Pseudorandomness of Decoding, Revisited: Adapting OHCP to Code-Based Cryptography

Maxime Bombar (CWI, The Netherlands)*, Alain Couvreur (Inria, France), Thomas Debris-Alazard (Inria, France)

ASIACRYPT 2023, Guangzhou, China

* Work conducted while M.B. was at Ecole Polytechnique, and Inria, France

December 07, 2023

Source of Hardness of Code-based cryptography

- Topic of this talk: Hardness of Decisional Decoding Problem (Pseudorandomness).
- More than the result: Proof techniques
- Recent trend: Getting inspired by euclidean lattices.
- OHCP: Modern reduction technique for lattices ([PRS17, RSW18, BJRW20, PS21]).

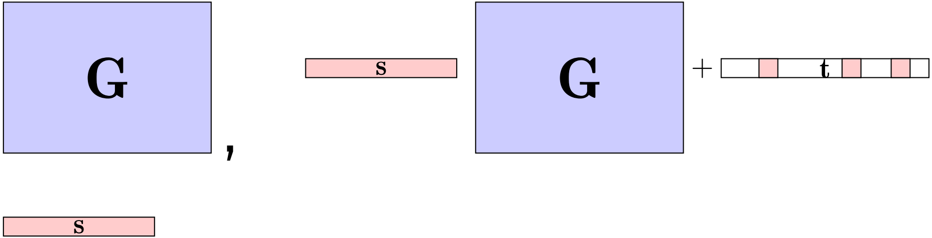
Results

- Adapt OHCP to the Hamming metric: Oracle with Hidden Support Problem (OHSP).
- Worst-case to Average-case, Search-to-Decision reduction (for non trivial parameters).
- An inch away reduction for structured codes.

Most of the talk generalise to arbitrary $\mathbb{F}_q$

Input:

Goal:



- Studied since (at least) the 1950s ✓
- Best algorithms[1] are exponential in $|\mathbf{t}|$ ✓

[1] For the rates we consider

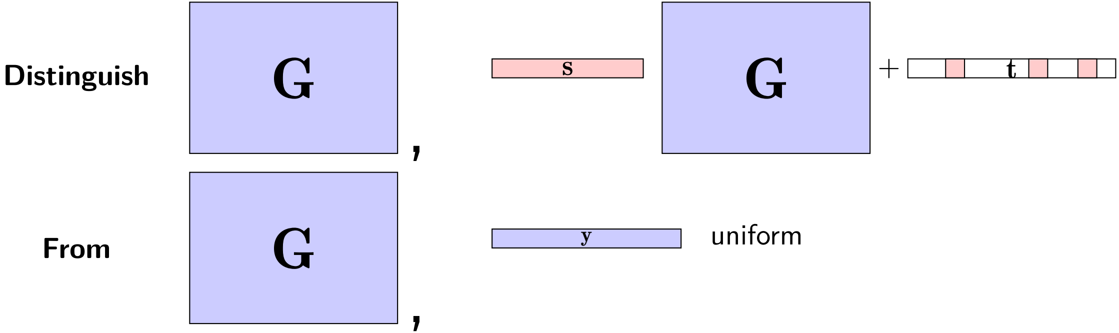
Learning Parity with Noise: LPN

Oracle $\mathsf{LPN}_{\mathbf{s}}(\omega)$: $\quad \mathbf{a}\ ,\quad \mathbf{s}\ \mathbf{a} + \mathsf{Ber}(\omega)$

LPN with N samples $\approx$ Average Decoding Problem for rate $\frac{k}{N}$

Worst-case to average-case reduction ([BLVW19, YZ21])

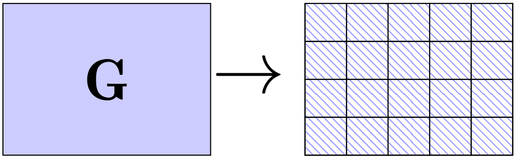
Decisional Version of the Decoding Problem



