(inefficient Prover) ZAPs from Hard-to-Invert Functions

Dana Dachman-Soled
UMD

Marshall Ball
NYU

30 Second Summary

one-way functions are NOT needed for a non-trivial notion of zero-knowledge proofs for NP

One Minute Summary

1. new primitive <u>below</u> one-way functions (OWF):

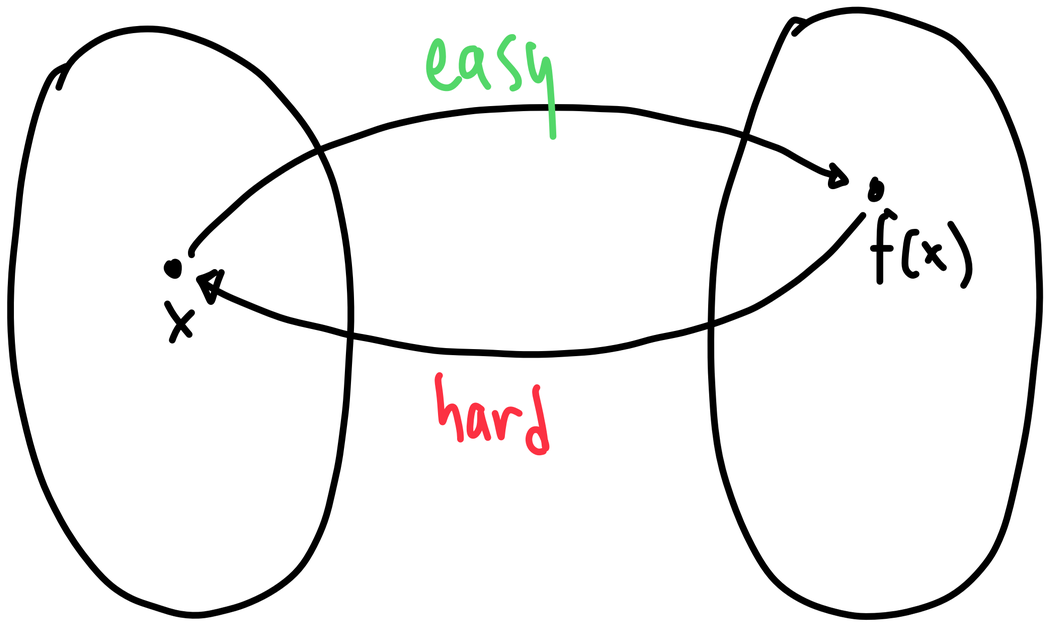
Hard-to-Invert Functions (HIF)

2. HIF $\Rightarrow$ ZAPs (2-message witness indistinguishable proofs) for NP with subexponential time prover

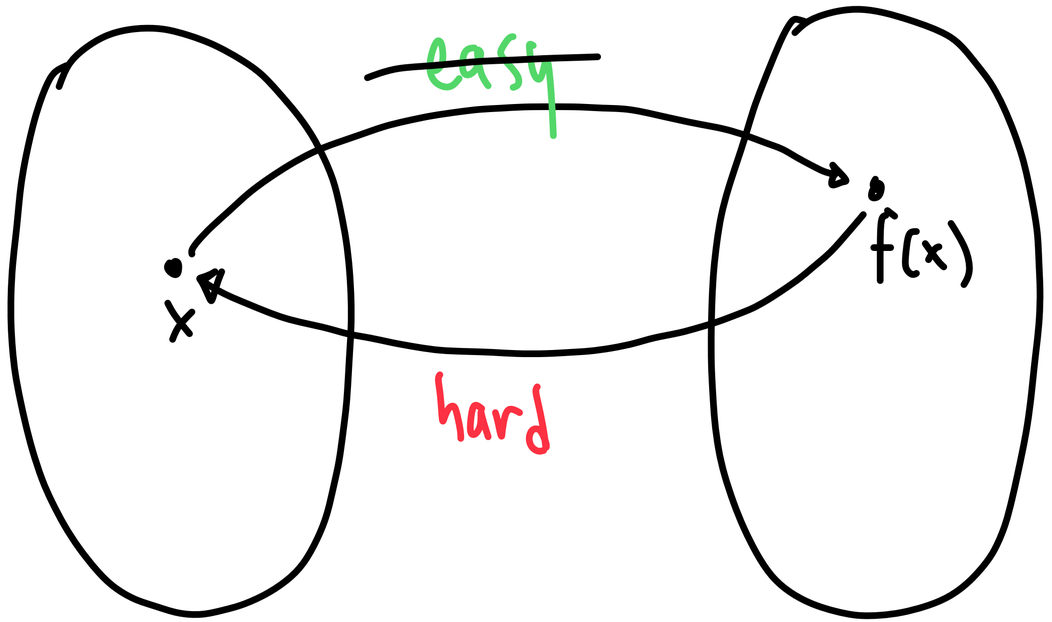
(3. HIF + more complexity assumptions $\Rightarrow$ hardness in $NP \cap coNP$, non-uniform setting)

Hard-to-Invert Function?

Recall: One-Way Function



Hard-to-Invert Function (HIF)



Can verify "$f(x) = y$" (given proof)

(i.e. $\{(x, f(x))\} \in NP$)

OWF $\Rightarrow$ HIF ($\Leftarrow$ TFUP hardness)

Zero Knowledge

Zero Knowledge

Completeness $x \in L \Rightarrow$ Verifier accepts (w.h.p.)

Zero Knowledge

Completeness $x \in L \implies$ Verifier accepts (w.h.p.)

soundness $x \notin L \implies$ Verifier rejects (w.h.p.), even if prover cheats

Zero Knowledge

Completeness $x \in L \implies$ Verifier accepts (w.h.p.)

soundness $x \notin L \implies$ Verifier rejects (w.h.p.), even if prover cheats

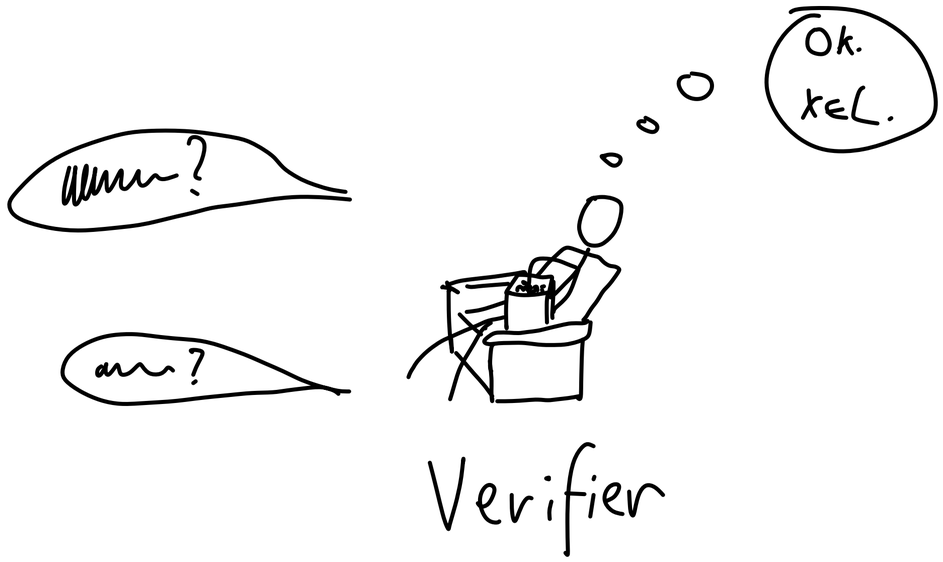
Zero-knowledge verifier only learns $x \in L$.

Zero Knowledge

Simulator

$\approx$

Prover



Completeness $x \in L \implies$ Verifier accepts (w.h.p.)

soundness $x \notin L \implies$ Verifier rejects (w.h.p.), even if prover cheats

Zero-knowledge verifier only learns $x \in L$.

($x \in L \implies$ interaction can be efficiently simulated.)

Zero Knowledge

[Goldreich Micali Wigderson '86] OWF ⇒ ZK for NP

Zero Knowledge

[Goldreich Micali Wigderson '86] OWF ⇒ ZK for NP

[Ostrovsky Wigderson '93] ¬OWF ⇒ no ZK for hard-on-average L

Zero Knowledge

Simulator Prover

[Goldreich Micali Wigderson '86] OWF ⇒ ZK for NP

[Ostrovsky Wigderson '93] ¬OWF ⇒ no ZK for hard-on-average L

[Ong Vadhan '08] ZK for L ⟺ L has instance dependent commitments

Zero Knowledge

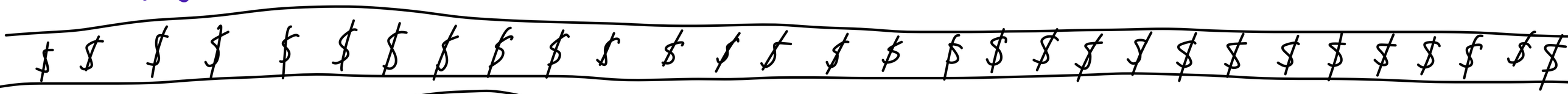
[Goldreich Micali Wigderson '86] OWF ⇒ ZK for NP

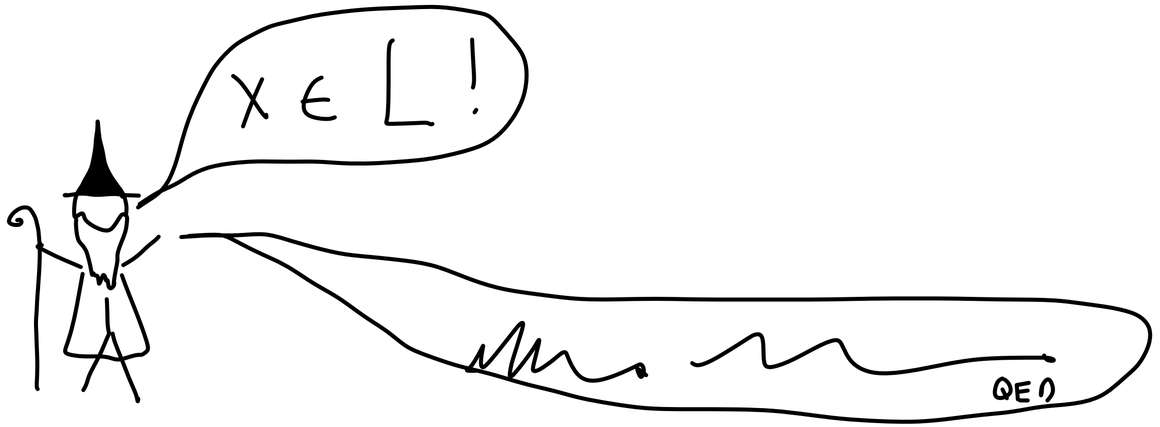
[Ostrovsky Wigderson '93] ¬OWF ⇒ no ZK for hard-on-average L

