroup of $\mathbb{R}^m$

[0] Image credit: Oded Regev

Lattice Defs

- A *lattice* is a discrete additive subgroup of $\mathbb{R}^m$
- Given a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$, define

$$\Lambda(\mathbf{B}) := \{\mathbf{B}^t \mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$$

[0] Image credit: Oded Regev

Lattice Defs

- A *lattice* is a discrete additive subgroup of $\mathbb{R}^m$
- Given a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$, define

$$\Lambda(\mathbf{B}) := \{\mathbf{B}^t \mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$$

- The *dual lattice* Λ^* is defined

$$\Lambda^* := \{\mathbf{x} \in \mathbb{R}^m : \forall \mathbf{y} \in \Lambda, \langle x, y \rangle \in \mathbb{Z}\}$$

[0] Image credit: Oded Regev

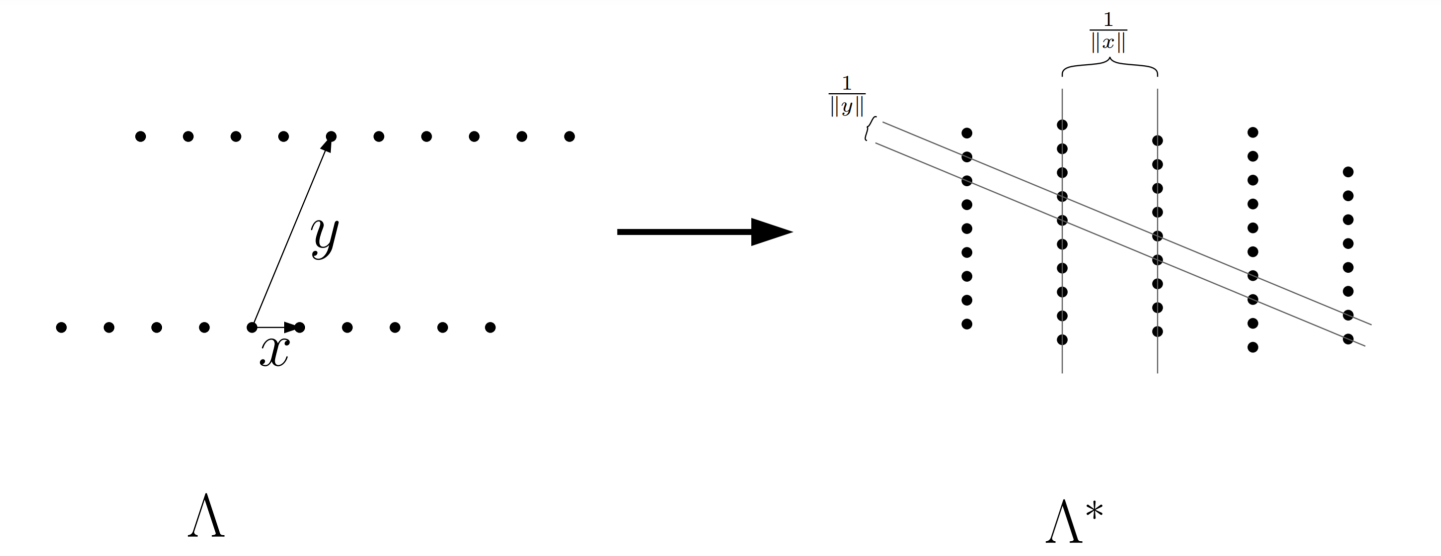
Lattice Defs

- A *lattice* is a discrete additive subgroup of $\mathbb{R}^m$
- Given a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$, define

$$\Lambda(\mathbf{B}) := \{\mathbf{B}^t \mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$$

- The *dual lattice* Λ^* is defined

$$\Lambda^* := \{\mathbf{x} \in \mathbb{R}^m : \forall \mathbf{y} \in \Lambda, \langle x, y \rangle \in \mathbb{Z}\}$$



[0] Image credit: Oded Regev

Lossy Encryption from Lattices (Intuition)