- Search-to-Decision reduction ([FS96])
- Fundamentally average-case to average-case

Adding Structure for Efficiency

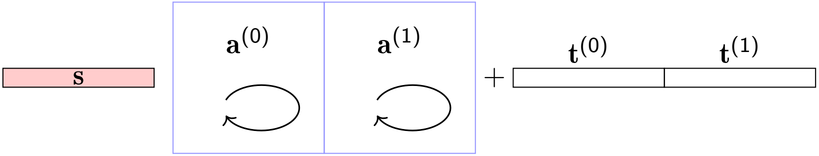
$\mathcal{R}$ ring, *e.g.* $\mathbb{F}_2[X]/(X^n - 1)$ Generalise to $\mathbb{F}_q[G]$ for an (abelian) group G



;

$$\begin{pmatrix} a_0 & a_1 & \dots & \dots & a_{n-1} \\ a_{n-1} & a_0 & \dots & \dots & a_{n-2} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_{n-1} & a_0 \end{pmatrix} \longleftrightarrow \mathbf{a} \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} a_i X^i \in \mathcal{R}$$

Example:



$$\longleftrightarrow \begin{cases} \mathbf{s}(X)\mathbf{a}^{(0)}(X) + \mathbf{t}^{(0)}(X) \in \mathcal{R} \\ \mathbf{s}(X)\mathbf{a}^{(1)}(X) + \mathbf{t}^{(1)}(X) \in \mathcal{R} \end{cases}$$

Ultimate Goal: Structured Variants

$\mathcal{R}$ ring, *e.g.* $\mathbb{F}_2[X]/(X^n - 1)$

Generalise to $\mathbb{F}_q[G]$ for an (abelian) group G

Search Version

Input. N samples of the form $(\mathbf{a}, \mathbf{s}\mathbf{a} + \mathbf{t})$ where $\mathbf{a} \leftarrow \mathcal{R}$, and $|\mathbf{t}| = t$.

Goal. Find $\mathbf{s}$.

Decision Version

Goal. Distinguish between $(\mathbf{a}, \mathbf{y}^{\text{unif}})$ and $(\mathbf{a}, \mathbf{s}\mathbf{a} + \mathbf{t})$, given N samples.

- BIKE and HQC (NIST 4th round).
- Used for some constructions in MPC ([BCGIKS20, **B**CCD23]).

Ultimate Goal: Structured Variants

$\mathcal{R}$ ring, *e.g.* $\mathbb{F}_2[X]/(X^n - 1)$

Generalise to $\mathbb{F}_q[G]$ for an (abelian) group G

Search Version

Input. N samples of the form $(\mathbf{a}, \mathbf{s}\mathbf{a} + \mathbf{t})$ where $\mathbf{a} \leftarrow \mathcal{R}$, and $|\mathbf{t}| = t$.

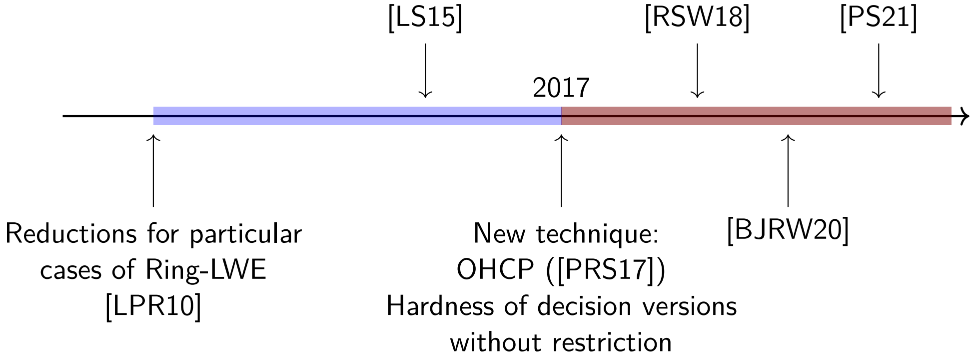
Goal. Find $\mathbf{s}$.

