Bounded Indistinguishability for Simple Sources

Andrej Bogdanov

CUHK

K. Dinesh

CUHK

Yuval Filmus

Technion

Yuval Ishai

Technion

Avi Kaplan

Technion

Akshay Srinivasan

TIFR

Cast of Characters

$X = (X_1, \ldots, X_n), Y = (Y_1, \ldots, Y_n)$ distributions on $\{0,1\}^n$

Cast of Characters

$X = (X_1, \ldots, X_n), Y = (Y_1, \ldots, Y_n)$ distributions on $\{0,1\}^n$

X is k-wise independent if every k coordinates look uniform

Cast of Characters

$X = (X_1, \ldots, X_n), Y = (Y_1, \ldots, Y_n)$ distributions on $\{0,1\}^n$

X is k-wise independent if every k coordinates look uniform

X, Y are k-wise indistinguishable if every k coordinates look the same

Cast of Characters

$X = (X_1, \ldots, X_n), Y = (Y_1, \ldots, Y_n)$ distributions on $\{0,1\}^n$

X is k-wise independent if every k coordinates look uniform

X, Y are k-wise indistinguishable if every k coordinates look the same

X is k-wise independent if X, U are k-wise indistinguishable

⇧
uniform
distribution

Examples

Examples

Uniform distribution on even parity vectors: $(n-1)$-wise independent

Examples

Uniform distribution on even parity vectors: $(n-1)$-wise independent

Uniform distribution on subspace is $(k-1)$-wise independent,
where k is dual distance (shortest linear relation)

Examples

Uniform distribution on even parity vectors: $(n-1)$-wise independent

Uniform distribution on subspace is $(k-1)$-wise independent,
where k is dual distance (shortest linear relation)

$X = (a_1, b_1, a_1 + b_1, \ldots, a_n, b_n, a_n + b_n)$ is 2-wise independent

Examples

Uniform distribution on even parity vectors: $(n-1)$-wise independent

Uniform distribution on subspace is $(k-1)$-wise independent, where k is dual distance (shortest linear relation)

$X = (a_1, b_1, a_1 + b_1, \ldots, a_n, b_n, a_n + b_n)$ is 2-wise independent

$X|_{a_1+\cdots+a_n=0}$ and $X|_{a_1+\cdots+a_n=1}$ are $(n-1)$-wise indistinguishable

Motivation

Motivation

k-wise independence: **derandomization**

Motivation

k-wise independence: **derandomization**

k-wise indistinguishability: **secret sharing schemes**

Motivation

k-wise independence: **derandomization**

k-wise indistinguishability: **secret sharing schemes**

any r parties can recover secret

no k keys leak *any* information

Motivation

k-wise independence: **derandomization**

k-wise indistinguishability: **secret sharing schemes**

any r parties can recover secret

no k keys leak *any* information

k-wise independent secret sharing schemes use linear reconstruction

AC^0 reconstruction requires k-wise indistinguishability

Motivation

k-wise independence: **derandomization**

k-wise indistinguishability: **secret sharing schemes**

any r parties can recover secret

no k keys leak *any* information

k-wise independent secret sharing schemes use linear reconstruction
AC^0 reconstruction requires k-wise indistinguishability

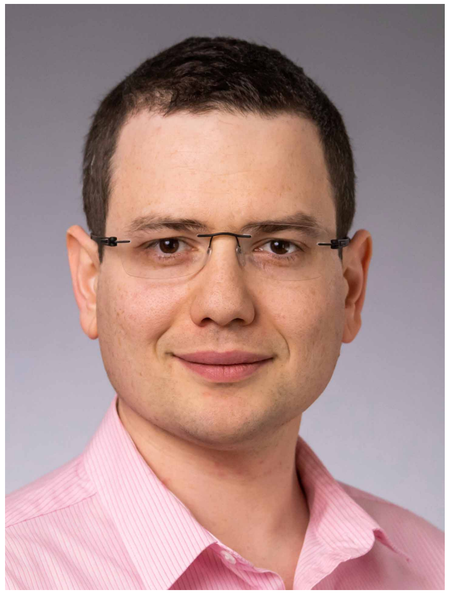
secure multiparty computation and **leakage-resilience** require share manipulation
breaks k-wise independence but not k-wise indistinguishability

Braverman for indistinguishability?

[Bogdanov–Ishai–Viola–Williamson 2016]

Braverman for indistinguishability?

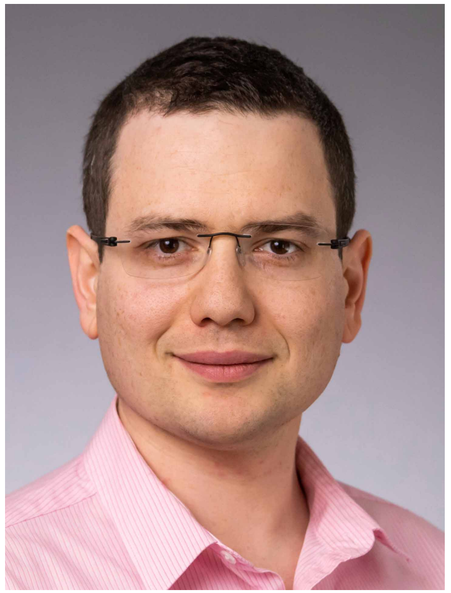
[Bogdanov–Ishai–Viola–Williamson 2016]



Braverman’s theorem:
polylog independence
fools AC^0

Braverman for indistinguishability?

[Bogdanov–Ishai–Viola–Williamson 2016]

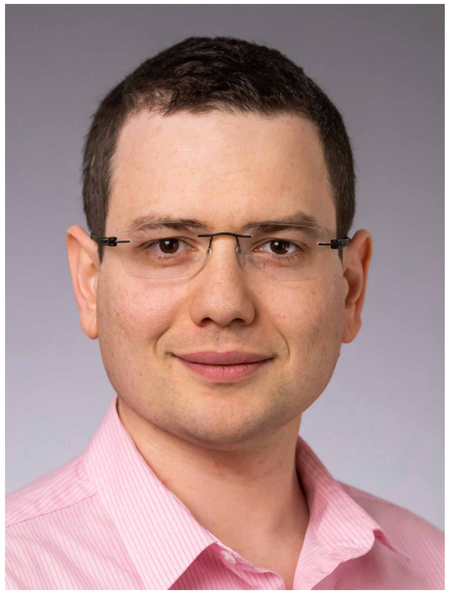


Braverman's theorem:
polylog independence
fools AC^0

"Fooling escalation"

Braverman for indistinguishability?

[Bogdanov–Ishai–Viola–Williamson 2016]

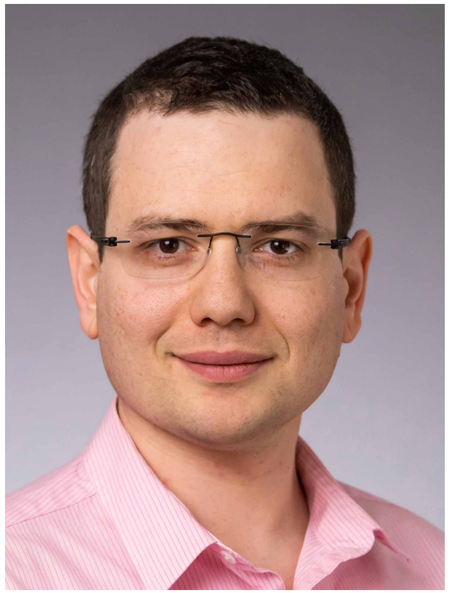


Braverman's theorem:
polylog independence
fools AC^0

"Fooling escalation"

Nisan–Szegedy:
approximate degree of OR is $\sqrt{n}$
so $\sqrt{n}$-wise indistinguishability
doesn't even fool OR!

