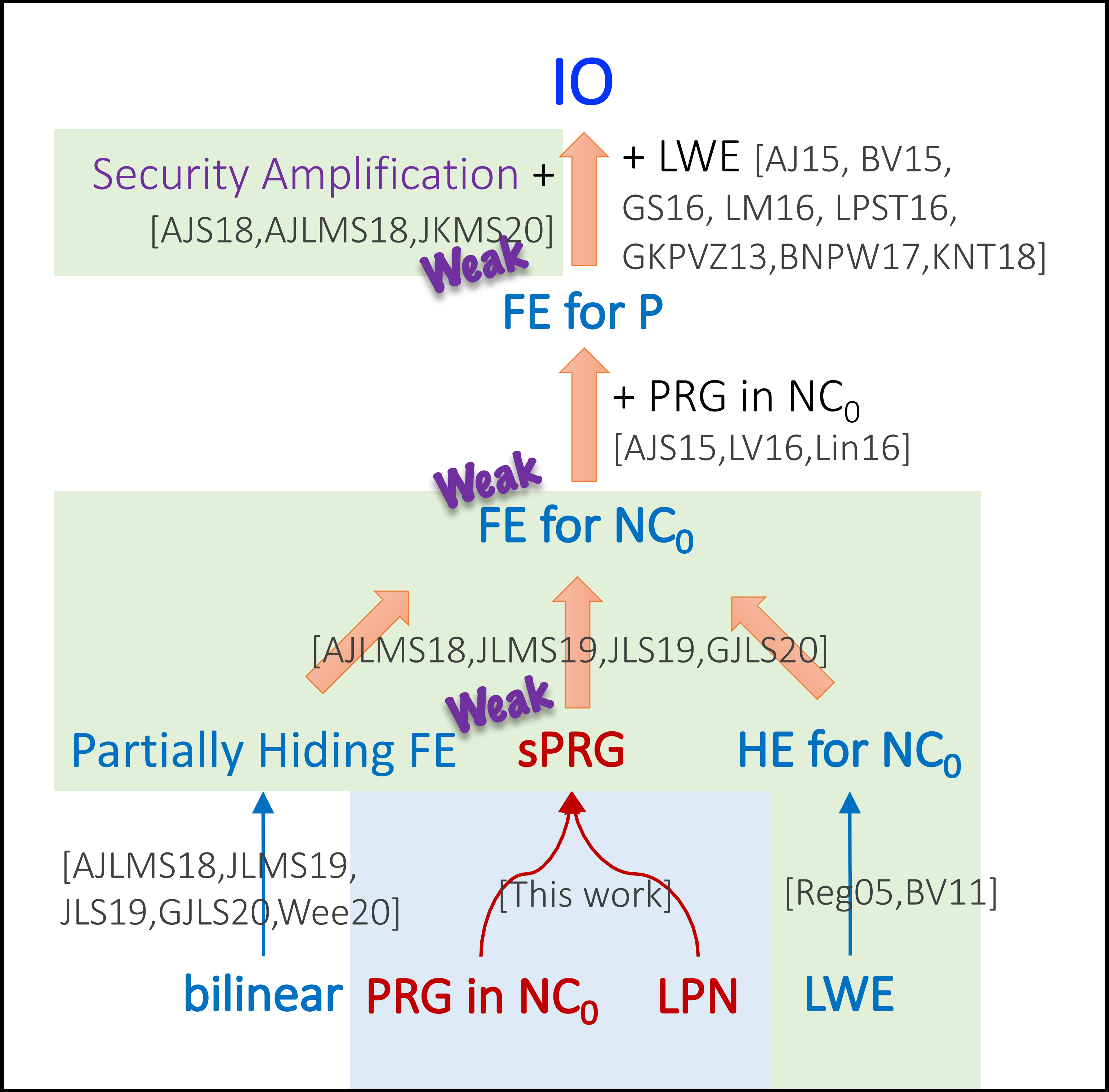
$i\mathcal{O}$ from Well-Founded Assumptions

Aayush Jain
UCLA

Huijia Lin
UW

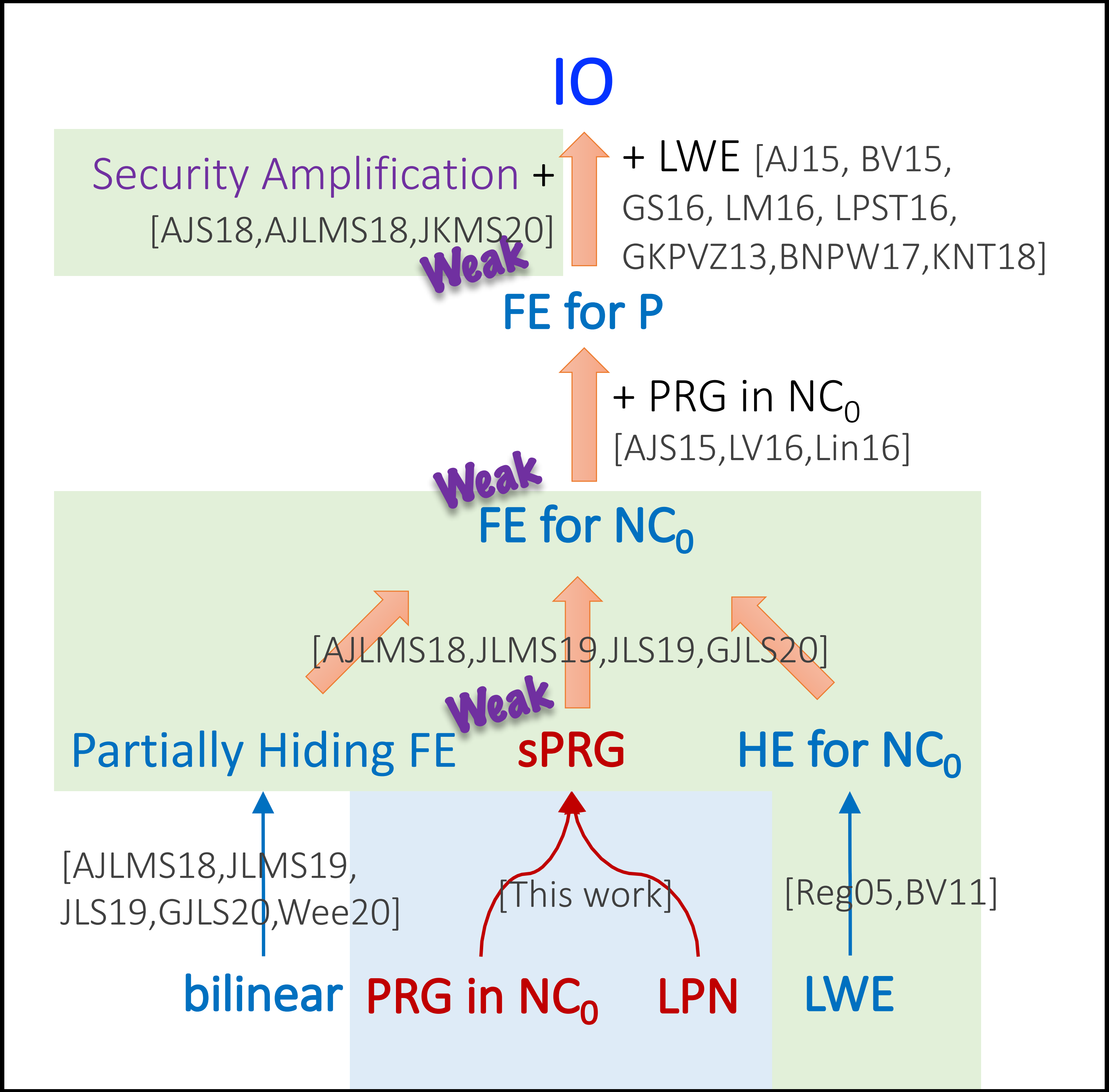
Amit Sahai
UCLA

From Rachel's Talk



Credits: Rachel Lin

From Rachel's Talk



This Talk: sPRG from

1. **LPN over $\mathbb{Z}_p$**
2. **PRGs in NC^0**

Credits: Rachel Lin

Learning Parity with Noise

[BFKL 93, IPS 09]

$$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i \mod p\}_{i\in[n]}$$

$$\approx_c$$

$$\{\overrightarrow{a}_i, u_i\}_{i\in[n]}$$

$n = \text{poly}(\ell)$

$\overrightarrow{a}_i, \vec{s} \leftarrow \mathbb{Z}_p^\ell$

$$e_i := \begin{cases} e_i \leftarrow \mathbb{Z}_p & \texttt{Pr.}\ \ell^{-\delta} \\ e_i = 0 & \texttt{Pr.}\ 1 - \ell^{-\delta} \end{cases}$$

$\delta \in (0,1)$

$\overrightarrow{u} \leftarrow \mathbb{Z}_p^{n\times 1}$

Search $\equiv$ Decision

[MP13, BFKL 93, Reg 05]

Within sub-exp factors.

Best Known Attack: $O(2^{\ell^{1-\delta}})$ **[EKM 17]**

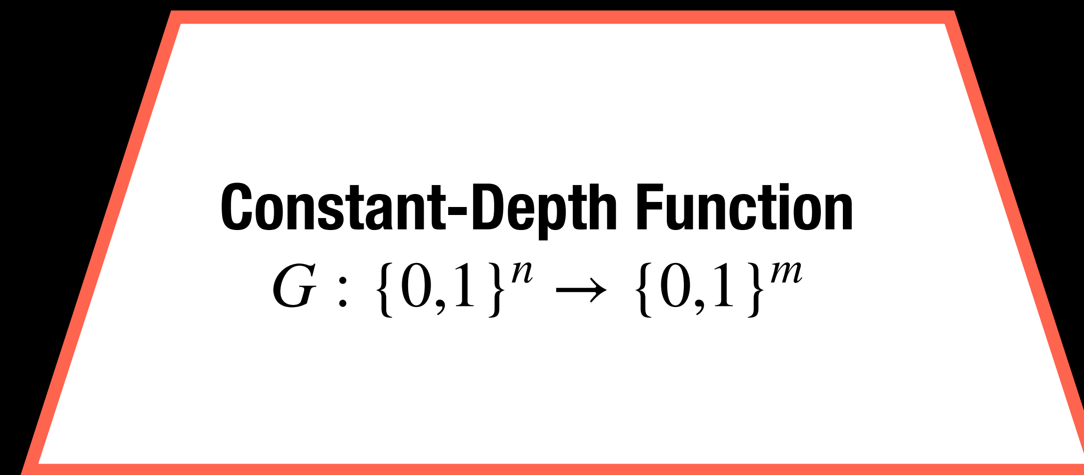
We use $\delta > 0$ arbitrarily small constant.

PRGs in NC^0

Computable by:

Constant-depth circuits.