Decision Version

Goal. Distinguish between $(\mathbf{a}, \mathbf{y}^{\text{unif}})$ and $(\mathbf{a}, \mathbf{s}\mathbf{a} + \mathbf{t})$, given N samples.

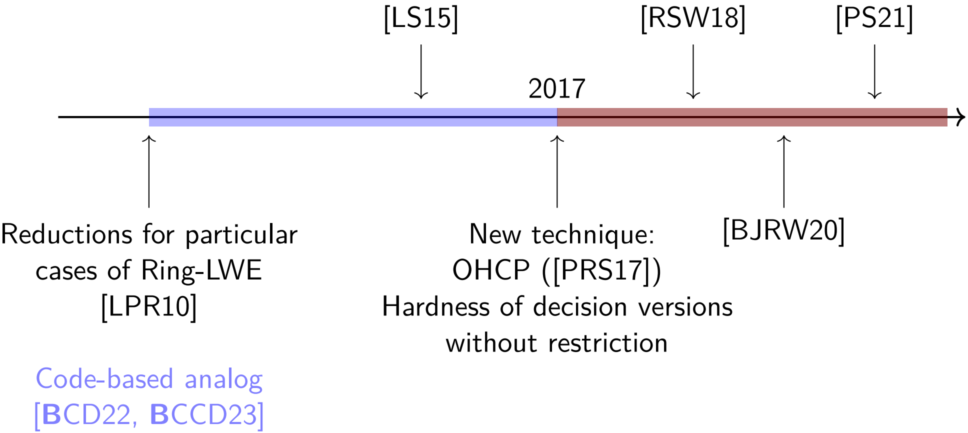
- Known search-to-decision reductions **do not** carry over ✗
- Very few specific reductions ([**B**CD22, **B**CCD23]).

Lattices: a History of Search-to-Decision Reductions[2]



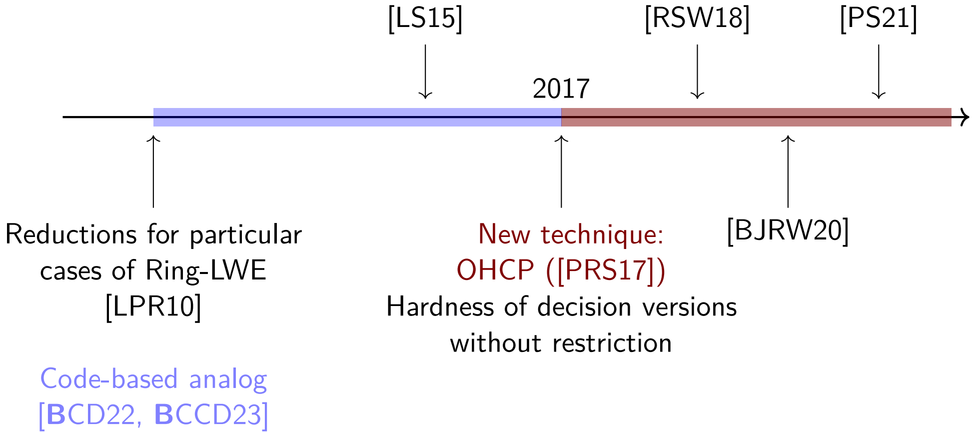
[2]Not exhaustive

Lattices: a History of Search-to-Decision Reductions[2]



[2] Not exhaustive

Lattices: a History of Search-to-Decision Reductions[2]



[2] Not exhaustive

Worst-case to Average-case Reduction

Adapting OHCP for plain Decoding Problem ✓

Decisional Decoding Problem with

$$\frac{k}{n} = \frac{1}{n^D} \qquad \text{and} \qquad \frac{t}{n} = \frac{1}{2}\left(1 - \frac{1}{n^{D(1+o(1))}}\right) \qquad \text{for some } D < 1/2,$$