- Given lattice $\Lambda(\mathbf{B})$, encode message $\mathbf{m} \in \mathbb{Z}_q^n$ as follows:

[0] Image credit: Oded Regev

Lossy Encryption from Lattices (Intuition)

- Given lattice $\Lambda(\mathbf{B})$, encode message $\mathbf{m} \in \mathbb{Z}_q^n$ as follows:
- Use $\mathbf{m}$ to select a lattice vector $\mathbf{B}^t\mathbf{m}$

[0] Image credit: Oded Regev

Lossy Encryption from Lattices (Intuition)

- Given lattice $\Lambda(\mathbf{B})$, encode message $\mathbf{m} \in \mathbb{Z}_q^n$ as follows:
- Use $\mathbf{m}$ to select a lattice vector $\mathbf{B}^t\mathbf{m}$
- Sample $\mathbf{e} \sim D_{\mathbb{Z}^m,\sigma}$

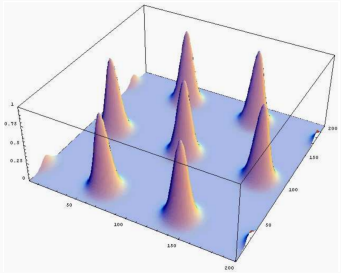
[0] Image credit: Oded Regev

Lossy Encryption from Lattices (Intuition)

- Given lattice $\Lambda(\mathbf{B})$, encode message $\mathbf{m} \in \mathbb{Z}_q^n$ as follows:
- Use $\mathbf{m}$ to select a lattice vector $\mathbf{B}^t\mathbf{m}$
- Sample $\mathbf{e} \sim D_{\mathbb{Z}^m,\sigma}$
- Return perturbed vector $\mathbf{B}^t\mathbf{m} + \mathbf{e}$

[0] Image credit: Oded Regev

Lossy Encryption from Lattices (Intuition)

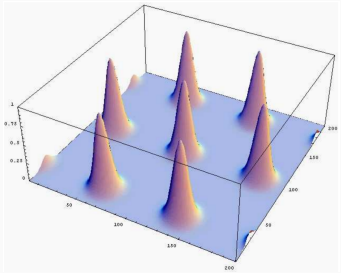


Low relative noise

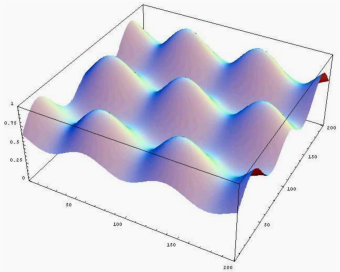
- Given lattice $\Lambda(\mathbf{B})$, encode message $\mathbf{m} \in \mathbb{Z}_q^n$ as follows:
- Use $\mathbf{m}$ to select a lattice vector $\mathbf{B}^t\mathbf{m}$
- Sample $\mathbf{e} \sim D_{\mathbb{Z}^m,\sigma}$
- Return perturbed vector $\mathbf{B}^t\mathbf{m} + \mathbf{e}$
- In a sufficiently sparse lattice with respect to σ, can efficiently recover $\mathbf{m}$ given a short basis $\mathbf{B}^*$ for Λ^* [Babai'86], [Aharanov-Regev'05], [Liu-Lyubashevsky-Micciancio'06]

[0] Image credit: Oded Regev

Lossy Encryption from Lattices (Intuition)