Input: $\overrightarrow{x} \in \{0,1\}^n$

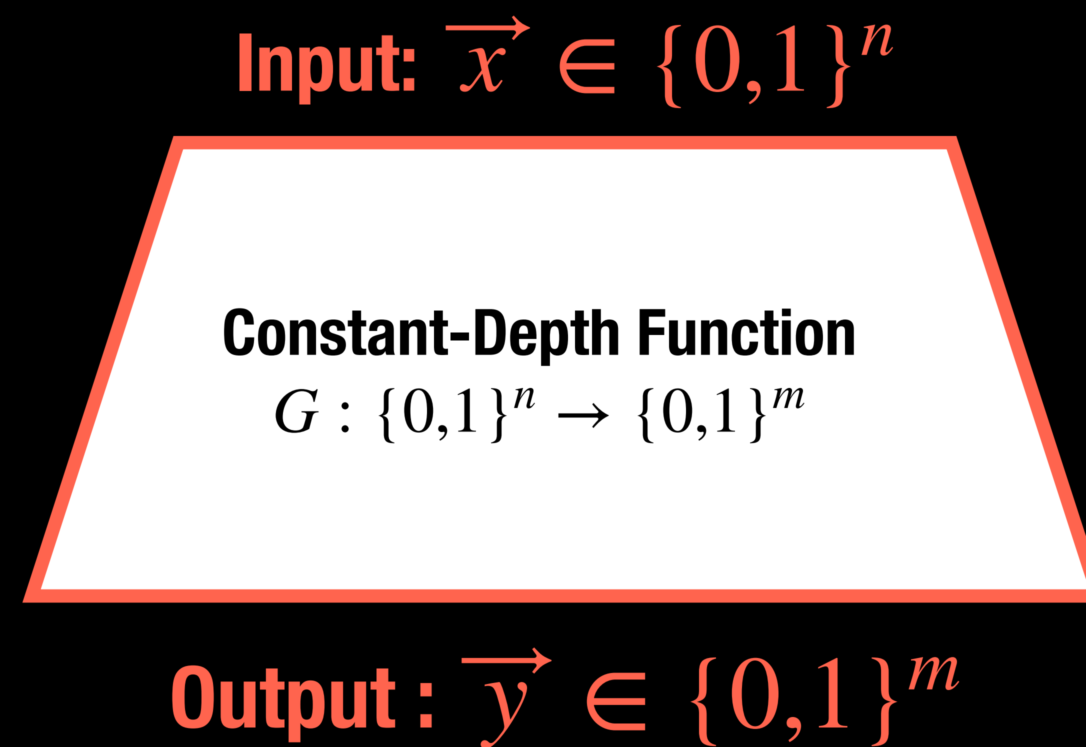


Output : $\overrightarrow{y} \in \{0,1\}^m$

PRGs in NC^0

Computable by:
Constant-depth circuits.

Stretch: $m \geq n^{1+\Omega(1)}$



PRGs in NC^0

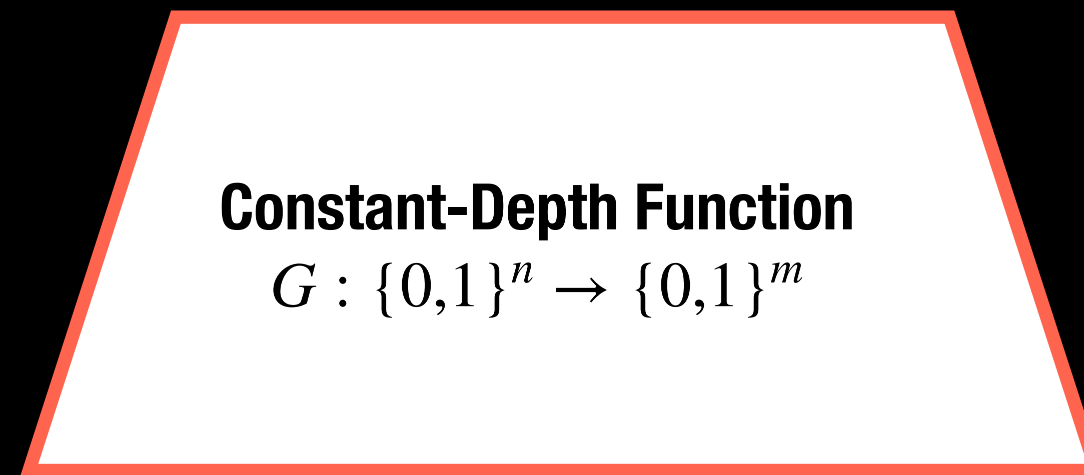
Computable by:
Constant-depth circuits.

Stretch: $m \geq n^{1+\Omega(1)}$

Security: **Let** $\overrightarrow{x} \leftarrow \{0,1\}^n$
and $\vec{r} \leftarrow \{0,1\}^m$

$\{G(\overrightarrow{x})\} \approx_c \{\vec{r}\}$

Input: $\overrightarrow{x} \in \{0,1\}^n$



Output : $\overrightarrow{y} \in \{0,1\}^m$

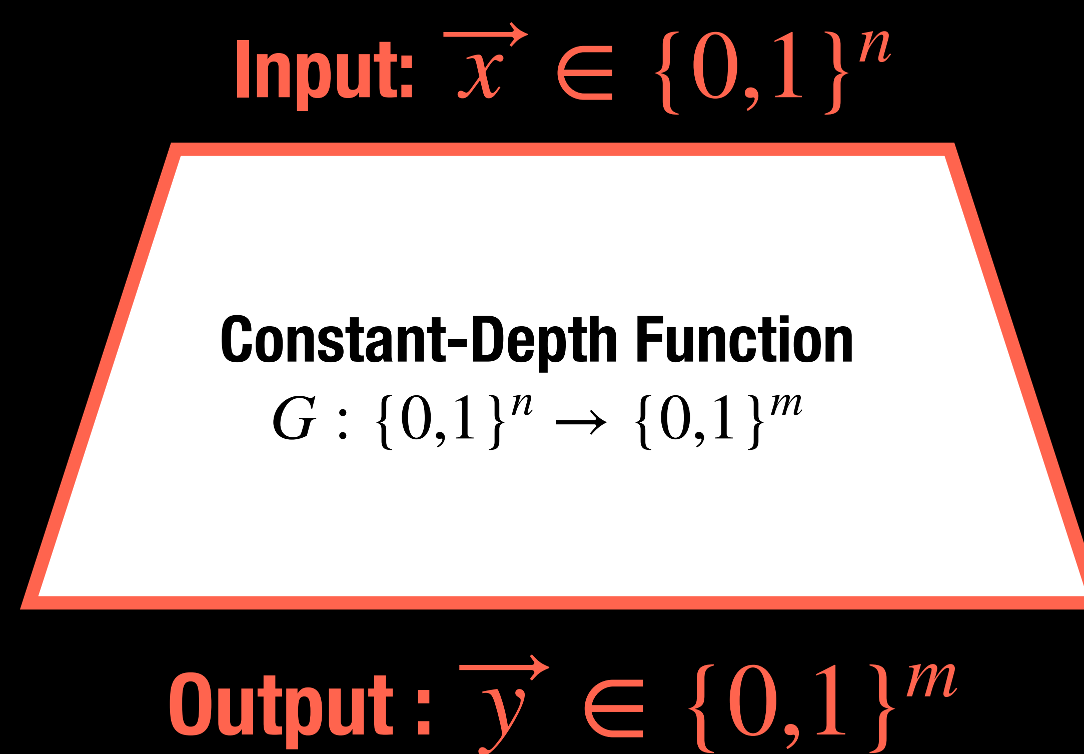
PRGs in NC^0

Computable by:
Constant-depth circuits.

Stretch: $m \geq n^{1+\Omega(1)}$

Security: **Let** $\overrightarrow{x} \leftarrow \{0,1\}^n$
and $\vec{r} \leftarrow \{0,1\}^m$

$$\{G(\overrightarrow{x})\} \approx_c \{\vec{r}\}$$



Extensively studied [Gol 00, CM 01, MST 03, IKOS 08, ABR 12, BQ 12, App 12, KMOW 17, CDM+18….].

PRGs in NC^0

Computable by:
Constant-depth circuits.

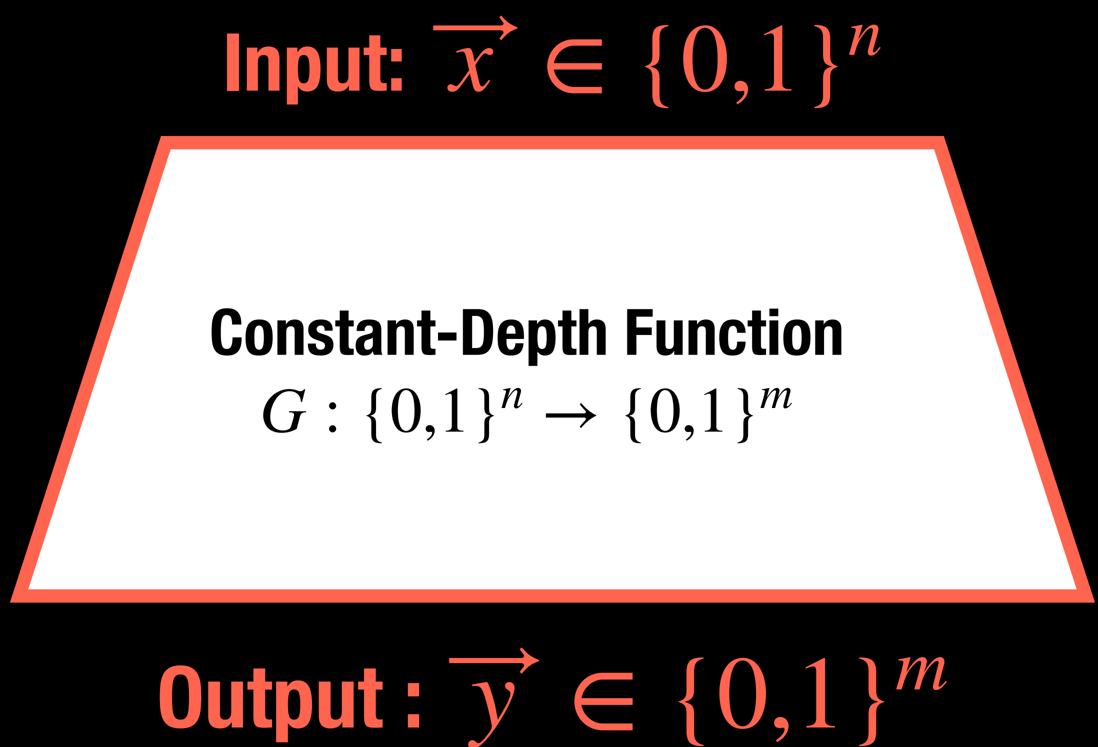
Stretch: $m \geq n^{1+\Omega(1)}$

Security: **Let** $\overrightarrow{x} \leftarrow \{0,1\}^n$
and $\vec{r} \leftarrow \{0,1\}^m$

$$\{G(\overrightarrow{x})\} \approx_c \{\vec{r}\}$$

Two Facts:

- Locality is constant.
- Can be written as constant degree multivariate polynomial over $\mathbb{R}$.



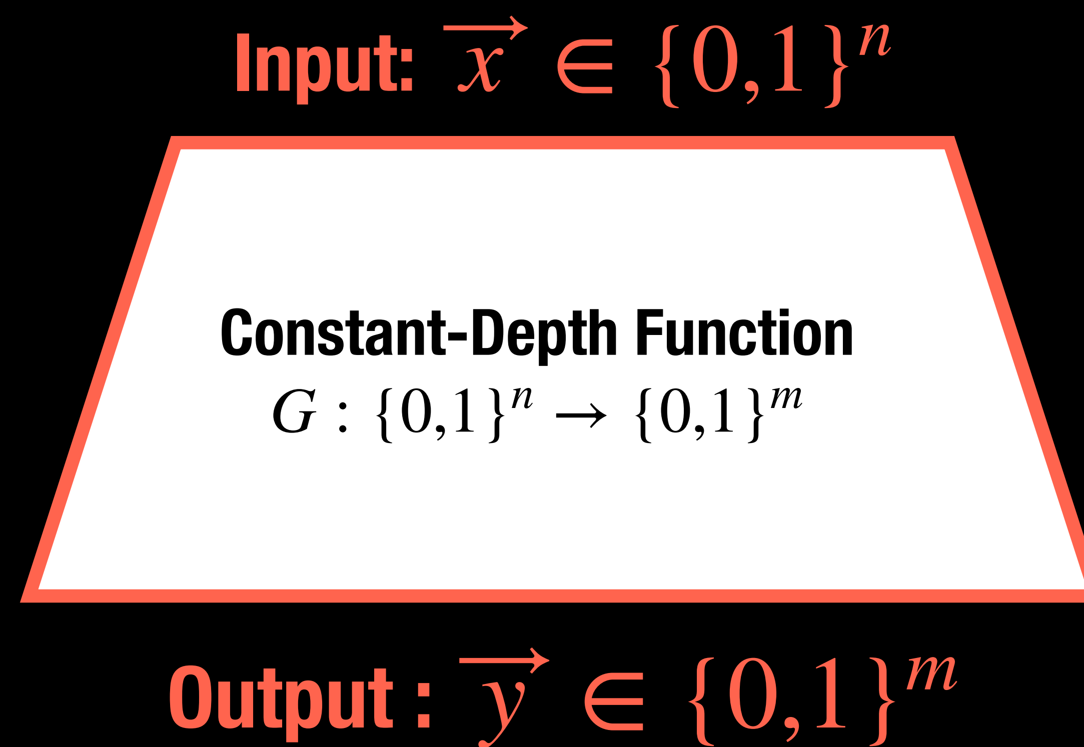
PRGs in NC^0

Computable by:
Constant-depth circuits.