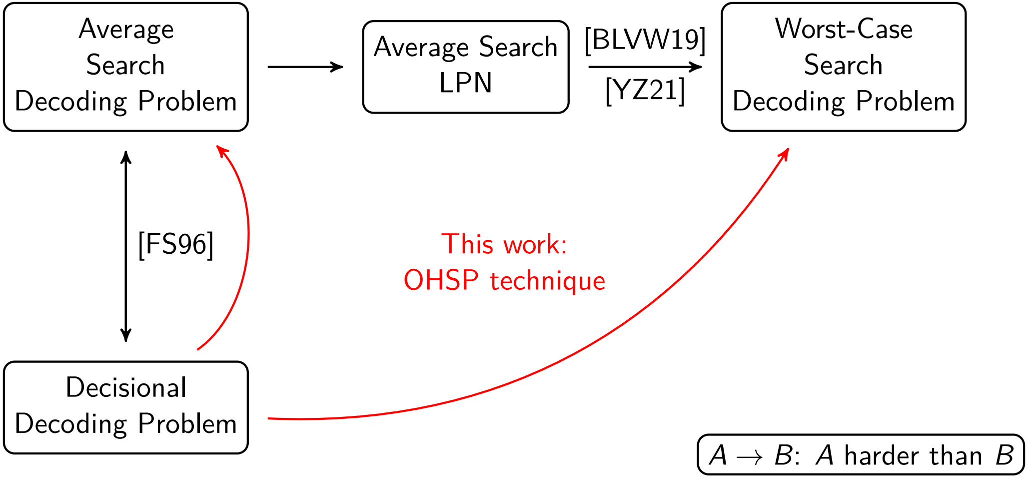
is harder than **worst-case** Decoding Problem with

$$\frac{k}{n} = \frac{1}{n^D} \qquad \text{and} \qquad \frac{t}{n} = \frac{\log^2(n)}{n^{1-D}}$$

(superpolynomial hardness, $\approx$ same parameters as [BLVW19])

Bypass earlier search-to-decision reductions

This work: Oracle with Hidden Support Problem (OHSP)

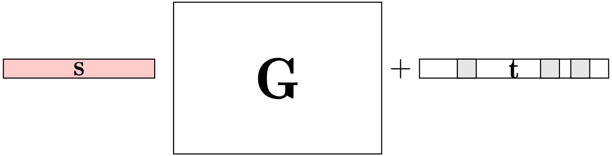


A non-positive result

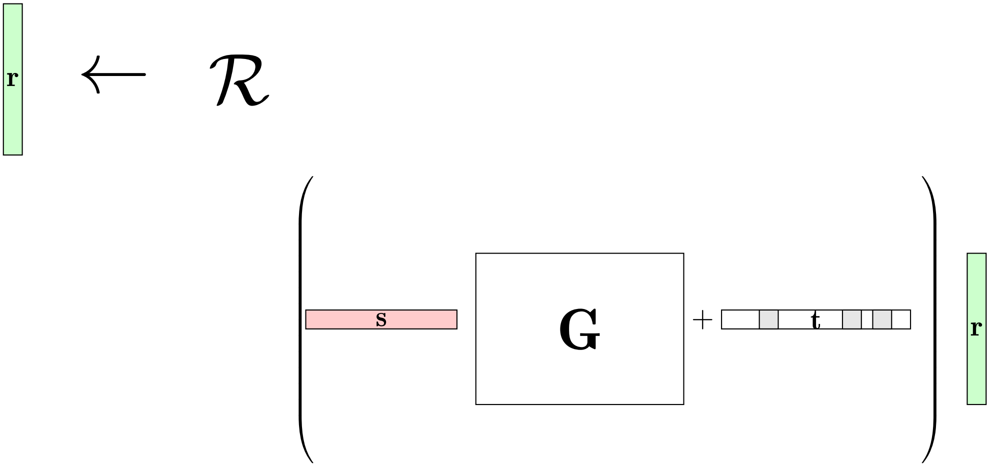
Structured variants

- Not as straightforward as for lattices ✗
- But only fails at the very end
- Identify a single point of failure which might be overcome in the future

