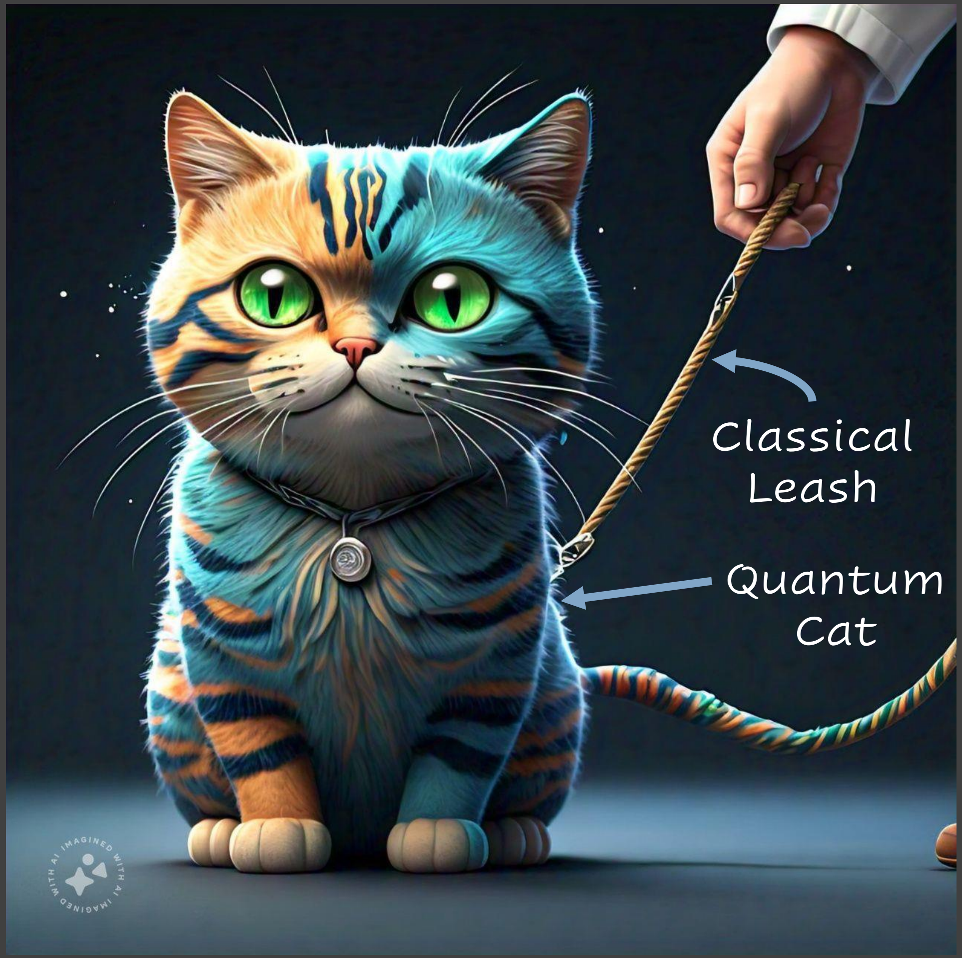


CLASSICAL VERIFICATION OF QUANTUM COMPUTATIONS

(PART2: FUNCTION CONSTRUCTIONS)

COL872: Lattices in CS

Anish Banerjee Shankh Gupta

Based on the [Mah23] of the same name

Main Results (Informal)

LWE is hard for a BQP machine

↓

There exists an **extended trapdoor claw-free family**.

↓

All decision problems in BQP can be verified by an efficient classical machine through interaction.

Measurement Protocol

Goal: Force the prover to behave as the verifier's trusted measurement device

Relation to this course

- ETCFs are built using LWE.
- Extensively used in the construction of several verification protocols.
- However, we only have **approximate constructions.**
- We want to study these constructions and understand why we don't have exact.