[Ong Vadhan '08] ZK for L ⟺ L has instance dependent commitments

[Hirahara Nanashima '24] if P≠NP, then ZK for NP ⇒ OWF

Non-Interactive Zero Knowledge

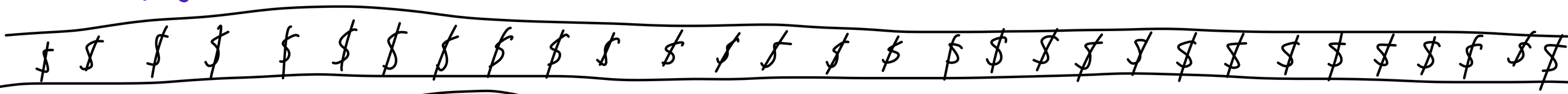


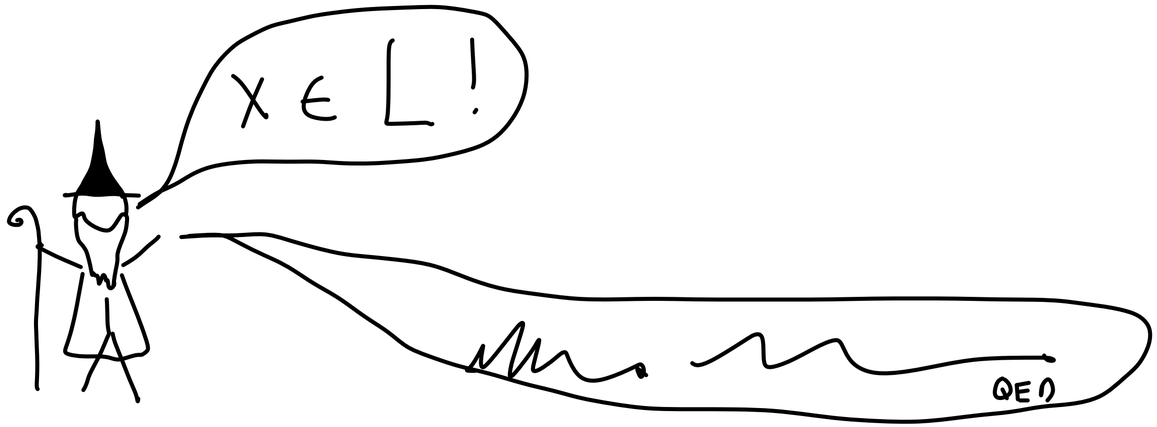


Prover

Verifier

Non-Interactive Zero Knowledge



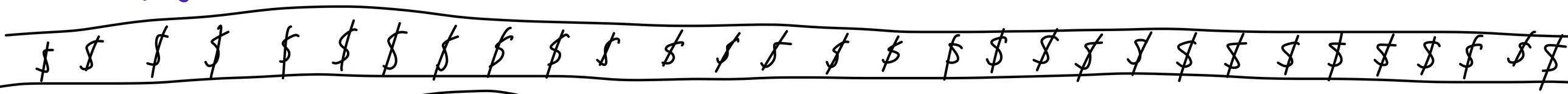


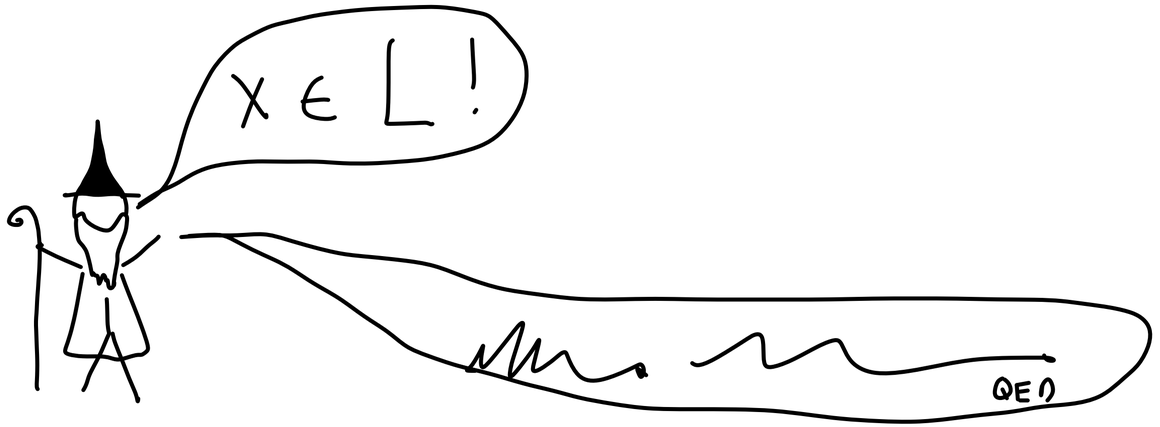
Prover

Verifier

Common "random" string (CRS) vs UNIFORM random string (URS)

Non-Interactive Zero Knowledge





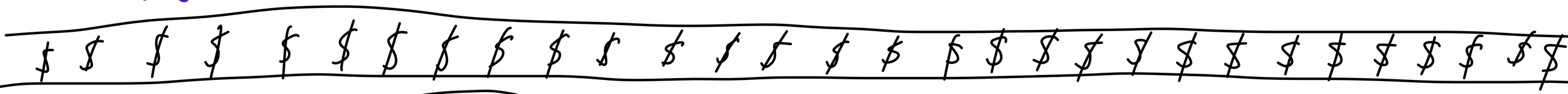
Prover

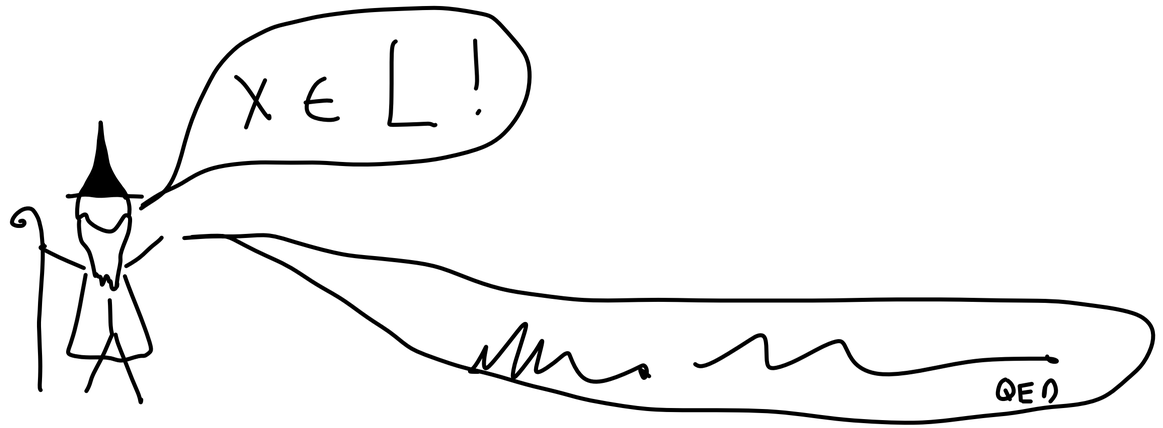
Verifier

Common "random" string (CRS) vs UNIFORM random string (URS)

[Passshelat '05] OWF ⇒ NIZK with CRS for NP (AM)

Non-Interactive Zero Knowledge





Prover

Verifier

Common "random" string (CRS)

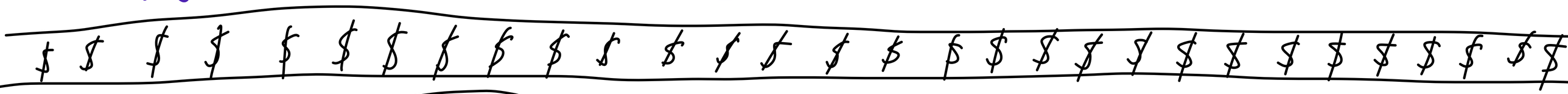
vs

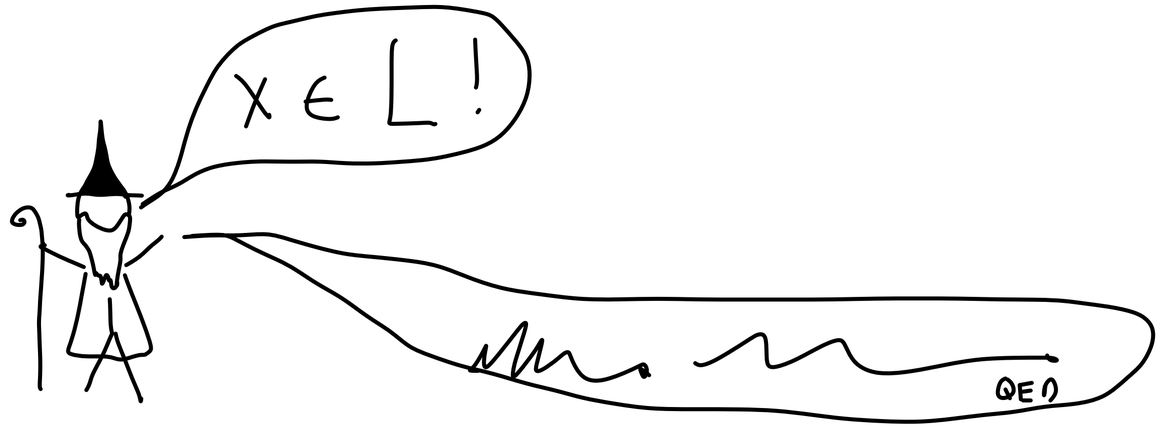
UNIFORM random string (URS)

[Passshelat '05] OWF ⇒ NIZK with CRS for NP (AM)

[FeigeLapidotShamir '99] OWP ⇒ NIZK with URS for NP (AM)

Non-Interactive Zero Knowledge





Prover

Verifier

Common "random" string (CRS) vs UNIFORM random string (URS)

[Passshelat '05] OWF ⇒ NIZK with CRS for NP (AM)

[FeigeLapidotShamir '99] OWP ⇒ NIZK with URS for NP (AM)

OPEN QUESTION:

OWF $\Rightarrow$ NIZK in URS model?

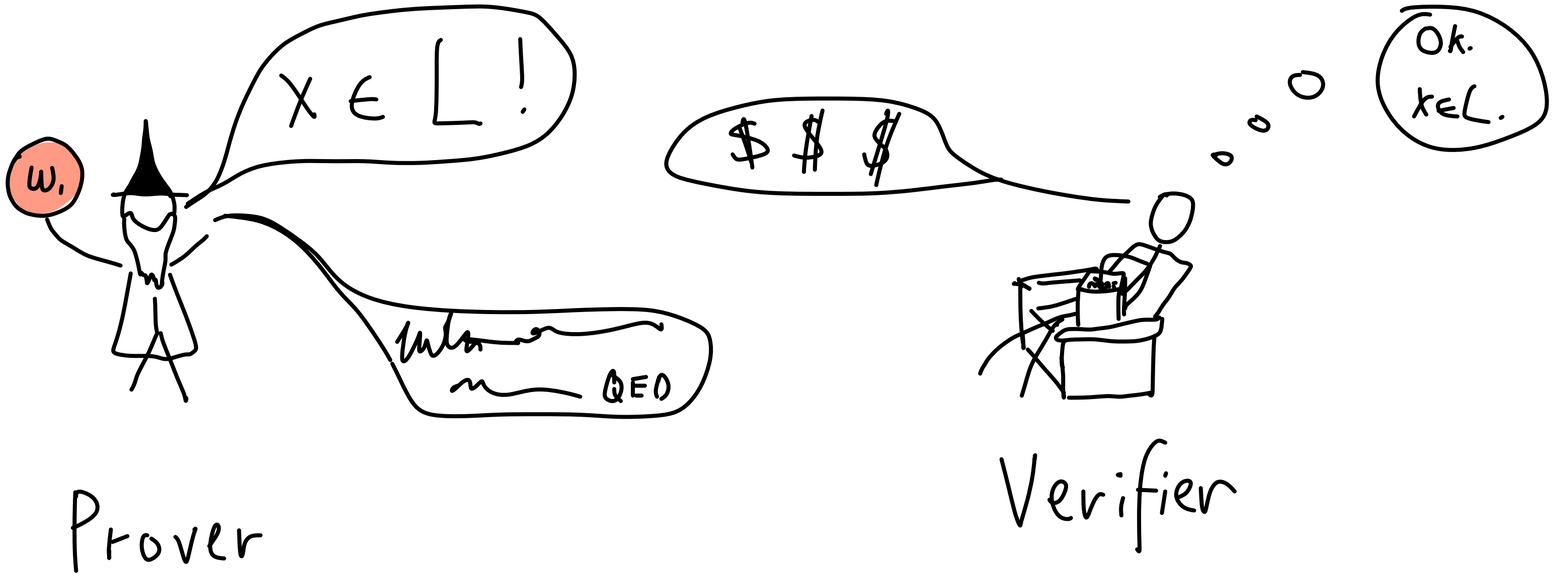
Partial progress: [Ghosal et al. '23] NIZK in ROM

this work: HIF $\Rightarrow$ ZAP (w/ inefficient prover)

(HIF < OWF) (ZAP < NIZK in URS)

ZAPs

[Dwork Naor '00]



Prover

Verifier

ZAPs

[Dwork Naor '00]