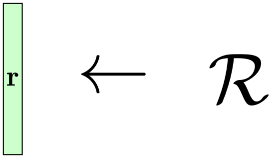
From Decoding to LPN [BLVW19, YZ21]

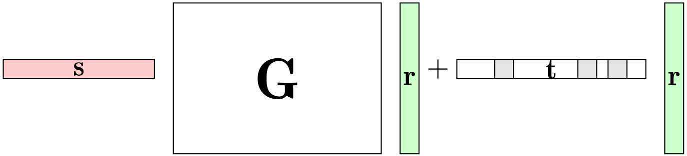


From Decoding to LPN [BLVW19, YZ21]

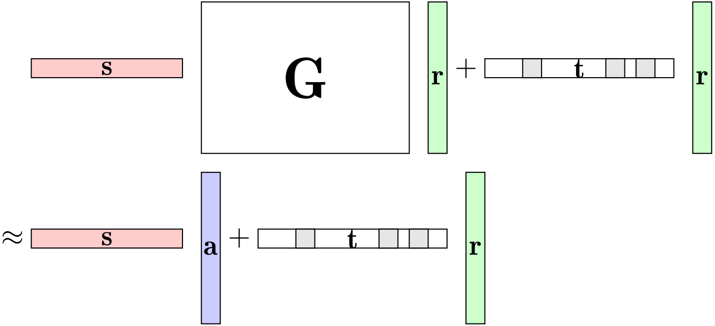


From Decoding to LPN [BLVW19, YZ21]



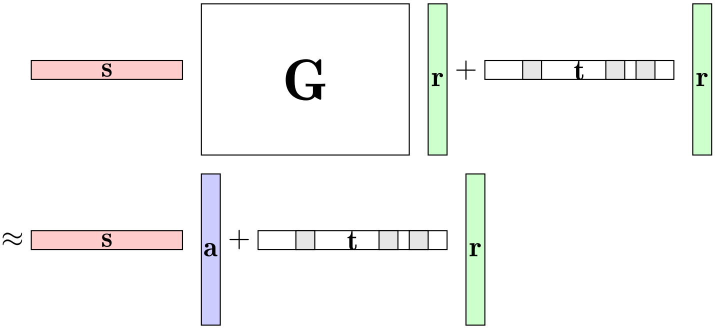


Building LPN-like Oracle



- $\mathbf{G}\mathbf{r} \approx^{?}$ uniform
- $(\mathbf{G}\mathbf{r}, \mathbf{t} \cdot \mathbf{r})$ are correlated ...

Building LPN-like Oracle



Statistically close

→ **Average-case:** Leftover hash lemma

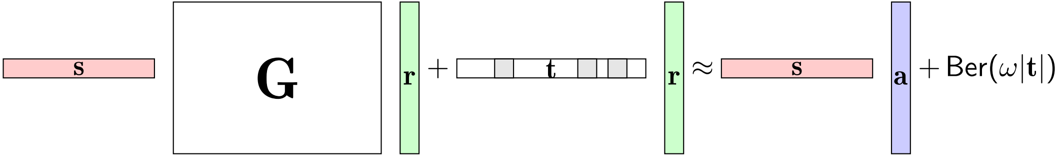
→ **Worst-case:** Notion of smoothing distribution ([BLVW19, YZ21, DDRT23, DR23])

Bernoulli Smoothing

(Non Standard) Notation

$$r_i \leftarrow \mathrm{Ber}(\omega) \text{ if } r_i \text{ Bernoulli with } \mathbb{P}(r_i = 1) = \frac{1}{2}(1 - 2^{-\omega}).$$

Remark: $\mathrm{Ber}(\omega_1) + \mathrm{Ber}(\omega_2) = \mathrm{Ber}(\omega_1 + \omega_2)$.