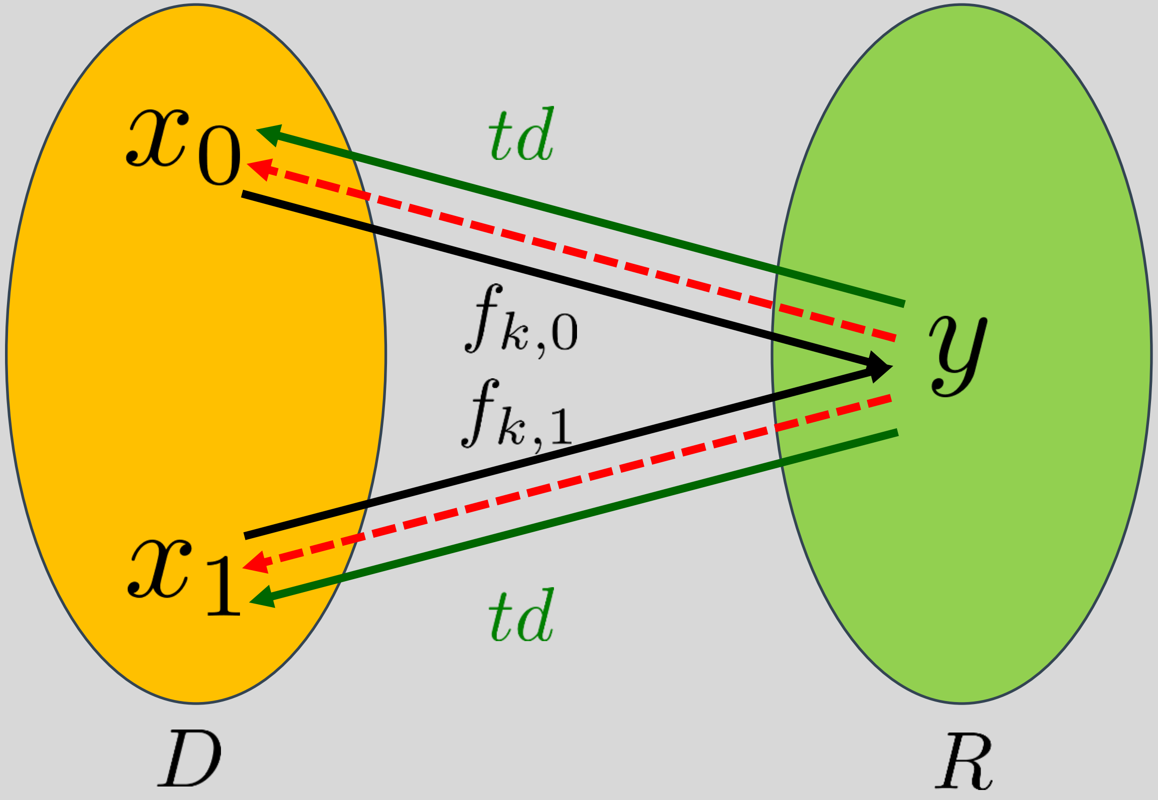
Why should you study this?

- **BQP Verification**
  - [BCM+21], [BKVV20], [Mah23] etc.
- **Deniable Encryption**
  - [CGV22]
- **QPKE with Certified Deletion**
  - [HMNY21]

Trapdoor Claw-free functions

$f_{k,0}, f_{k,1} : D \to R$

Injective, same range



Hard to find a **claw** (x_0, x_1) such that

$$f_{k,0}(x_0) = f_{k,1}(x_1)$$

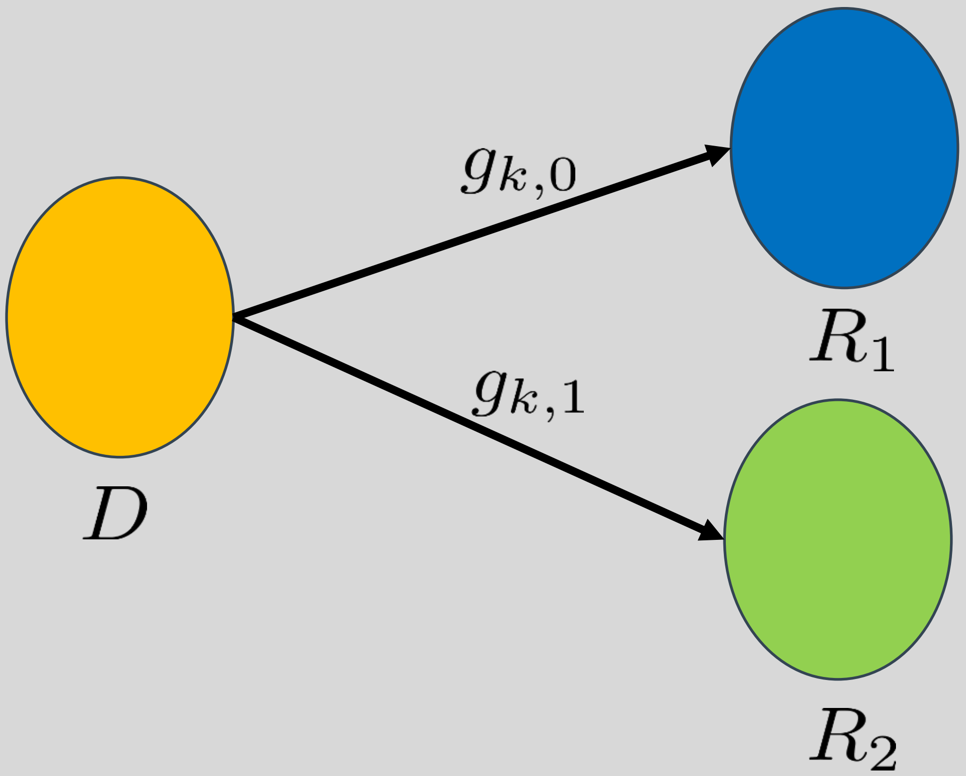
without td.