Braverman for indistinguishability?
[Bogdanov–Ishai–Viola–Williamson 2016]



Braverman's theorem:
polylog independence
fools AC^0

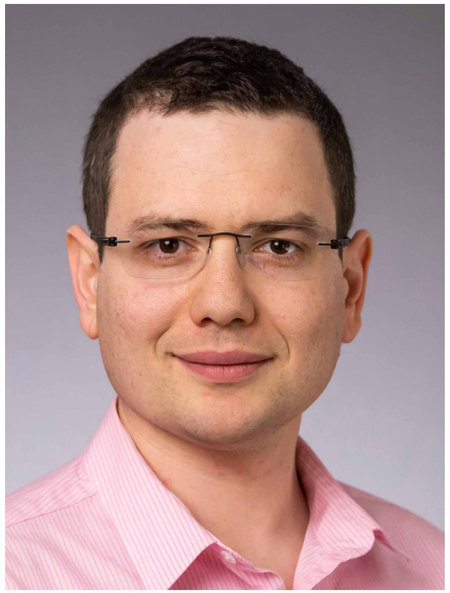
"Fooling escalation"

LP duality ⇒

Nisan–Szegedy:
approximate degree of OR is $\sqrt{n}$
so $\sqrt{n}$-wise indistinguishability
doesn't even fool OR!

Braverman for indistinguishability?

[Bogdanov–Ishai–Viola–Williamson 2016]



Does Braverman hold for polylog indistinguishable *simple* sources?

Braverman's theorem: polylog independence fools AC^0

"Fooling escalation"

LP duality ⇒

Nisan–Szegedy: approximate degree of OR is $\sqrt{n}$ so $\sqrt{n}$-wise indistinguishability doesn't even fool OR!

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Yes

Leakage-resilience of secure multiparty computation (also secure hardware etc.)

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Yes

Leakage-resilience of secure multiparty computation (also secure hardware etc.)

“Resilience escalation”

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Yes

Leakage-resilience of secure multiparty computation (also secure hardware etc.)

“Resilience escalation”

AC^0 **models realistic leakage**

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Yes

Leakage-resilience of secure multiparty computation (also secure hardware etc.)

“Resilience escalation”

AC^0 **models realistic leakage**

No

Low-complexity secret sharing

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Yes

Leakage-resilience of secure multiparty computation (also secure hardware etc.)

“Resilience escalation”

AC^0 **models realistic leakage**

No

Low-complexity secret sharing

Generating shares is simple

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Yes

Leakage-resilience of secure multiparty computation (also secure hardware etc.)

“Resilience escalation”

AC^0 models realistic leakage

No

Low-complexity secret sharing

Generating shares is simple

Secret recovery in AC^0

Motivation

Does Braverman hold for polylog-wise indistinguishable simple sources?

Yes

Win–Win!

No

Leakage-resilience of secure multiparty computation (also secure hardware etc.)

Low-complexity secret sharing

“Resilience escalation”

Generating shares is simple

AC^0 **models realistic leakage**

Secret recovery in AC^0

Simple sources

Sources that are easy to sample given iid uniform random bits $r_1, r_2, r_3, \ldots$

Simple sources

Sources that are easy to sample given iid uniform random bits $r_1, r_2, r_3, \ldots$

- local sources

Simple sources

Sources that are easy to sample given iid uniform random bits $r_1, r_2, r_3, \ldots$

- local sources
- linear sources: **linear secret sharing with easy reconstruction**

Simple sources

Sources that are easy to sample given iid uniform random bits $r_1, r_2, r_3, \ldots$

- local sources
- linear sources: **linear secret sharing with easy reconstruction**
- linear sources: **proactive secret sharing**

Simple sources

Sources that are easy to sample given iid uniform random bits $r_1, r_2, r_3, \ldots$

- local sources
- linear sources: **linear secret sharing with easy reconstruction**
- linear sources: **proactive secret sharing**
- quadratic sources: **secure multiparty computation**

Simple sources

Sources that are easy to sample given iid uniform random bits $r_1, r_2, r_3, \ldots$

- local sources
- linear sources: **linear secret sharing with easy reconstruction**
- linear sources: **proactive secret sharing**
- quadratic sources: **secure multiparty computation**

] Arise in natural crypto protocols when combining different shares

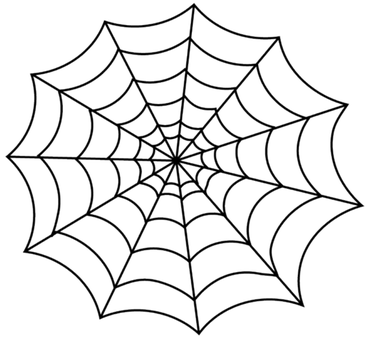
Simple sources

Sources that are easy to sample given iid uniform random bits $r_1, r_2, r_3, \ldots$

- local sources
- linear sources: **linear secret sharing with easy reconstruction**
- linear sources: **proactive secret sharing**
- quadratic sources: **secure multiparty computation**

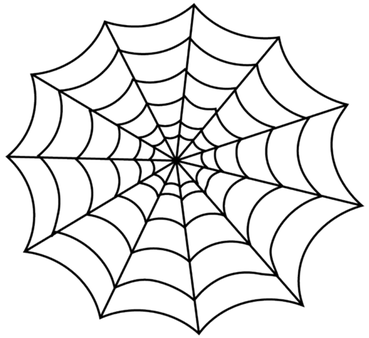
] Arise in natural crypto protocols when combining different shares

Some instances reducible to Braverman; others (e.g. LDPC codes) not



Web of Conjectures

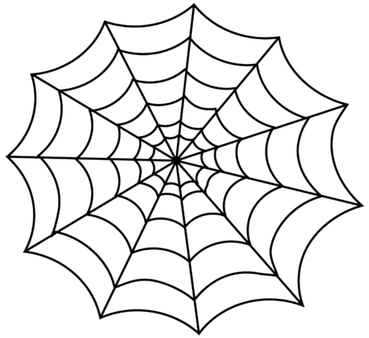
Given: class of sources (e.g. affine), class of circuits (e.g. AC^0)



Web of Conjectures

Given: class of sources (e.g. affine), class of circuits (e.g. AC^0)

Circuits cannot distinguish k-wise indistinguishable sources

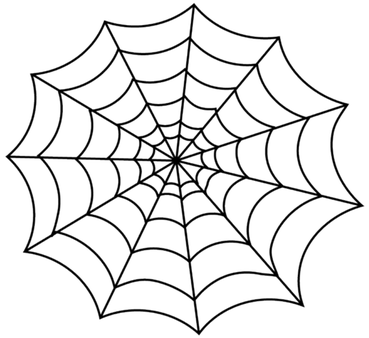


Web of Conjectures

Given: class of sources (e.g. affine), class of circuits (e.g. AC^0)

Circuits cannot distinguish k-wise indistinguishable sources

Circuits cannot distinguish k-wise indistinguishable sources of the form $X|_{r_1=0}$ and $X|_{r_1=1}$ (“cosets”)



Web of Conjectures

Given: class of sources (e.g. affine), class of circuits (e.g. AC^0)

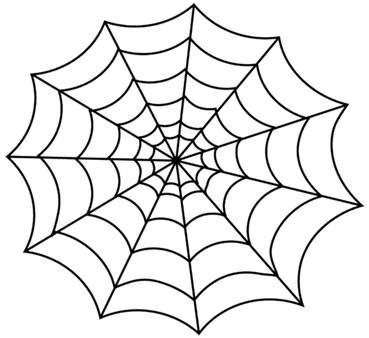
Circuits cannot distinguish k-wise indistinguishable sources

↓

Circuits cannot distinguish k-wise indistinguishable sources of the form $X|_{r_1=0}$ and $X|_{r_1=1}$ (“cosets”)

↕

No k source bits contain any information on r_1 ⇒ Circuits cannot predict r_1



Web of Conjectures

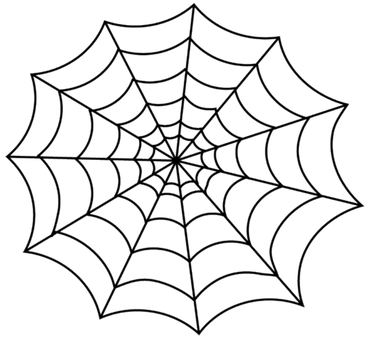
Given: class of sources (e.g. affine), class of circuits (e.g. AC^0)

Linear sources

Circuits cannot distinguish k-wise indistinguishable sources

Circuits cannot distinguish k-wise indistinguishable sources of the form $X|_{r_1=0}$ and $X|_{r_1=1}$ (“cosets”)