Low relative noise

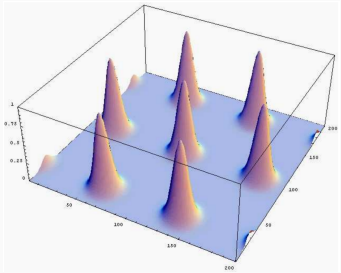


Moderate noise

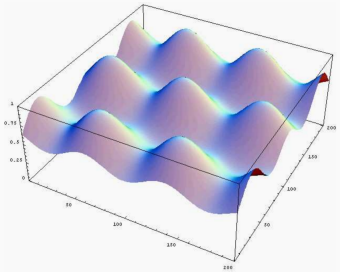
- Given lattice $\Lambda(\mathbf{B})$, encode message $\mathbf{m} \in \mathbb{Z}_q^n$ as follows:
- Use $\mathbf{m}$ to select a lattice vector $\mathbf{B}^t\mathbf{m}$
- Sample $\mathbf{e} \sim D_{\mathbb{Z}^m,\sigma}$
- Return perturbed vector $\mathbf{B}^t\mathbf{m} + \mathbf{e}$
- In a sufficiently sparse lattice with respect to σ, can efficiently recover $\mathbf{m}$ given a short basis $\mathbf{B}^*$ for Λ^* [Babai'86], [Aharanov-Regev'05], [Liu-Lyubashevsky-Micciancio'06]
- But for large σ, even maximum likelihood decoding doesn't work

[0] Image credit: Oded Regev

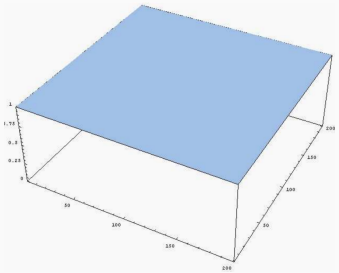
Lossy Encryption from Lattices (Intuition)



Low relative noise



Moderate noise



Oof...

- Given lattice $\Lambda(\mathbf{B})$, encode message $\mathbf{m} \in \mathbb{Z}_q^n$ as follows:
- Use $\mathbf{m}$ to select a lattice vector $\mathbf{B}^t\mathbf{m}$
- Sample $\mathbf{e} \sim D_{\mathbb{Z}^m,\sigma}$
- Return perturbed vector $\mathbf{B}^t\mathbf{m} + \mathbf{e}$
- In a sufficiently sparse lattice with respect to σ, can efficiently recover $\mathbf{m}$ given a short basis $\mathbf{B}^*$ for Λ^* [Babai'86], [Aharanov-Regev'05], [Liu-Lyubashevsky-Micciancio'06]
- But for large σ, even maximum likelihood decoding doesn't work

[0] Image credit: Oded Regev

Lossy Encryption from Lattices (Intuition)

Algorithm 1 $\mathrm{Rec}^{(1)}$

Input: $b \in \{0,1\}$

if $b = 0$ then
  $\mathbf{B} \leftarrow$ basis for sparse lattice
else
  $\mathbf{B} \leftarrow$ basis for dense lattice
return $\mathbf{B}$

Algorithm 2 Send

Input: $\mathbf{B}, m_0, m_1 \in \{0,1\}^n$

$\mathbf{y}_0 \leftarrow \mathsf{Encode}(\Lambda(\mathbf{B}), m_0)$
$\mathbf{y}_1 \leftarrow \mathsf{Encode}(\Lambda^*(\mathbf{B}), m_1)$
return $(\mathbf{y}_0, \mathbf{y}_1)$

Algorithm 3 $\mathrm{Rec}^{(2)}$ Input: $b \in \{0,1\}$, ST, (μ_0, μ_1)

if $b = 0$ then
  $m \leftarrow \mathsf{Decode}(\Lambda, \mathbf{y}_0)$
else
  $m \leftarrow \mathsf{Decode}(\Lambda^*, \mathbf{y}_1)$
return m

Lossy Encryption from Lattices (Intuition)

$\mathsf{Rec}^{(1)}$
Input: $b \in \{0,1\}$

if $b = 0$ then
  $\mathbf{B} \leftarrow$ basis for sparse lattice
else
  $\mathbf{B} \leftarrow$ basis for dense lattice
return $\mathbf{B}$

Send
Input: $\mathbf{B}, m_0, m_1 \in \{0,1\}^n$

$\mathbf{y}_0 \leftarrow \mathsf{Encode}(\Lambda(\mathbf{B}), \sigma, m_0)$
$\mathbf{y}_1 \leftarrow \mathsf{Encode}(\Lambda^*(\mathbf{B}), \sigma, m_1)$
return $(\mathbf{y}_0, \mathbf{y}_1)$

$\mathsf{Rec}^{(2)}$
Input: $b \in \{0,1\}, \mathbf{B}, y_0, y_1$

if $b = 0$ then
  $m \leftarrow \mathsf{Decode}(\Lambda, \mathbf{y}_0)$
else
  $m \leftarrow \mathsf{Decode}(\Lambda^*, \mathbf{y}_1)$
return m

But is it lossy... enough?