Also satisfies two other **adaptive hardcore bit** properties.

Trapdoor Injective Functions

$g_{k,0}, g_{k,1} : D \to R$

Injective, disjoint range



Given $y = gk_{\ b}(x)$, hard to find (b, x) without td.

ETCF=TCF+TIF+Injective Invariance

Hard to distinguish between (f_0,f_1) and (g_0,g_1)

Unfortunately, we don't have **exact** constructions!

Truncated Discrete Gaussian

$$D_{\mathbb{Z}_q,B}(x) = \frac{e^{\frac{-\pi\|x\|^2}{B^2}}}{\sum_{x\in\mathcal{D}} e^{\frac{-\pi\|x\|^2}{B^2}}}$$

$$\mathcal{D} = \{x \in \mathbb{Z}_q \mid \|x\| \leq B\}$$

$$D_{\mathbb{Z}_q^m,B}(\mathbf{x}) = D_{\mathbb{Z}_q,B}(x_1)D_{\mathbb{Z}_q,B}(x_2)\dots D_{\mathbb{Z}_q,B}(x_m)$$

$$\mathcal{D}^m = \{\mathbf{x} \in \mathbb{Z}_q^m \mid \|\mathbf{x}\| \leq B\sqrt{m}\}$$

Trapdoors from Lattices

Theorem [MP11]

There is an efficient algorithm

$$(\boldsymbol{A}, td_{\boldsymbol{A}}) \leftarrow \mathrm{GenTrap}()$$

- Distribution of $\boldsymbol{A}$ $\approx$ Uniform Distribution
- Efficient Inversion

$$(\boldsymbol{s}, \boldsymbol{e}) \leftarrow \mathrm{Invert}(\boldsymbol{A}, td_{\boldsymbol{A}}, \boldsymbol{As} + \boldsymbol{e})$$

$$||\boldsymbol{e}|| \leq \frac{q}{C_T\sqrt{n \log \mathrm{q}}} = 2B_P\sqrt{m}$$

q: Modulus, $\boldsymbol{A}$ is of dimension m x n

Parameters

$$m = \Omega(n \log q)$$

$$B_L < B_V < B_P$$

$$B_P = \frac{q}{2C_T\sqrt{mn \log \mathrm{q}}}$$

$\frac{B_P}{B_V}, \frac{B_V}{B_L}$ are super-polynomial

Noisy-TCF Family

- The range of the functions is a probability density D_Y over Y

$$(f_{k,b}(\mathbf{x}))(\mathbf{y}) = D_{\mathbb{Z}_q^m, B_P}(\mathbf{y} - \mathbf{A}\mathbf{x} - b\mathbf{A}\mathbf{s})$$

- The trapdoor injective pair property is defined in terms of support of the densities
  - claw: identical supports
- We require an QPT procedure which generates the state

$$\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f_{k,b}(x))(y)}\, |x\rangle\, |y\rangle$$

  - Not possible!
  - We will create an approximation of this using a related family

Efficient Function Generation

- $(k, td) \leftarrow \mathrm{Gen}()$

- $(\boldsymbol{A}, td_{\boldsymbol{A}}) \leftarrow \mathrm{GenTrap}()$
- $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ $\quad\quad$ $\boldsymbol{e} \leftarrow_{D_{\mathbb{Z}_q^m}, B_V} \mathbb{Z}_q^m$
- $k = (\boldsymbol{A}, \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}), td = td_{\boldsymbol{A}}$

