Indistinguishability Obfuscation and Learning Problems

Aayush Jain
NTT Research, CMU (Fall 2022)

Indistinguishability Obfuscation ($i\mathcal{O}$)

[DH 76, BGIRSVY 01]

(same input-output behavior)

(Polynomially slower)

Hides implementation differences!

Indistinguishability Obfuscation ($i\mathcal{O}$)

[DH 76, BGIRSVY 01]

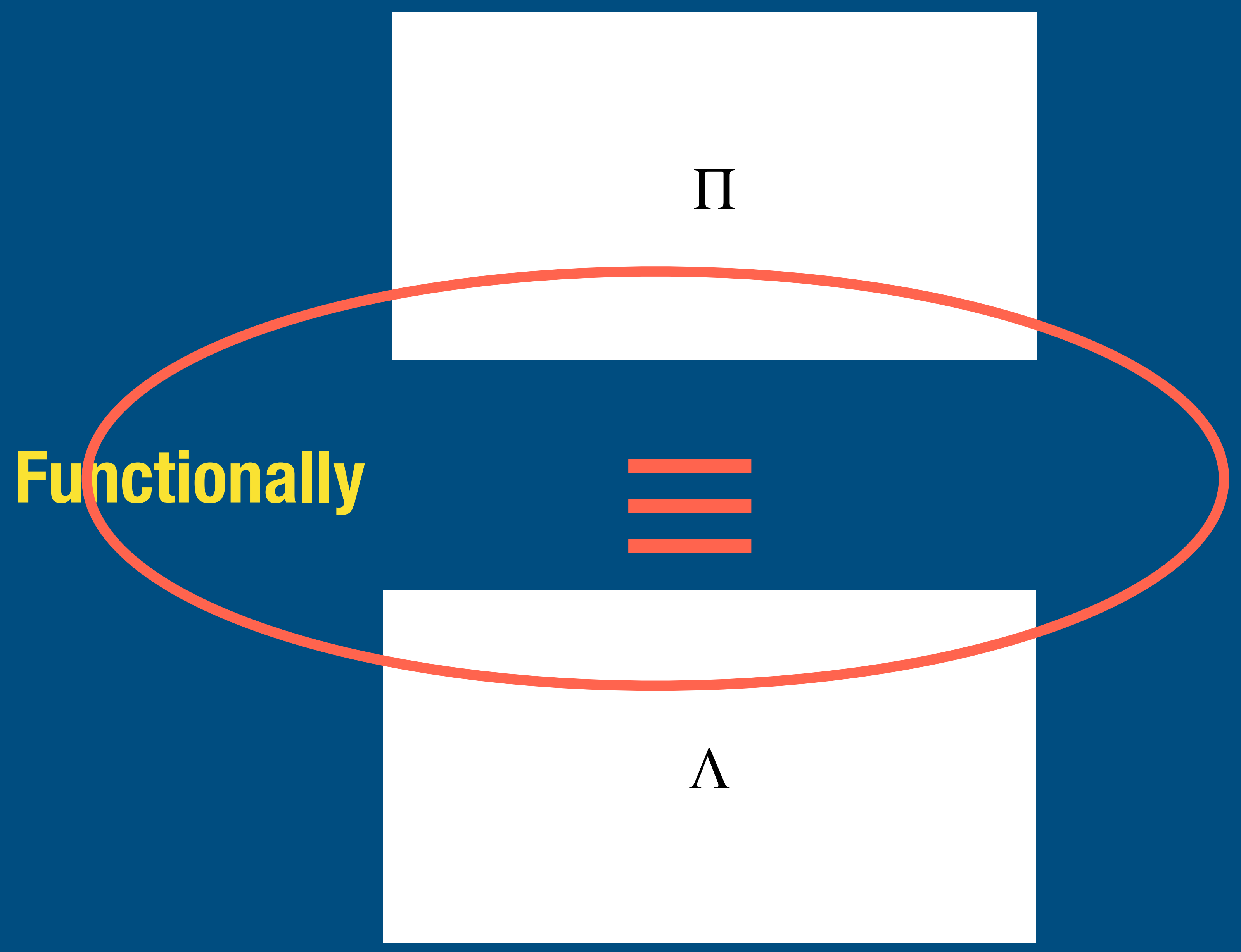
Π

Functionally $\equiv$

Λ

Indistinguishability Obfuscation ($i\mathcal{O}$)

[DH 76, BGIRSVY 01]



Same Input-Output Behavior

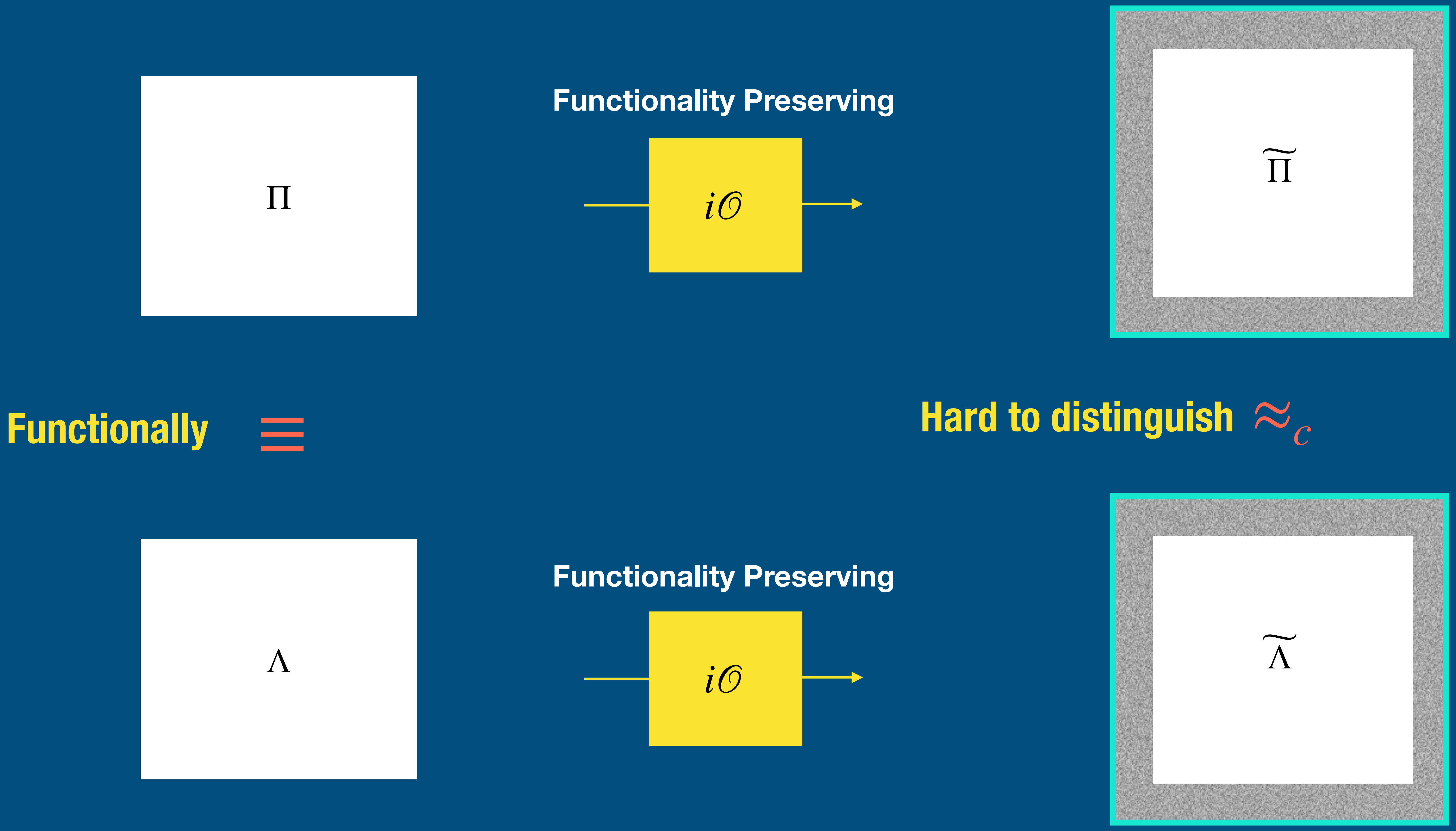
Common Sense Requirements:

- **Running times**
- **Description size**

Different Implementations

Indistinguishability Obfuscation ($i\mathcal{O}$)

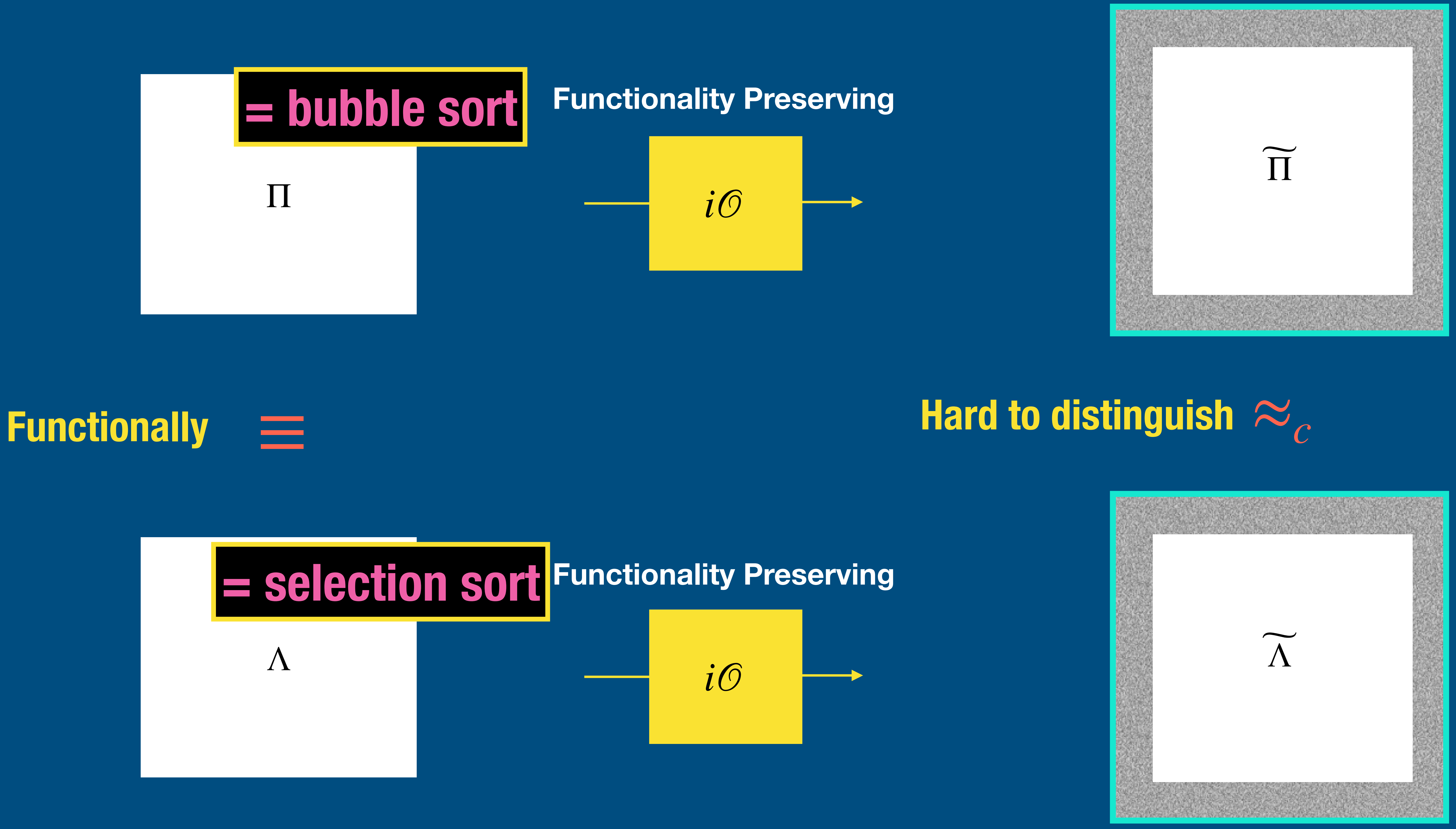
[DH 76, BGIRSVY 01]



Hides implementation differences!

Indistinguishability Obfuscation ($i\mathcal{O}$)

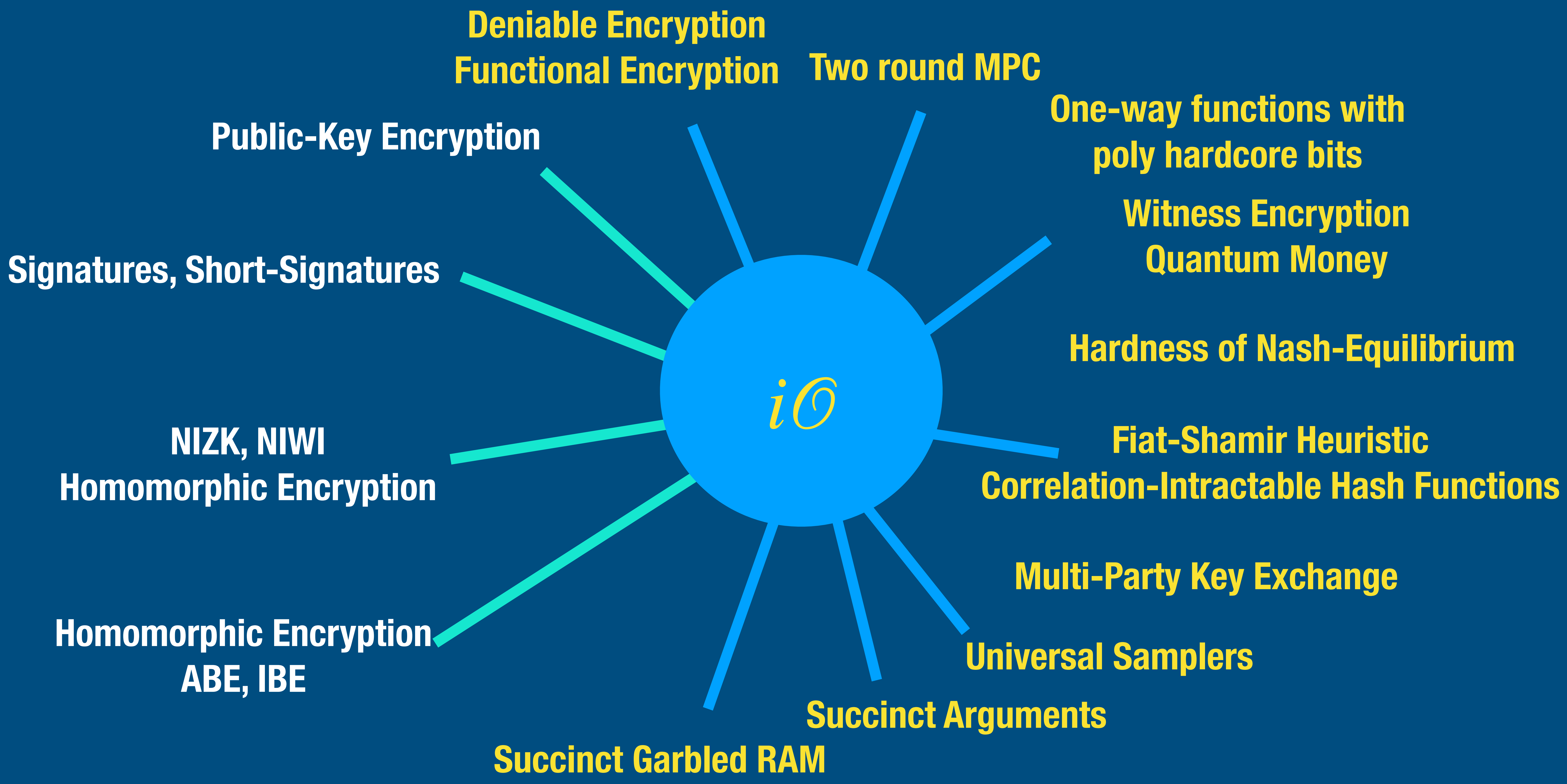
[DH 76, BGIRSVY 01]