Stretch: $m \geq n^{1+\Omega(1)}$

Security: **Let** $\overrightarrow{x} \leftarrow \{0,1\}^n$
and $\vec{r} \leftarrow \{0,1\}^m$

$$\{G(\overrightarrow{x})\} \approx_c \{\vec{r}\}$$



Two Facts:

- **Locality is constant.**
- **Can be written as constant degree multivariate polynomial over** $\mathbb{R}$.

We need stretch of $n^{1+\tau}$ **for arbitrarily small constant** $\tau > 0$.

What's sPRG?

Computable using Bilinear Maps- Bilinear Maps Friendly.

Implicit in [AJS 18, AJLMS 19, JLMS 19, GJLS 20]:

PRGs implementable as follows.

What's sPRG?

Computable using Bilinear Maps- Bilinear Maps Friendly.

Implicit in [AJS 18, AJLMS 19, JLMS 19, GJLS 20]:
PRGs implementable as follows.

Structured Seed

Public Seed $P \in \mathbb{Z}_p^k$

Secret Seed $S \in \mathbb{Z}_p^k$

Output:

$$\overrightarrow{y} \in \{0,1\}^m$$

$$m \gg |P| + |S|$$

$$y_i = \sum_{j,k} f_{j,k}(P) \cdot S_j \cdot S_k \mod p$$

Any constant degree

p **is the order of bilinear group.**

What's sPRG?

Structured Seed

Public Seed
$P \in \mathbb{Z}_p^k$

Secret Seed
$S \in \mathbb{Z}_p^k$

Output:

$\overrightarrow{y} \in \{0,1\}^m$

$m \gg |P| + |S|$

$$y_i = \sum_{j,k} f_{j,k}(P) \cdot S_j \cdot S_k \mod p$$

Any constant degree

Security: $(P, \overrightarrow{y}) \approx_c (P, \vec{r} \leftarrow \{0,1\}^m)$

What's sPRG?

Structured Seed

Public Seed
$P \in \mathbb{Z}_p^k$

Secret Seed
$S \in \mathbb{Z}_p^k$

Output:

$\overrightarrow{y} \in \{0,1\}^m$

$m \gg |P| + |S|$

$$y_i = \sum_{j,k} f_{j,k}(P) \cdot S_j \cdot S_k \mod p$$

Any constant degree

Is a public-seed (leakage) necessary?
Does degree-2 in secret seed suffice?

Previous degree-2 PRGs (without public seed) attacked [LT 17, AJS 18a, Agr 18, LM 18a]

Is Public Seed Necessary?

Degree-2 PRGs [LT 17, AJS 18, Agr 18, LM 18]:
Implaussible due to [BBKK 18, LV 18, BHJKS 19]

Is Public Seed Necessary?

Degree-2 PRGs [LT 17, AJS 18, Agr 18, LM 18]:
Implaussible due to [BBKK 18, LV 18, BHJKS 19]

Major Culprit: Sum-of-Squares Hierarchy [Lassere, Parillo]

Is Public Seed Necessary?

Degree-2 PRGs [LT 17, AJS 18, Agr 18, LM 18]:
Implaussible due to [BBKK 18, LV 18, BHJKS 19]

Major Culprit: Sum-of-Squares Hierarchy [Lassere, Parillo]

Takeaway: Avoid Degree-2 computation over $\mathbb{R}$.

Is Public Seed Necessary?

Degree-2 PRGs [LT 17, AJS 18, Agr 18, LM 18]:
Implaussible due to [BBKK 18, LV 18, BHJKS 19]

Major Culprit: Sum-of-Squares Hierarchy [Lassere, Parillo]

Takeaway: Avoid Degree-2 computation over $\mathbb{R}$.

Thank you, Sum-of-Squares!

sPRG Overview

Desired Input:

$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$

Desired Output:

$G(\overrightarrow{\sigma}) \in \{0,1\}^m$

PRG in NC^0

$$m = n^{1+\tau} \gg n$$

sPRG Overview

Desired Input:

$$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$$

Desired Output:

$$G(\overrightarrow{\sigma}) \in \{0,1\}^m$$

PRG in NC^0

Problem: This is not a degree-2 computation.

sPRG Overview

Desired Input:

$$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$$

Desired Output:

$$G(\overrightarrow{\sigma}) \in \{0,1\}^m$$

PRG in NC^0

Public Seed P**:**

$$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$$

sPRG Overview

Desired Input:

$$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$$

Desired Output:

$G(\overrightarrow{\sigma}) \in \{0$

PRG in

LPN Error

Public Se

$$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$$

Sampling Details:

$$\overrightarrow{a}_i, \vec{s} \leftarrow \mathbb{Z}_p^{\ell}$$

$$\Pr[e_i \neq 0] = \ell^{-\delta}$$

sPRG Overview

Desired Input:

$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$

Private Seed S:

$\mathsf{PreProc}(\vec{s}, \overrightarrow{e})$

Desired Output:

$G(\overrightarrow{\sigma}) \in \{0,1\}^m$

PRG in NC^0

Public Seed P:

$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$

sPRG Overview

Desired Input:

$$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$$

Private Seed S:

$\text{PreProc}(\vec{s}, \overrightarrow{e})$

Desired Output:

$$G(\overrightarrow{\sigma}) \in \{0,1\}^m$$

PRG in NC^0

Public Seed P:

$$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$$

$$|P| + |S| \ll m$$

sPRG Overview

Desired Input:

$$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$$

Private Seed S:

$\text{PreProc}(\vec{s}, \overrightarrow{e})$

Desired Output:

$$G(\overrightarrow{\sigma}) \in \{0,1\}^m$$

PRG in NC^0

Public Seed P:

$$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$$

$$|P| + |S| \ll m$$

$$G_i(\overrightarrow{\sigma}) = \sum_{j,k} f_{i,j,k}(P) S_j S_k \mod p$$

sPRG Overview

Desired Input:

$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$

Private Seed S:

PreProc($\vec{s}$, $\overrightarrow{e}$)

Desired Output:

$G(\overrightarrow{\sigma}) \in \{0,1\}^m$

PRG in NC^0

Public Seed P:

$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$

Known to Attacker

sPRG Overview

Desired Input:

$\overrightarrow{\sigma} \leftarrow \{0,1\}^n$

Private Seed S:

PreProc$(\vec{s}, \overrightarrow{e})$

Desired Output:

$G(\overrightarrow{\sigma}) \in \{0,1\}^m$

PRG in NC^0

Public Seed P:

$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$

Known to Attacker

Missing Piece:

- **How does S look like?**
- **How to Evaluate it?**

sPRG Construction Details

How to Construct sPRG?

sPRG Desiderata:

Index I	Public Seed P	Private Seed S
$\{\vec{a}_i\}_{i\in[n]}$	$\{b_i = \langle \vec{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	**?**

Goal: Find S such that,

Computation: $\vec{y} = G(\sigma)$

$$y_i = \sum_{j,k} f_{i,j,k}(P) \cdot S_j \cdot S_k \mod p$$

Size: $|P| + |S| \ll m$

Sampling Details

$\{\vec{a}_i\}_{i\in[n]}, \vec{s} \leftarrow \mathbb{Z}_p^\ell$

$e_i \leftarrow \mathrm{Ber}(\ell^{-\delta}) \cdot \mathbb{Z}_p$

$\sigma \leftarrow \{0,1\}^n$

G is a degree-d PRG with stretch $m = n^{1+\tau}$

Key Intuition: Sparsity Helps

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\mid 1)^{\otimes \lceil \frac{d}{2} \rceil}$

Key Intuition: Sparsity Helps

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s} \,|\, 1)^{\otimes\lceil \frac{d}{2}\rceil}$

Stretch Calculation

Set ℓ such that $\ell^{\lceil \frac{d}{2}\rceil} = n$

$\implies |S| = n\log_2 p$

Key Intuition: Sparsity Helps

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes\lceil\frac{d}{2}\rceil}$

Computation:

$\overrightarrow{y}' = G(b_1 - \langle \overrightarrow{a}_1, \vec{s}\rangle, \ldots, b_n - \langle \overrightarrow{a}_n, \vec{s}\rangle) \mod p$

Stretch Calculation

Set ℓ such that $\ell^{\lceil\frac{d}{2}\rceil} = n$

$\implies |S| = n\log_2 p$

Key Intuition: Sparsity Helps

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes \lceil \frac{d}{2} \rceil}$

Computation:

$$\overrightarrow{y}' = G(b_1 - \langle \overrightarrow{a}_1, \vec{s}\rangle, \ldots, b_n - \langle \overrightarrow{a}_n, \vec{s}\rangle) \mod p$$

$$= G(\sigma_1 + e_1, \ldots, \sigma_n + e_n) \mod p$$

Stretch Calculation

Set ℓ such that $\ell^{\lceil \frac{d}{2} \rceil} = n$

$\implies |S| = n \log_2 p$

Key Intuition: Sparsity Helps

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes\lceil \frac{d}{2}\rceil}$

Computation:

$$\overrightarrow{y}' = G(b_1 - \langle \overrightarrow{a}_1, \vec{s}\rangle, \ldots, b_n - \langle \overrightarrow{a}_n, \vec{s}\rangle) \mod p$$

$$= G(\sigma_1 + e_1, \ldots, \sigma_n + e_n) \mod p$$

Stretch Calculation

Set ℓ such that $\ell^{\lceil \frac{d}{2}\rceil} = n$

$\implies |S| = n \log_2 p$

$$\Pr[y_i' = y_i] \geq \left(1 - \frac{1}{\ell^\delta}\right)^{\text{Locality}} \geq 1 - O(\ell^{-\delta})$$

Key Intuition: Sparsity Helps

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,\vert\,1)^{\otimes\lceil\frac{d}{2}\rceil}$

Takeaway: Any given output is already correct with prob. $1 - \ell^{-\delta}$

Key Intuition: Sparsity Helps

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s} \mid 1)^{\otimes \lceil \frac{d}{2} \rceil}$

 # Error Locations $\mathsf{BAD} = \{i \mid y_i \neq y_i'\}$**:**

Expectation: $\mathbb{E}[\,|\mathsf{BAD}|\,] \leq O(\frac{m}{\ell^{\delta}})$

Key Intuition: Sparsity Helps

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,\|\,1)^{\otimes\lceil \frac{d}{2}\rceil}$

 # Error Locations $\mathsf{BAD} = \{i \mid y_i \neq y_i'\}$**:**

Expectation: $\mathbb{E}[\,|\mathsf{BAD}|\,] \leq O(\frac{m}{\ell^\delta})$

Via Markov Inequality,

Probability: $\Pr[\,|\mathsf{BAD}| \geq \lambda\frac{m}{\ell^\delta}] \leq O(\frac{1}{\lambda})$