Trapdoor Injective Pair

- **Trapdoor:** For every $y \in \mathrm{Supp}\left(\mathrm{f_{k,b}}(\mathrm{x})\right)$

$$x \leftarrow \mathrm{Inv}_F(\mathrm{k}, \mathrm{td}, \mathrm{b}, \mathrm{y})$$

- **Injective Pair:** Perfect matching R_k

$$f_{k,0}(x_0) = f_{k,1}(x_1) \Leftrightarrow (x_0, x_1) \in \mathcal{R}_k$$

$$(f_{k,b}(\mathbf{x}))(\mathbf{y}) = D_{\mathbb{Z}_q^m, B_P}(\mathbf{y} - \mathbf{A}\mathbf{x} - b\mathbf{A}\mathbf{s})$$

$$\textsc{Supp}(f_{k,0}(x)) = \left\{\mathbf{A}\mathbf{x} + \mathbf{e}_0 \,|\, \|\mathbf{e}_0\| \leq B_p\sqrt{m}\right\}$$

$$\textsc{Supp}(f_{k,1}(x)) = \left\{\mathbf{A}(\mathbf{x} + \mathbf{s}) + \mathbf{e}_0 \,|\, \|\mathbf{e}_0\| \leq B_p\sqrt{m}\right\}$$

$$(\mathbf{x} + b\mathbf{s}, \mathbf{e}_0) \leftarrow \textsc{Invert}(\mathbf{A}, \mathsf{td}, \mathbf{y})$$

The inversion works due to our choice of B_P

Perfect matching: $(\boldsymbol{x}, \boldsymbol{x} - \boldsymbol{s}) \in R_k$

Efficient Range Superposition

- **Inversion:** For all $(x_0, x_1) \in R_k$ and $y \in \mathrm{Supp}(f'_{k,b}(x_b))$

$$x_b \leftarrow \textsc{Inv}_{\mathcal{F}}(\mathsf{td}, b, y) \qquad x_{b\oplus 1} \leftarrow \textsc{Inv}_{\mathcal{F}}(\mathsf{td}, b \oplus 1, y)$$

- **Check:** $\mathrm{Chk}_F(\mathrm{k}, \mathrm{b}, \mathrm{x}, \mathrm{y})$ tells if $y \in \mathrm{Supp}\left(f'_{k,b}(x)\right)$

- **Close to *F*:**

$$\mathop{\mathbb{E}}_{x \leftarrow \mathcal{X}}[H^2(f_{k,b}(x), f'_{k,b}(x)] \leq \mathrm{negl}(\lambda)$$

- **Efficient Sampling:**

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{f'_{k,b}(x)}\, |x\rangle\, |y\rangle \leftarrow \textsc{Samp}_{\mathcal{F}}(k, b)$$

No longer have the perfect matching property!

Efficient Range Superposition (Construction)

$$(f'_{k,b}(\mathbf{x}))(\mathbf{y}) = D_{\mathbb{Z}_q^m,B_V}(\mathbf{y} - \mathbf{A}\mathbf{x} - b(\mathbf{A}\mathbf{s} + \mathbf{e}))$$

Inversion

$$\textsc{Supp}(f'_{k,0}(x)) = \left\{\mathbf{A}\mathbf{x} + \mathbf{e}_0 \middle| \|\mathbf{e}_0\| \leq B_p\sqrt{m}\right\}$$

$$\textsc{Supp}(f'_{k,1}(x)) = \left\{\mathbf{A}(\mathbf{x} + \mathbf{s}) + \mathbf{e}_0 + \mathbf{e} \middle| \|\mathbf{e}_0\| \leq B_p\sqrt{m}\right\}$$

Invert still works!

Check(k,b,x,y)

$$\left\|\mathbf{y} - \mathbf{A}\mathbf{x} - b'(\mathbf{A}\mathbf{s} + \mathbf{e})\right\| \leq B_P\sqrt{m}$$