Hides implementation differences!

Applications: Indistinguishability Obfuscation ($i\mathcal{O}$)

[SW 14, 100's of works]



Pre-$i\mathcal{O}$ applications!

Brave new world!

Problems Used to Construct $i\mathcal{O}$

Constructions of
Indistinguishability Obfuscation [GGHRSW 13 ++]

Both styles, not feasible for implementation yet.

Using Pairing Groups / Elliptic Curves

[LT 18, AJLMS 19, Agr 19, JLMS 19….]

[JLS 20, JLS 21]

Computational Problems:
Boolean PRG in NC^0
Learning Parity with Noise over $\mathbb{Z}_p$
Elliptic Curve Cryptography

- Well studied assumptions
- Elliptic curve crypto broken in quantum polynomial time

Lattice Decoding Only

[Mmaps, BDGM 20]

[WW 21, GP 21, BDGM 21, HJL 21 DQVWW 21]

Computational Problems:
LWE ++ (LWE + structured leakage)

- New, exciting and needs analysis
- Holy grail: a construction from LWE alone
- Also important: LWE+well understood leakage

Problems Used to Construct $i\mathcal{O}$

Constructions of
Indistinguishability Obfuscation [GGHRSW 13 ++]

Both styles, not feasible for implementation yet.

Using Pairing Groups / Elliptic Curves

[LT 18, AJLMS 19, Agr 19, JLMS 19….]

[JLS 20, JLS 21]

Computational Problems:
Boolean PRG in NC^0
Learning Parity with Noise over $\mathbb{Z}_p$
Elliptic Curve Cryptography

- Well studied assumptions
- Elliptic curve crypto broken in quantum polynomial time

Lattice Decoding Only

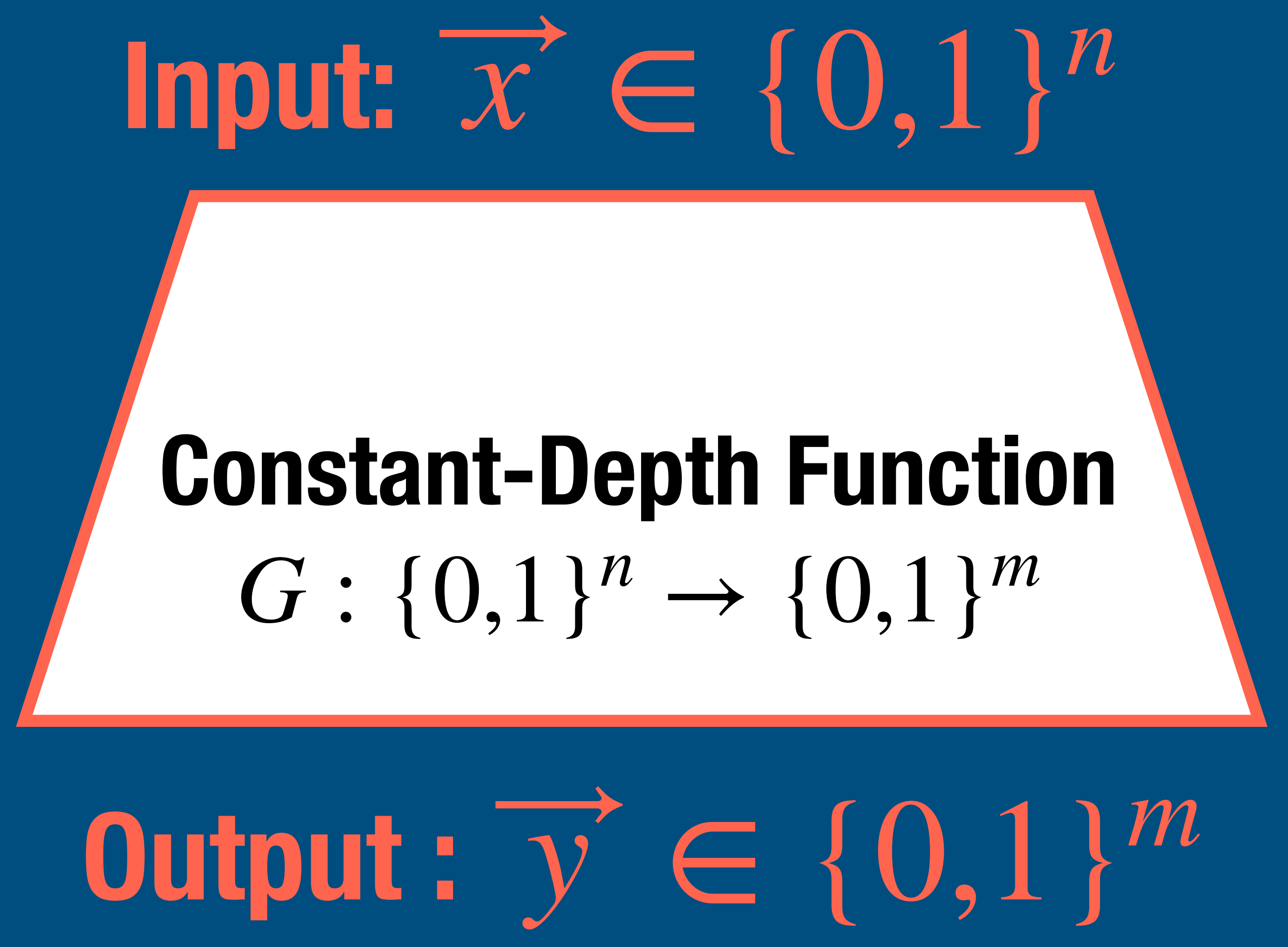
[Mmaps, BDGM 20]

[WW 21, GP 21, BDGM 21, HJL 21 DQVWW 21]

Computational Problems:
LWE ++ (LWE + structured leakage)

- New, exciting and needs analysis
- Holy grail: a construction from LWE alone
- Also important: LWE+well understood leakage

Boolean PRGs in NC^0



Computable by: Constant-depth circuits.