Lossy Encryption from Lattices (Intuition)

$\mathsf{Rec}^{(1)}$
Input: $b \in \{0,1\}$

if $b = 0$ then
  $\mathsf{B} \leftarrow$ basis for sparse lattice
else
  $\mathsf{B} \leftarrow$ basis for dense lattice
return B

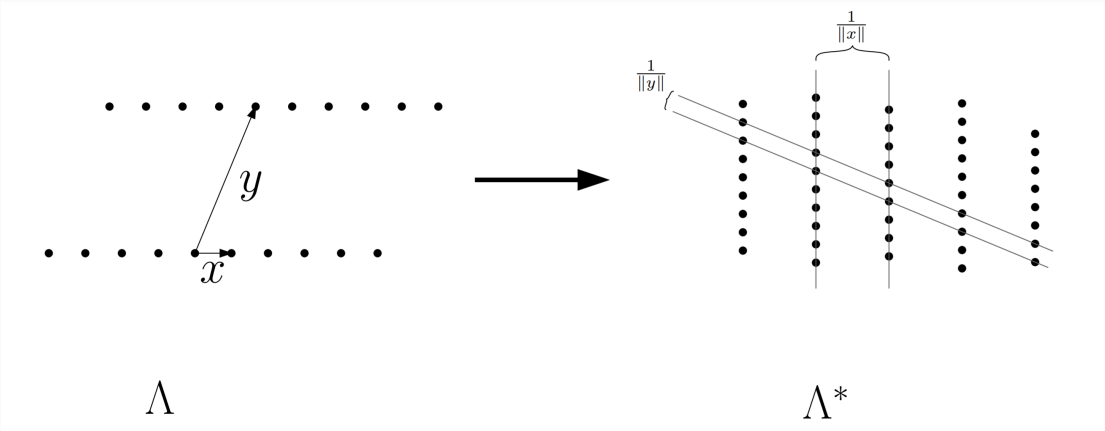
Send
Input: $\mathsf{B}, m_0, m_1 \in \{0,1\}^n$

$\mathsf{y}_0 \leftarrow \mathsf{Encode}(\Lambda(\mathsf{B}), \sigma, m_0)$
$\mathsf{y}_1 \leftarrow \mathsf{Encode}(\Lambda^*(\mathsf{B}), \sigma, m_1)$
return $(\mathsf{y}_0, \mathsf{y}_1)$

$\mathsf{Rec}^{(2)}$
Input: $b \in \{0,1\}, \mathsf{B}, y_0, y_1$

if $b = 0$ then
  $m \leftarrow \mathsf{Decode}(\Lambda, \mathsf{y}_0)$
else
  $m \leftarrow \mathsf{Decode}(\Lambda^*, \mathsf{y}_1)$
return m

But is it lossy... enough?



Module Lattices

- Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$

Module Lattices

- Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$
- Ideals of $\mathcal{R}$ embed into $\mathbb{Z}^n$ as a lattice under the *coefficient embedding* σ_c:

$$\sigma_c\left(\sum_{i=0}^{n-1} a_i X^i\right) \mapsto (a_0, \ldots, a_{n-1})$$

Module Lattices

- Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$
- Ideals of $\mathcal{R}$ embed into $\mathbb{Z}^n$ as a lattice under the *coefficient embedding* σ_c:

$$\sigma_c\left(\sum_{i=0}^{n-1} a_i X^i\right) \mapsto (a_0, \ldots, a_{n-1})$$

- Given a matrix $\mathbf{B} \in \mathcal{R}_q^{\ell \times m}$, define the *q-ary module lattice*

$$\Lambda_q(\mathbf{B}) := \{\mathbf{x} \in \mathcal{R}^m : \mathbf{x} = \mathbf{B}^t\mathbf{y} \bmod q, \mathbf{y} \in \mathcal{R}^\ell\}$$