Check for which b' this is true

Close to F

$f_{k,1}(x)$ and $f'_{k,1}(x)$ are Discrete Gaussians separated by $\boldsymbol{e}$

$$H^2(f_{k,1}(\mathbf{x}), f'_{k,1}(\mathbf{x})) \leq 1 - e^{\frac{-2\pi m B_V}{B_P}} \leq \frac{2\pi m B_V}{B_P}$$

$$|\psi_0\rangle = \sum_{\mathbf{e}_0 \in \mathbb{Z}_q^m} \sqrt{D_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{e}_0\rangle \xrightarrow{\text{Add an auxiliary registers}} \sum_{\mathbf{e}_0 \in \mathbb{Z}_q^m} \sqrt{D_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{e}_0\rangle\, |0\rangle\, |0\rangle$$

$$\xrightarrow{\text{Compute uniform superposition over second register}} \frac{1}{\sqrt{q^n}} \sum_{\mathbf{x} \in \mathbb{Z}_q^n \mathbf{e}_0 \in \mathbb{Z}_q^m} \sqrt{D_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{e}_0\rangle\, |\mathbf{x}\rangle\, |0\rangle$$

$$\xrightarrow{\text{Compute third register}} \frac{1}{\sqrt{q^n}} \sum_{\mathbf{x} \in \mathbb{Z}_q^n \mathbf{e}_0 \in \mathbb{Z}_q^m} \sqrt{D_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{e}_0\rangle\, |\mathbf{x}\rangle\, |\mathbf{e}_0 - \mathbf{A}\mathbf{x} - b(\mathbf{A}\mathbf{s} + \mathbf{e})\rangle$$

$$\xrightarrow{\text{Uncompute and discard first register}} \frac{1}{\sqrt{q^n}} \sum_{\mathbf{x} \in \mathbb{Z}_q^n \mathbf{e}_0 \in \mathbb{Z}_q^m} \sqrt{D_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{x}\rangle\, |\mathbf{e}_0 + \mathbf{A}\mathbf{x} + b(\mathbf{A}\mathbf{s} + \mathbf{e})\rangle$$

$$= \frac{1}{\sqrt{q^n}} \sum_{\mathbf{x} \in \mathbb{Z}_q^n, \mathbf{y} \in \mathbb{Z}_q^m} \sqrt{D_{\mathbb{Z}_q^m, B_P}(\mathbf{y} - \mathbf{A}\mathbf{x} - b(\mathbf{A}\mathbf{s} + \mathbf{e}))}\, |\mathbf{x}\rangle\, |\mathbf{y}\rangle = \frac{1}{\sqrt{q^n}} \sum_{\mathbf{x} \in \mathbb{Z}_q^n, \mathbf{y} \in \mathbb{Z}_q^m} \sqrt{(f'_{k,b}(x))(y)}\, |\mathbf{x}\rangle\, |\mathbf{y}\rangle$$

[Reg05]: This can be efficiently sampled

EFFICIENT RANGE SUPERPOSITION (SAMPLING)

TIF FAMILY

$$k = (\boldsymbol{A}, \boldsymbol{u})$$

$$(g_{k,b}(x))(y) = D_{\mathbb{Z}_q^m, B_P}(\mathbf{y} - \mathbf{A}\mathbf{x} - b\mathbf{u})$$

Efficient Function Generation

- $(k, td) \leftarrow \mathrm{Gen}()$

- $(\boldsymbol{A}, td_{\boldsymbol{A}}) \leftarrow \mathrm{GenTrap}()$
- $\boldsymbol{u} \leftarrow \mathbb{Z}_q^m$. If

  $(\boldsymbol{s}, \boldsymbol{e}) \leftarrow \mathrm{Invert}(\boldsymbol{A}, td_{\boldsymbol{A}}, \boldsymbol{u})$

  such that $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$ and $||\boldsymbol{e}|| \leq 2B_P\sqrt{m}$ then reject and resample.
- $k = (\boldsymbol{A}, \boldsymbol{u}), td = td_{\boldsymbol{A}}$

Disjoint Trapdoor Injective Pair

- **Trapdoor:** For every $y \in \mathrm{Supp}\left(\mathrm{g_{k,b}(x)}\right)$

$$(b, x) \leftarrow \mathrm{Inv}_G(\mathrm{k}, \mathrm{td}, \mathrm{y})$$

- **Disjoint Injective Pair:**

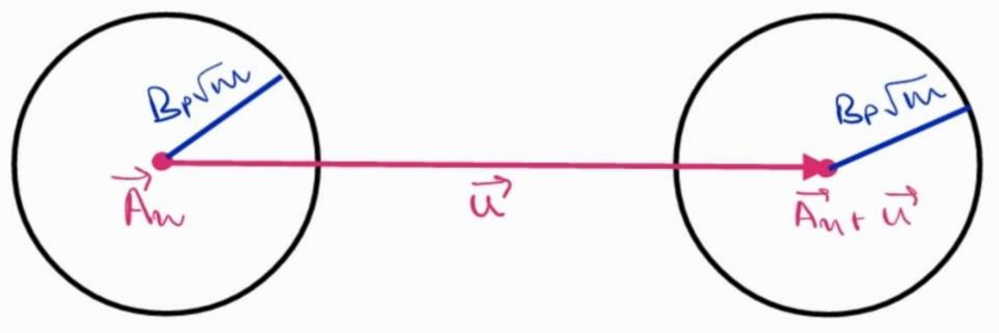
$$(b, x) \neq (b', x') \Leftrightarrow$$