Smoothing bounds from [DR23]

A continuous hybrid argument

- $(\mathbf{G}, \mathbf{y} \stackrel{\text{def}}{=} \mathbf{sG} + \mathbf{t})$
- Distinguisher $\mathscr{A}$ between LPN(ω_0) and LPN(∞).
- $\mathscr{A}$ makes N queries to the oracle and has advantage ε.

We build LPN($\omega|\mathbf{t}|$) oracle.

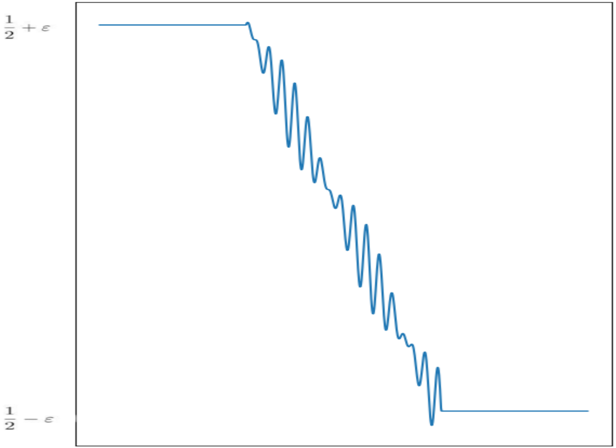
- $\mathscr{A}$ can be given any LPN(ω)-like oracle.
- Will accept with some probability $p(\omega)$.

- $p(\omega_0) = \frac{1}{2} + \varepsilon$
- $p(\omega) \to \frac{1}{2} - \varepsilon$ as $\omega \to \infty$

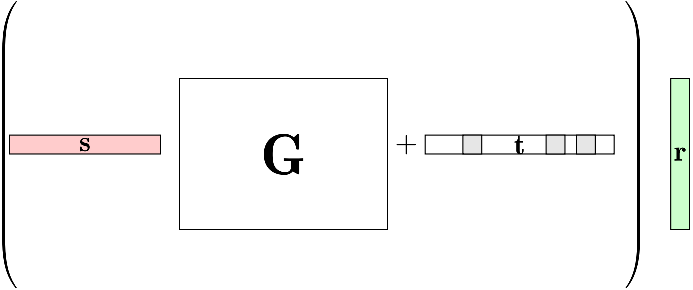
- $p(\omega)$ unknown for $\omega \in (\omega_0, \infty)$
- But can be estimated via statistical methods.

Acceptance behaviour of $\mathscr{A}^{\mathsf{LPN}(\omega)}$ **must** change as $\omega \to \infty$.