No k source bits contain any information on r_1 ⇒ Circuits cannot predict r_1



Web of Conjectures

Given: class of sources (e.g. affine), class of circuits (e.g. AC^0)

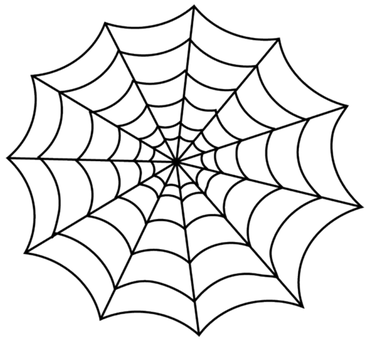
Linear sources

Circuits cannot distinguish k-wise indistinguishable sources

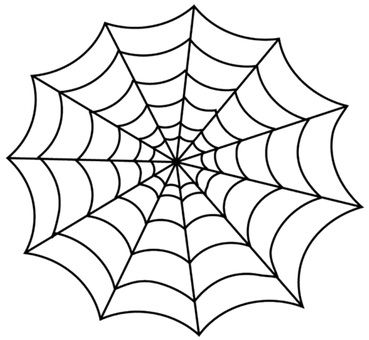
Circuits cannot distinguish k-wise indistinguishable sources of the form $X|_{r_1=0}$ and $X|_{r_1=1}$ (“cosets”)

No k source bits contain any information on r_1 ⇒ Circuits cannot predict r_1

Special case: compute parity of codewords belonging to LDPC code

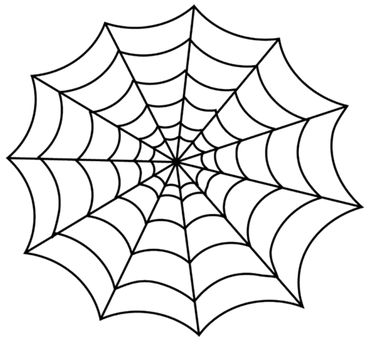


Inner Product w/ Preprocessing



Inner Product w/ Preprocessing

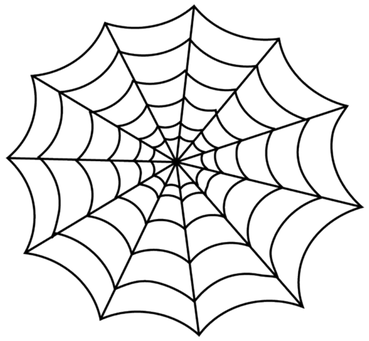
IPPP: Compute $\langle x, y \rangle$ in AC^0 given $f_i(x), g_j(y)$



Inner Product w/ Preprocessing

IPPP: Compute $\langle x, y \rangle$ in AC^0 given $f_i(x), g_j(y)$

Compute IP in $\mathsf{PH}^{\mathsf{cc}}$

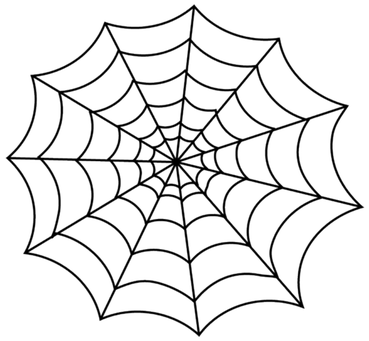


Inner Product w/ Preprocessing

IPPP: Compute $\langle x, y \rangle$ in AC^0 given $f_i(x), g_j(y)$

Compute IP in $\mathsf{PH}^{\mathsf{cc}}$

Linear IPPP: Compute $\langle x, y \rangle$ in $\mathsf{AC}^0 \circ \oplus$ (equivalently, $f_i(x), g_j(y)$ linear)



Inner Product w/ Preprocessing

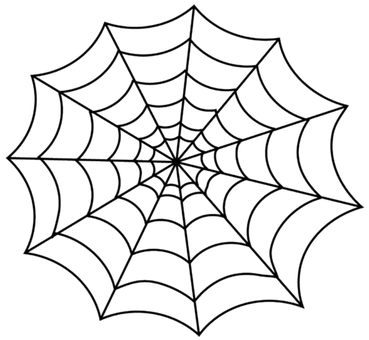
IPPP: Compute $\langle x, y \rangle$ in AC^0 given $f_i(x), g_j(y)$

Compute IP in PH^{cc}

Linear IPPP: Compute $\langle x, y \rangle$ in $\mathrm{AC}^0 \circ \oplus$ (equivalently, $f_i(x), g_j(y)$ linear)

Linear sources, AC^0 circuits

No k source bits contain any information on r_1 $\Rightarrow$ Circuits cannot predict r_1



Inner Product w/ Preprocessing

IPPP: Compute $\langle x, y \rangle$ in AC^0 given $f_i(x), g_j(y)$

Compute IP in $\mathrm{PH}^{\mathrm{cc}}$

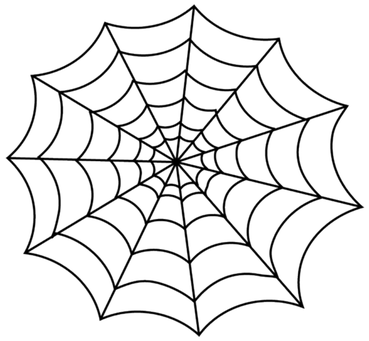
Linear IPPP: Compute $\langle x, y \rangle$ in $\mathrm{AC}^0 \circ \oplus$ (equivalently, $f_i(x), g_j(y)$ linear)

Linear sources, AC^0 circuits

No k source bits contain any information on r_1 $\Rightarrow$ Circuits cannot predict r_1

Cannot compute $\langle x, y \rangle$ for all x in AC^0 given linear $g_j(y)$

Cannot compute $\langle x, y \rangle$ in AC^0 given linear $f_i(x), g_j(y)$



Inner Product w/ Preprocessing

IPPP: Compute $\langle x, y \rangle$ in AC^0 given $f_i(x), g_j(y)$

Compute IP in $\mathrm{PH}^{\mathrm{cc}}$

Linear IPPP: Compute $\langle x, y \rangle$ in $\mathrm{AC}^0 \circ \oplus$ (equivalently, $f_i(x), g_j(y)$ linear)

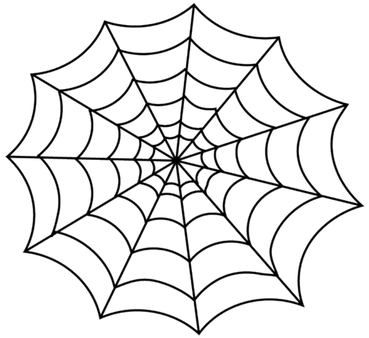
Linear sources, AC^0 circuits

No k source bits contain any information on r_1 ⇒ Circuits cannot predict r_1

Cannot compute $\langle x, y \rangle$ for all x in AC^0 given linear $g_j(y)$

Cannot compute $\langle x, y \rangle$ in AC^0 given linear $f_i(x), g_j(y)$

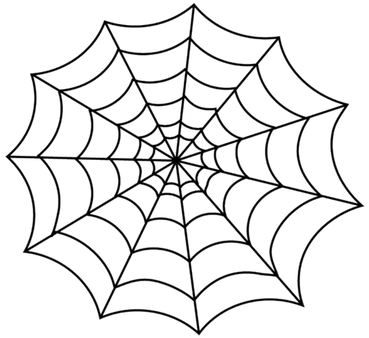
Arbitrary preprocessing



Barrier

AC^0 circuits cannot distinguish k-wise indistinguishable linear sources

Cannot compute $\langle x, y \rangle$ in $\mathrm{AC}^0 \circ \oplus$

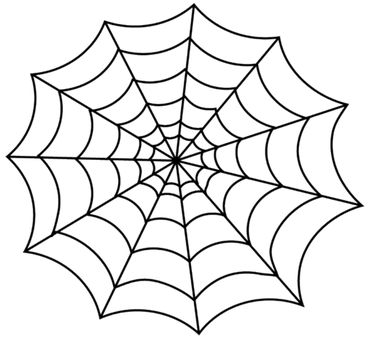


Barrier

AC^0 circuits cannot distinguish k-wise indistinguishable linear sources

Cannot compute $\langle x, y \rangle$ in $\mathrm{AC}^0 \circ \oplus$

Hard!