Key Intuition: Sparsity Helps

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,\|\,1)^{\otimes\lceil\frac{d}{2}\rceil}$

 # Error Locations $\mathsf{BAD} = \{i \mid y_i \neq y_i'\}$**:**

Expectation: $\mathbb{E}[\,|\mathsf{BAD}|\,] \leq O(\frac{m}{\ell^\delta})$

Via Markov Inequality,

Probability: $\Pr[\,|\mathsf{BAD}| \geq \lambda\frac{m}{\ell^\delta}] \leq O(\frac{1}{\lambda})$

Goal: **Fix** $T = \frac{m\lambda}{\ell^\delta}$ **errors, while ensuring security and expansion.**

Correct T Errors (Failed First Attempt)

Define: $\text{Corr} = G(\vec{\sigma}) - G(\sigma_1 + e_1, \ldots, \sigma_n + e_n)$

sPRG Components:

Index I

$\{\vec{a}_i\}_{i \in [n]}$

Public Seed P

$\{b_i = \langle \vec{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$

Private Seed S

$(\vec{s} \mid 1)^{\otimes \lceil \frac{d}{2} \rceil}$

Corr

Correct T Errors (Failed First Attempt)

Define: $\text{Corr} = G(\overrightarrow{\sigma}) - G(\sigma_1 + e_1, \ldots, \sigma_n + e_n)$

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,|\,1)^{\otimes\lceil \frac{d}{2}\rceil}$ Corr

$$\overrightarrow{y}'_i = G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \text{Corr}_i$$

Problem: No Stretch. **Private Seed too big.**

Correct T Errors (Failed First Attempt)

Define: $\text{Corr} = G(\vec{\sigma}) - G(\sigma_1 + e_1, \ldots, \sigma_n + e_n)$

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\vec{a}_i\}_{i \in [n]}$	$\{b_i = \langle \vec{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$	$(\vec{s} \mid 1)^{\otimes \lceil \frac{d}{2} \rceil}$ Corr

$$\vec{y}_i' = G_i(\vec{\sigma} + \vec{e}) + \text{Corr}_i$$

Problem: No Stretch. Private Seed too big.

Use Corr **is sparse.**

Problem: Can't Reveal BAD

Define: $\text{Corr} = G(\overrightarrow{\sigma}) - G(\sigma_1 + e_1, \ldots, \sigma_n + e_n)$

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i \in [n]}$

BAD?

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$

Private Seed S

$(\vec{s} \mid 1)^{\otimes \lceil \frac{d}{2} \rceil}$

$\{\text{Corr}_i\}_{i \in \text{BAD}}$

Problem: Can't Reveal BAD

Define: $\text{Corr} = G(\overrightarrow{\sigma}) - G(\sigma_1 + e_1, \ldots, \sigma_n + e_n)$

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,|\,1)^{\otimes\lceil\frac{d}{2}\rceil}$
BAD?		$\{\text{Corr}_i\}_{i\in\text{BAD}}$

$$\overrightarrow{y}'_i = \begin{cases} G_i(\overrightarrow{\sigma} + \overrightarrow{e}), \textbf{ when } i \notin \text{BAD} \\ G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \text{Corr}_i, \textbf{ when } i \in \text{BAD} \end{cases}$$

Problem: Can't Reveal BAD

Define: $\text{Corr} = G(\vec{\sigma}) - G(\sigma_1 + e_1, \ldots, \sigma_n + e_n)$

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\vec{a}_i\}_{i\in[n]}$ BAD?	$\{b_i = \langle \vec{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,\vert\,1)^{\otimes\lceil\frac{d}{2}\rceil}$ $\{\text{Corr}_i\}_{i\in\text{BAD}}$

$$\vec{y}'_i = \begin{cases} G_i(\vec{\sigma} + \vec{e}), \textbf{ when } i \notin \text{BAD} \\ G_i(\vec{\sigma} + \vec{e}) + \text{Corr}_i, \textbf{ when } i \in \text{BAD} \end{cases}$$

Problem: No Security.

BAD reveals locations of errors in LPN Samples.

Problem: Can't Reveal BAD

Define: $\text{Corr} = G(\overrightarrow{\sigma}) - G(\sigma_1 + e_1, \ldots, \sigma_n + e_n)$

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i \in [n]}$ BAD?	$\{b_i = \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$	$(\vec{s} \mid 1)^{\otimes \lceil \frac{d}{2} \rceil}$ $\{\text{Corr}_i\}_{i \in \text{BAD}}$

Main Idea: Encode Corr as a sparse matrix and use matrix factorization.

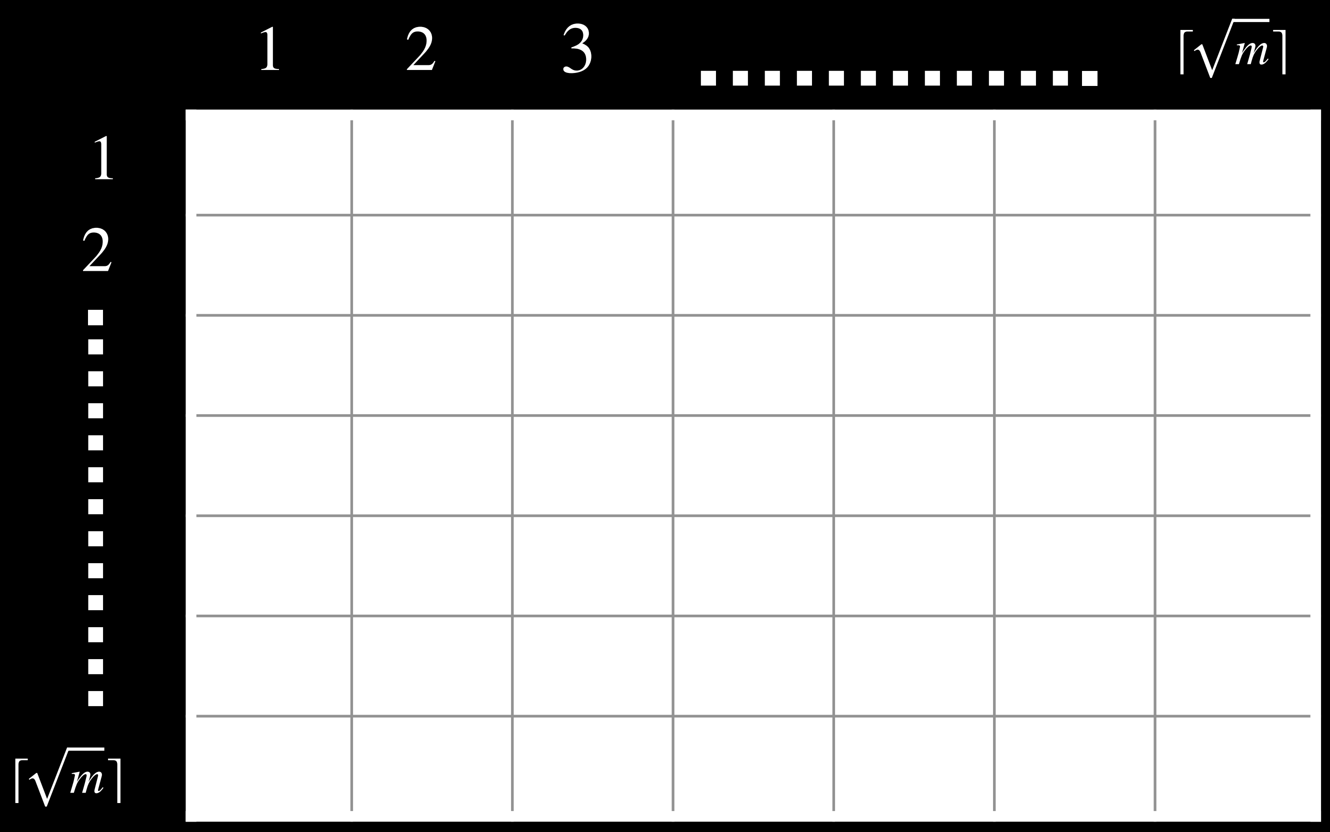
Example: Correcting One Error

Key Idea: Use the fact that Corr is sparse. Can compress using matrix factorization.

Example: Correcting One Error

Key Idea: Use the fact that Corr is sparse. Can compress using matrix factorization.

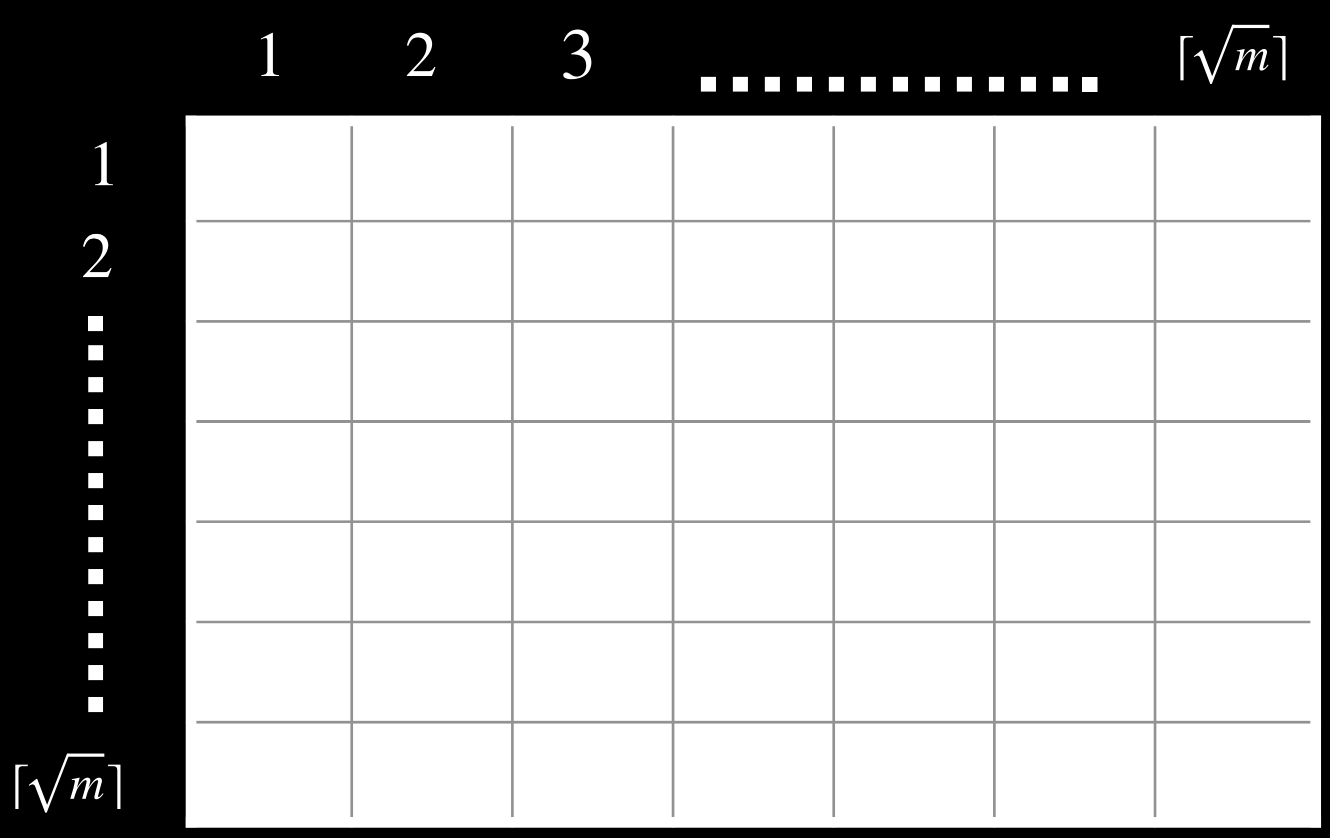
Assume: $T = |\mathsf{BAD}| = 1$. Arrange m outputs in a grid.