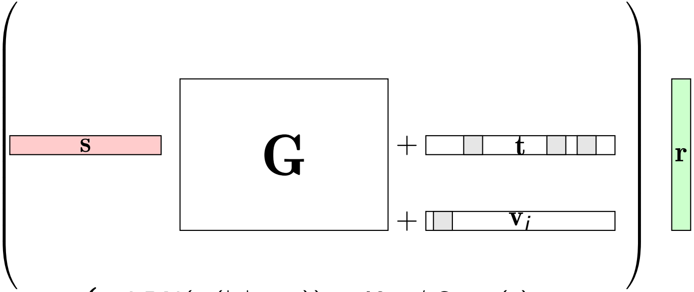
Estimating $p(\omega)$



Wishful thinking: Testing Support Membership

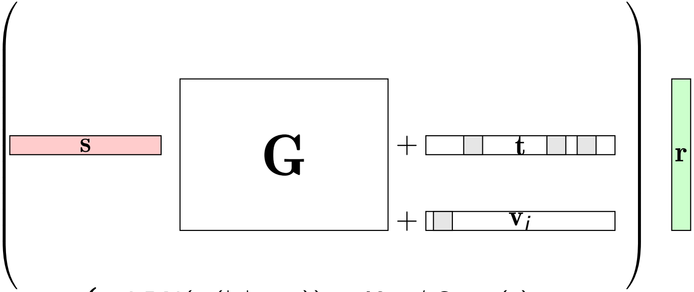


Wishful thinking: Testing Support Membership



$$\approx \begin{cases} \mathsf{LPN}(\omega(|\mathbf{t}|+1)) & \text{If } i \notin \mathsf{Supp}(\mathbf{t}) \\ \mathsf{LPN}(\omega(|\mathbf{t}|-1)) \text{ or } \mathsf{LPN}(\omega|\mathbf{t}|) & \text{Else} \end{cases}$$

Wishful thinking: Testing Support Membership



$$\approx \begin{cases} \mathsf{LPN}(\omega(|\mathbf{t}|+1)) & \text{If } i \notin \mathsf{Supp}(\mathbf{t}) \\ \mathsf{LPN}(\omega(|\mathbf{t}|-1)) \\ \text{ or } \mathsf{LPN}(\omega|\mathbf{t}|) & \text{Else} \end{cases}$$

Not so easy to distinguish those two situations...

Shift your oracles

Idea: *Zoom in* and sample $\mathbf{r} \leftarrow \mathsf{Ber}^{\otimes n}(2^x \omega_0)$.

$$\mathcal{O}_0(x) \approx \mathsf{LPN}(2^x \omega_0 |\mathbf{t}|) \quad \text{and} \quad \mathcal{O}_{\mathbf{v}_i}(x) \approx \mathsf{LPN}(2^x \omega_0 |\mathbf{t} + \mathbf{v}_i|).$$

Define $p(x) \stackrel{\text{def}}{=} \mathbb{P}(\mathcal{A}^{\mathcal{O}_0(x)} \text{ accepts})$.

$$\mathbb{P}(\mathcal{A}^{\mathcal{O}_{\mathbf{v}_i}(x)} \text{accepts}) = p\left(x + \log \frac{|\mathbf{t} + \mathbf{v}_i|}{|\mathbf{t}|}\right)$$

where

$$\log \frac{|\mathbf{t} + \mathbf{v}_i|}{|\mathbf{t}|} = \begin{cases} \log(1 + \frac{1}{t}) > 0 & \text{if } i \notin \mathsf{Supp}(\mathbf{t}) \\ \leqslant 0 & \text{if } i \in \mathsf{Supp}(\mathbf{t}). \end{cases}$$

Shift your oracles (Cont'd)

Change of behaviour in $\mathbb{P}(\mathcal{A}^{\mathcal{O}_0(x)}$ accepts) should happen at some point x_0.

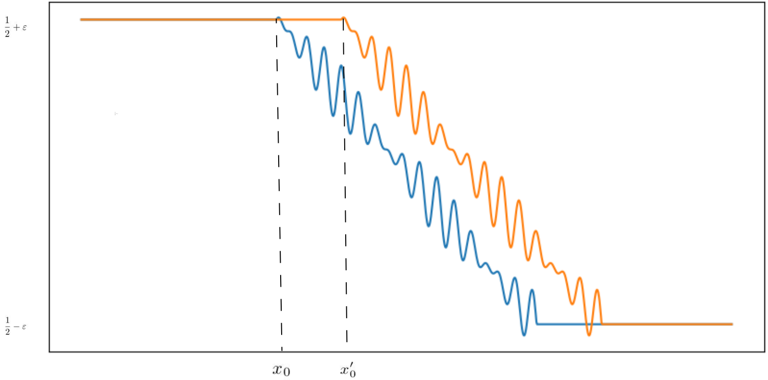
If $i \notin \mathrm{Supp}(\mathbf{t})$, behaviour of $\mathbb{P}(\mathcal{A}^{\mathcal{O}_{\mathbf{v}_i}(x)}$accepts) changes at some x_0' such that