Prover

$\approx$

Prover

Verifier

WITNESS INDISTINGUISHABILITY

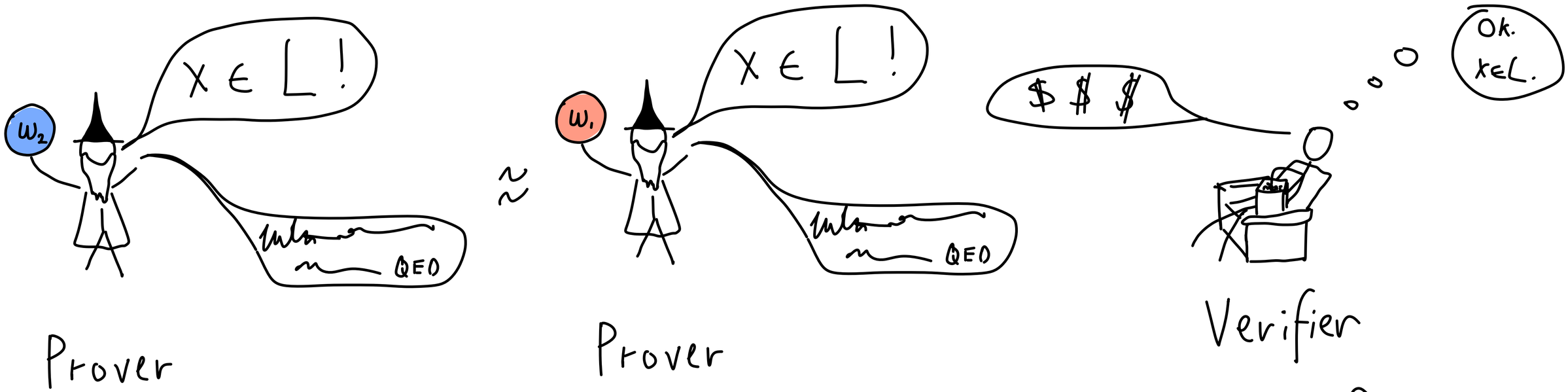
[Feige Shamir '90]

for any witnesses w_1, w_2 for x :

$$\text{Prover}(w_1) \rightleftarrows \text{Verifier} \approx \text{Prover}(w_2) \rightleftarrows \text{Verifier}$$

ZAPs

[Dwork Naor '00]



Prover ≈ Prover

Verifier

WITNESS INDISTINGUISHABILITY

[Feige Shamir '90]

for any witnesses w_1, w_2 for x :

$$\text{Prover}(w_1) \rightleftarrows \text{Verifier} \approx \text{Prover}(w_2) \rightleftarrows \text{Verifier}$$

NOTE: <u>trivial</u> if witnesses unique or prover <u>arbitrarily</u> inefficient (find & send 'canonical' witness)

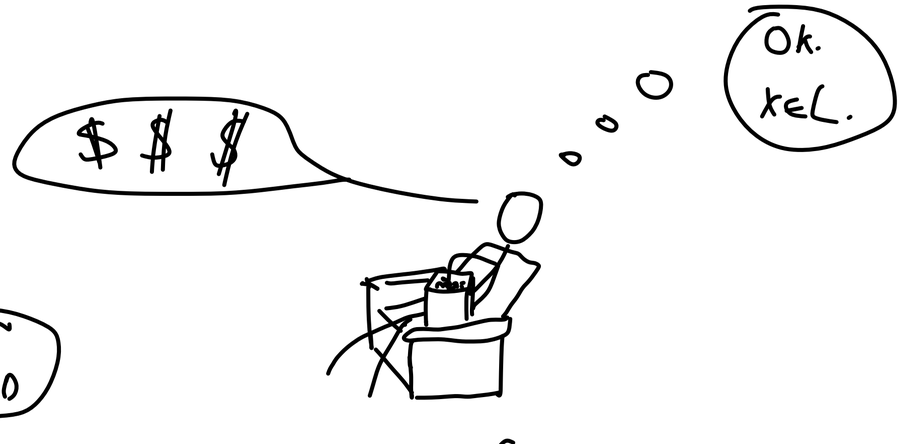
ZAPs

[Dwork Naor '00]

Prover

$\approx$

Prover



Verifier

WITNESS INDISTINGUISHABILITY

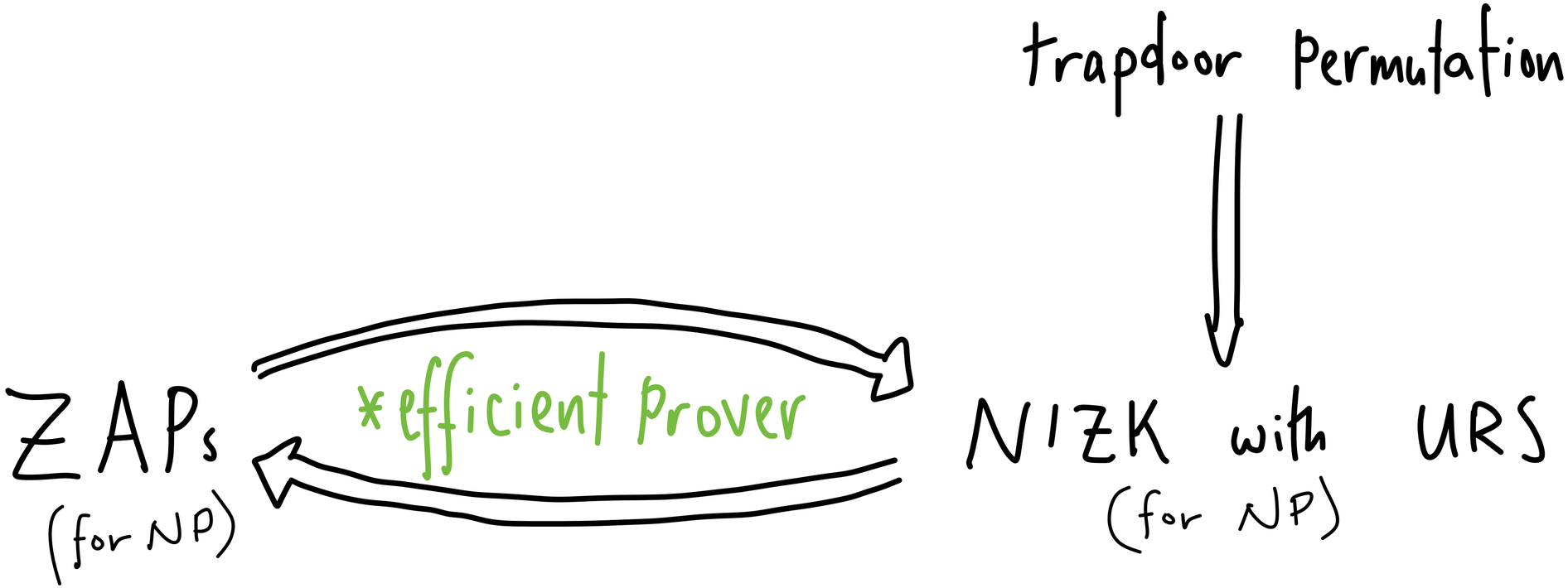
[Feige Shamir '90]

for any witnesses w_1, w_2 for x :

$$\text{Prover}(w_1) \rightleftarrows \text{Verifier} \approx \text{Prover}(w_2) \rightleftarrows \text{Verifier}$$

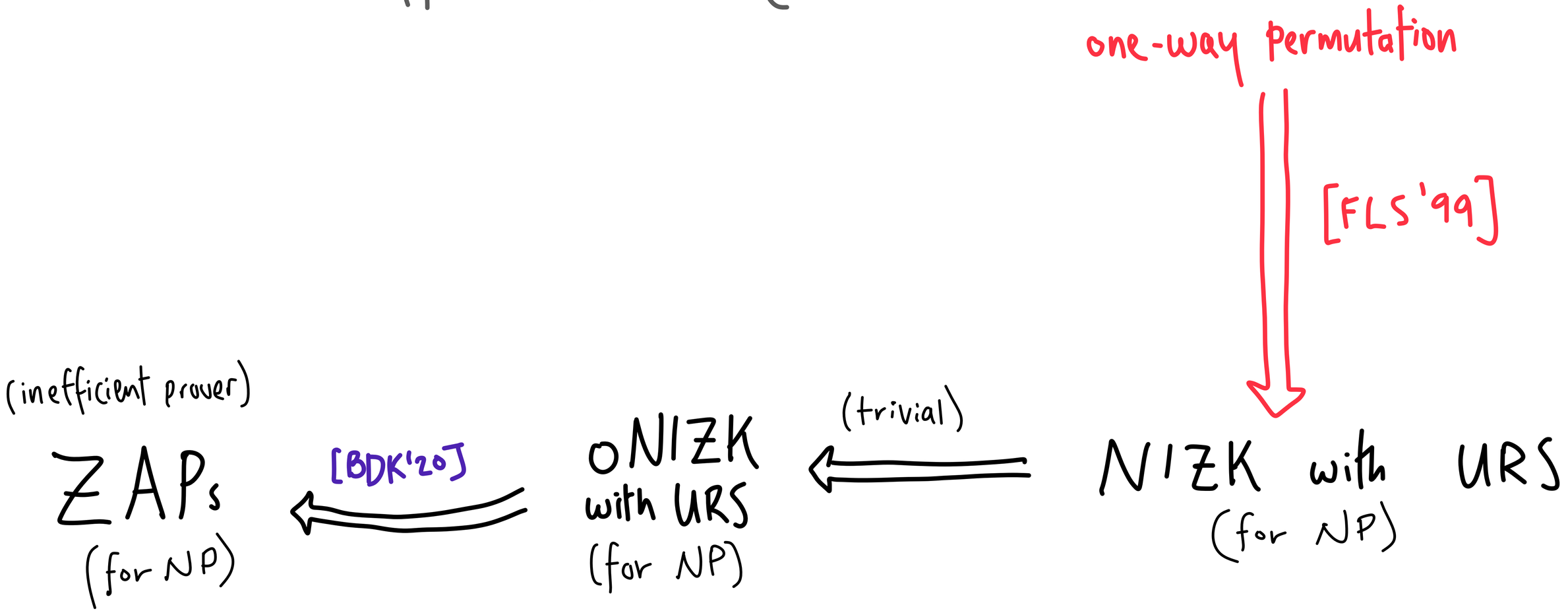
NOTE: non-trivial if prover runs in 2^{n^ε} time $(\varepsilon < 1)$.

[Dwork Naor '00]