Example: Correcting One Error

Key Idea: Use the fact that Corr is sparse. Can compress using matrix factorization.

Assume: $T = |\mathsf{BAD}| = 1$. Arrange m outputs in a grid.



Sampling Details

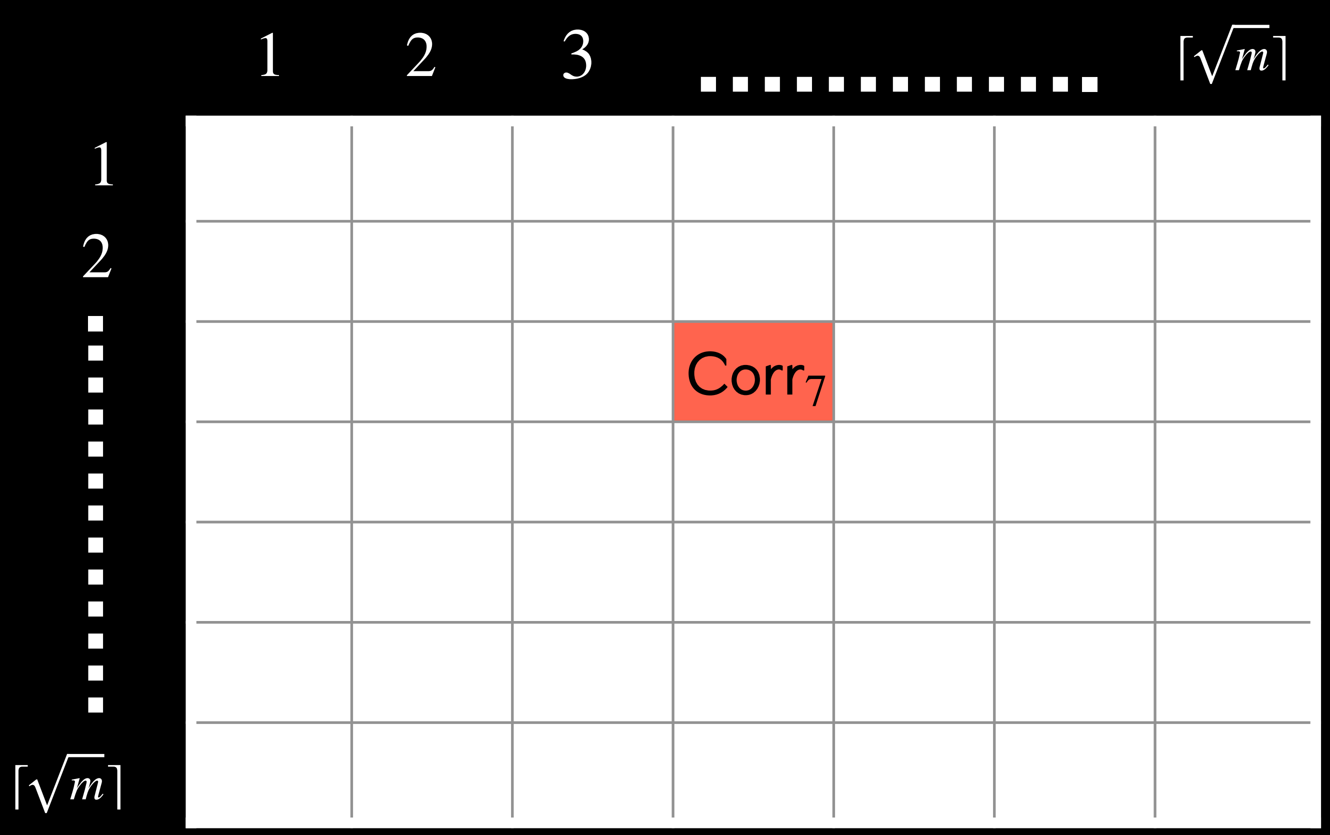
1. Obvious Bijection

$$\phi : [m] \to [\sqrt{m}] \times [\sqrt{m}]$$

Example: Correcting One Error

Key Idea: Use the fact that Corr is sparse. Can compress using matrix factorization.

Assume: $T = |\mathsf{BAD}| = 1$. Arrange m outputs in a grid.



Sampling Details

1. Obvious Bijection

$$\phi : [m] \rightarrow [\sqrt{m}] \times [\sqrt{m}]$$

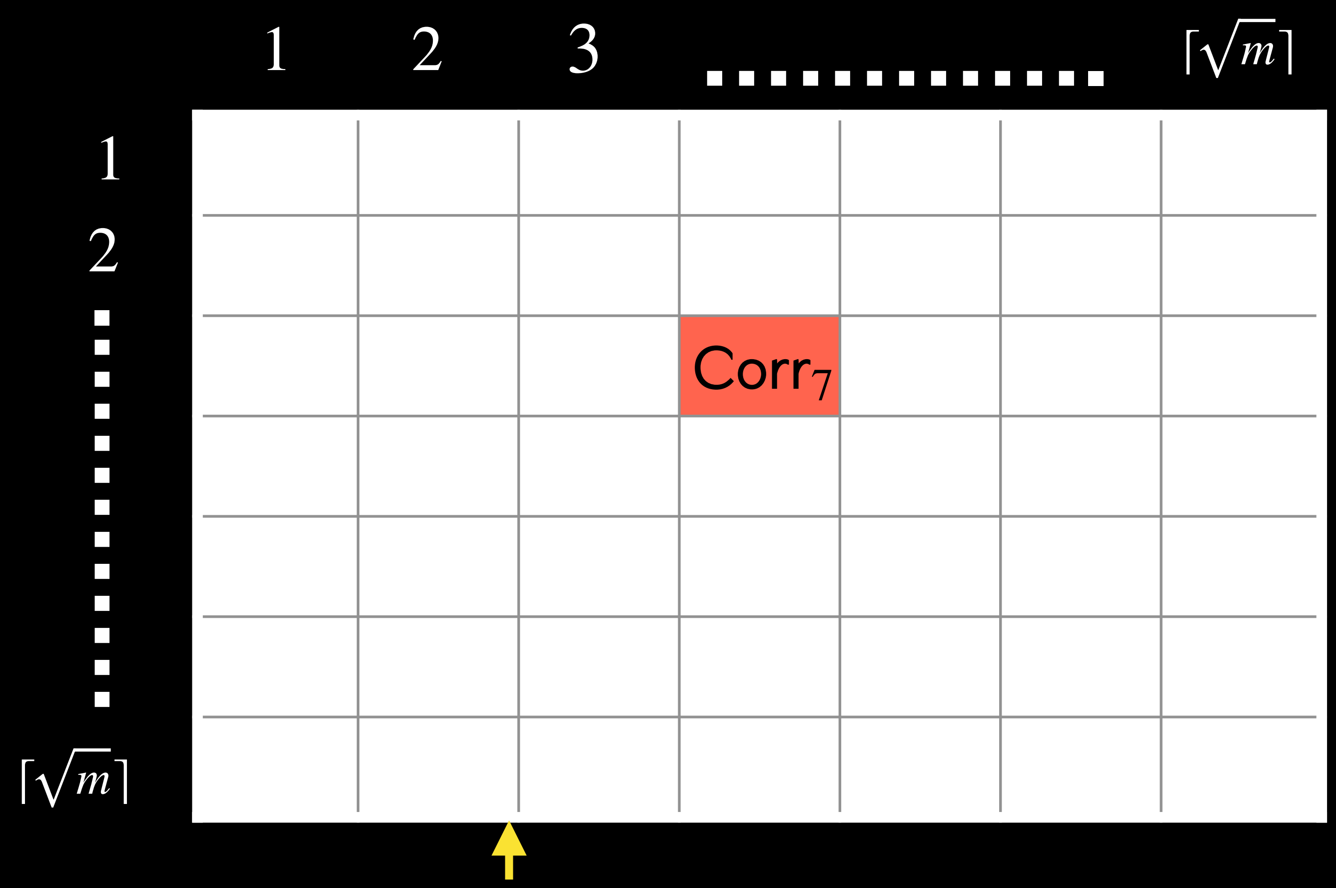
2. Form Matrix

$$M_{\phi(i)} = \mathsf{Corr}_i$$

Example: Correcting One Error

Key Idea: Use the fact that Corr is sparse. Can compress using matrix factorization.

Assume: $T = |\mathsf{BAD}| = 1$. Arrange m outputs in a grid.



Rank 1 Matrix

Sampling Details

1. Obvious Bijection

$$\phi : [m] \to [\sqrt{m}] \times [\sqrt{m}]$$

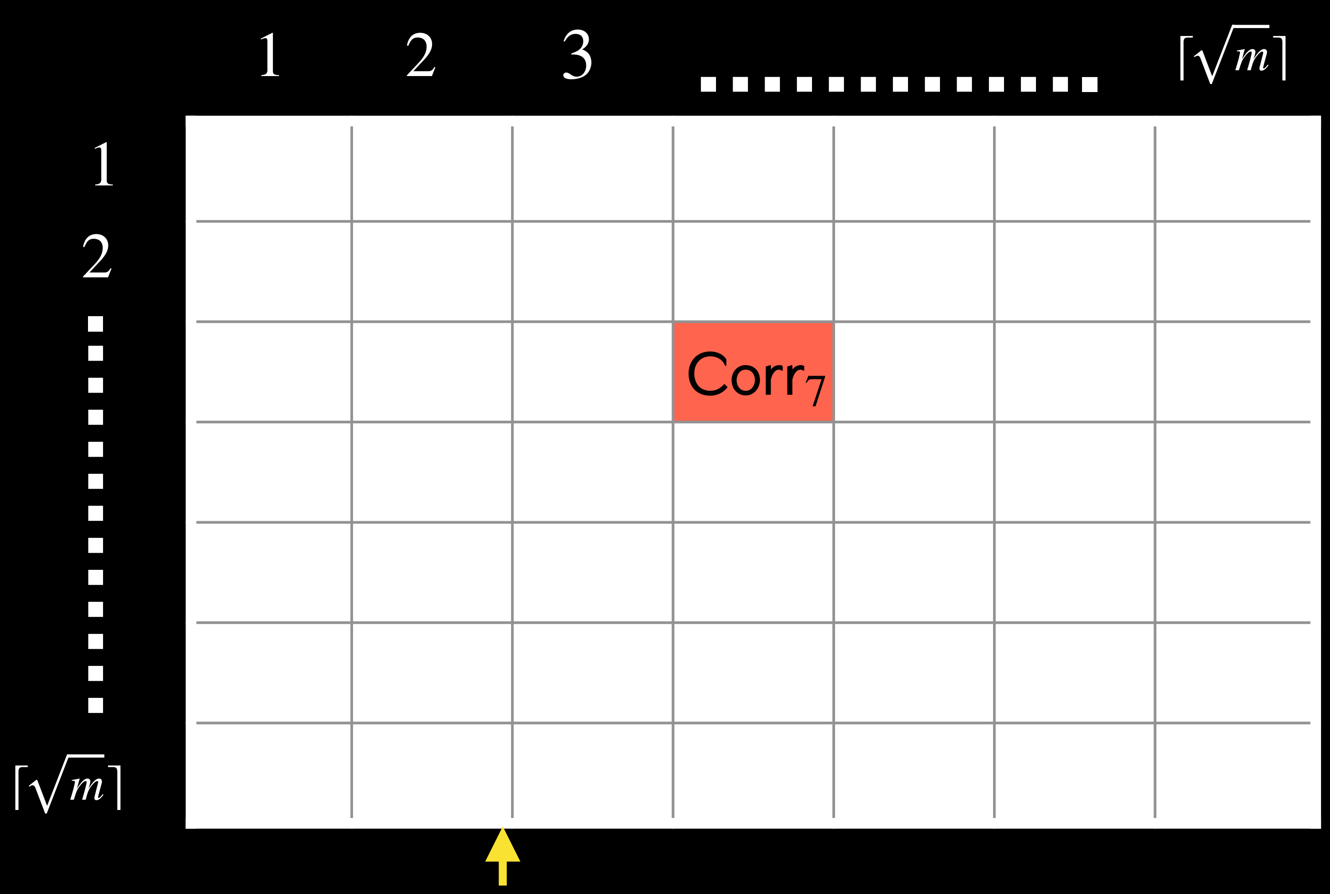
2. Form Matrix

$$M_{\phi(i)} = \mathsf{Corr}_i$$

Example: Correcting One Error

Key Idea: Use the fact that Corr is sparse. Can compress using matrix factorization.