Barrier

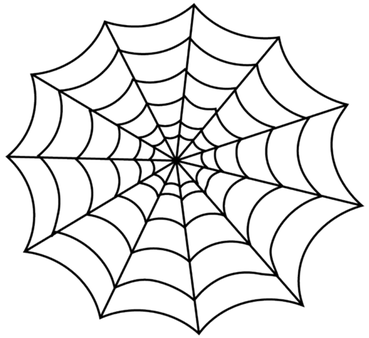
AC^0 circuits cannot distinguish k-wise indistinguishable linear sources

↓

Cannot compute $\langle x, y \rangle$ in $\mathrm{AC}^0 \circ \oplus$

Hard!

Best result: lower bound for DNF $\circ \oplus$ with error $1/\mathrm{poly}(n)$



Barrier

AC^0 circuits cannot distinguish k-wise indistinguishable linear sources

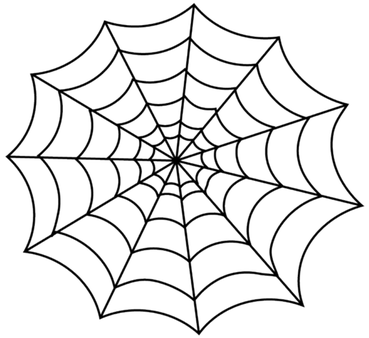
↓

Cannot compute $\langle x, y \rangle$ in $AC^0 \circ \oplus$

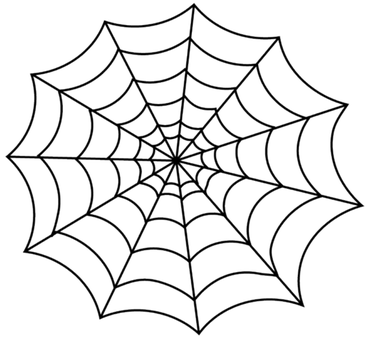
Hard!

Best result: lower bound for DNF $\circ \oplus$ with error $1/\mathrm{poly}(n)$

Concentrate on OR, decision trees, DNFs

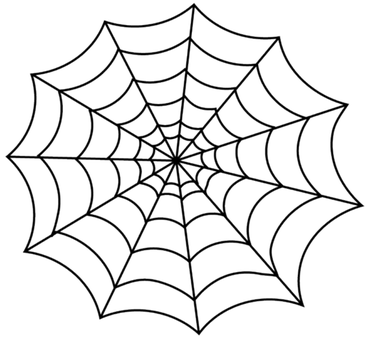


OR is interesting!



OR is interesting!

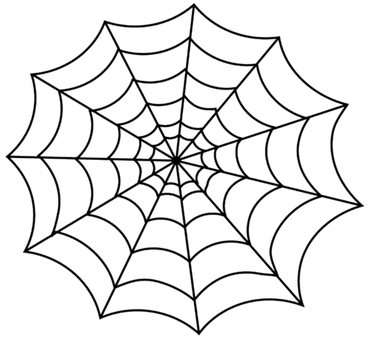
selective failure attacks



OR is interesting!

selective failure attacks

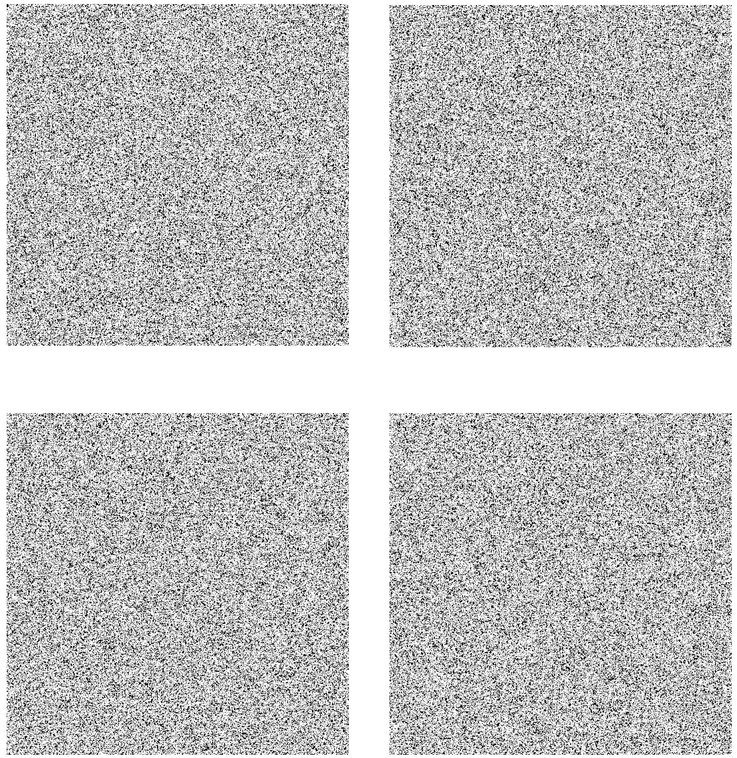
visual secret sharing
[Naor–Shamir 1994]



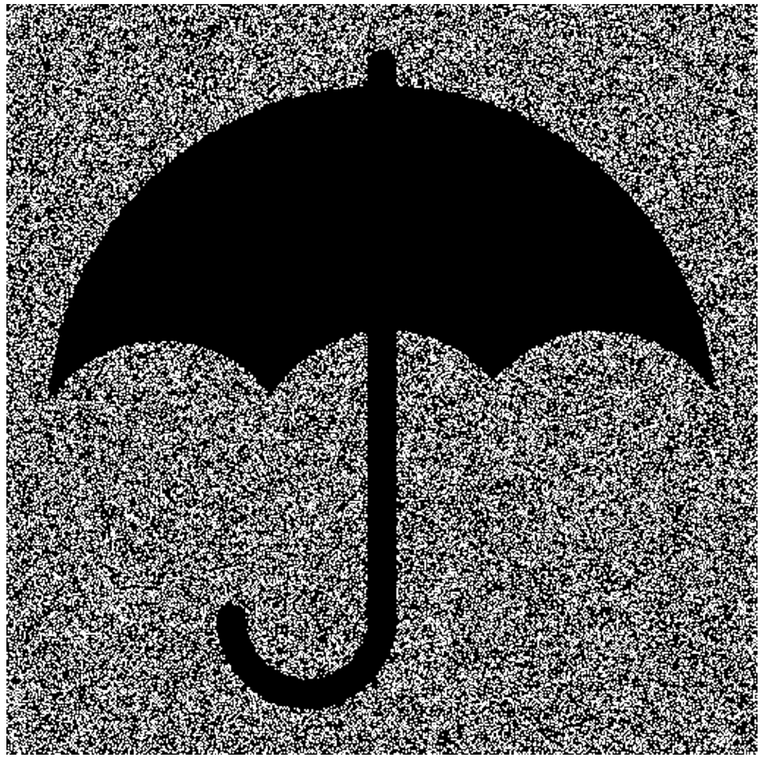
OR is interesting!