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

Cryptographic Security:

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

For any polynomial time attacker $\mathscr{A}$,

$$\left| \Pr_{x \leftarrow \{0,1\}^n} [\mathscr{A}(G(x)) = 1] - \Pr_{r \leftarrow \{0,1\}^m} [\mathscr{A}(r) = 1] \right| \leq \text{CRYPTOSMALL} = 2^{-n^{\Omega(1)}}$$

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

How to Build Boolean PRGs in NC^0

A general recipe by Goldreich in 2001.

A balanced constant local predicate

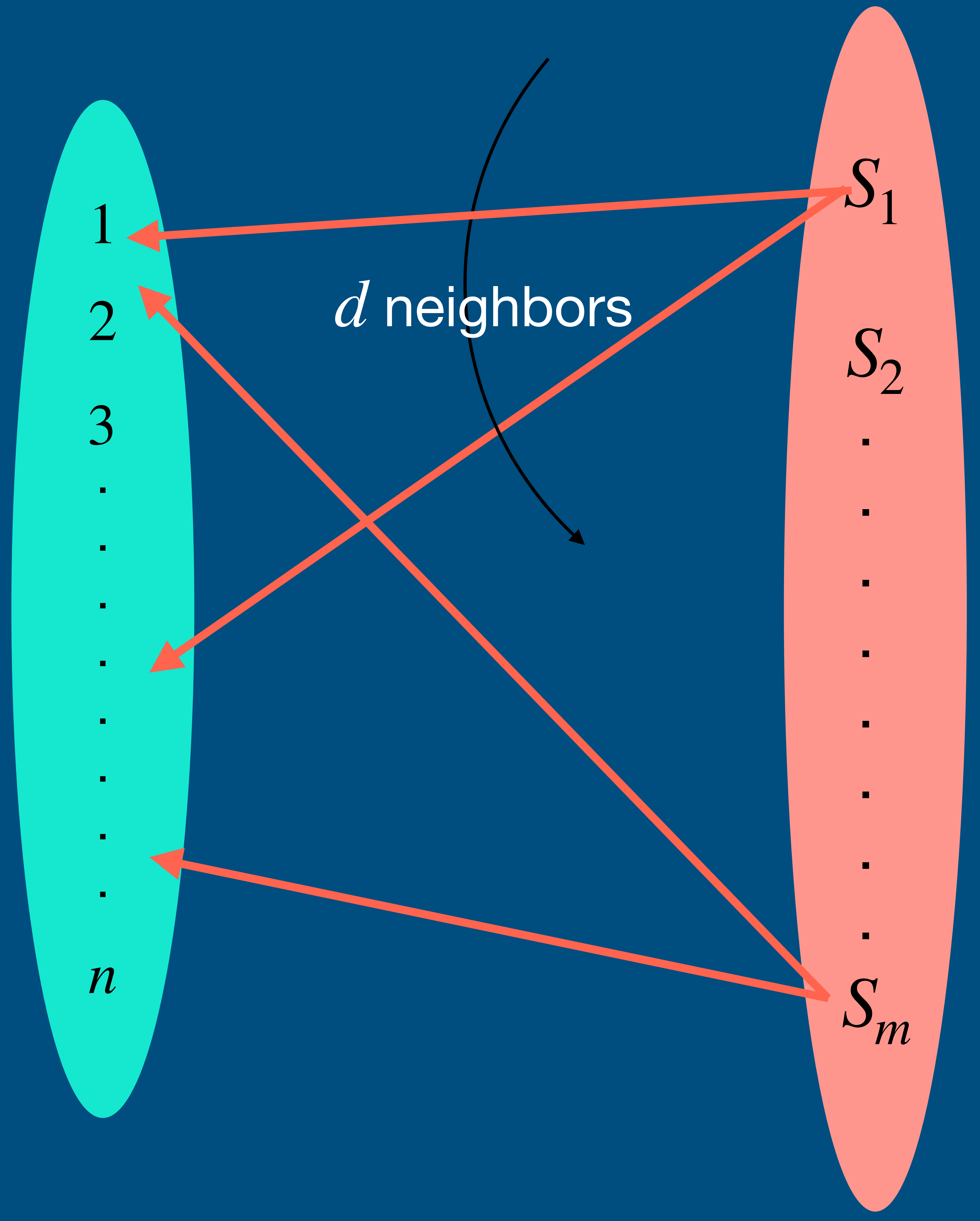
$$P : \{0,1\}^d \to \{0,1\}$$

$$f_{P,H}(\vec{x} \in \{0,1\}^n) = (y_1, \ldots, y_m)$$

$$y_i = P(x_{i_1}, \ldots, x_{i_d}) \text{ where } S_i = \{i_1, \ldots, i_d\}$$

PRG Conjecture:

Properly chosen H and $P \implies f_{P,H}$ is a secure PRG



Hypergraph $H = (S_1, \ldots, S_m)$

Random d-CSPs

A balanced constant local predicate

$P : \{0,1\}^d \to \{0,1\}$ and a random H

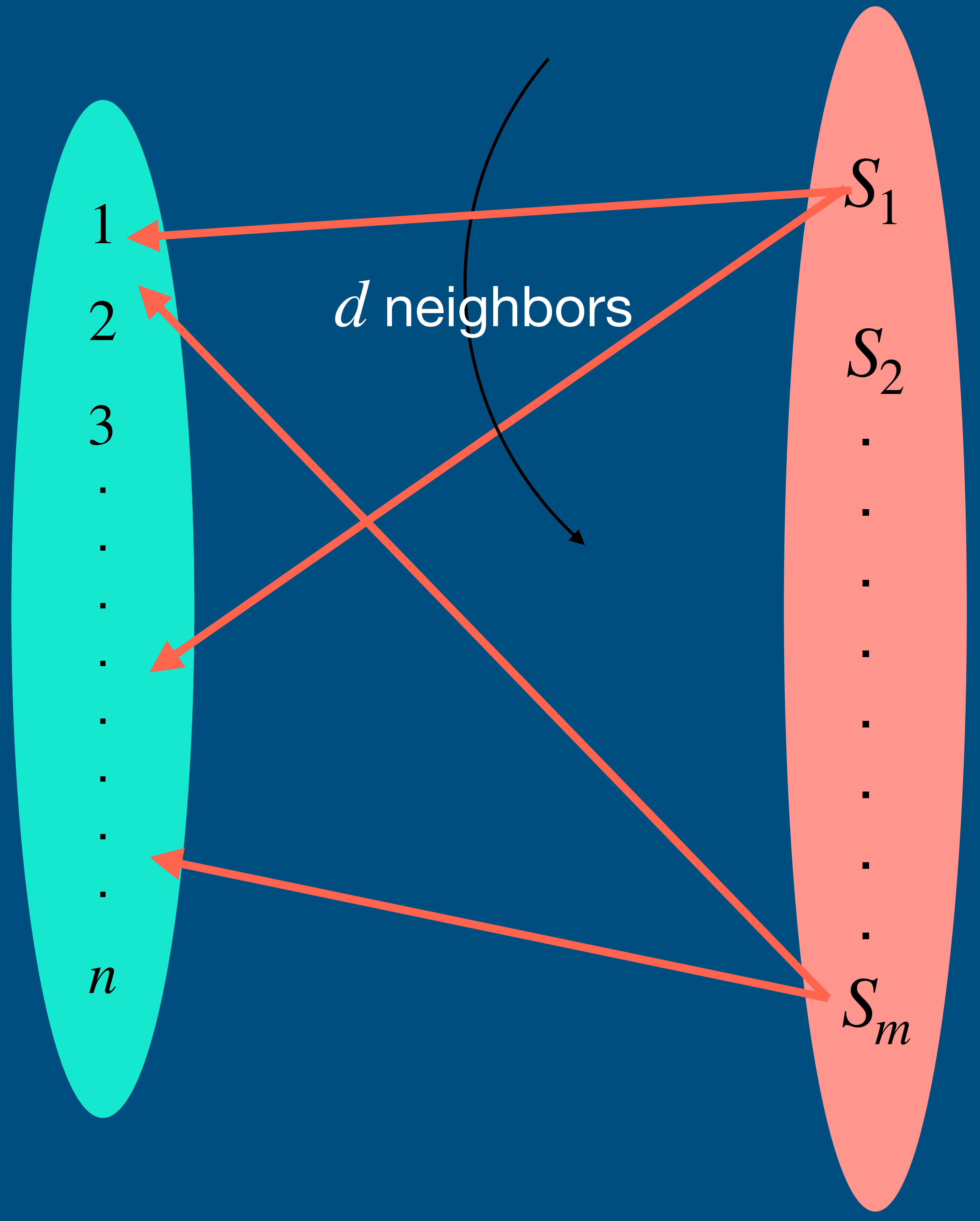
$d \geq 3$

Planted Distribution:

- Sample $x^* \leftarrow \{0,1\}^n$
- m constraints, one per $S_i = \{i_1, \ldots, i_d\}$.
  1. Sample $\vec{c}_i \leftarrow \{0,1\}^d$, and flip_i from $Ber(\rho)$
  2. Output $\vec{c}_i, b_i = P(\vec{c}_i \oplus x^* |_{S_i}) \oplus \text{flip}_i$

Random Distribution:

- m constraints, one per $S_i = \{i_1, \ldots, i_d\}$.
  1. Sample $\vec{c}_i \leftarrow \{0,1\}^d$, and r_i from $Ber(0.5)$
  2. Output $\vec{c}_i, b_i = r_i$



Hypergraph $H = (S_1, \ldots, S_m)$

$m = \Delta n$

Problems about Random d-CSPs

Objective: Val(x) = Number of constraints satisfied by x

OPT = $max_x Val(x)$

OPT[planted] $\geq m(1 - \rho - o(1))$ with high probability
OPT[random] $\leq m(0.5 + o(1))$ with high probability

Search:
Find x' s.t.
Val(x') $\geq$OPT [planted]

Refutation:
Certify random instances
Find an algorithm R that on input Ψ:
Output $v \geq$ OPT
If Random: with $\Omega(1)$ probability
$v \leq m(1 - \delta)$ for $\delta > \rho$

Distinguishing:
Distinguish planted vs random
with $\Omega(1)$ probability

Problems about Random d-CSPs

Search:
Find x' s.t.
Val(x') $\geq$OPT [planted]

Refutation:
Certify random instances
Find an algorithm R that on input Ψ:
Output $v \geq$ OPT
If Random: with $\Omega(1)$ probability
$v \leq m(1-\delta)$ for $\delta > \rho$

Distinguishing:
Distinguish planted vs random
with $\Omega(1)$ probability

Hardness:
- SEARCH>DISTINGUISHING
- REFUTATION>DISTINGUISHING
- DISTINGUISHING>SEARCH (see Benny's talk)

Feige's Hypothesis:
"When $m \geq \Delta n$ for a constant Δ, then there is no polynomial time refutation for random 3-SAT"
- Exist P such that best known algorithms subexponential when $m = n^{1+\Omega(1)}$ (even $m = n^{d/2-\epsilon}$)

Building PRGs from CSP

High level idea: Use an appropriate CSP to build a PRG, constant $d \geq 3$, $m \geq n^{1+\Omega(1)}$

Issue 1: CSP where distinguishing success is cryptographically SMALL

Random H do not satisfy required expansion properties with probability $\frac{1}{n^{O(1)}}$

For example, $S_1 = S_2$ with noticeable probability, and y_1, y_2 might be correlated.

Reasonable to expect SMALL probability if Graph is "nice".

Issue 2: Which predicate to use?

$d-$XOR, as hard as any $d-$CSP.

One predicate to rule them all: d-XOR

Consider $P(x_1, \ldots, x_k)$ there exists $S \subseteq [k]$ with $|S| = d$ such that:

$$\left| \mathbb{E}_{x \in \{0,1\}^k} P(x_1, \ldots, x_k) \oplus_{i \in S} x_i - \frac{1}{2} \right| \geq 2^{-k/2}$$

Can transform planted instance with $m(1 - \rho - o(1))$ satisfied constraints
to a $d-$XOR instance with $m(0.5 + 2^{-k/2} - \rho - o(1))$ satisfied constraints

Strong Refutation for $d-$XOR$\implies$weak refutation for P

Random $d-$XOR

Long history of study. Let's say $m = n^{d/2-\epsilon}$

CSP Algorithms:

- Sum-of-Squares: [G 01, S08, OW14, AOW 15, KMOW 17]
- Statistical Query Model: [FPV 15]
- Restricted models (such as AC^0 circuits, myopic models): [ABR 12, App 15]

Runtime: $2^{n^{\Omega_d(\epsilon)}}$

Does not care about noise (any d wise independent predicate suffices)

Threshold behavior: Easily broken when $m = \tilde{\Omega}(n^{d/2})$

First candidate: Use noiseless d XOR!

Will avoid these attacks for $m = n^{d/2-\epsilon}$

Problems due to lack of noise: Algebra strikes

$$P(x_1, \ldots, x_d) = x_1 \oplus \ldots \oplus x_d$$

Equations are non-noisy. Gaussian elimination can just invert. Prone to Algebra.

Didn't apply to CSPs because of "noise".

Mimic CSP noise.

Idea: Adding Non-Linearity [MST 03]:

$$P(x_1, \ldots, x_{2d}) = x_1 \oplus \ldots \oplus x_d \oplus \mathrm{NL}(x_{d+1}, \ldots, x_{2d})$$

Examples of NL: AND, OR, Majority….

Algebraic Attacks

$$P(x_1, \ldots, x_{2d}) = x_1 \oplus \ldots \oplus x_d \oplus \mathsf{NL}(x_{d+1}, \ldots, x_{2d})$$

Polynomial time CSP algorithms fail even when $m = n^{d/2-\epsilon}$

Question: How to choose, NL, to prove security against Linear Algebra?
We need $m = n^{1+\Omega(1)}$ but preferably we'd like to support $m = n^{\Omega(d)}$.
Ideally if $n^{d/2}$ possible?

Types of Algebraic Attacks

Algebraic Attacks

Linear Bias [CM01, MST 03,…]

Goal: Find Test $\subseteq [m]$ such that $\bigoplus_{i \in \text{Test}} P(x_{S_i})$ is biased.

Polynomial Calculus [AL 16]

Goal: Refutations via high degree algebraic manipulations

$f_{H,P}$ is small bias generator, $\forall \text{Test} \subseteq [m]$

$$\left| \mathbb{E}_x[\bigoplus_{i \in \text{Test}} y_i] - 0.5 \right| \leq 2^{-n^{\Omega(1)}}$$

Prove algebraic refutation lower-bounds.