Assume: $T = |\text{BAD}| = 1$. Arrange m outputs in a grid.



Rank 1 Matrix

Sampling Details

1. Obvious Bijection

$$\phi : [m] \to [\sqrt{m}] \times [\sqrt{m}]$$

2. Form Matrix

$$M_{\phi(i)} = \text{Corr}_i$$

3. Factor Matrix

$$M = U \cdot V$$

$$U, V^\top \in \mathbb{Z}_p^{\sqrt{m} \times 1}$$

Example: Correcting One Error

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

ϕ

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes\lceil\frac{d}{2}\rceil}$

U, V

Example: Correcting One Error

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

ϕ

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes \lceil \frac{d}{2} \rceil}$

U, V

Computation:

$$\overrightarrow{y}'_i = G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \underbrace{[U \cdot V]_{\phi(i)}}_{=M_{\phi(i)}}$$

$$= G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \mathsf{Corr}_i$$

Example: Correcting One Error

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

ϕ

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes\lceil\frac{d}{2}\rceil}$

U, V

Computation:

$$\overrightarrow{y}'_i = G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \underbrace{[U\cdot V]_{\phi(i)}}_{=M_{\phi(i)}}$$

$$= G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \mathsf{Corr}_i$$

Stretch:

$$|P| + |S| \leq O(n\log_2 p + \sqrt{m}\log_2 p)$$
$$\ll m$$

Example: Correcting One Error

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$ ϕ	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,\vert\,1)^{\otimes\lceil \frac{d}{2}\rceil}$ U, V

Computation:

$$\overrightarrow{y}'_i = G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \underbrace{[U \cdot V]_{\phi(i)}}_{=M_{\phi(i)}}$$

$$= G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \mathsf{Corr}_i$$

Stretch: ✓

$$|P| + |S| \leq O(n \log_2 p + \sqrt{m} \log_2 p) \ll m$$

Security: ✓

BAD is not leaked

Example: Correcting One Error

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,|\,1)^{\otimes\lceil \frac{d}{2}\rceil}$
ϕ		U, V

Computation:

$$\overrightarrow{y}'_i = G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \underbrace{[U \cdot V]_{\phi(i)}}_{=M_{\phi(i)}}$$

$$= G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \mathsf{Corr}_i$$

Stretch: ✓

$$|P| + |S| \leq O(n \log_2 p + \sqrt{m} \log_2 p)$$
$$\ll m$$

Security: ✓

BAD is not leaked

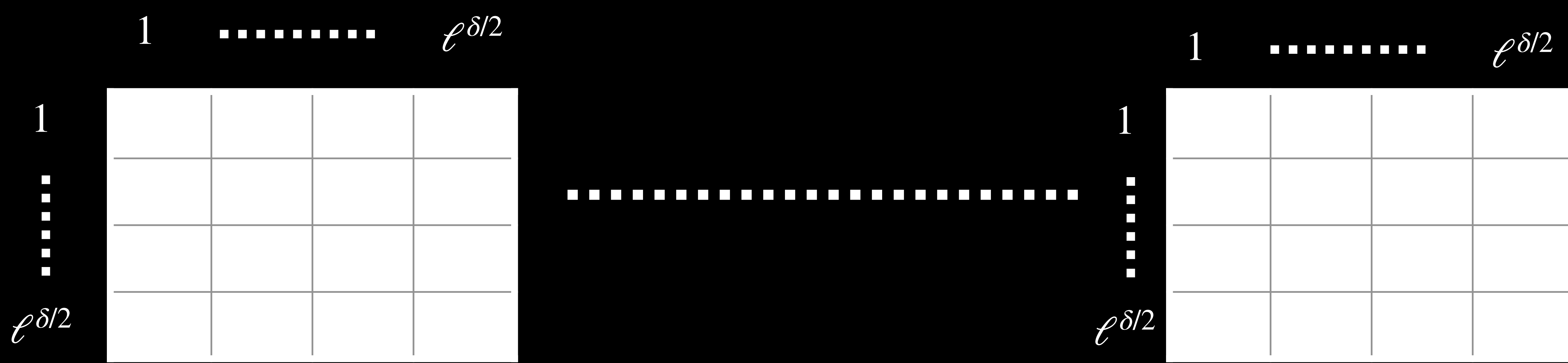
Takeaway:

Works for $m^{0.49}$ errors.

What about $T = \frac{m\lambda}{\ell^\delta}$ errors? ✗

Correcting $T = \frac{m\lambda}{\ell^\delta}$ Errors

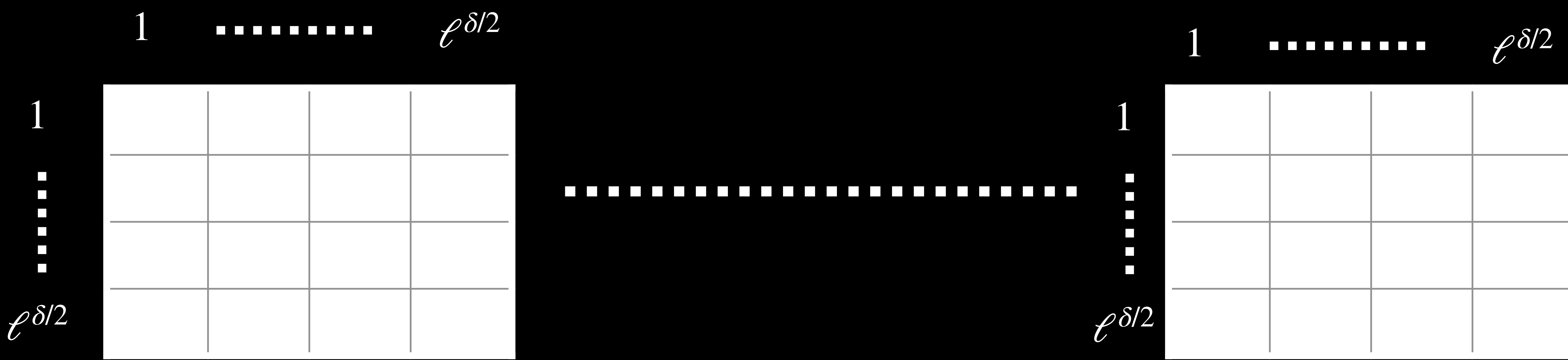
First Idea: Set up T matrices. On average each matrix get a constant number of errors.



Matrix M_1 Matrix M_T

Correcting $T = \frac{m\lambda}{\ell^\delta}$ Errors

First Idea: Set up T matrices. On average each matrix get a constant number of errors.



Matrix M_1 Matrix M_T

1. Sample Maps: Sample two random maps.

$$\phi_{bkt} : [m] \rightarrow [T].$$

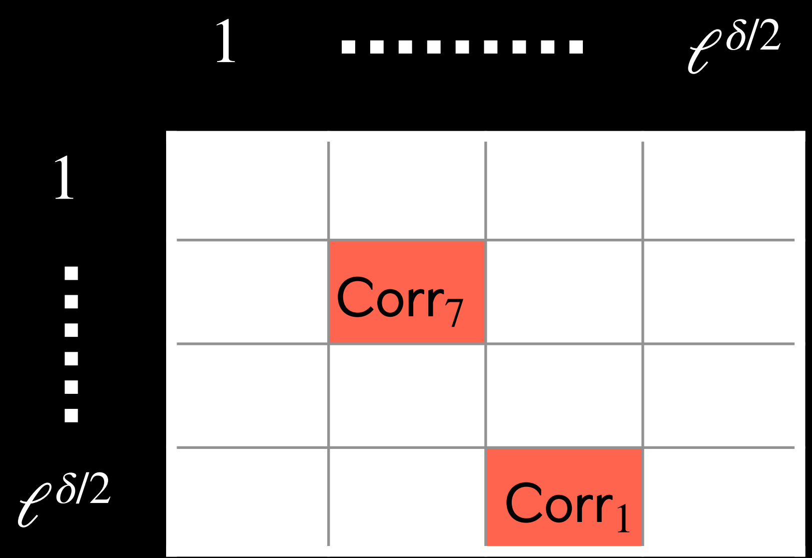
To assign an output bit to a matrix.

$$\phi_{ind} : [m] \rightarrow [\ell^{\delta/2}] \times [\ell^{\delta/2}].$$

To assign an output bit to a position in the matrix.

Correcting $T = \frac{m\lambda}{\ell^{\delta}}$ Errors

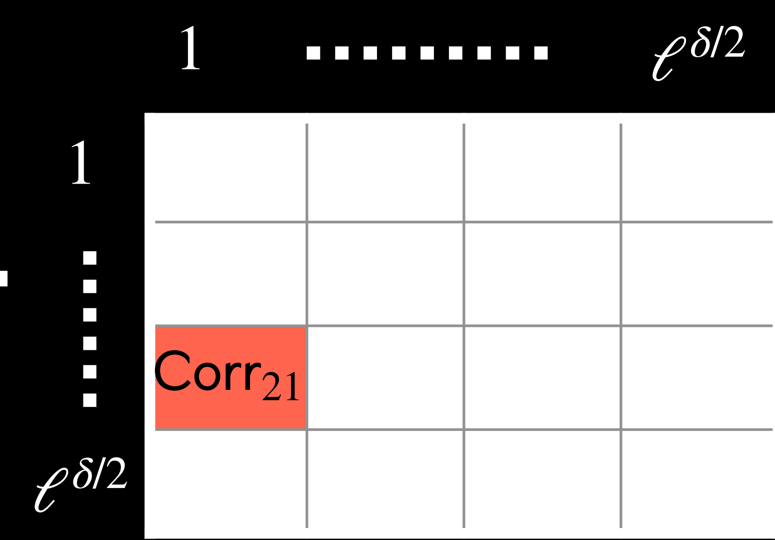
First Idea: Set up T matrices. On average each matrix get a constant number of errors.



Matrix M_1

2. Form Matrices:

$$M_{\phi_{bkt(i)}}[\phi_{ind}(i)] = \text{Corr}_i$$



Matrix M_T

1. Sample Maps: Sample two random maps.

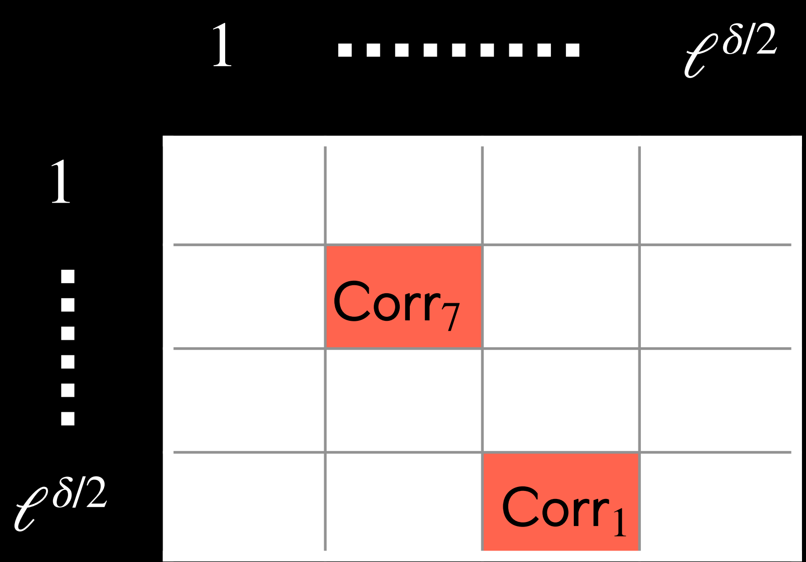
$$\phi_{bkt} : [m] \rightarrow [T].$$

To assign an output bit to a matrix.

$$\phi_{ind} : [m] \rightarrow [\ell^{\delta/2}] \times [\ell^{\delta/2}].$$

To assign an output bit to a position in the matrix.