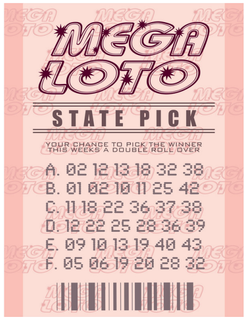
selective failure attacks

visual secret sharing
[Naor–Shamir 1994]



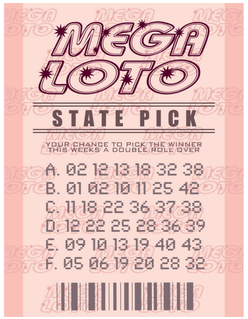
OR





Our Results

k-wise indistinguishable sources

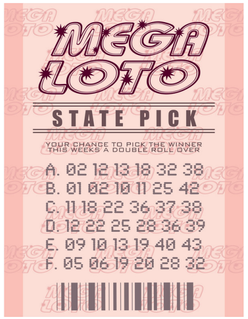


Our Results

k-wise indistinguishable sources

Constant degree/locality

Constant k fools OR



Our Results

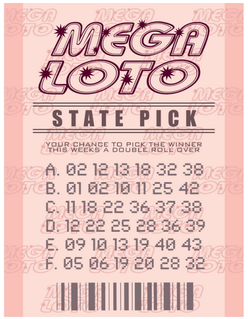
k-wise indistinguishable sources

Constant degree/locality

Constant k fools OR

Quadratic sources

$k = \mathrm{polylog}(n)$ fools decision trees



Our Results

k-wise indistinguishable sources

Constant degree/locality

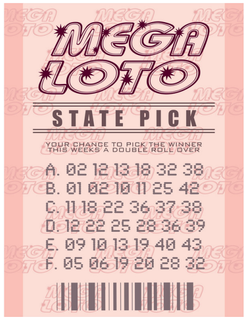
Quadratic sources

Linear sources

Constant k fools OR

$k = \mathrm{polylog}(n)$ fools decision trees

$k = \mathrm{polylog}(n)$ fools local DNFs



Our Results

k-wise indistinguishable sources

Constant degree/locality

Quadratic sources

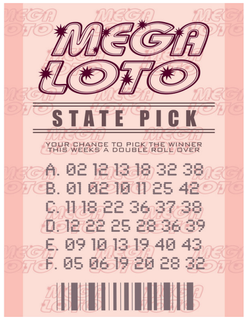
Linear sources

Constant k fools OR

$k = \mathrm{polylog}(n)$ fools decision trees

$k = \mathrm{polylog}(n)$ fools local DNFs

Suffices for *one* source to be simple!



Our Results

k-wise indistinguishable sources

Constant degree/locality

Quadratic sources

Linear sources

Degree $\log n$

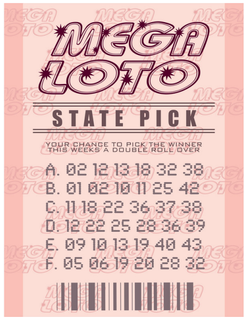
Constant k fools OR

$k = \mathrm{polylog}(n)$ fools decision trees

$k = \mathrm{polylog}(n)$ fools local DNFs

Suffices for *one* source to be simple!

OR distinguishes $k = \sqrt{n}$



Our Results

k-wise indistinguishable sources

Constant degree/locality

Quadratic sources

Linear sources

Constant k fools OR

$k = \mathrm{polylog}(n)$ fools decision trees

$k = \mathrm{polylog}(n)$ fools local DNFs

Suffices for *one* source to be simple!

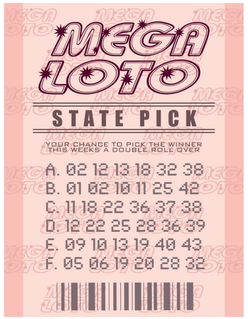
Degree $\log n$

Mixture of iid

OR distinguishes $k = \sqrt{n}$

Application to visual secret sharing


Techniques



Our Results

k-wise indistinguishable sources

Constant degree/locality

Quadratic sources

Linear sources

Constant k fools OR

$k = \mathrm{polylog}(n)$ fools decision trees

$k = \mathrm{polylog}(n)$ fools local DNFs

Suffices for *one* source to be simple!

Degree $\log n$

Mixture of iid

OR distinguishes $k = \sqrt{n}$

Application to visual secret sharing

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

Convert sources to poly size decision trees

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

Convert sources to poly size decision trees

Convert to degree $\log n$ using randomized encoding

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

Given arbitrary source X on $\{0,1\}^n$, construct mixture of iid X' on $\{0,1\}^m$

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

Given arbitrary source X on $\{0,1\}^n$, construct mixture of iid X' on $\{0,1\}^m$

Sample $x \sim X$, sample $i_1, \ldots, i_m \in [n]$, output $x_{i_1}, \ldots, x_{i_m}$

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

Given arbitrary source X on $\{0,1\}^n$, construct mixture of iid X' on $\{0,1\}^m$

Sample $x \sim X$, sample $i_1, \ldots, i_m \in [n]$, output $x_{i_1}, \ldots, x_{i_m}$

If X, Y are k-wise indistinguishable, so are X', Y'

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Starting point: $\sqrt{n}$-wise indistinguishable sources distinguished by OR

Resampling: each source is mixture of iid

Given arbitrary source X on $\{0,1\}^n$, construct mixture of iid X' on $\{0,1\}^m$

Sample $x \sim X$, sample $i_1, \ldots, i_m \in [n]$, output $x_{i_1}, \ldots, x_{i_m}$

If X, Y are k-wise indistinguishable, so are X', Y'

Distinguishing advantage of OR reduces by arbitrarily small constant

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Resampling: each source is mixture of iid

Convert sources to poly size decision trees

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Resampling: each source is mixture of iid

Convert sources to poly size decision trees

Technicality 1: Can only sample exactly from dyadic distributions

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Resampling: each source is mixture of iid

Convert sources to poly size decision trees

Technicality 1: Can only sample exactly from dyadic distributions

Sample with small failure probability

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Resampling: each source is mixture of iid

Convert sources to poly size decision trees

Technicality 1: Can only sample exactly from dyadic distributions

Sample with small failure probability

Technicality 2: Size of decision tree depends on complexity of mixture probabilities

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Resampling: each source is mixture of iid

Convert sources to poly size decision trees

Technicality 1: Can only sample exactly from dyadic distributions

Sample with small failure probability

Technicality 2: Size of decision tree depends on complexity of mixture probabilities

Turns out complexity is low enough

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Convert sources to poly size decision trees

Convert to degree $\log n$ using randomized encoding

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Convert sources to poly size decision trees

Convert to degree $\log n$ using randomized encoding

Express each output bit as sum of s many 1-leaves of decision tree

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Convert sources to poly size decision trees

Convert to degree $\log n$ using randomized encoding

Express each output bit as sum of s many 1-leaves of decision tree

Encode leaf $\ell_1 \wedge \cdots \wedge \ell_w$ as $\prod_{k=1}^{d} \left[1 + \sum_{j=1}^{w} (1 + \ell_j) r_{k,j} \right]$

OR distinguishes $\sqrt{n}$-wise indistinguishable *simple* sources

Convert sources to poly size decision trees

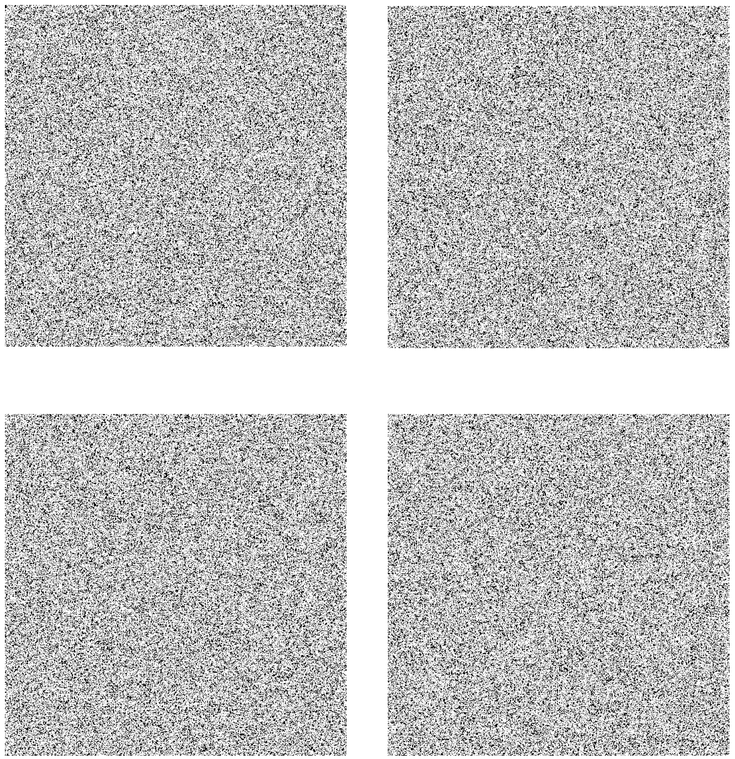
Convert to degree $\log n$ using randomized encoding