$$\mathrm{Supp}\left(\mathrm{g_{k,b}(x)}\right) \cap \mathrm{Supp}\left(\mathrm{g_{k,b'}(x')}\right) = \phi$$

$$(g_{k,b}(x))(y) = D_{\mathbb{Z}_q^m, B_P}(\mathbf{y} - \mathbf{A}\mathbf{x} - b\mathbf{u})$$

$$\textsc{Supp}(g_{k,0}(x)) = \left\{\mathbf{A}\mathbf{x} + \mathbf{e}_0 \mid \|\mathbf{e}_0\| \leq B_p\sqrt{m}\right\}$$

$$\textsc{Supp}(g_{k,1}(x)) = \left\{\mathbf{A}\mathbf{x} + \mathbf{e}_0 + \mathbf{u} \mid \|\mathbf{e}_0\| \leq B_p\sqrt{m}\right\}$$



Efficient Range Superposition

- Check: $\mathrm{Chk}_G(\mathrm{k}, \mathrm{b}, \mathrm{x}, \mathrm{y})$ tells if $y \in \mathrm{Supp}\left(g_{k,b}\,(x)\right)$
- Efficient Sampling:

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{g_{k,b}(x)}\, |x\rangle\, |y\rangle \leftarrow \textsc{Samp}_{\mathcal{G}}(k, b)$$

Use the same functions as in NTCF family.

Injective Invariance

- The functions $\mathbf{Chk}_F, \mathbf{Samp}_F$ are the same as $\mathbf{Chk}_G, \mathbf{Samp}_G$
- No QPT adversary can distinguish between the outputs of the generation algorithms of F and G

$$\mathcal{D}_0 = \{(\mathbf{A}, \mathbf{As} + \mathbf{e}) \leftarrow \mathrm{GEN}_{\mathcal{F}_{\mathrm{LWE}}}(1^\lambda)\}$$
$$\mathcal{D}_1 = \{(\mathbf{A}, \mathbf{u}) \leftarrow \mathrm{GEN}_{\mathcal{G}_{\mathrm{LWE}}}(1^\lambda)\}$$

Reduces to hardness of LWE!

Hardcore Bit Properties - Overview

Adaptive Hardcore Bit

Hard to find (x_b, d) such that

$(d \neq 0)$ and $d \cdot (x_0 + x_1) = 0$

Hardcore Bit 2

There exists a string $\boldsymbol{d}$ such that for all claws $(\boldsymbol{x_0}, \boldsymbol{x_1})$