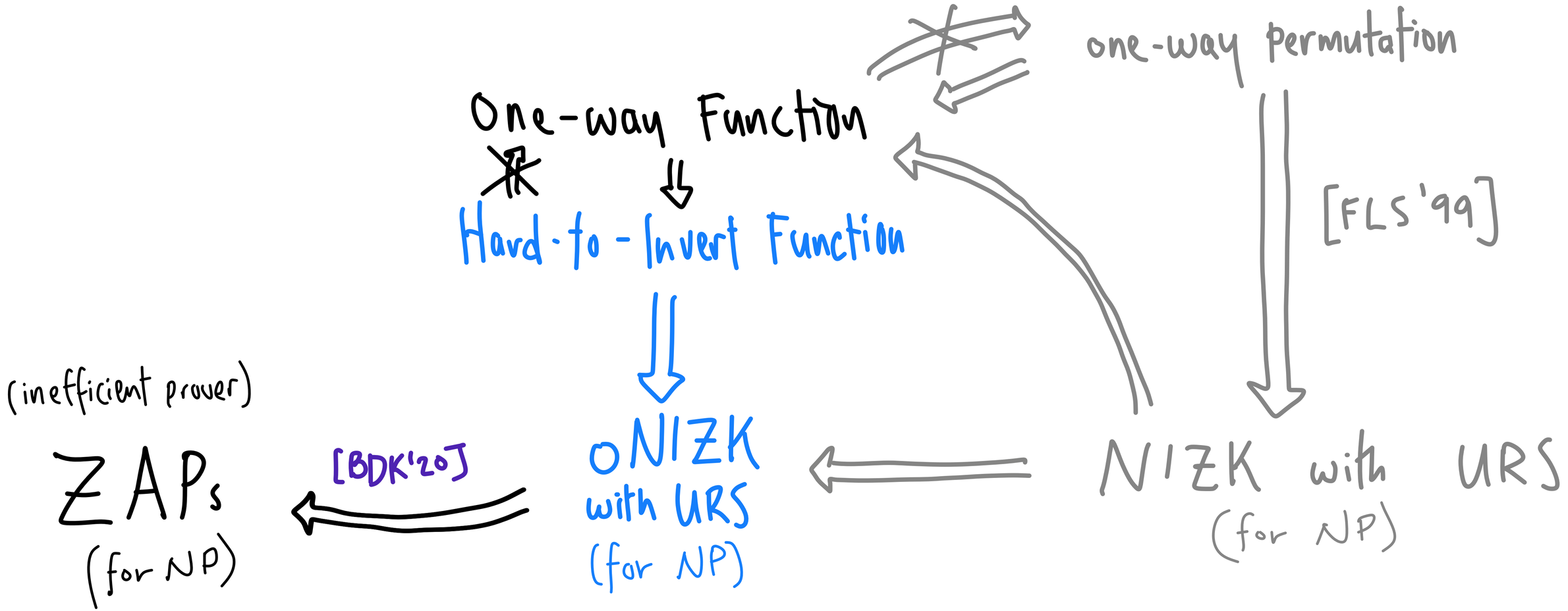


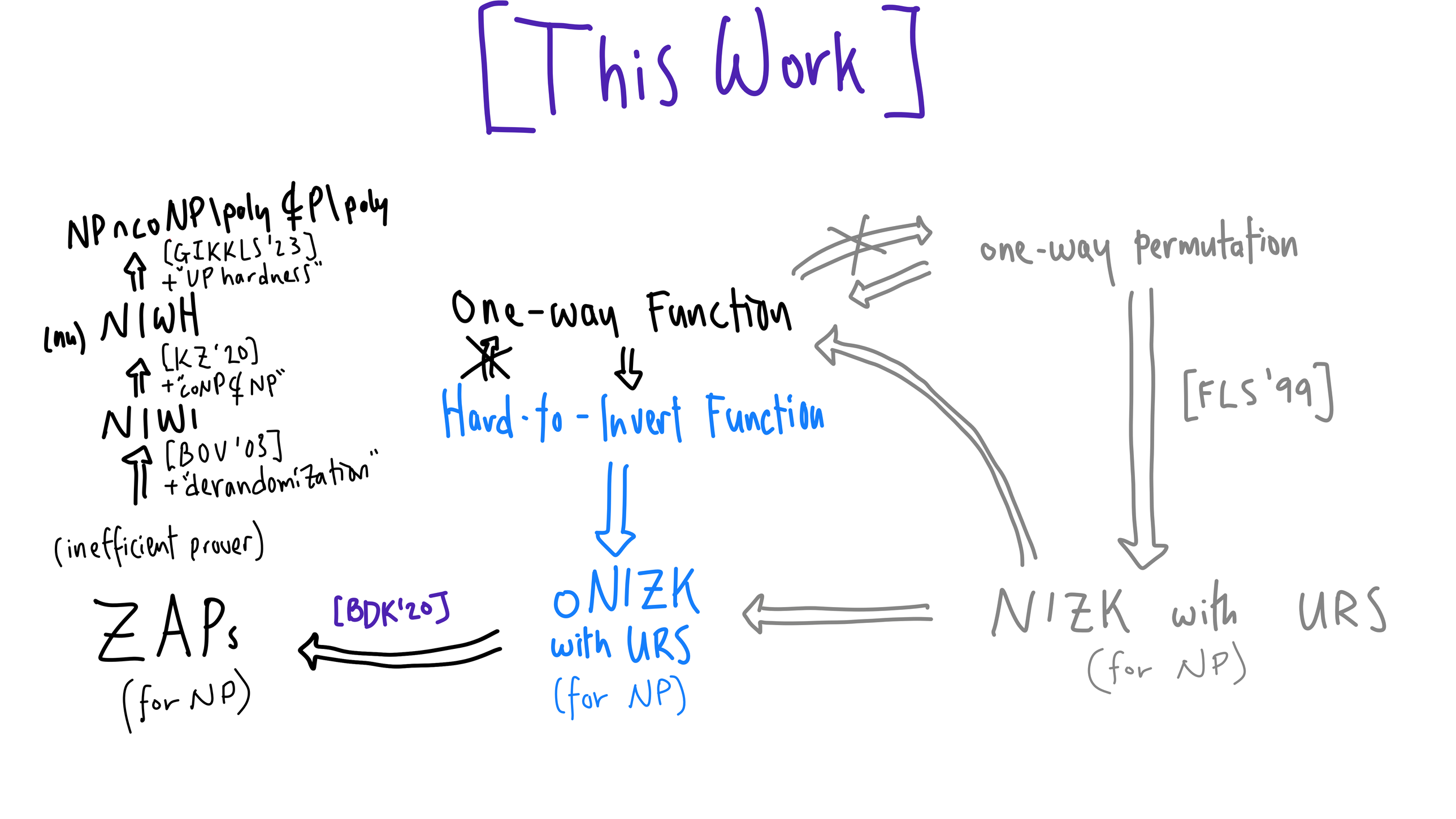
[Ball Dachman-Soled Kulkarni '20]

inefficient provers (for better assumptions)?



[This Work]

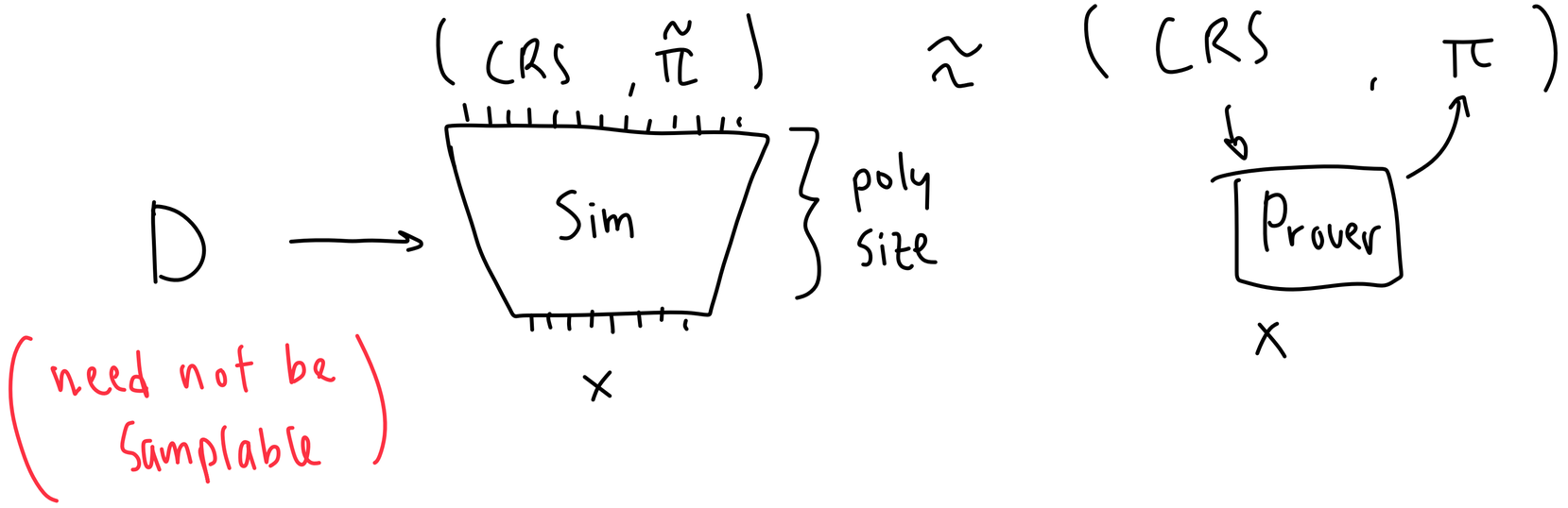

one-way permutation
One-way Function
Hard-to-Invert Function
[FLS'99]
(inefficient prover)
ZAPs
(for NP)
[BDK'20]
oNIZK
with URS
(for NP)
NIZK with URS
(for NP)


[This Work]
NP∩coNP/poly ⊄ P/poly
⇑ [GIKKLS'23] + "UP hardness"
(nu) NIWH
⇑ [KZ'20] + "coNP ⊄ NP"
NIWI
⇑ [BOV'03] + "derandomization"
(inefficient prover)
ZAPs
(for NP)
[BDK'20]
One-way Function
Hard-to-Invert Function
oNIZK with URS
(for NP)
one-way permutation
[FLS'99]
NIZK with URS
(for NP)

oNIZK?

"Offline Simulation"

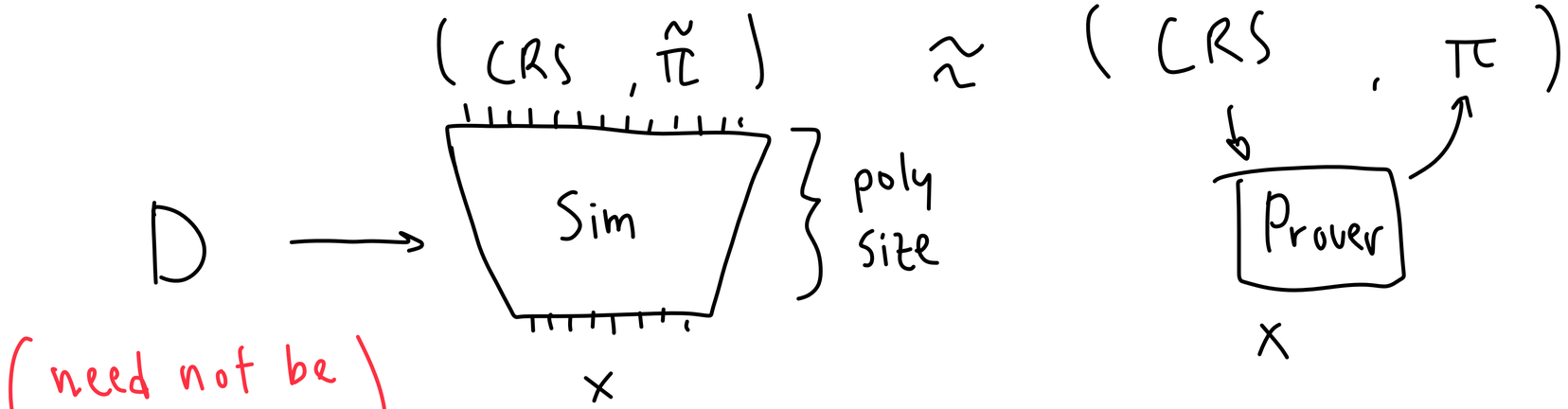
$\forall x \in L$.



oNIZK?

"Offline Simulation"

$\forall x \in L$.



(need not be Samplable)

(NIZK: D → Sim (Samplable))

Technical Overview

how to build oNIZK from HIF?

[Feige Lapidot Shamir '99] framework for NIZK

① NIZK for NP in "hidden bits" model

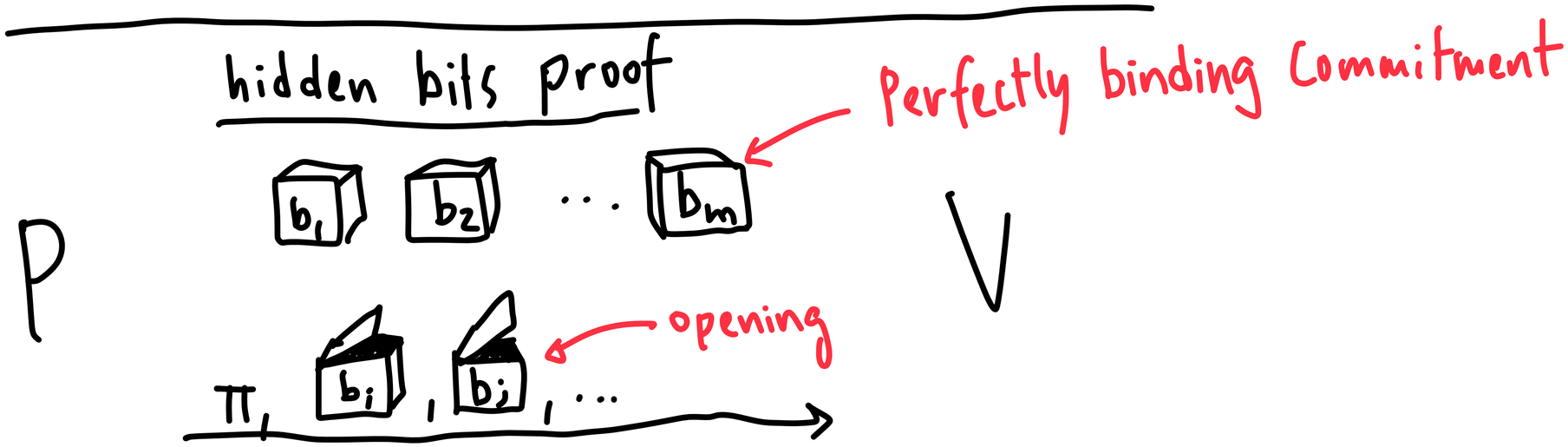
② hidden bits from URS + OWP

how to build oNIZK from HIF?

[Feige Lapidot Shamir '99] framework for NIZK

① NIZK for NP in "hidden bits" model

② hidden bits from URS + OWP



how to build ∘NIZK from HIF?