Express each output bit as sum of s many 1-leaves of decision tree

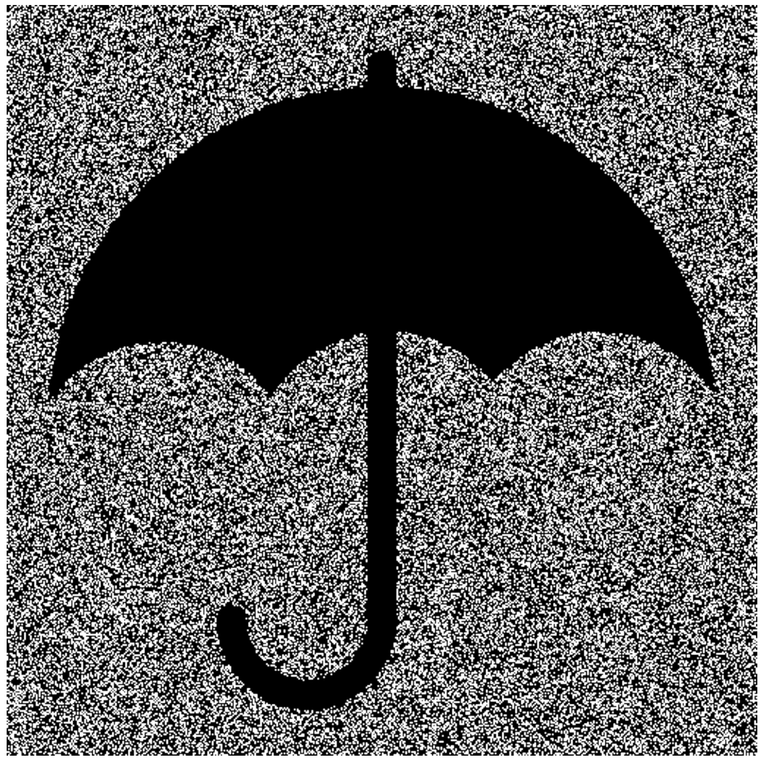
Encode leaf $\ell_1 \wedge \cdots \wedge \ell_w$ as $\prod_{k=1}^{d} \left[1 + \sum_{j=1}^{w} (1 + \ell_j) r_{k,j} \right]$