$$d \cdot (x_0 + x_1)$$

is the same bit and is hard to compute

Adaptive Hardcore Bit

For any QPT Adversary $\mathcal{A}$,

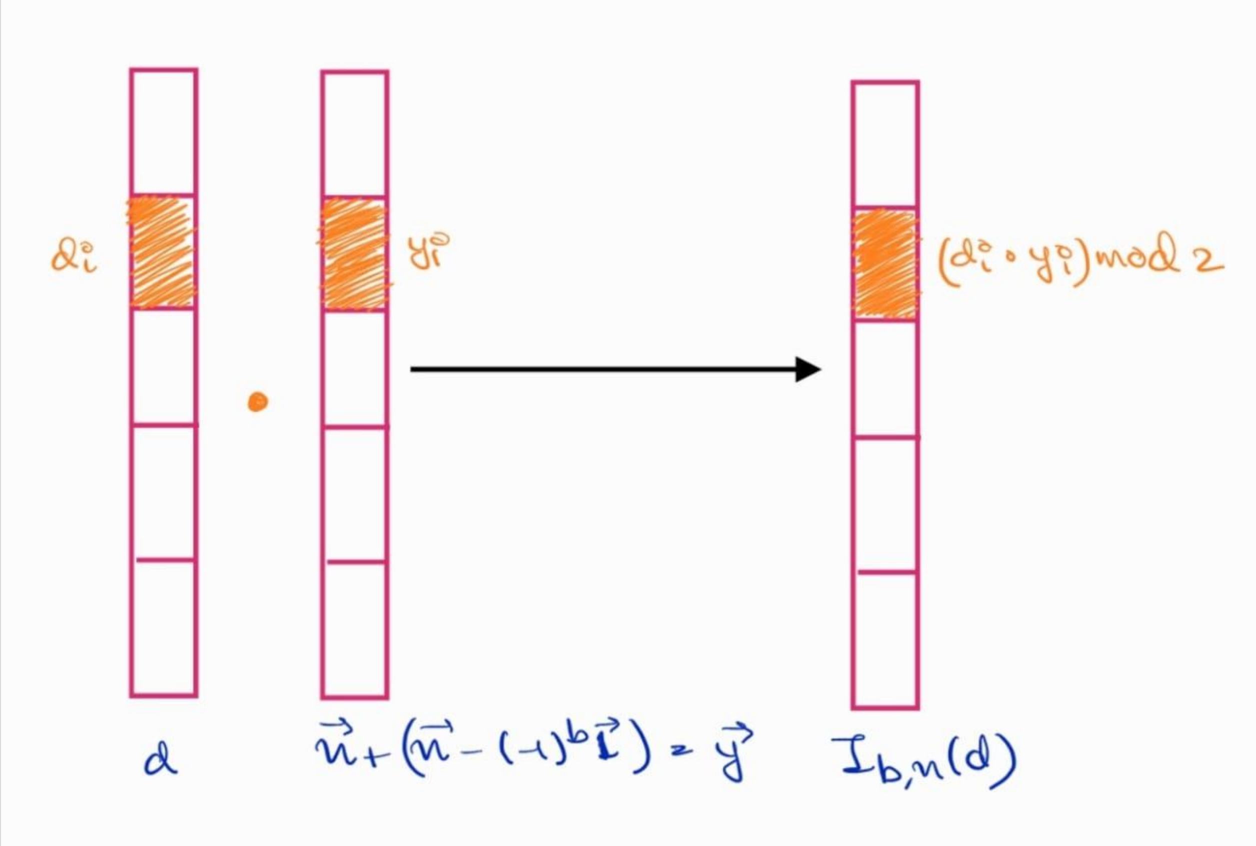
$$\left|\Pr_{(k,\mathsf{td})\leftarrow\mathsf{GEN}(1^\lambda)}[\mathcal{A}(k)\in H_s]-\Pr_{(k,\mathsf{td})\leftarrow\mathsf{GEN}(1^\lambda)}[\mathcal{A}(k)\in \bar{H}_s]\right|\leq \mathsf{negl}(\lambda)$$