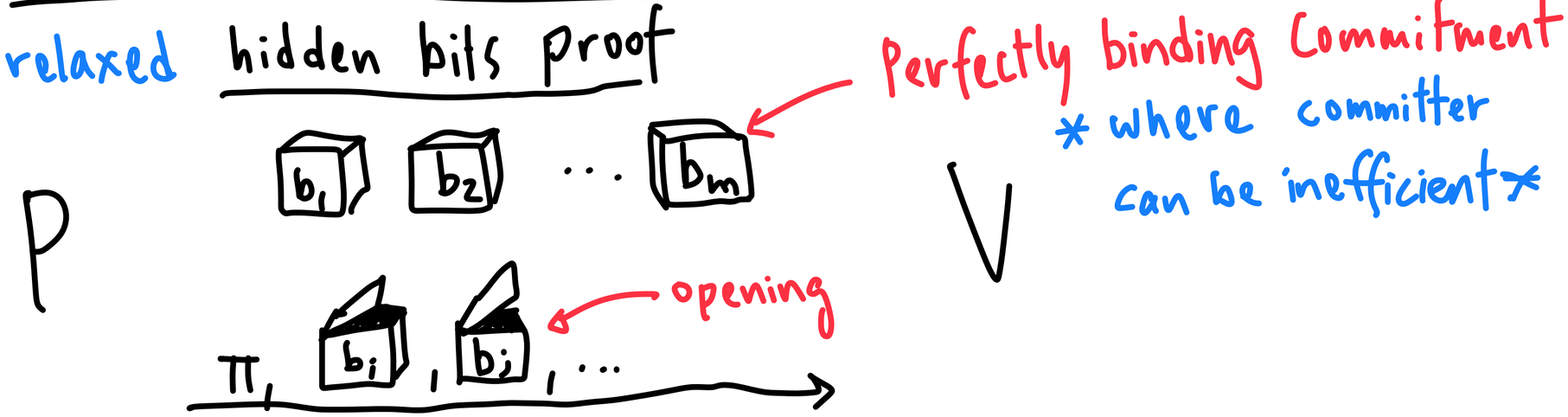
[Feige Lapidot Shamir '99] framework for ∘NIZK

① ∘NIZK for NP in "relaxed hidden bits" model (no change) ✓

② relaxed hidden bits from URS + HIF ← main contribution

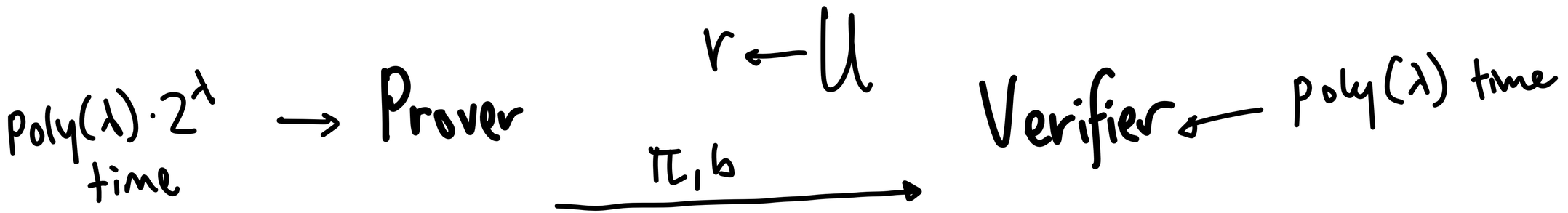
---

relaxed hidden bits proof



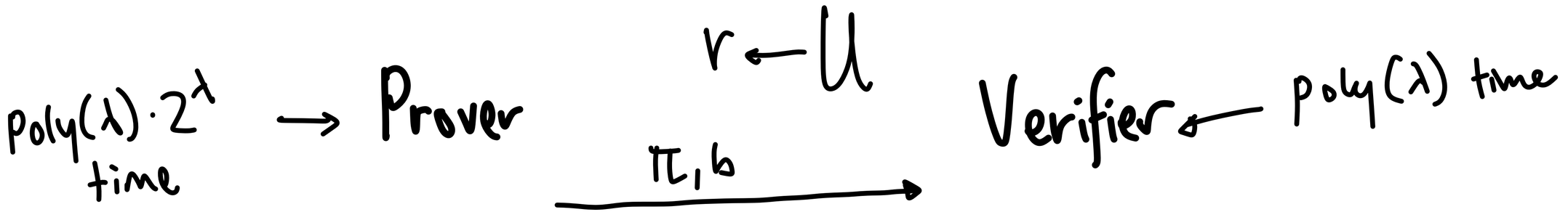
Relaxed Hidden Bits in URS model (informal)

"Verifiable Hardcore of Random String"



Relaxed Hidden Bits in URS model (informal)

"Verifiable Hardcore of Random String"

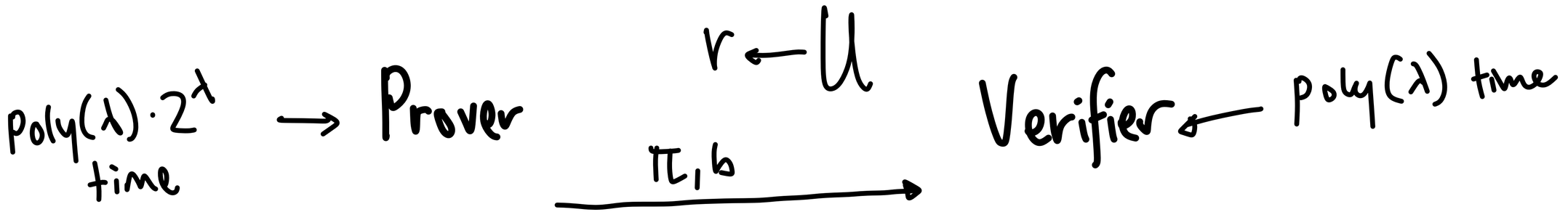


with high probability over r:

1. (total) perfect binding: $\exists$ unique b that verifier will accept (w/ π)

Relaxed Hidden Bits in URS model (informal)

"Verifiable Hardcore of Random String"

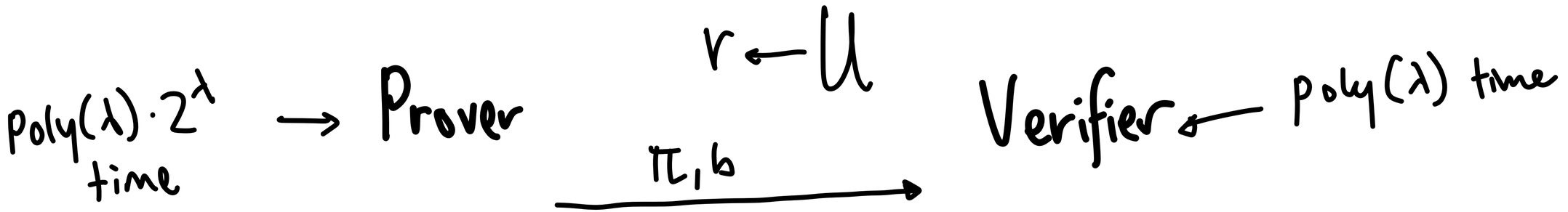


with high probability over r:

1. (total) perfect binding: ∃ unique b that verifier will accept (w/ π)
2. hiding: $\left(r \mid \text{Prover}(r)=(\pi,0)\right) \approx \left(r \mid \text{Prover}(r)=(\pi,1)\right)$

Relaxed Hidden Bits in URS model (informal)

"Verifiable Hardcore of Random String"



with high probability over r:

1. (total) perfect binding: $\exists$ unique b that verifier will accept (w/ π)
2. hiding: $\left(r \mid \text{Prover}(r) = (\pi, 0)\right) \approx \left(r \mid \text{Prover}(r) = (\pi, 1)\right)$

non-relaxed version: can efficiently sample (r, π, b)

Recall: [FLS'99] hidden bits

tools:
- one-way permutation: $f : \{0,1\}^\lambda \xrightarrow{\sim} \{0,1\}^\lambda$
- hardcore predicate for f: $\phi : \{0,1\}^\lambda \to \{0,1\}$

Recall: [FLS'99] hidden bits

tools: • one-way permutation: $f : \{0,1\}^\lambda \xrightarrow{\sim} \{0,1\}^\lambda$

• hardcore predicate for f: $\phi : \{0,1\}^\lambda \to \{0,1\}$

Construction:

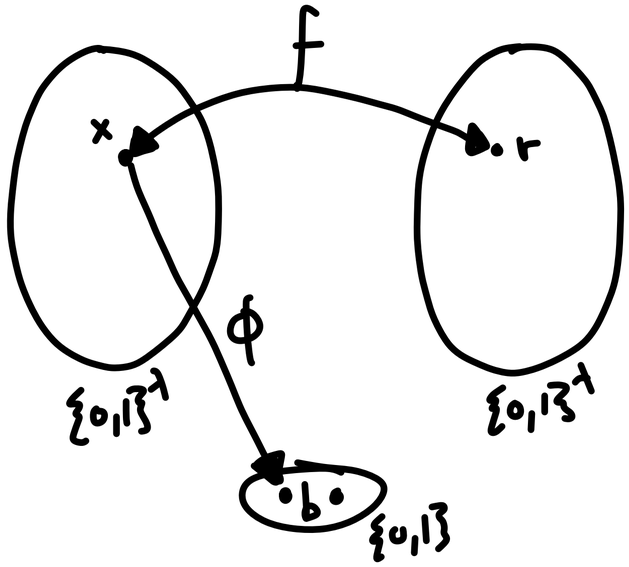
$r \leftarrow U_\lambda$

Prover		Verifier
compute: 1. $x = f^{-1}(r)$	$\xrightarrow{x, b}$	check $f(x) = r$
2. $b = \phi(x)$		& $\phi(x) = b$

Recall: [FLS'99] hidden bits

tools: • one-way permutation: $f : \{0,1\}^\lambda \xrightarrow{\sim} \{0,1\}^\lambda$

• hardcore predicate for f: $\phi : \{0,1\}^\lambda \to \{0,1\}$



Construction:

$r \leftarrow U_\lambda$

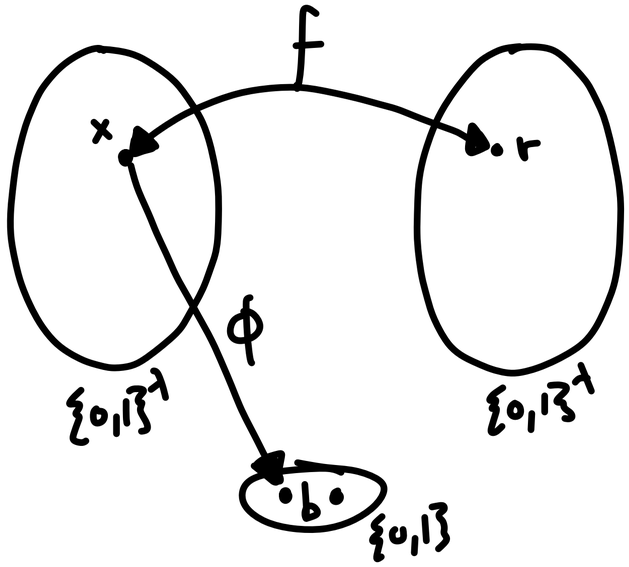
Prover		Verifier
compute: 1. $x = f^{-1}(r)$	$\xrightarrow{x, b}$	check $f(x) = r$
2. $b = \phi(x)$		& $\phi(x) = b$

(total) perfect binding: $x = f^{-1}(r)$ always exists and is unique (sim. $\phi(x) = b$)

Recall: [FLS'99] hidden bits

tools: • one-way permutation: $f : \{0,1\}^\lambda \xrightarrow{\sim} \{0,1\}^\lambda$

• hardcore predicate for f: $\phi : \{0,1\}^\lambda \to \{0,1\}$



Construction:

$r \leftarrow U_\lambda$

Prover		Verifier
compute: 1. $x = f^{-1}(r)$	$\xrightarrow{x, b}$	check $f(x) = r$
2. $b = \phi(x)$		& $\phi(x) = b$