Error is 2^{-d}, so need degree $d = O(\log s)$

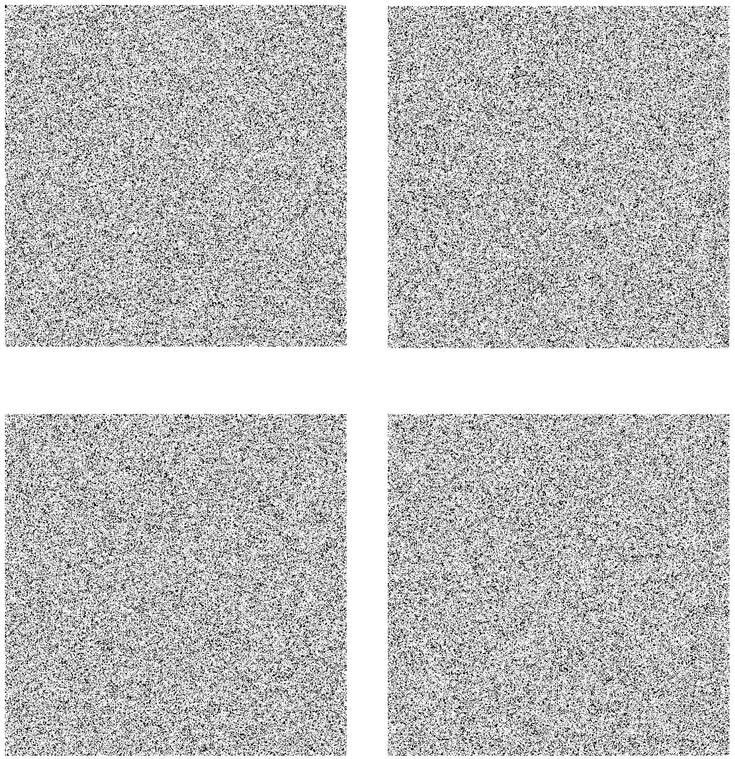
Application: Visual Secret Sharing



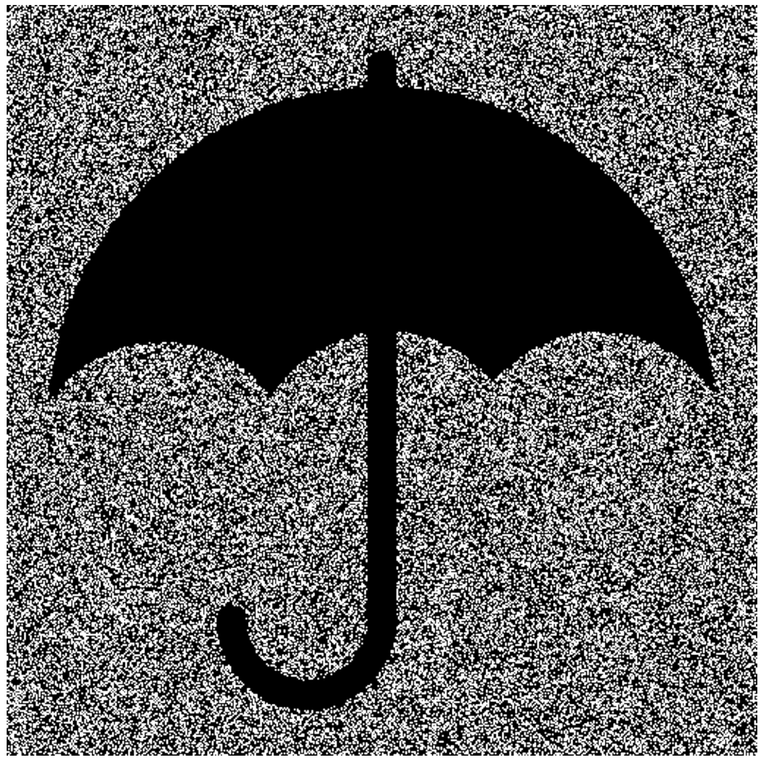
OR



Application: Visual Secret Sharing

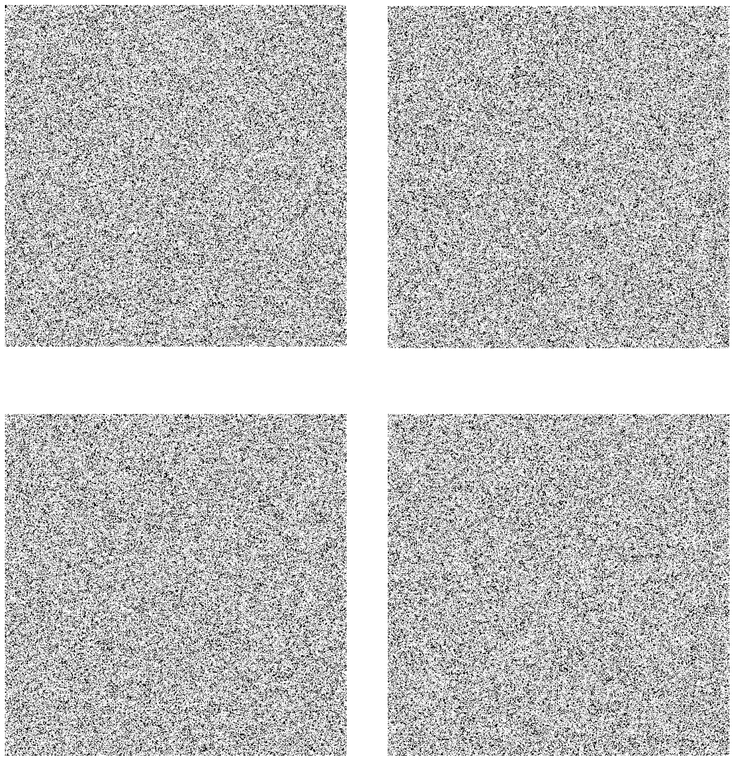


OR

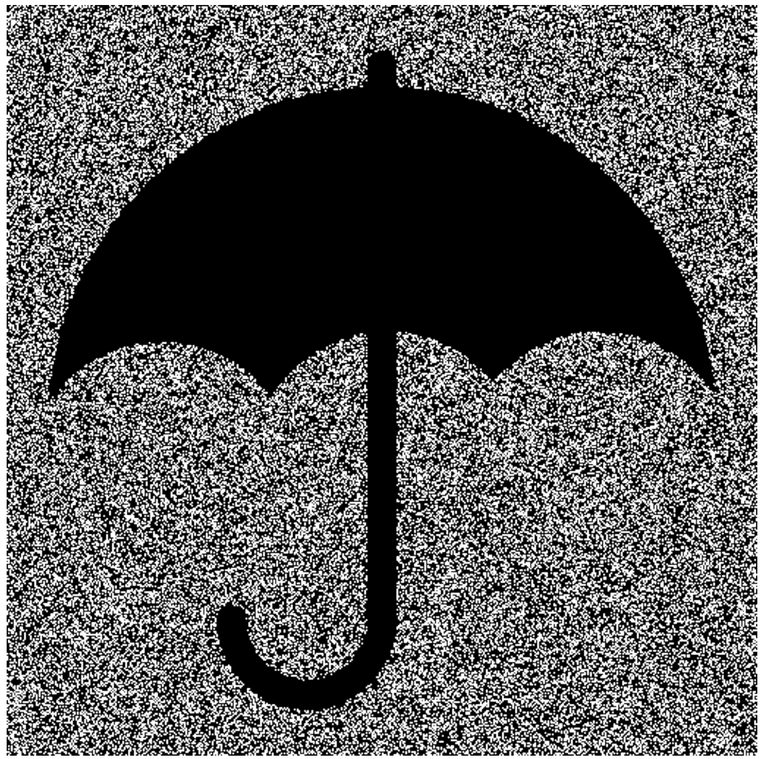


Given two distributions on biases: *X* for white, *Y* for black

Application: Visual Secret Sharing



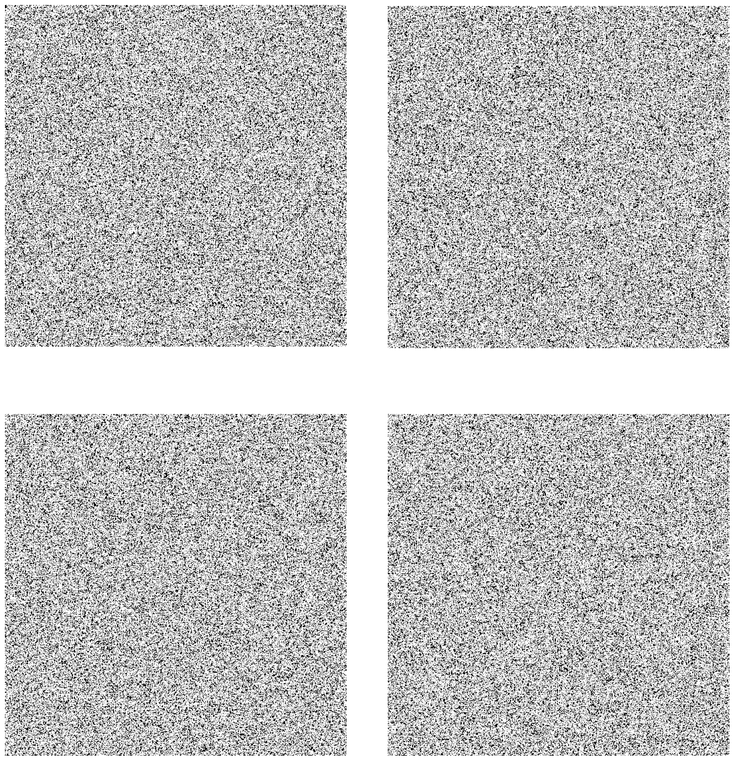
OR



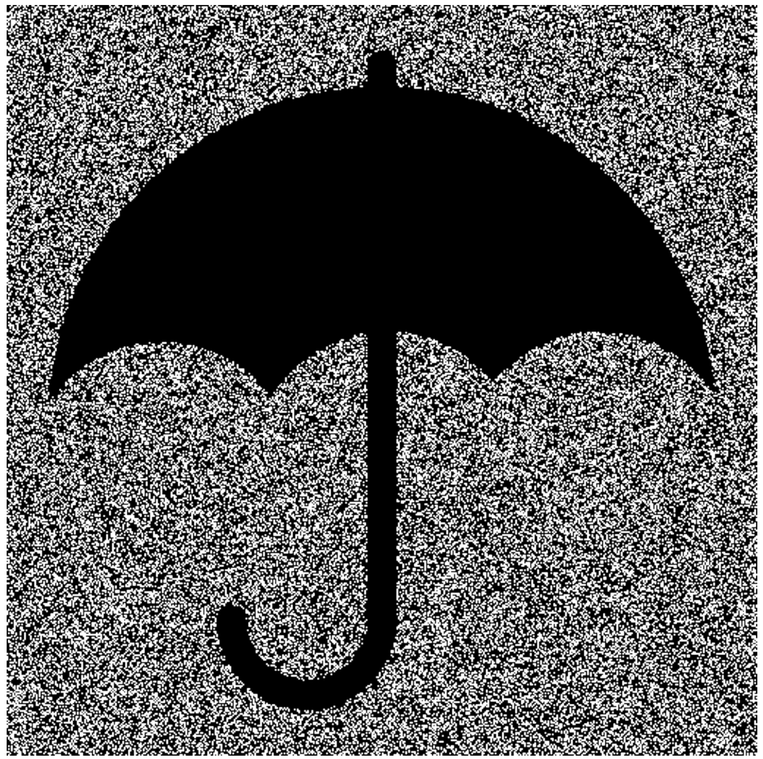
Given two distributions on biases: *X* for white, *Y* for black

Sample once and for all bias for each pixel

Application: Visual Secret Sharing



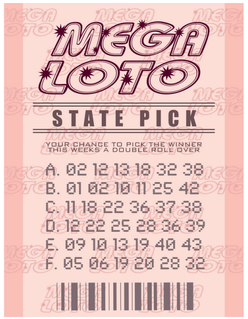
OR



Given two distributions on biases: *X* for white, *Y* for black

Sample once and for all bias for each pixel

Generate new share by sampling each pixel according to its bias



Our Results

k-wise indistinguishable sources

Constant degree/locality

Quadratic sources

Linear sources

Constant k fools OR

$k = \mathrm{polylog}(n)$ fools decision trees

$k = \mathrm{polylog}(n)$ fools local DNFs

Suffices for *one* source to be simple!

Degree $\log n$

Mixture of iid

OR distinguishes $k = \sqrt{n}$

Application to visual secret sharing

Predictability

Predictability

Source: Distribution $X = (X_1, \ldots, X_n)$ on $\{0,1\}^n$

Predictability

Source: Distribution $X = (X_1, \ldots, X_n)$ on $\{0,1\}^n$

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

Predictability

Source: Distribution $X = (X_1, \ldots, X_n)$ on $\{0,1\}^n$

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Predictability

Source: Distribution $X = (X_1, \ldots, X_n)$ on $\{0,1\}^n$

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Linear and quadratic sources are predictable

Predictability

Source: Distribution $X = (X_1, \ldots, X_n)$ on $\{0,1\}^n$

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Linear and quadratic sources are predictable

If X, Y are $\mathrm{polylog}(n)$-indistinguishable and Y is predictable then X, Y fool decision trees

Linear sources are predictable

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Linear sources are predictable

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Linear source: each X_i is linear function of $r_1, r_2, r_3, \ldots$

Linear sources are predictable

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Linear source: each X_i is linear function of $r_1, r_2, r_3, \ldots$

Linear sources are $(\log_2(1/\epsilon), \epsilon)$-predictable

Linear sources are predictable

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Linear source: each X_i is linear function of $r_1, r_2, r_3, \ldots$

Linear sources are $(\log_2(1/\epsilon), \epsilon)$-predictable

Case 1: $X_1, \ldots, X_n$ has a basis S of size at most $\log_2(1/\epsilon)$

Linear sources are predictable

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

Linear source: each X_i is linear function of $r_1, r_2, r_3, \ldots$

Linear sources are $(\log_2(1/\epsilon), \epsilon)$-predictable

Case 1: $X_1, \ldots, X_n$ has a basis S of size at most $\log_2(1/\epsilon)$

$$X|_S = 0 \Longrightarrow X = 0$$

Linear sources are predictable

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\text{polylog}(1/\epsilon), \epsilon)$-predictable