Module Lattices

- Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$
- Ideals of $\mathcal{R}$ embed into $\mathbb{Z}^n$ as a lattice under the *coefficient embedding* σ_c:

$$\sigma_c\left(\sum_{i=0}^{n-1} a_i X^i\right) \mapsto (a_0, \ldots, a_{n-1})$$

- Given a matrix $\mathbf{B} \in \mathcal{R}_q^{\ell \times m}$, define the *q-ary module lattice*

$$\Lambda_q(\mathbf{B}) := \{\mathbf{x} \in \mathcal{R}^m : \mathbf{x} = \mathbf{B}^t\mathbf{y} \bmod q, \mathbf{y} \in \mathcal{R}^\ell\}$$

- Fact: for q-ary module lattice Λ_q over $\mathcal{R}$ of dimension n,

$$\lambda_1(\Lambda_q) = \lambda_2(\Lambda_q) = \cdots = \lambda_n(\Lambda_q)$$

Module Lattices

- Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$
- Ideals of $\mathcal{R}$ embed into $\mathbb{Z}^n$ as a lattice under the *coefficient embedding* σ_c:

$$\sigma_c\left(\sum_{i=0}^{n-1} a_i X^i\right) \mapsto (a_0, \ldots, a_{n-1})$$

- Given a matrix $\mathbf{B} \in \mathcal{R}_q^{\ell \times m}$, define the *q-ary module lattice*

$$\Lambda_q(\mathbf{B}) := \{\mathbf{x} \in \mathcal{R}^m : \mathbf{x} = \mathbf{B}^t\mathbf{y} \bmod q, \mathbf{y} \in \mathcal{R}^\ell\}$$

- Fact: for q-ary module lattice Λ_q over $\mathcal{R}$ of dimension n,

$$\lambda_1(\Lambda_q) = \lambda_2(\Lambda_q) = \cdots = \lambda_n(\Lambda_q)$$

Module Lattices

- Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$
- Ideals of $\mathcal{R}$ embed into $\mathbb{Z}^n$ as a lattice under the *coefficient embedding* σ_c:

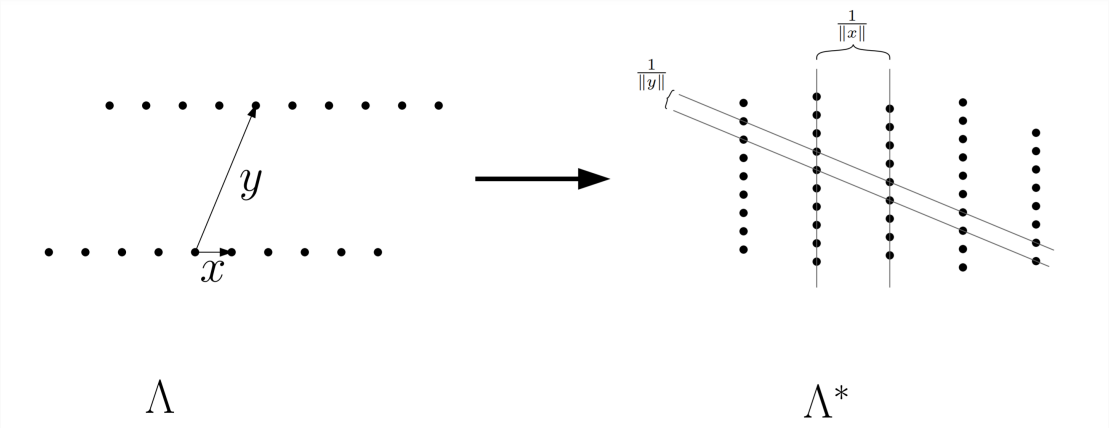
$$\sigma_c\left(\sum_{i=0}^{n-1} a_i X^i\right) \mapsto (a_0, \ldots, a_{n-1})$$