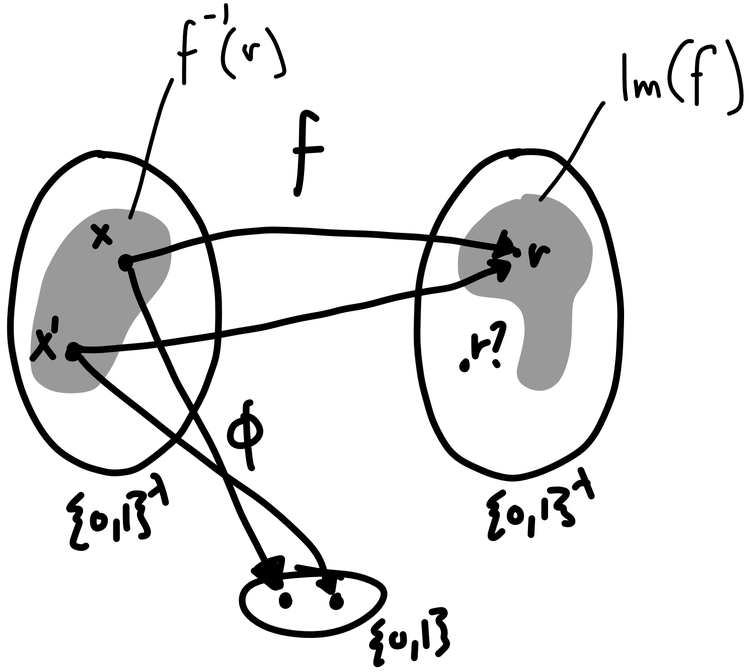
(total) perfect binding: $x = f^{-1}(r)$ always exists and is unique (sim. $\phi(x) = b$)

hiding: ϕ is hardcore predicate

Attempt from OWF (HIF same)

tools: • one-way ~~permutation~~ function: $f : \{0,1\}^\lambda \xrightarrow{\sim} \{0,1\}^\lambda$

• hardcore predicate for f: $\phi : \{0,1\}^\lambda \to \{0,1\}$



Construction:

$r \leftarrow U_\lambda$

Prover		Verifier
compute: 1. $x = f^{-1}(r)$	$\xrightarrow{x, b}$	check $f(x) = r$
2. $b = \phi(x)$		& $\phi(x) = b$

(total) perfect binding: $x = f^{-1}(r)$ ~~always exists~~ and is ~~unique~~ (sim. $\phi(x) = b$) ☹

hiding: ϕ is hardcore predicate ✓

Attempt 2: making total

given (H, r)

Prover

compute: 1. $x = (H \circ f)^{-1}(r)$

2. $b = \phi(x)$

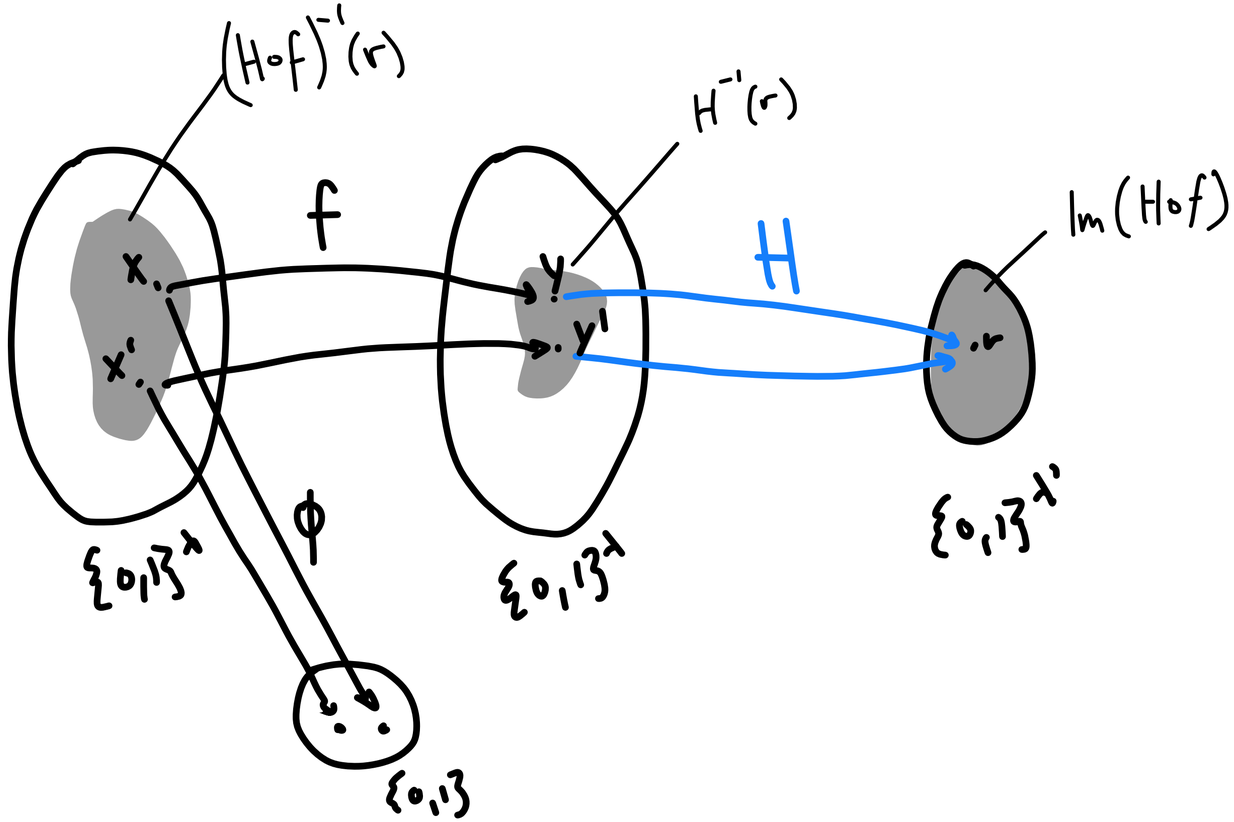
send (x, b) to open

Verifier

check $H \circ f(x) = r$

& $\phi(x) = b$

H: t-wise independent hash



(total) perfect binding: $x \in (H \circ f)^{-1}(r)$ always exists and is ~~unique~~

hiding: ϕ is hardcore predicate ✓

Attempt 3: making unique?

given (H, r, h)

Prover

compute: 1. $x \in (H \circ f)^{-1}(r) \cap h^{-1}(0)$

2. $b = \phi(x)$

send (x, b) to open

Verifier

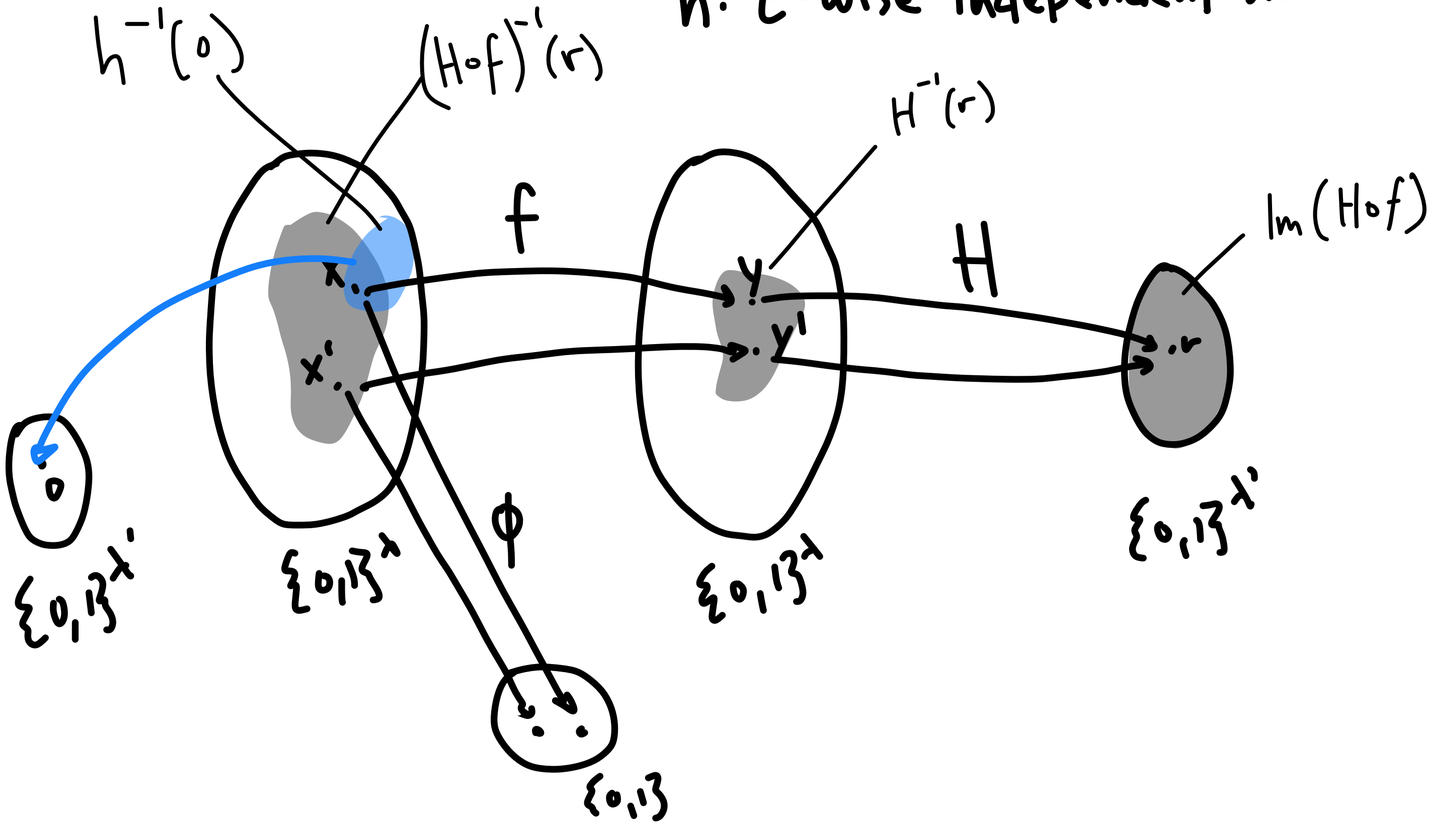
check $H \circ f(x) = r$

& $\phi(x) = b$

& $h(x) = 0$

H: t-wise independent hash

h: t-wise independent hash



(total) perfect binding: $x \in (H \circ f)^{-1}(r) \cap h^{-1}(0)$ always exists and is unique?

hiding: ϕ is hard core predicate ✓

Attempt 3: making unique?

given (H, r, h)

Prover

compute: 1. $x \in (H \circ f)^{-1}(r) \cap h^{-1}(0)$

2. $b = \phi(x)$

send (x, b) to open

Verifier

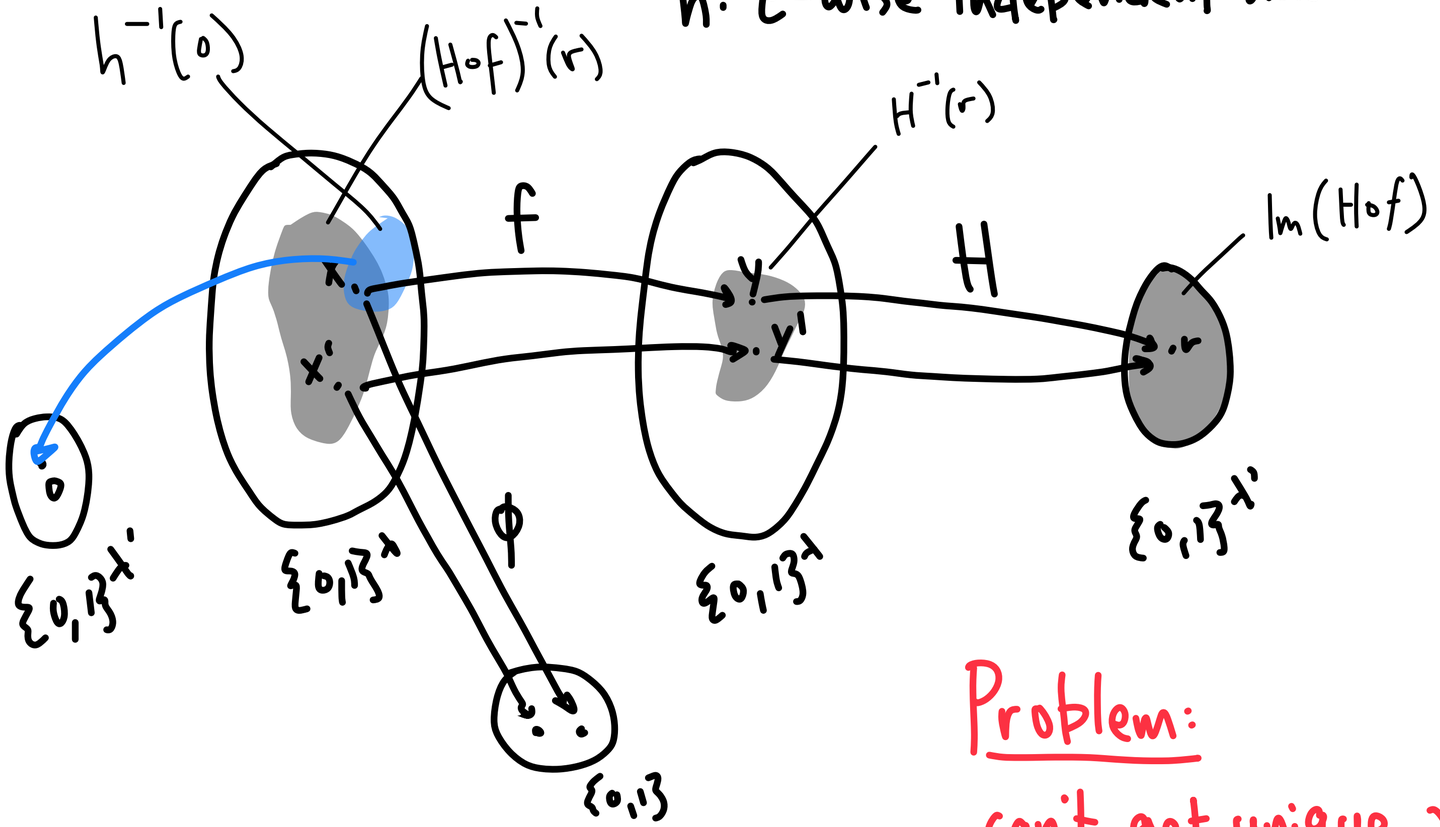
check $H \circ f(x) = r$

& $\phi(x) = b$

& $h(x) = 0$

H: t-wise independent hash

h: t-wise independent hash



Problem:

can't get unique x w.h.p.