Linear Attacks: Choice of NL is important

Recall: $f_{H,P}$ is secure against linear attacks if (small bias generator),

$$\forall \text{Test} \subseteq [m] \left| \mathbb{E}_x[\oplus_{i \in \text{Test}} y_i] - 0.5 \right| \leq 2^{-n^{\Omega(1)}}$$

$$P(x_1, \ldots, x_{2d}) = x_1 \oplus \ldots \oplus x_d \oplus \text{NL}(x_{d+1}, \ldots, x_{2d}) \quad m = n^{\Omega(d)}$$

Proofs exploit structure of NL and expansion of the graph in a crucial manner.

Linear Attacks: How to Choose NL?

Example:

$P(x_1, \ldots, x_{2d}) = x_1 \oplus \ldots \oplus x_d \oplus \mathsf{NL}(x_{d+1}, \ldots, x_{2d})$

Arbitrary NL? Partially yes.

[ABR 12]: $d \geq 3$ and arbitrary NL $\Longrightarrow$ security for $m = n^{1.25-\epsilon}$

If NL is degree c, no security when $m \geq n^c$.

Question: Large degree? What about NL $= x_{d+1} \ldots x_{2d}$?

Large degree does not imply small Bias [AL 16]

Recall:

$$P(x_1, \ldots, x_{2d}) = x_1 \oplus \ldots \oplus x_d \oplus x_{d+1} \cdots x_{2d}$$

Broken by linear attacks when $m = n^{2.1}$ (independent of d)

Step 1: Collect $t = \Omega(n^{1.1})$ outputs $y_1, \ldots, y_t$ where $y_i = P(x|_{S_i})$ and

$$S_i = \{i_1, \ldots, i_d, 1, i_{d+2}, i_{d+3}, \ldots, i_{2d}\}$$

$$y_i = x_{i_1} \oplus \ldots \oplus x_{i_d} \oplus x_1 x_{i_{d+2}} \ldots x_{i_{2d}}$$

Step 2: If $x_1 = 0$ (w.p. 0.5) then, becomes a linear equation in rest of the variables. Solve for x

What Criteria is Needed for Small Bias?

r-Bit-Fixing degree needs to be high.
r-Bit-Fixing degree (P)= e if minimum degree of P for any fixing of r bits is e

E.g. 1-Bit-Fixing degree of P with $NL = x_{d+1}x_{d+2}\ldots x_{2d}$ is 1.

Thm [AL 16]: If r-bit fixing degree of P is e, then $f_{H,P}$ Broken by linear attacks $m > n^{r+e}$.

Thm [AL 16]: If r-bit fixing degree of P is e where, $r, e = \Omega(d)$ then, $f_{H,P}$ is small bias generator when $m = n^{\Omega(d)}$.

Conclusion: Use NL with large bit fixing degree such as majority $d/4$ bit fixing degree $d/4$.

A huge gap between attacks, and what we can prove secure.

Algebraic Refutation Attacks [AL 16]

Is Small Bias enough to argue security?
No!

What if $P = \oplus_{i \in [d]} x_i \oplus \mathrm{NL}(x_{d+1}, \ldots, x_{2d})$ has large bit fixing degree but,

Can find low degree e Q, R such that:

Minimum such: rational degree

$$PQ = R$$

$$OR(x_1, x_2, \ldots, x_d) \cdot x_1 = x_1$$

Form equations: $y_i Q(x|_{S_i}) = R(x|_{S_i})$

Thm [AL 16]: Broken when $m = n^e$; Use linearization/polynomial calculus refutations

Algebraic Refutation Attacks [AL 16]

How to build counterexamples?

Observation: Use OR

$$P(x_1, \ldots, x_{d+d^2}) = x_1 \oplus \ldots \oplus x_d \oplus \mathrm{OR}_{i \in [d]}(\oplus_{j \in [d]} x_{d+(i-1)d+j})$$

$d - 1$ bit fixing degree d

Thm [AL 16]: $f_{H,P}$ is small bias generator when $m = n^{\Omega(d)}$.

But broken when $m \geq n^2$

[AL 16]: For any predicate with Rational degree e, $f_{H,P}$ secure when $m \leq n^{\Omega(e)}$.

Gap exists between attacks and lower bounds

Summary

OPTIMAL PREDICATE

$$P(x_1, \dots, x_{2d}) = x_1 \oplus \dots \oplus x_d \oplus \mathsf{NL}(x_{d+1}, \dots, x_{2d})$$

1. d wise-independence, CSP attack fails when $m < n^{d/2-\epsilon}$

2. NL must have high bit fixing and Rational Degree

High rational degree $\implies$ high bit fixing degree.

Use Majority. Rational degree of $\lceil d/2 \rceil$

$$P(x_1, \dots, x_{2d}) = x_1 \oplus \dots \oplus x_d \oplus \mathsf{MAJ}(x_{d+1}, \dots, x_{2d})$$

No known heuristic attacks: $m = n^{d/2-\epsilon}$

Provable bounds much weaker: $m \approx n^{d/38}$

Open Questions

Formal connections between Random CSP and PRGs

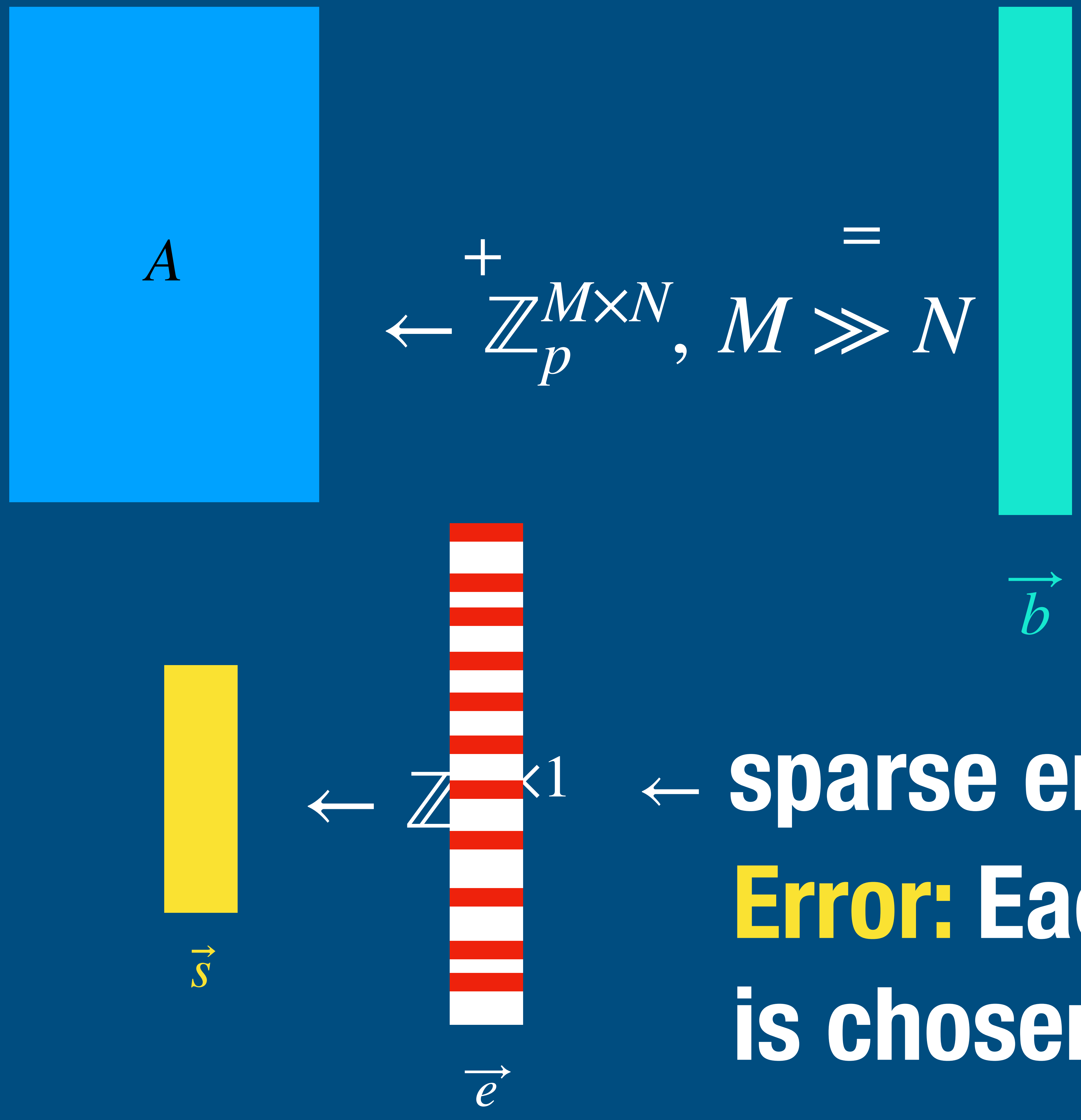
PRGs as secure as CSPs?