- Given a matrix $\mathbf{B} \in \mathcal{R}_q^{\ell \times m}$, define the *q-ary module lattice*

$$\Lambda_q(\mathbf{B}) := \{\mathbf{x} \in \mathcal{R}^m : \mathbf{x} = \mathbf{B}^t\mathbf{y} \bmod q, \mathbf{y} \in \mathcal{R}^\ell\}$$

- Fact: for q-ary module lattice Λ_q over $\mathcal{R}$ of dimension n,

$$\lambda_1(\Lambda_q) = \lambda_2(\Lambda_q) = \cdots = \lambda_n(\Lambda_q)$$



Algorithm 4 $\mathsf{Rec}^{(1)}$

Input: $b \in \{0,1\}$

if $b = 0$ then
  $\mathbf{B} \leftarrow$ basis for sparse ($\mathcal{R}$-module) lattice
else
  $\mathbf{B} \leftarrow$ basis for dense ($\mathcal{R}$-module) lattice
return $\mathbf{B}$

Algorithm 5 Send

Input: $\mathbf{B}, m_0, m_1 \in \{0,1\}^n$

$\mathbf{x} \leftarrow \mathcal{R}_q^\ell$
$\mathbf{y}_0 \leftarrow \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})$
$\mathbf{r} \leftarrow \{0,1\}^\ell$
$\mathtt{mask} \leftarrow \mathtt{Ext}(\mathbf{r}, \mathbf{x})$
$\mu_0 \leftarrow \mathbf{m}_0 \oplus \mathtt{mask}$
$\mathbf{y}_1 \leftarrow \mathsf{Encode}(\Lambda_q^*(\mathbf{B}), m_1)$
return $(\mathbf{y}_0, \mathbf{r}, \mu_0, \mathbf{y}_1)$

Algorithm 6 $\mathsf{Rec}^{(2)}$ Input: $b \in \{0,1\}$, $\mathbf{B}$, $(\mathbf{y}_0, \mathbf{r}, \mu_0, y_1)$

if $b = 0$ then
  $\mathbf{x} \leftarrow \mathsf{Decode}(\Lambda, \mathbf{y}_0)$
  $\mathtt{mask} \leftarrow \mathtt{Ext}(\mathbf{r}, \mathbf{x})$
  $\mathbf{m} \leftarrow \mathtt{mask} \oplus \mu_0$
else
  $\mathbf{m}_1 \leftarrow \mathsf{Decode}(\Lambda^*, \mathbf{y}_1)$
return m

Proof Techniques

Need to show that either

1. m_0 is statistically hidden, so $H_\infty(\mathsf{x} \mid \mathsf{y} = \mathsf{Encode}(\Lambda_q(\mathsf{B}), \mathsf{x})) \geq n$
2. m_1 is statistically hidden by encoding

Proof Techniques

Need to show that either

1. m_0 is statistically hidden, so $H_\infty(\mathbf{x} \mid \mathbf{y} = \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})) \geq n$
2. m_1 is statistically hidden by encoding

We use known results on Gaussian measure over lattices

$$\rho_\sigma(\Lambda) := \sum_{\mathbf{v} \in \Lambda} e^{-\pi(\|v\|/\sigma)^2}$$

Proof Techniques

Need to show that either

1. m_0 is statistically hidden, so $H_\infty(\mathbf{x} \mid \mathbf{y} = \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})) \geq n$
2. m_1 is statistically hidden by encoding

We use known results on Gaussian measure over lattices

$$\rho_\sigma(\Lambda) := \sum_{\mathbf{v} \in \Lambda} e^{-\pi(\|v\|/\sigma)^2}$$

The *smoothing parameter* η_ϵ of a lattice Λ is defined

$$\eta_\epsilon(\Lambda) := \mathsf{min}\{\sigma \in \mathbb{R} : \rho_{1/\sigma}(\Lambda^*) \leq 1 + \epsilon\}$$