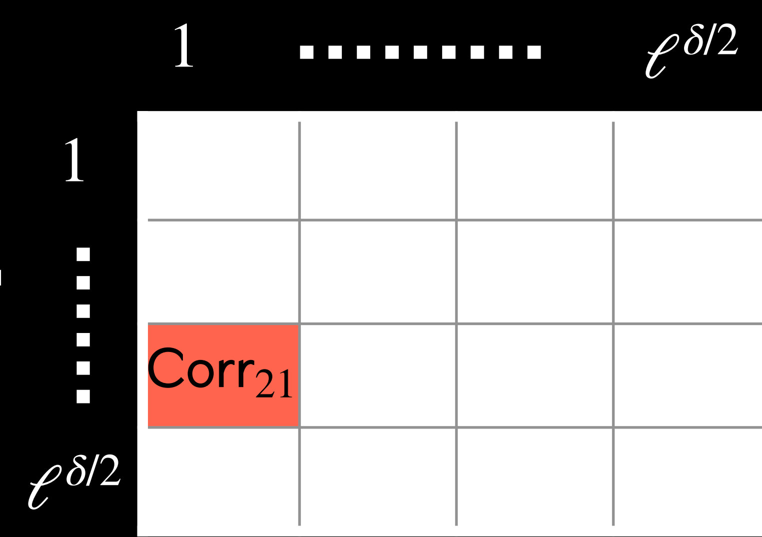
Correcting $T = \frac{m\lambda}{\ell^{\delta}}$ Errors



Matrix M_1

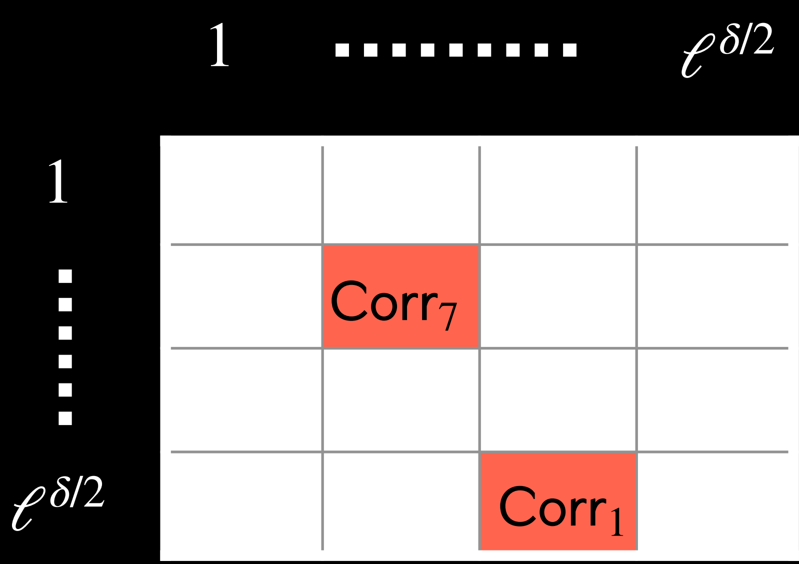
Set :

$M_{\phi_{bkt(i)}}[\phi_{ind}(i)] = \mathsf{Corr}_i$



Matrix M_T

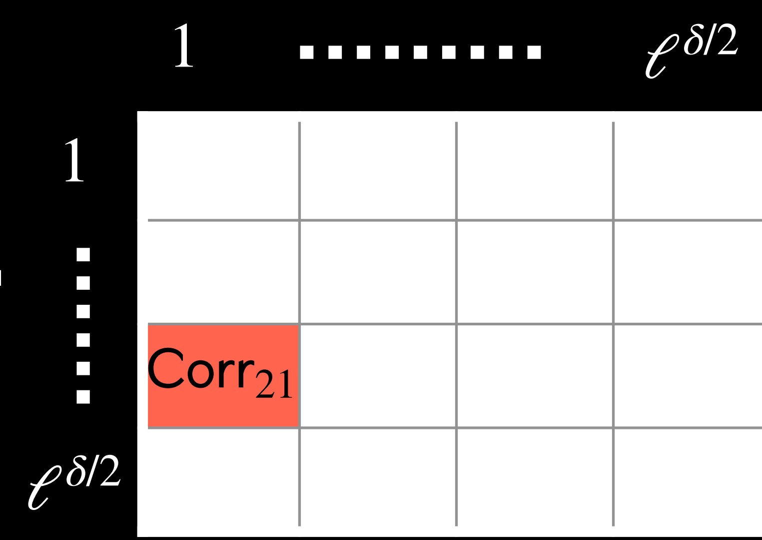
Correcting $T = \frac{m\lambda}{\ell^\delta}$ Errors



Matrix M_1

Set :

$M_{\phi_{bkt(i)}}[\phi_{ind}(i)] = \text{Corr}_i$

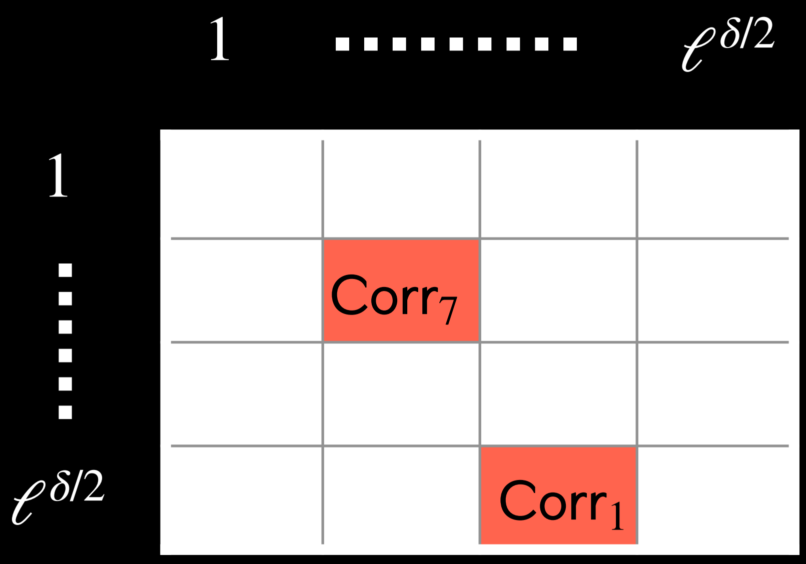


Matrix M_T

Matrices are Low Rank.

Corr **is T-sparse and $\overrightarrow{\phi} = (\phi_{bkt}, \phi_{rand})$ is random.**

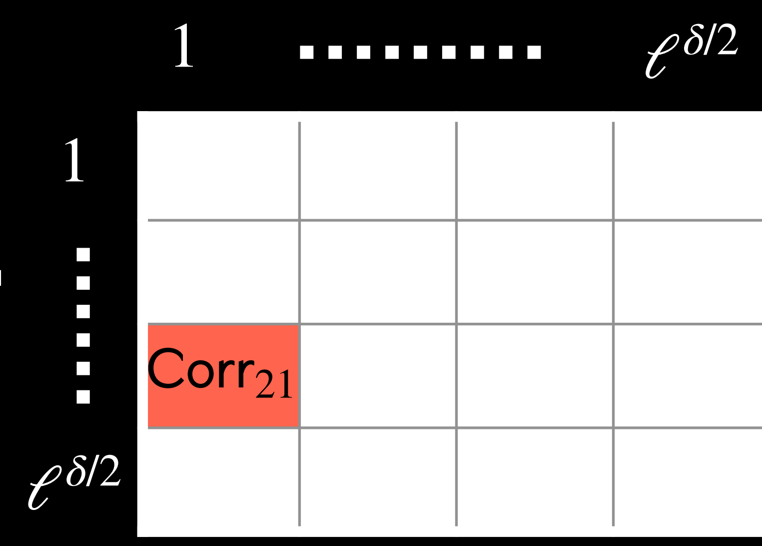
Correcting $T = \frac{m\lambda}{\ell^\delta}$ Errors



Matrix M_1

Set:

$M_{\phi_{bkt(i)}}[\phi_{ind}(i)] = \text{Corr}_i$



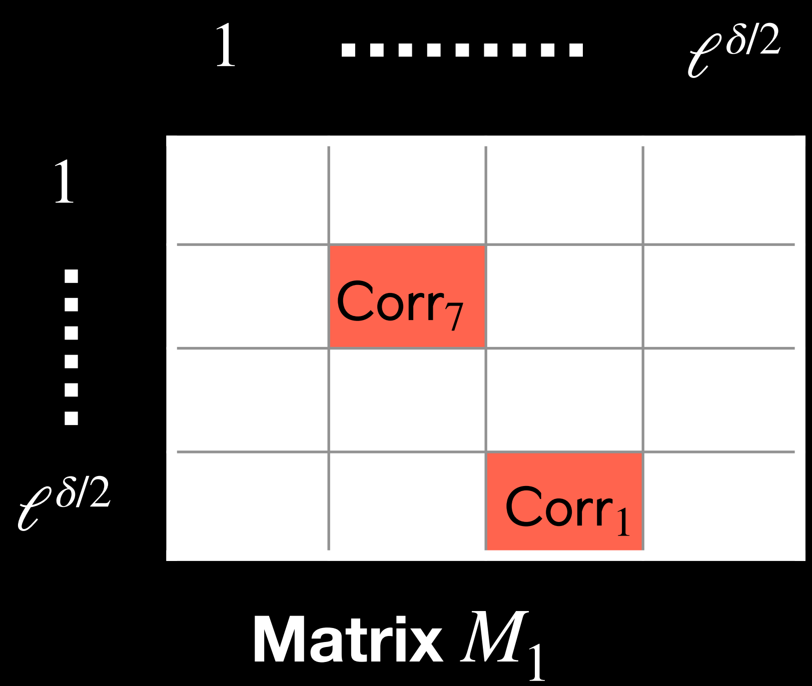
Matrix M_T

Matrices are Low Rank.

Corr **is** T**-sparse and** $\overrightarrow{\phi} = (\phi_{bkt}, \phi_{rand})$ **is random.**

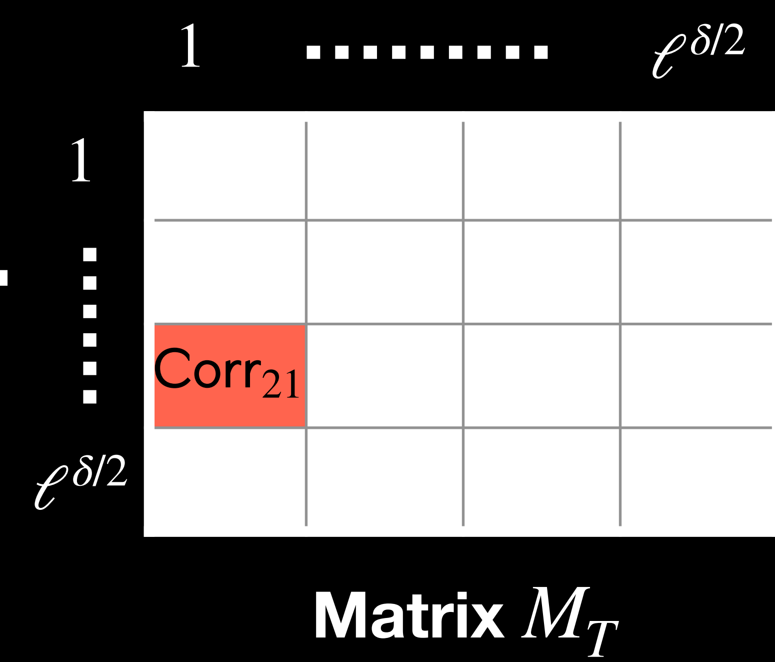
With probability $1 - e^{-\lambda}$, density$(M_i) \leq \lambda$, **(** $\implies$ rank$(M_i) \leq \lambda$**)**

Correcting $T = \frac{m\lambda}{\ell^{\delta}}$ Errors



Set :

$M_{\phi_{bkt(i)}}[\phi_{ind}(i)] = \text{Corr}_i$



Matrices are Low Rank.