Tighter Characterization?

Fill differences between attacks and proofs?

Other attacks?

Learning Parity with Noise [Hamming 1950, BFKL 94, IPS 09]



mod p

$$M \gg N$$

$$A \leftarrow \mathbb{Z}_p^{M\times N}$$

$$\vec{s} \leftarrow \mathbb{Z}_p^{N}$$

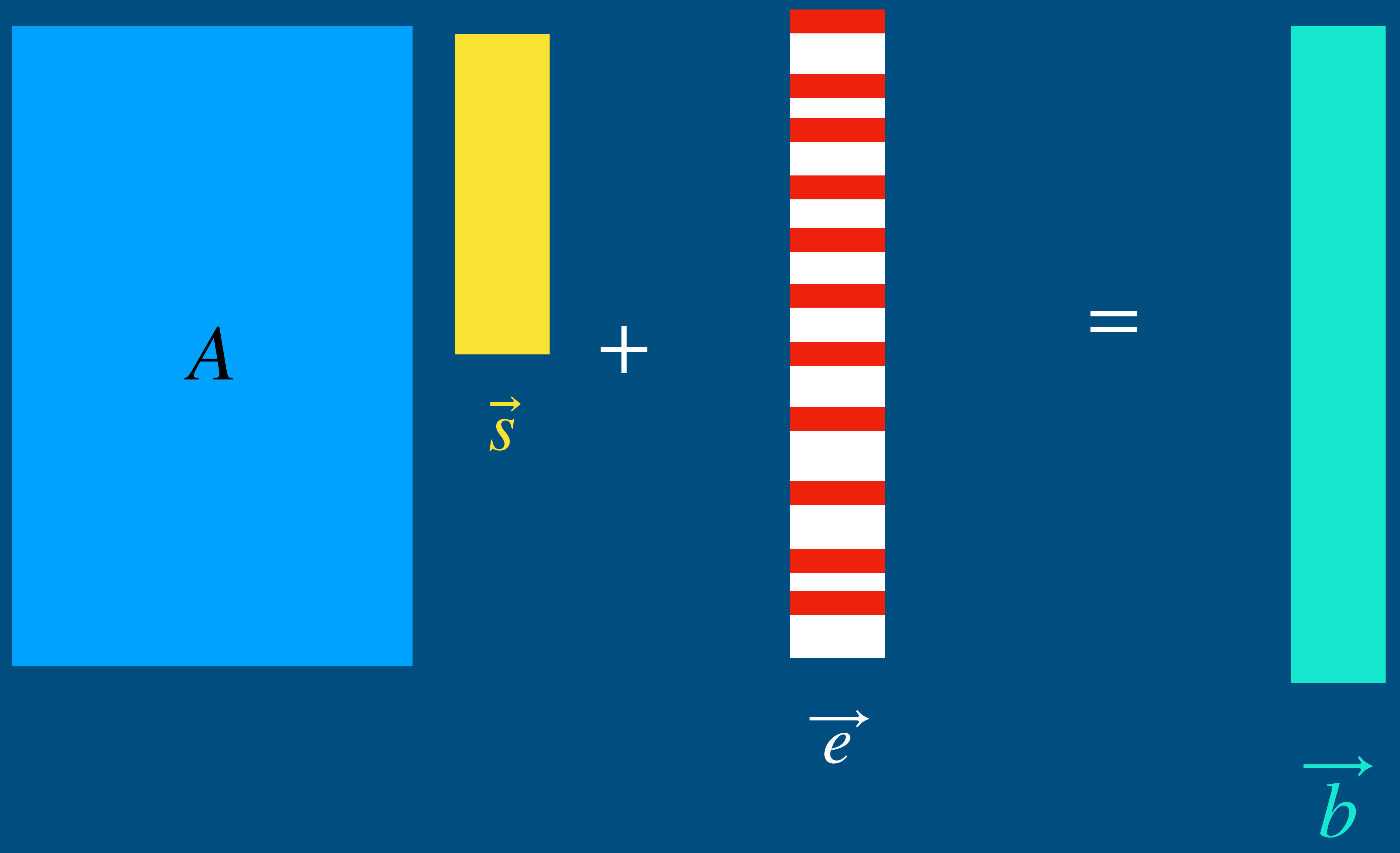
$$e_i := \begin{cases} e_i \leftarrow \mathbb{Z}_p & \texttt{Pr.}\ \rho \\ e_i = 0 & \texttt{Pr.}\ 1-\rho \end{cases}$$

$$\rho \in (0, 0.99)$$

← **sparse error**

Error: Each coordinate e_i is chosen sparsely.

Learning Parity with Noise [Hamming 1950, BFKL 94, IPS 09]



$\bmod p$

$$M \gg N$$

$$A \leftarrow \mathbb{Z}_p^{M \times N}$$

$$\vec{s} \leftarrow \mathbb{Z}_p^N$$

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

$$\rho \in (0, 0.99)$$

(N, M, ρ, p)-Search LPN: Decoding problem. Find $\vec{s}$.

Unique when $M = O_\rho(N)$.

(N, M, ρ, p)-Decision LPN: Distinguish between (A, b) and (A, u).

Use in Cryptography [BFKL 93, IPS 09]

$\mathbb{F}_2$ is used more (but $\mathbb{F}_p$ is also common). Typically samples are $M = N^{\Omega(1)}$

$\rho = O\left(\frac{1}{N}\right)$, broken in polynomial time

$\rho = 1$, perfectly indistinguishable

Typical Cryptography

High—noise
$\rho = \Theta(1)$

Low—noise
$\rho = \frac{1}{N^{\delta}}$ for $\delta \in (0,1)$

Extremely low—noise
$\rho = \frac{\log^{\Omega(1)} N}{N}$

For iO $\rho = \frac{1}{N^{0.00001}}$

For Public-Key Encryption $\rho = O(N^{-0.5})$

Search vs Distinguishing

Claim: Distinguishing > Decoding/Search [BFKL 94, Reg 05, MM 10, MP 13]

Simple approach: Using Distinguisher to guess bits of secret $\vec{s}$

Each LPN sample: $\vec{a} = (a_1, \ldots, a_N), \langle \vec{a}, \vec{s} \rangle + e \mod 2$

$$\vec{a}', \langle \vec{a}, \vec{s} \rangle + e - a_1 s_{1,guess}$$

$$\vec{a}' = (a_2, \ldots, a_N)$$

If guess is correct, we get LPN samples in dimension $N-1$, else we get random.