Linear source: each X_i is linear function of $r_1, r_2, r_3, \ldots$

Linear sources are $(\log_2(1/\epsilon), \epsilon)$-predictable

Case 1: $X_1, \ldots, X_n$ has a basis S of size at most $\log_2(1/\epsilon)$

$$X|_S = 0 \Longrightarrow X = 0$$

Case 2: Let S be $\log_2(1/\epsilon)$ linearly independent output bits

Linear sources are predictable

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\text{polylog}(1/\epsilon), \epsilon)$-predictable

Linear source: each X_i is linear function of $r_1, r_2, r_3, \ldots$

Linear sources are $(\log_2(1/\epsilon), \epsilon)$-predictable

Case 1: $X_1, \ldots, X_n$ has a basis S of size at most $\log_2(1/\epsilon)$

$$X|_S = 0 \Longrightarrow X = 0$$

Case 2: Let S be $\log_2(1/\epsilon)$ linearly independent output bits

$$\Pr[X|_S = 0] = \epsilon$$

Predictability and decision trees

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

If X, Y are $\mathrm{polylog}(n)$-indistinguishable and Y is predictable then X, Y fool decision trees

Predictability and decision trees

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

If X, Y are $\mathrm{polylog}(n)$-indistinguishable and Y is predictable then X, Y fool decision trees

Y is $(k, \epsilon/n)$-predictable, X, Y are $(k+1)$-indistinguishable $\implies$ X is (k, ϵ)-predictable

Predictability and decision trees

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

If X, Y are $\mathrm{polylog}(n)$-indistinguishable and Y is predictable then X, Y fool decision trees

Y is $(k, \epsilon/n)$-predictable, X, Y are $(k+1)$-indistinguishable $\implies$ X is (k, ϵ)-predictable

X, Y are (k, ϵ)-predictable and $2k$-indistinguishable $\implies$ OR has advantage $\leq \epsilon$

Predictability and decision trees

X is (k, ϵ)-*predictable* if there exists $S \subseteq [n]$ of size k such that

$$\Pr[X|_S = 0 \text{ and } X \neq 0] \leq \epsilon$$

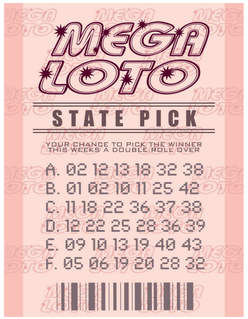
A class of sources is *predictable* if every source is $(\mathrm{polylog}(1/\epsilon), \epsilon)$-predictable

If X, Y are $\mathrm{polylog}(n)$-indistinguishable and Y is predictable then X, Y fool decision trees

Y is $(k, \epsilon/n)$-predictable, X, Y are $(k+1)$-indistinguishable $\implies$ X is (k, ϵ)-predictable

X, Y are (k, ϵ)-predictable and $2k$-indistinguishable $\implies$ OR has advantage $\leq \epsilon$

OR has advantage $\leq \epsilon \implies$ Size s decision trees have advantage $\leq s\epsilon$



Our Results

k-wise indistinguishable sources

Constant degree/locality

Quadratic sources

Linear sources

Constant k fools OR

$k = \mathrm{polylog}(n)$ fools decision trees

$k = \mathrm{polylog}(n)$ fools local DNFs

Suffices for *one* source to be simple!

Degree $\log n$

Mixture of iid

OR distinguishes $k = \sqrt{n}$

Application to visual secret sharing

Predictability for local DNFs

Predictability for local DNFs

X: n-bit source, f: n-bit function

Predictability for local DNFs

X: n-bit source, f: n-bit function

X is (k, ϵ)-*predictable* for f if there exists a depth k decision tree T with leaves labeled $0,1, \perp$ such that $T(x) \in \{f(x), \perp\}$ and $\Pr[T(X) = \perp] \leq \epsilon$

Predictability for local DNFs

X: n-bit source, f: n-bit function

X is (k, ϵ)-*predictable* for f if there exists a depth k decision tree T with leaves labeled $0,1, \perp$ such that $T(x) \in \{f(x), \perp\}$ and $\Pr[T(X) = \perp] \leq \epsilon$

w-local DNF: disjunction of functions depending on w input bits

Predictability for local DNFs

X: n-bit source, f: n-bit function

X is (k, ϵ)-*predictable* for f if there exists a depth k decision tree T with leaves labeled $0,1, \perp$ such that $T(x) \in \{f(x), \perp\}$ and $\Pr[T(X) = \perp] \leq \epsilon$

w-local DNF: disjunction of functions depending on w input bits

Linear sources are $(O(w2^w \log(1/\epsilon)), \epsilon)$-predictable for w-local DNFs

Predictability for local DNFs

X: n-bit source, f: n-bit function

X is (k, ϵ)-*predictable* for f if there exists a depth k decision tree T with leaves labeled $0,1, \perp$ such that $T(x) \in \{f(x), \perp\}$ and $\Pr[T(X) = \perp] \leq \epsilon$

w-local DNF: disjunction of functions depending on w input bits

Linear sources are $(O(w2^w \log(1/\epsilon)), \epsilon)$-predictable for w-local DNFs

If X, Y are k-indistinguishable and Y is (k, ϵ)-predictable for f then f is ϵ-fooled by X, Y

Open Questions

Open Questions

Beyond OR

Results on DNFs or AC^0? No barriers for local sources!

Open Questions

Beyond OR

Results on DNFs or AC^0? No barriers for local sources!

Application: secret-sharing with sharing in NC^0 and reconstruction in AC^0 (current best: sharing using decision trees and reconstruction using OR)

Open Questions

Beyond OR

Results on DNFs or AC^0? No barriers for local sources! **NC^0/AC^0 secret-sharing**

Web of conjectures

Given linear preprocessing $g_j(y)$, which parities of y are computable in AC^0?

Linear IPPP: not all — Our conjecture: short linear combinations — Equivalent?

Open Questions

Beyond OR

Results on DNFs or AC^0? No barriers for local sources! **NC^0/AC^0 secret-sharing**

Web of conjectures

Given linear preprocessing $g_j(y)$, which parities of y are computable in AC^0?

Linear IPPP: not all — Our conjecture: short linear combinations — Equivalent?

Conjectures about linear sources imply conjectures about quadratic sources?

Open Questions

Beyond OR

Results on DNFs or AC^0? No barriers for local sources! **NC^0/AC^0 secret-sharing**

Web of conjectures

Given linear preprocessing $g_j(y)$, which parities of y are computable in AC^0?

Linear IPPP: not all — Our conjecture: short linear combinations — Equivalent?

Conjectures about linear sources imply conjectures about quadratic sources?

More on OR

Best degree? (know: $O(\log n)$ and $\omega(1)$)

Best locality? (reduced precision implies locality 4; ruled out for mixture of iid)

Open Questions

Beyond OR

Results on DNFs or AC^0? No barriers for local sources! **NC^0/AC^0 secret-sharing**

Web of conjectures

Given linear preprocessing $g_j(y)$, which parities of y are computable in AC^0?

Linear IPPP: not all — Our conjecture: short linear combinations — Equivalent?

Conjectures about linear sources imply conjectures about quadratic sources?

More on OR

Best degree? (know: $O(\log n)$ and $\omega(1)$)

Best locality? (reduced precision implies locality 4; ruled out for mixture of iid)

Beyond Boolean

$(n-1)$-wise indistinguishable distributions over Σ^n distinguished by AC^0?

Connection to approximate degree breaks down

Open Questions

Beyond OR

Results on DNFs or AC^0? No barriers for local sources! **NC^0/AC^0 secret-sharing**

Web of conjectures

Given linear preprocessing $g_j(y)$, which parities of y are computable in AC^0?

Linear IPPP: not all — Our conjecture: short linear combinations — Equivalent?

Conjectures about linear sources imply conjectures about quadratic sources?

More on OR

Best degree? (know: $O(\log n)$ and $\omega(1)$)

Best locality? (reduced precision implies locality 4; ruled out for mixture of iid)

Beyond Boolean

$(n-1)$-wise indistinguishable distributions over $(\{0,1\}^n)^n$ distinguished by AC^0?

Application: secret sharing scheme in AC^0 with "sharp threshold"