(total) perfect binding: $x \in (H \circ f)^{-1}(r) \cap h^{-1}(0)$ always exists and is unique?

hiding: ϕ is hard core predicate ✓

Attempt 4: use everything

given (H, r, h)

Prover

compute: 1. $S = (H \circ f)^{-1}(r) \cap h^{-1}(0)$

2. $b = \phi(S)$

send (S, b) to open

Verifier

check $\forall x \in S$,

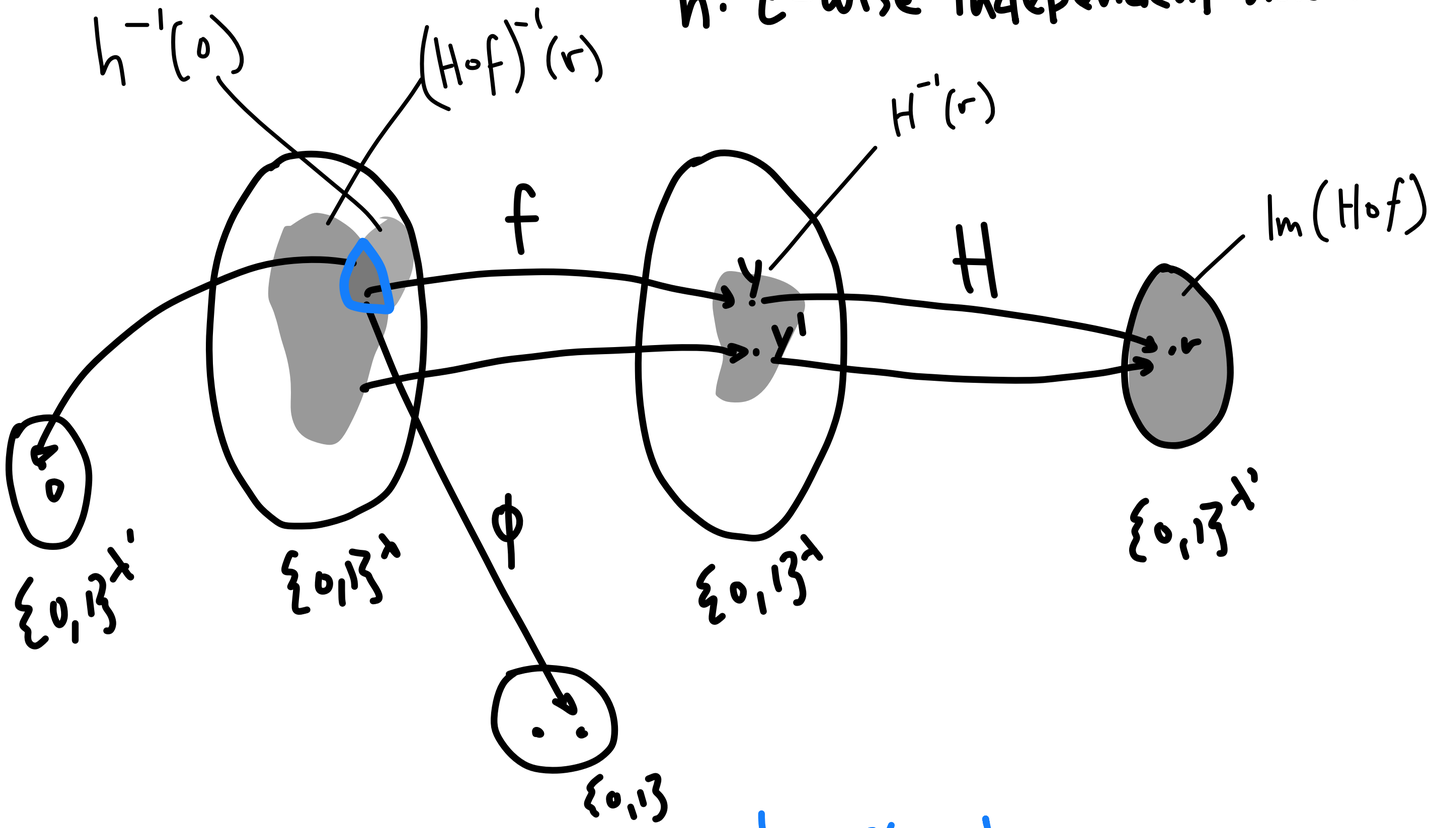
$H \circ f(x) = r$

& $h(x) = 0$,

& $\phi(S) = b$

H: t-wise independent hash

h: t-wise independent hash



(total) perfect binding: $S = (H \circ f)^{-1}(r) \cap h^{-1}(0)$ always exists and is ~~unique~~ poly-sized w.h.p.

hiding: ϕ is hard core predicate ✓

Attempt 4: use everything

given (H, r, h)

Prover

compute: 1. $S = (H \circ f)^{-1}(r) \cap h^{-1}(0)$

2. $b = \phi(S)$

send (S, b) to open

Verifier

check $\forall x \in S$,

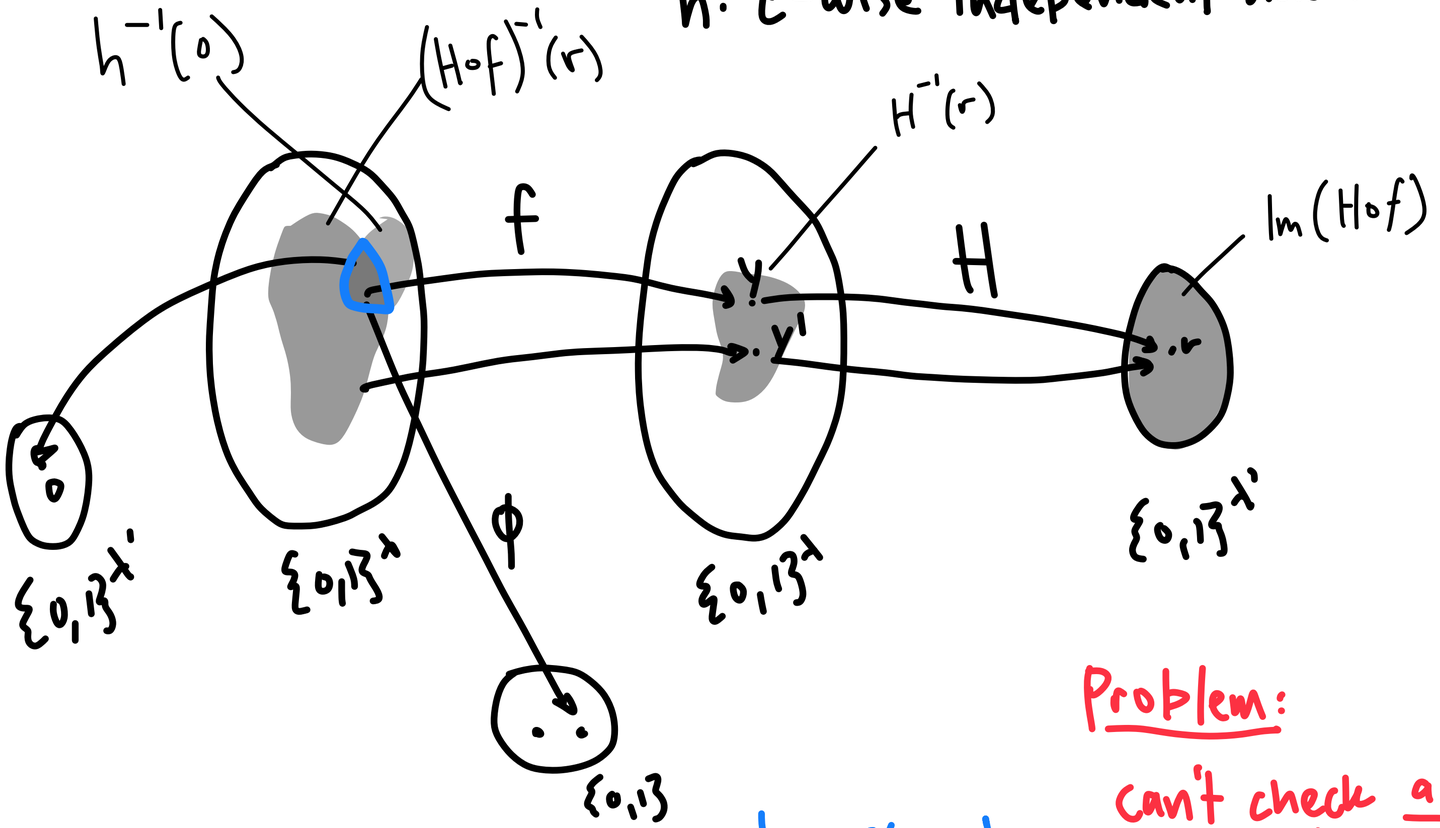
$H \circ f(x) = r$

& $h(x) = 0$,

& $\phi(S) = b$

H: t-wise independent hash

h: t-wise independent hash



Problem:

can't check all of S sent!

$\phi(S) \neq \phi(S \setminus \{x\})$

(total) perfect binding: $S = (H \circ f)^{-1}(r) \cap h^{-1}(0)$ always exists and is ~~unique~~ poly-sized w.h.p.

hiding: ϕ is hard core predicate ✓

Attempt 5: repeat & use statistics

given $(H_1, r_1, h_1) \cdots (H_m, r_m, h_m)$

Prover

compute: for $i = 1 \cdots m$:

1. $S_i = (H_i \circ f_i)^{-1}(r_i) \cap h^{-1}(0)$
2. $b_i = \phi(S_i)$

set $b = MAJ(b_1, \cdots b_m)$

send $(S_1, \cdots, S_m, b)$

Verifier

check $\forall i, \forall x \in S_i$

$H_i \circ f_i(x) = r_i$

& $h_i(x) = 0$,

& $MAJ(\phi(S_1) \cdots \phi(S_m)) = b$

& $|S_1| \cdots |S_m|$ distributed correctly

Attempt 5: repeat & use statistics

given $(H_1, r_1, h_1) \cdots (H_m, r_m, h_m)$

Prover

compute: for $i = 1 \cdots m$:

1. $S_i = (H_i \circ f_i)^{-1}(r_i) \cap h^{-1}(0)$
2. $b_i = \phi(S_i)$

set $b = MAJ(b_1 \cdots b_m)$

send $(S_1, \cdots, S_m, b)$

Verifier

check $\forall i, \forall x \in S_i$

$H_i \circ f_i(x) = r_i$

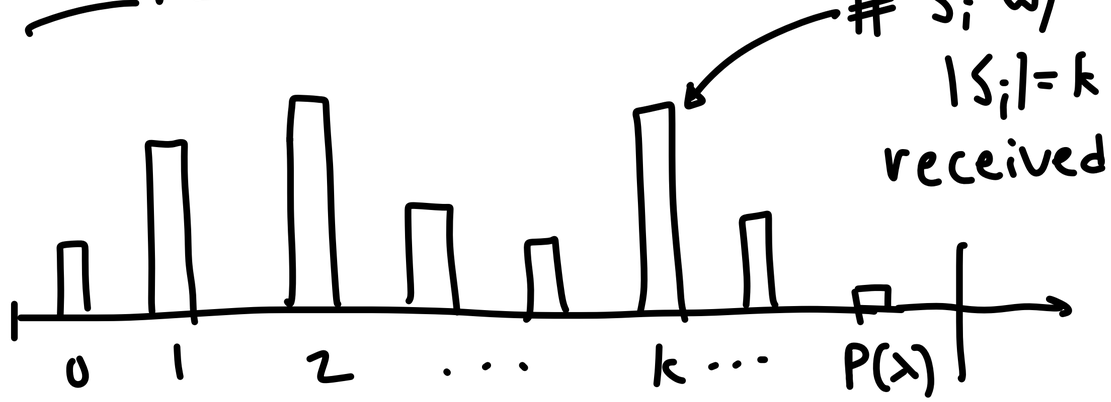
& $h_i(x) = 0$,

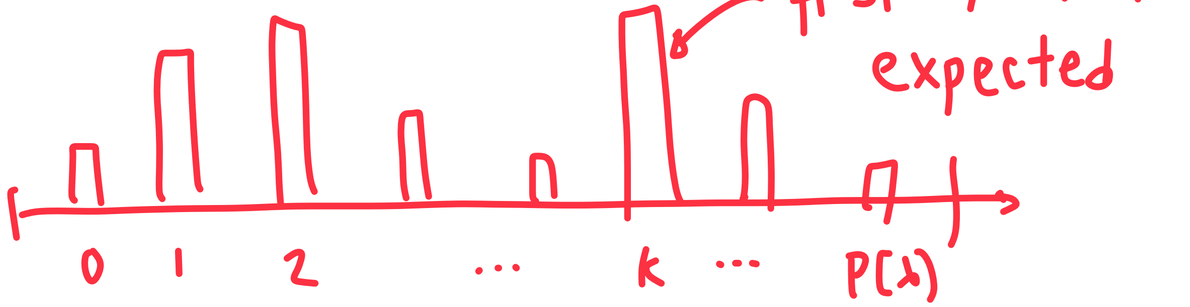
& $MAJ(\phi(S_1) \cdots \phi(S_m)) = b$

& $|S_1| \cdots |S_m|$ distributed correctly

[Akavia Goldreich Goldwasser Moshkovitz '06]:

Compare (acceptif close):





Attempt 5: repeat & use statistics

given $(H_1, r_1, h_1) \cdots (H_m, r_m, h_m)$

Prover

compute: for $i = 1 \cdots m$:

1. $S_i = (H_i \circ f_i)^{-1}(r_i) \cap h^{-1}(0)$
2. $b_i = \phi(S_i)$

set $b = MAJ(b_1 \cdots b_m)$

send $(S_1, \cdots, S_m, b)$

Verifier

check $\forall i, \forall x \in S_i$

$H_i \circ f_i(x) = r_i$

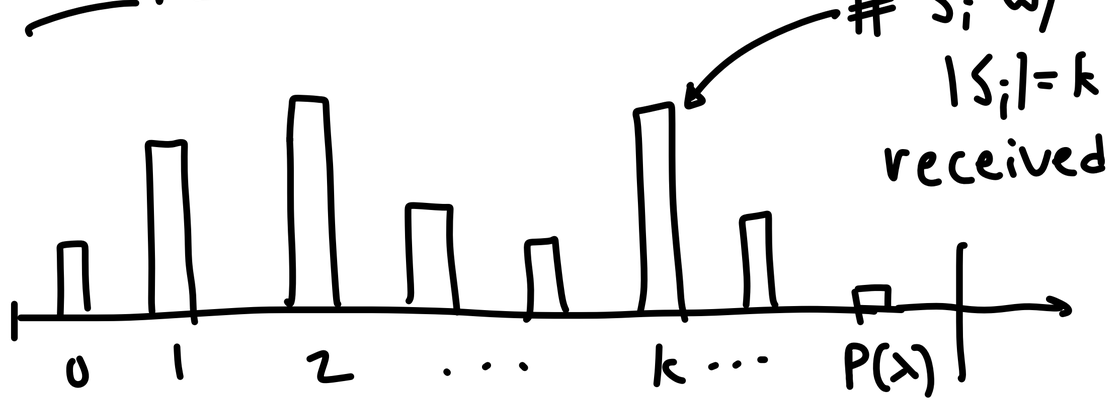
& $h_i(x) = 0$,

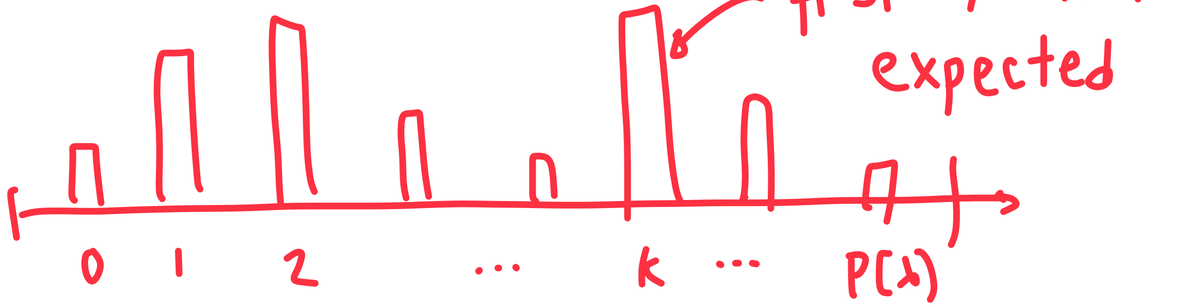
& $MAJ(\phi(S_1) \cdots \phi(S_m)) = b$

& $|S_1| \cdots |S_m|$ distributed correctly

[Akavia Goldreich Goldwasser Moshkowitz '06]:

Compare (acceptif close):





Problem: where to get statistics about f?

Attempt 5: repeat & use statistics

given $(H_1, r_1, h_1) \cdots (H_m, r_m, h_m)$

Prover

compute: for $i = 1 \cdots m$:

1. $S_i = (H_i \circ f_i)^{-1}(r_i) \cap h^{-1}(0)$
2. $b_i = \phi(S_i)$

set $b = MAJ(b_1 \cdots b_m)$

send $(S_1, \cdots, S_m, b)$

Verifier

check $\forall i, \forall x \in S_i$:

$H_i \circ f_i(x) = r_i$

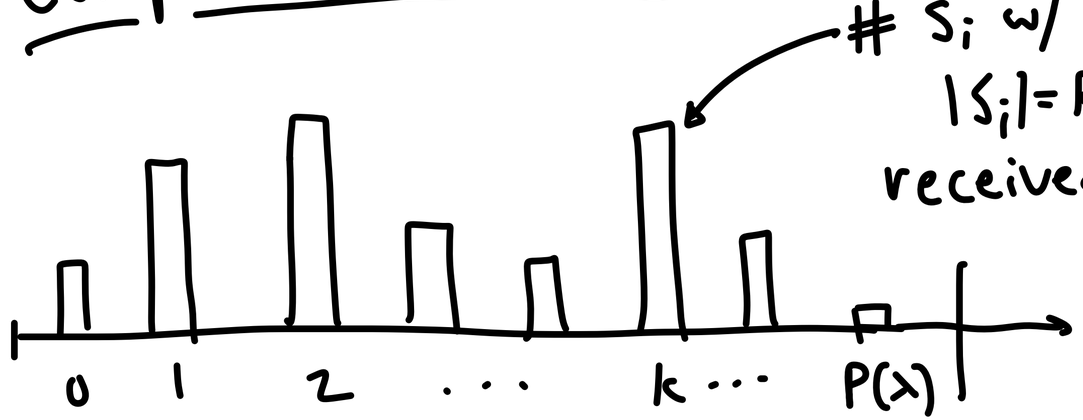
& $h_i(x) = 0$,

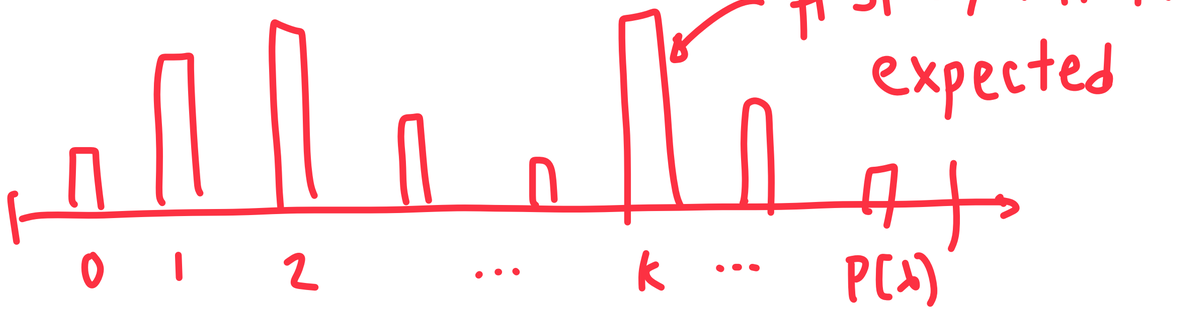
& $MAJ(\phi(S_1) \cdots \phi(S_m)) = b$

& $|S_1| \cdots |S_m|$ distributed correctly

[Akavia Goldreich Goldwasser Moshkovitz '06]:

Compare (acceptif close):





problem: where to get statistics about f?

Prover (w/ proof) [Haitner Mahmoody Xiao '10]

Hardness in $NP \cap coNP$

[Brassard '79] OWP $\Rightarrow$ $NP \cap coNP \neq P$.

Hardness in $NP \cap coNP$

[Brassard '79] OWP $\Rightarrow$ $NP \cap coNP \neq P$.

[BitanskyDeqwekerVaikuntanathan'17] injective OWF $\overset{BB}{\not\Rightarrow}$ $NP \cap coNP \neq P$.

Hardness in $NP \cap coNP$

[Brassard '79] OWP $\Rightarrow$ $NP \cap coNP \neq P$.

[BitanskyDegweker Vaikuntanathan '17] injective OWF $\overset{BB}{\not\Rightarrow}$ $NP \cap coNP \neq P$.

[Ghosal Ishai Korb Kushilevitz Lou Sahai '23] $UP \not\subseteq RP \Rightarrow NP \cap coNP \neq P$ relative to Random Oracle
(non BB)

Hardness in $NP \cap coNP$

[Brassard '79] OWP $\Rightarrow$ $NP \cap coNP \neq P$.

[BitanskyDegweker Vaikuntanathan '17] injective OWF $\overset{BB}{\not\Rightarrow}$ $NP \cap coNP \neq P$.

[Ghosal Ishai Korb Kushilevitz Lou Sahai '23] $UP \not\subseteq RP \Rightarrow NP \cap coNP \neq P$ relative to Random Oracle
(non BB)

[This Work] assuming (nonuniform):
1. HIF
2. derandomization assumption (for AM=NP)
3. coNP hard for subexponential nondeterministic time
4. UP is hard for subexponential time

then $P \neq NP \cap coNP$ in nonuniform setting

Hardness in $NP \cap coNP$

[Brassard '79] OWP $\Rightarrow$ $NP \cap coNP \neq P$.

[BitanskyDegweker Vaikuntanathan '17] injective OWF $\overset{BB}{\not\Rightarrow}$ $NP \cap coNP \neq P$.

[Ghosal Ishai Korb Kushilevitz Lou Sahai '23] $UP \not\subseteq RP \Rightarrow NP \cap coNP \neq P$ relative to Random Oracle
(non BB)

[This Work] assuming (nonuniform):
① HIF
② derandomization assumption (for AM=NP)
③ coNP hard for subexponential nondeterministic time
④ UP is hard for subexponential time

then $P \neq NP \cap coNP$ in nonuniform setting

(Sketch: ① $\Rightarrow$ ZAP $\overset{+②}{\Rightarrow}$ NIWI [BOV'03] $\overset{+③}{\Rightarrow}$ NIWH [KZ'20] $\overset{+④}{\Rightarrow}$ $P \neq NP \cap coNP$ [GIKKLS'23])

Open Questions

1. minimal assumptions for...
   - (a) hard-to-invert functions?
   - (b) oNIZK?
   - (c) inefficient prover ZAPs?

2. OWF $\Rightarrow$ NIZK in URS model?

3. explicit hardness in NP $\cap$ coNP without OWP?

thanks!