Reduction run time/sample complexity poly($p, \frac{1}{\epsilon}, N, M$)

Sample preserving [MM 10]

Credit: Geffroy Couteau

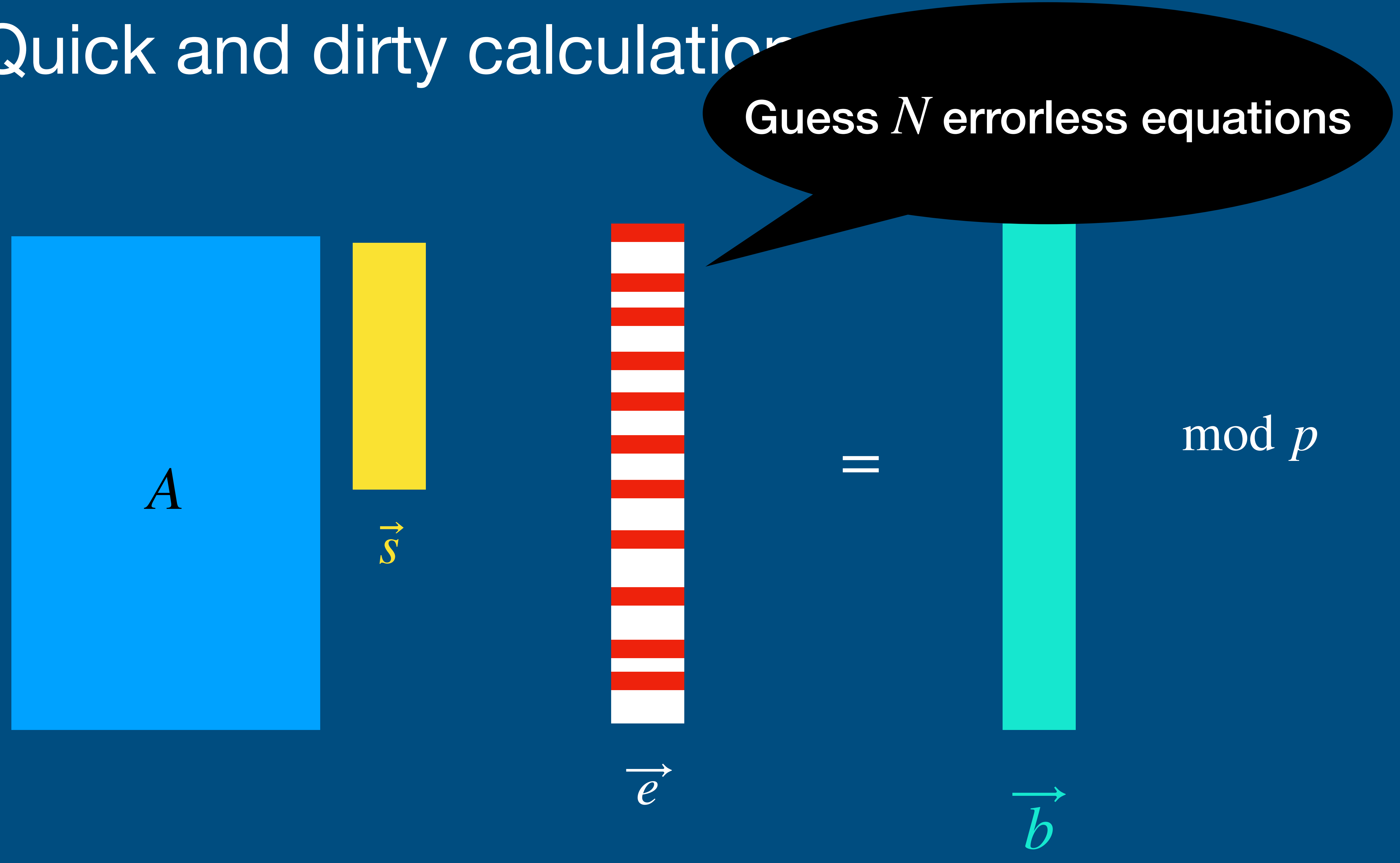
Security of LPN over Large Fields

A tremendous number of attacks on LPN has been published in the literature

- **Statistical Decoding Attacks**
  - Jabri's attack [ICCC:Jab01]
  - Overbeck's variant [ACISP:Ove06]
  - FKI's variant [Trans.IT:FKI06]
  - Debris-Tillich variant [ISIT:DT17]
- **Information Set Decoding Attacks**
  - Prange's algorithm [Prange62]
  - Stern's variant [ICIT:Stern88]
  - Finiasz and Sendrier's variant [AC:FS09]
  - BJMM variant [EC:BJMM12]
  - May-Ozerov variant [EC:MO15]
  - Both-May variant [PQC:BM18]
  - MMT variant [AC:MMT11]
  - Well-pooled MMT [CRYPTO:EKM17]
  - BLP variant [CRYPTO:BLP11]
- **Classical Techniques**
  - Low-deg approx [ITCS:ABGKR17]
- **Gaussian Elimination attacks**
  - Standard gaussian elimination
  - Blum-Kalai-Wasserman [J.ACM:BKW03]
  - Sample-efficient BKW [A-R:Lyu05]
  - Pooled Gauss [CRYPTO:EKM17]
  - Well-pooled Gauss [CRYPTO:EKM17]
  - Leviel-Fouque [SCN:LF06]
  - Covering codes [JC:GJL19]
  - Covering codes+ [BTV15]
  - Covering codes++ [BV:AC16]
  - Covering codes+++ [EC:ZJW16]
- **Other Attacks**
  - Generalized birthday [CRYPTO:Wag02]
  - Improved GBA [Kirchner11]
  - Linearization [EC:BM97]
  - Linearization 2 [INDO:Saa07]
  - Low-weight parity-check [Zichron17]

How to Solve LPN: Guessing Algorithm

Quick and dirty calculatio



$\Pr[N$ equations are errorless$] = (1-\rho)^N$

$\rho = \Omega(1)$	$2^{O(N)}$
$\rho = 1/N^{\delta}$	$2^{O(N^{1-\delta})}$
$\rho = \log^2 N/N$	$2^{O(\log^2 N)}$

Expected run time= $(1-\rho)^{-N}\text{poly}(N)$

Blum-Kalai-Wasserman [2003]

Main Result: Can solve $\mathbb{F}_2$ LPN with constant ρ ... $O(N/\log N)$.

biased with $0.5 + (1 - \rho/2)^t$

$a_1, \langle a_1, s \rangle + e_1$

.
.
.

$a_M, \langle a_M, s \rangle + e_1$

$t = O(\sqrt{N})$

$s_1 + \sum_{i \in [m]} x_i e_i$

sparse vector $\vec{x} \in \{0,1\}^N$
Such that $\sum_i x_i a_i = (1,0,..,0)$

Modifications:
[Lyu 05] $2^{O(N/\log \log N)}$ time algorithm for $M = N^{1+\epsilon}$

Can be found whp if $M \geq 2^{O(N/\log N)}$
In time poly(m)

Open question: Algorithm for large fields?

Open Questions

- Matching result for large fields?
- Other algorithms?
- Worst-case hardness? [BLVW 19, YZ 19]
- How do LPN with various prime fields relate?