$$x_0' = x_0 + \log\left(1 + \frac{1}{t}\right) \approx x_0 + \frac{1}{t}.$$

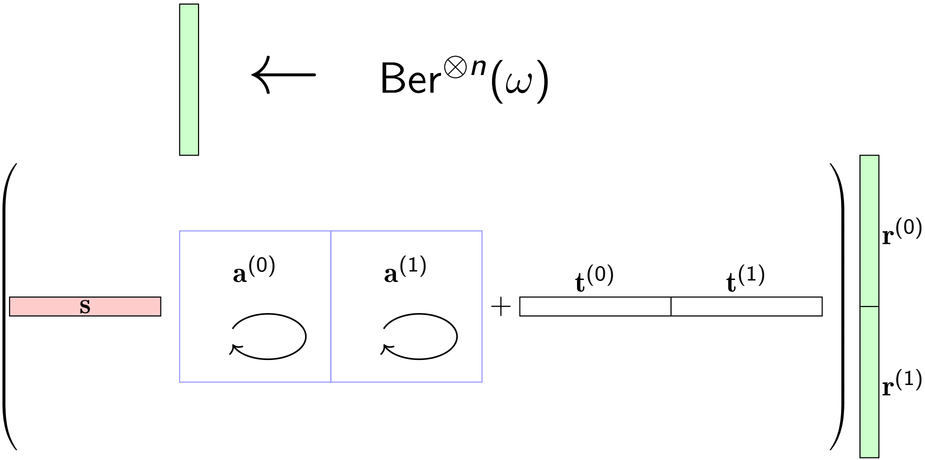
Oracle Comparison Problem from [PRS17]

p is very constrained (Lipschitz etc...) $\Rightarrow$ This can actually be detected in **polynomial time**!

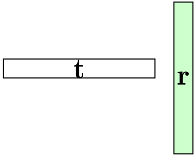
Shifted hybrid argument



What about Structured Variants ?

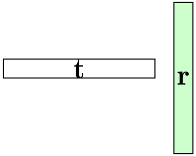


What about Structured Variants ?



$$\longleftrightarrow \sum_k \underbrace{\left(\sum_{i+j\equiv k \mod n} t_i r_j\right)}_{\sim \mathrm{Ber}(\omega|\mathbf{t}|)} X^k$$

What about Structured Variants ?



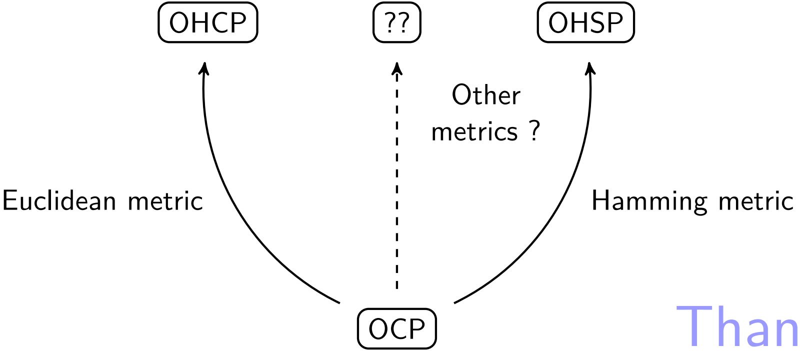
$$\longleftrightarrow \sum_k \underbrace{\left(\sum_{i+j \equiv k \mod n} t_i r_j \right)}_{\sim \mathrm{Ber}(\omega|\mathbf{t}|)} X^k$$

NOT independent ...

Open questions

→ How to make the reduction work in the structured case ?

→ Find better smoothing bounds to improve the reduction ?



Thanks! eprint.iacr.org/2022/1751