where,

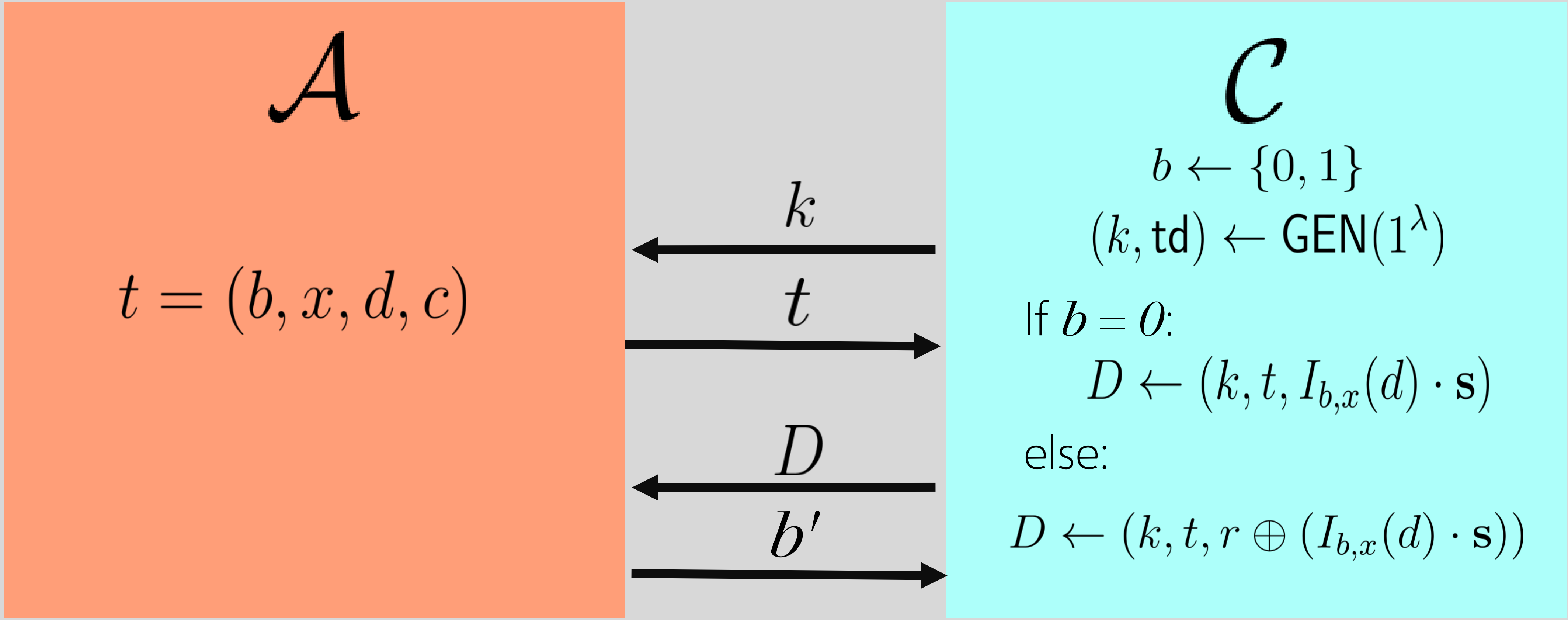
$$H_s=\{(b,x,d,d\cdot(x+(x-(-1)^b s)))\}$$
$$\bar{H}_s=\{(b,x,d,c\oplus 1|(b,x,d,c)\in H_s\}$$

Mapping $I_{b,x}(d)$

- Defined to be the inner product of d and $(x + (x - (-1)^b \mathbf{1}))$
- Each entry of x belongs to $\mathbb{Z}_q$. So first convert it into binary.



Adaptive Hardcore Bit – Security Game



Adaptive Hardcore Bit – Security Game

Claim : The **AHB security game** implies the former definition

Proof: Just trust me 🥴

Intuition: Observe $d \cdot (x + (x - (-1)^b)s) = I_{b,x}(d) \cdot s$

We know prove that any QPT adversary cannot have non-negligible advantage in our security game.

Moderate Matrix Lemma

Given a close to uniform matrix **C** (fixed) and a vector **Cs** the following holds with a very high probability:

$$(d \cdot s) \mod 2 \approx r, \quad \text{where } r \leftarrow_{\$} \{0,1\}$$

The distributions are statistically indistinguishable

Adaptive Hardcore Bit – Security Game

Thus, using the Moderate Matrix Lemma, we can directly say that the two distributions

$$D_0 = ((\mathbf{A}, \mathbf{A}s + e), (b, x, d, c) \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}s + e), I_{b,x}(d) \cdot s)$$

and

$$D_1 = ((\mathbf{A}, \mathbf{A}s + e), (b, x, d, c) \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}s + e), r \oplus (I_{b,x}(d) \cdot s))$$

are computationally indistinguishable.

Hardcore Bit 2

There exists a string d, such that for all Quantum poly-time adversaries $\mathcal{A}$,

$$\left| \mathsf{Pr}_{(k,\mathsf{td})\leftarrow \mathsf{GEN}(1^\lambda)}[\mathcal{A}(k) = b] - \frac{1}{2} \right| \leq \mathsf{negl}(1^\lambda)$$

where,

$$b = d \cdot (x_0 + x_1), \quad (x_0, x_1) \in \mathcal{R}_k$$

Hardcore Bit 2 – Alternative Version

For all strings d, for any QPT adversary $\mathcal{A}$, the distributions

$$D_0 = ((\mathbf{A}, \mathbf{A}s + e), d \cdot s) \quad \text{and}$$

$$D_1 = ((\mathbf{A}, \mathbf{A}s + e), r), \quad \text{where } r \leftarrow_{\$} \{0,1\},$$

are computationally indistinguishable.

- The above definition implies the former one. (for any choice of string d)
- The distributions D_0 and D_1 above are computationally indistinguishable using the Moderate Matrix Lemma.

Our Contributions

- We simplified the proof of Hardcore-Bit properties by slightly tweaking the Moderate Matrix Lemma.
- We attempted to construct exact TCFs.

THANK YOU