Corr **is T-sparse and $\overrightarrow{\phi} = (\phi_{bkt}, \phi_{rand})$ is random.**

With probability $1 - e^{-\lambda}$, density$(M_i) \leq \lambda$, ($\implies$ rank$(M_i) \leq \lambda$)

3. Factor Matrix

$M_i = U_i \cdot V_i$

$U_i, V_i^{\top} \in \mathbb{Z}_p^{\sqrt{\ell^{\delta}} \times \lambda}$

Correcting $T = \frac{m\lambda}{\ell^\delta}$ Errors

sPRG Components:

Index I	Public Seed P	Private Seed S
$\{\overrightarrow{a}_i\}_{i\in[n]}$ $\overrightarrow{\phi} = (\phi_{bkt}, \phi_{ind})$	$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$	$(\vec{s}\,\vert\, 1)^{\otimes\lceil \frac{d}{2}\rceil}$ $\{U_i, V_i\}_{i\in[T]}$

Correcting $T = \frac{m\lambda}{\ell\delta}$ Errors

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

$\overrightarrow{\phi} = (\phi_{bkt}, \phi_{ind})$

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes\lceil\frac{d}{2}\rceil}$

$\{U_i, V_i\}_{i\in[T]}$

Computation:

$$\overrightarrow{y}'_i = G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \underbrace{[U_{\phi_{bkt}(i)} \cdot V_{\phi_{bkt}(i)}]_{\phi_{ind}(i)}}_{=M_{\phi_{bkt}(i)}}$$

$$= G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \mathsf{Corr}_i$$

Correcting $T = \frac{m\lambda}{\ell\delta}$ Errors

sPRG Components:

Index I

$\{\vec{a}_i\}_{i\in[n]}$

$\vec{\phi} = (\phi_{bkt}, \phi_{ind})$

Public Seed P

$\{b_i = \langle \vec{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes\lceil \frac{d}{2}\rceil}$

$\{U_i, V_i\}_{i\in[T]}$

Computation:

$$\vec{y}'_i = G_i(\vec{\sigma} + \vec{e}) + \underbrace{[U_{\phi_{bkt}(i)} \cdot V_{\phi_{bkt}(i)}]_{\phi_{ind}(i)}}_{=M_{\phi_{bkt}(i)}}$$

$$= G_i(\vec{\sigma} + \vec{e}) + \mathsf{Corr}_i$$

Security: ✓

Corrects T errors.

Correcting $T = \frac{m\lambda}{\ell^{\delta}}$ Errors

sPRG Components:

Index I

$\{\overrightarrow{a}_i\}_{i\in[n]}$

$\overrightarrow{\phi} = (\phi_{bkt}, \phi_{ind})$

Public Seed P

$\{b_i = \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}$

Private Seed S

$(\vec{s}\,|\,1)^{\otimes \lceil \frac{d}{2} \rceil}$

$\{U_i, V_i\}_{i\in[T]}$

Computation:

$$\overrightarrow{y}'_i = G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \underbrace{[U_{\phi_{bkt}(i)} \cdot V_{\phi_{bkt}(i)}]_{\phi_{ind}(i)}}_{=M_{\phi_{bkt}(i)}}$$

$$= G_i(\overrightarrow{\sigma} + \overrightarrow{e}) + \mathsf{Corr}_i$$

Security: ✓

Corrects T errors.

Stretch: ✓

$$|P| + |S| = O(n \log_2 p + T \cdot |U_i|)$$

$$= O(n \log_2 p + \frac{m\lambda^2}{\ell^{\delta/2}} \log_2 p)$$

$$\ll m$$

(as $\ell \gg \lambda$)

sPRG: Simple Proof

$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$ $\quad G(\overrightarrow{\sigma})$

sPRG: Simple Proof

$$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]} \qquad G(\overrightarrow{\sigma})$$

$\approx_c$ **By LPN**

$$\{\overrightarrow{a}_i, u_i + \sigma_i \mod p\}_{i \in [n]} \qquad G(\overrightarrow{\sigma})$$

sPRG: Simple Proof

$$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]} \qquad G(\overrightarrow{\sigma})$$

$\approx_c$ **By LPN**

$$\{\overrightarrow{a}_i, u_i + \sigma_i \mod p\}_{i \in [n]} \qquad G(\overrightarrow{\sigma})$$

$\approx_{id}$ **Identical Distribution**

$$\{\overrightarrow{a}_i, u_i \mod p\}_{i \in [n]} \qquad G(\overrightarrow{\sigma})$$

sPRG: Simple Proof

$\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p\}_{i \in [n]}$ $\quad G(\overrightarrow{\sigma})$

$\approx_c$ **By LPN**

$\{\overrightarrow{a}_i, u_i + \sigma_i \mod p\}_{i \in [n]}$ $\quad G(\overrightarrow{\sigma})$

$\approx_{id}$ **Identical Distribution**

$\{\overrightarrow{a}_i, u_i \mod p\}_{i \in [n]}$ $\quad G(\overrightarrow{\sigma})$

$\approx_c$ **PRG Security**

$\{\overrightarrow{a}_i, u_i \mod p\}_{i \in [n]}$ $\quad r \leftarrow \{0,1\}^m$

Q.E.D. ∎

How to get iO?

Learn $\left(\{ \overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p \}_{i \in [n]}, G(\overrightarrow{\sigma}) \right)$ **with probability** $1 - \dfrac{1}{\lambda}$**.**

How to get iO?

Learn $\left(\{ \overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s} \rangle + e_i + \sigma_i \mod p \}_{i \in [n]}, G(\overrightarrow{\sigma}) \right)$ **with probability** $1 - \dfrac{1}{\lambda}$**.**

$1 - \dfrac{1}{\lambda}$ **security immediate from LPN and PRG security.**

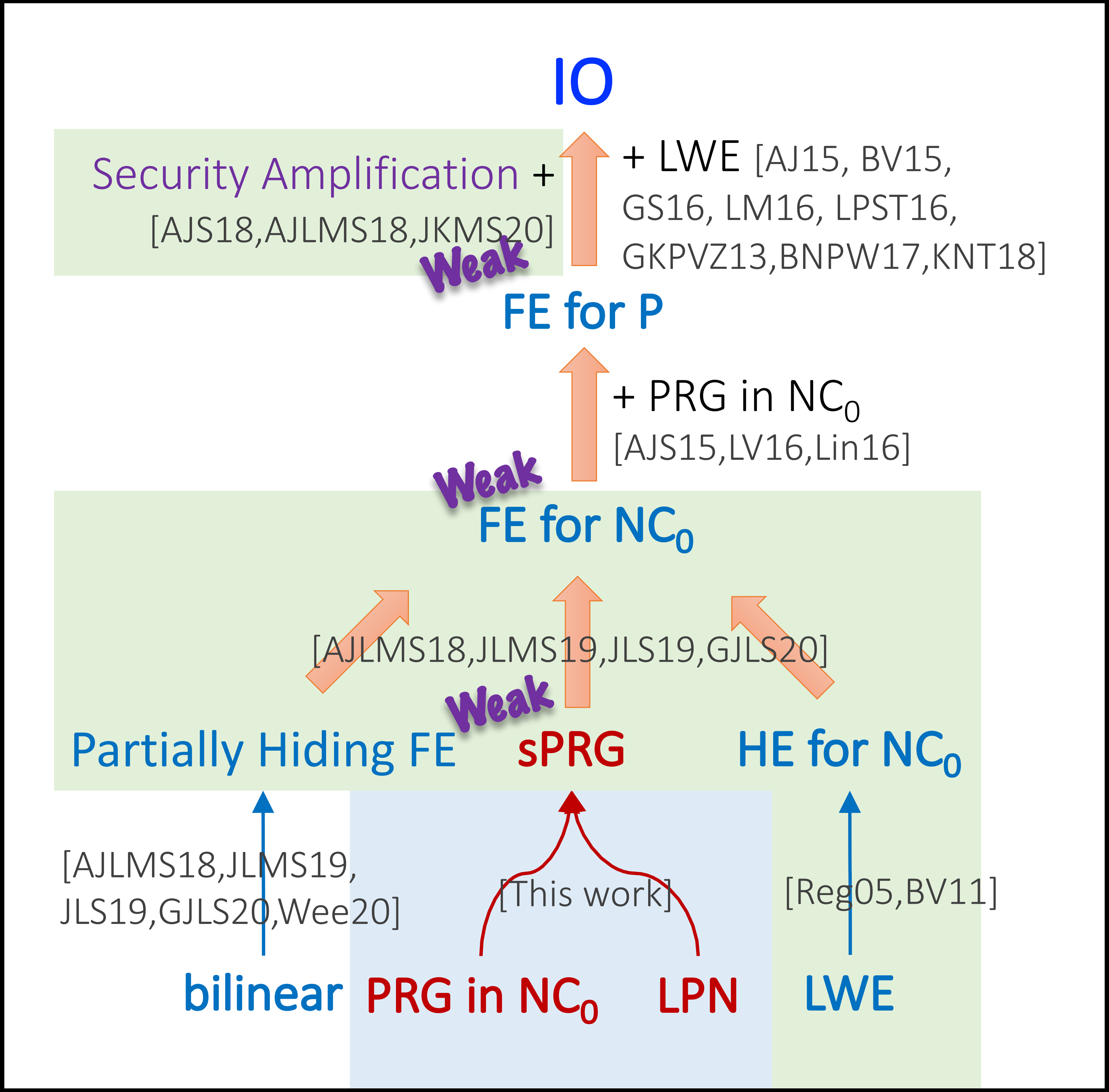
How to get iO?

Learn $\left(\{\overrightarrow{a}_i, \langle \overrightarrow{a}_i, \vec{s}\rangle + e_i + \sigma_i \mod p\}_{i\in[n]}, G(\overrightarrow{\sigma})\right)$ **with probability** $1 - \dfrac{1}{\lambda}$**.**

$1 - \dfrac{1}{\lambda}$ **security immediate from LPN and PRG security.**

Construct Weakly secure sPRG.

Bird's Eye View



Credits: Rachel Lin

Open Problems?

Construct provably secure post-quantum secure $i\mathcal{O}$ from well-founded assumptions?

Our Line: Remove reliance on SXDH.

New Directions: [BDGM 20a, GP 20, WW 20, BDGM 20b]

Open Problems?

Construct provably secure post-quantum secure $i\mathcal{O}$ from well-founded assumptions?

Our Line: Remove reliance on SXDH.

New Directions: [BDGM 20a, GP 20, WW 20, BDGM 20b]

Efficiency Question? [GJK 19, BIJMSZ 20]

Open Problems?

Construct provably secure post-quantum secure $i\mathcal{O}$ from well-founded assumptions?

Our Line: Remove reliance on SXDH.

New Directions: [BDGM 20a, GP 20, WW 20, BDGM 20b]

Efficiency Question? [GJK 19, BIJMSZ 20]

Other approaches yielding $i\mathcal{O}$ from well-founded assumptions?