Theorem (Lyubashevsky-Peikert-Regev'13)

Let D_σ denote the Gaussian distribution with parameter σ. If $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$, and $\mathbf{x} \leftarrow D_\sigma^m$, then

$$\mathbf{Bx} \approx_\Delta \mathcal{U}(\mathcal{R}_q^\ell)$$

Proof Techniques

Need to show that either

1. m_0 is statistically hidden, so $H_\infty(\mathbf{x} \mid \mathbf{y}_0 = \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})) \geq n$
2. m_1 is statistically hidden by encoding

Theorem (Lyubashevsky-Peikert-Regev'13)

Let D_σ denote the Gaussian distribution with parameter σ. If $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$, and $\mathbf{x} \leftarrow D_\sigma^m$, then

$$\mathbf{B}\mathbf{x} \approx_\Delta \mathcal{U}(\mathcal{R}_q^\ell)$$

Proof Techniques

Need to show that either

1. m_0 is statistically hidden, so $H_\infty(\mathbf{x} \mid \mathbf{y}_0 = \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})) \geq n$
2. m_1 is statistically hidden by encoding

Theorem (Lyubashevsky-Peikert-Regev'13)

Let D_σ denote the Gaussian distribution with parameter σ. If $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$, and $\mathbf{x} \leftarrow D_\sigma^m$, then

$$\mathbf{B}\mathbf{x} \approx_\Delta \mathcal{U}(\mathcal{R}_q^\ell)$$

On the one hand, if $\Lambda(\mathbf{B})$ has no short vectors, then $\eta_\epsilon(\Lambda^*)$ is small $\rightarrow$ encode m_1 by $\mathbf{B}\mathbf{x} + m_1$ for $\mathbf{x} \sim D_\sigma$ and $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$

Proof Techniques

Need to show that either

1. m_0 is statistically hidden, so $H_\infty(\mathbf{x} \mid \mathbf{y}_0 = \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})) \geq n$
2. m_1 is statistically hidden by encoding

Theorem (Lyubashevsky-Peikert-Regev'13)

Let D_σ denote the Gaussian distribution with parameter σ. If $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$, and $\mathbf{x} \leftarrow D_\sigma^m$, then

$$\mathbf{B}\mathbf{x} \approx_\Delta \mathcal{U}(\mathcal{R}_q^\ell)$$

On the one hand, if $\Lambda(\mathbf{B})$ has no short vectors, then $\eta_\epsilon(\Lambda^*)$ is small $\rightarrow$ encode m_1 by $\mathbf{B}\mathbf{x} + m_1$ for $\mathbf{x} \sim D_\sigma$ and $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$

Proof Techniques

Need to show that either

1. m_0 is statistically hidden, so $H_\infty(\mathbf{x} \mid \mathbf{y}_0 = \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})) \geq n$
2. m_1 is statistically hidden by encoding

Theorem (Lyubashevsky-Peikert-Regev'13)

Let D_σ denote the Gaussian distribution with parameter σ. If $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$, and $\mathbf{x} \leftarrow D_\sigma^m$, then

$$\mathbf{B}\mathbf{x} \approx_\Delta \mathcal{U}(\mathcal{R}_q^\ell)$$

On the one hand, if $\Lambda(\mathbf{B})$ has no short vectors, then $\eta_\epsilon(\Lambda^*)$ is small $\rightarrow$ encode m_1 by $\mathbf{B}\mathbf{x} + m_1$ for $\mathbf{x} \sim D_\sigma$ and $\sigma > q\eta_\epsilon(\Lambda^*(\mathbf{B}))$

On the other hand, if $\Lambda(\mathbf{B})$ has at least one short vector, it has n of them. So $H_\infty(\mathbf{x} \mid \mathbf{y}_0 = \mathsf{Encode}(\Lambda_q(\mathbf{B}), \mathbf{x})) \geq n$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathsf{B}, m)$:

- $\mathsf{x} \leftarrow \mathcal{R}_q^\ell$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathsf{B}, m)$:

- $\mathsf{x} \leftarrow \mathcal{R}_q^\ell$
- $\mathsf{e} \leftarrow D_{\mathcal{R}^m, \sigma}$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathsf{B}, m)$:

- $\mathsf{x} \leftarrow \mathcal{R}_q^\ell$
- $\mathsf{e} \leftarrow D_{\mathcal{R}^m, \sigma}$
- $\mathsf{y} \leftarrow \mathsf{B}^t\mathsf{x} + \mathsf{e} \mod q$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathsf{B}, m)$:

- $\mathsf{x} \leftarrow \mathcal{R}_q^\ell$
- $\mathsf{e} \leftarrow D_{\mathcal{R}^m, \sigma}$
- $\mathsf{y} \leftarrow \mathsf{B}^t\mathsf{x} + \mathsf{e} \mod q$

In this case, we'll show
$H_\infty(\mathsf{x} \mid \mathsf{y}) \geq n$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathsf{B}, m)$:

- $\mathsf{x} \leftarrow \mathcal{R}_q^\ell$
- $\mathsf{e} \leftarrow D_{\mathcal{R}^m, \sigma}$
- $\mathsf{y} \leftarrow \mathsf{B}^t\mathsf{x} + \mathsf{e} \mod q$

In this case, we'll show
$H_\infty(\mathsf{x} \mid \mathsf{y}) \geq n$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathsf{B}, m)$:

- $\mathsf{x} \leftarrow \mathcal{R}_q^\ell$
- $\mathsf{e} \leftarrow D_{\mathcal{R}^m, \sigma}$
- $\mathsf{y} \leftarrow \mathsf{B}^t\mathsf{x} + \mathsf{e} \mod q$

In this case, we'll show
$H_\infty(\mathsf{x} \mid \mathsf{y}) \geq n$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathsf{B}, m)$:

- $\mathsf{x} \leftarrow \mathcal{R}_q^\ell$
- $\mathsf{e} \leftarrow D_{\mathcal{R}^m, \sigma}$
- $\mathsf{y} \leftarrow \mathsf{B}^t\mathsf{x} + \mathsf{e} \mod q$

In this case, we'll show
$H_\infty(\mathsf{x} \mid \mathsf{y}) \geq n$

Case 1: Λ has at least n short vectors

$\mathsf{Encode}(\mathbf{B}, m)$:

- $\mathbf{x} \leftarrow \mathcal{R}_q^\ell$
- $\mathbf{e} \leftarrow D_{\mathcal{R}^m, \sigma}$
- $\mathbf{y} \leftarrow \mathbf{B}^t\mathbf{x} + \mathbf{e} \mod q$

In this case, we'll show $H_\infty(\mathbf{x} \mid \mathbf{y}) \geq n$

Proof idea (following BD'18):

- Mostly likely $\mathbf{x}$ given $\mathbf{y}$ is the one that minimizes $\mathbf{e}$
- Means closest lattice point to $\mathbf{y}$ is $\mathbf{B}^t\mathbf{x}$
- So $\mathbf{e} \in \mathcal{V}(\Lambda(\mathbf{B}))$
- But if $\Lambda(\mathbf{B})$ has many short vectors, $\Pr[\mathbf{e} \in \mathcal{V}(\Lambda(\mathbf{B}))]$ can't be too large
- Can show $\Pr[\mathbf{e} \in \mathcal{V}(\Lambda(\mathbf{B}))]$ down to 2^{-n} and so $H_\infty(\mathbf{x} \mid \mathbf{y}) \geq 3n/2$

Summary

- Efficient statistically sender private oblivious transfer from $\mathcal{R}$-module lattices
- Used structure of $\mathcal{R}$-module lattices to get improvements in efficiency above and beyond what is standard when moving to the algebraically structured lattice setting.
- We get $O(\log \lambda)$ communication overhead for messages of length λ. Is $O(1)$ possible?