Uniform Black-Box Separations via Non-Malleable Extractors

Marshall Ball
(NYU)

Dana Dachman-Soled
(UMD)

ROADMAP

I. (seedless) t-Non-Malleable Extractors

II. an Application: UNIFORM Black-Box Separations

I. (Seedless) t-Non-Malleable Extractors

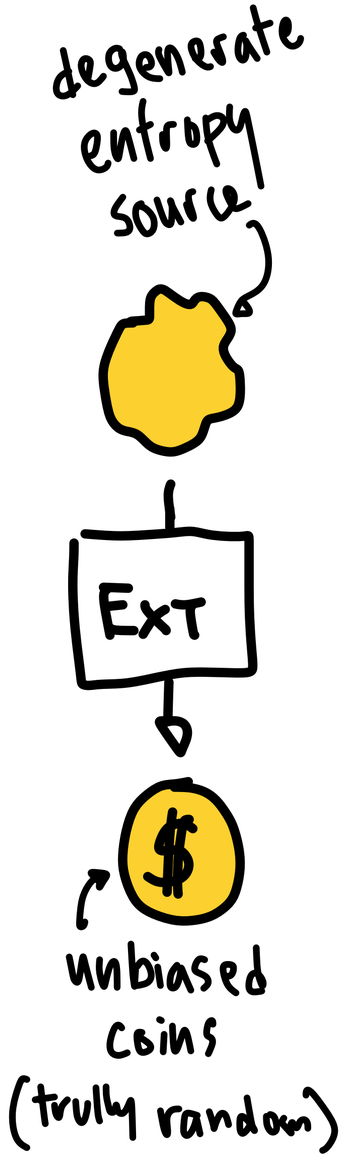
for poly size samplable sources
and poly size tampering

Randomness Extractors

$S = \{X\}$ a class of sources

all sources must have sufficient <u>min-entropy</u>

$(\forall x,\ \Pr[X = x] \leq 2^{-k}$ for some $k)$



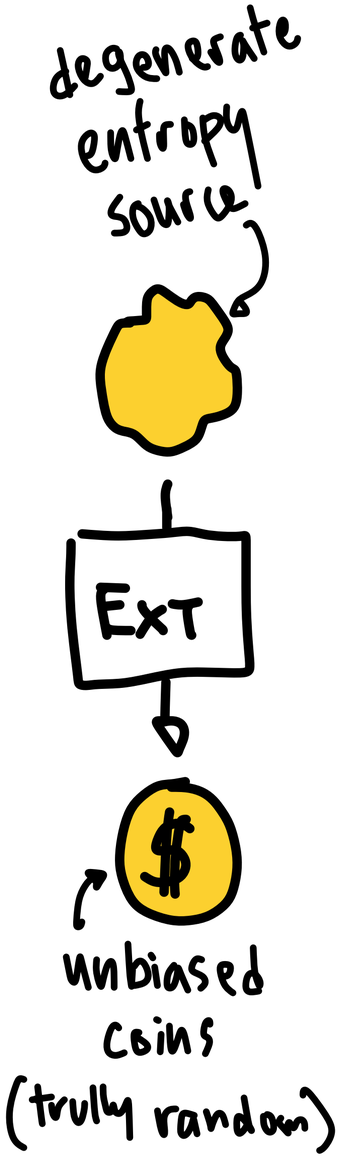
Randomness Extractors

$S = \{X\}$ a class of sources

all sources must have sufficient <u>min-entropy</u>

$(\forall x,\ \Pr[X = x] \le 2^{-k}$ for some $k)$

$$\boxed{\begin{array}{c}\text{Ext is an } \underline{\text{extractor}} \text{ for } S \text{ if} \\ \forall X \in S, \quad \mathrm{Ext}(X) \approx U\end{array}}$$



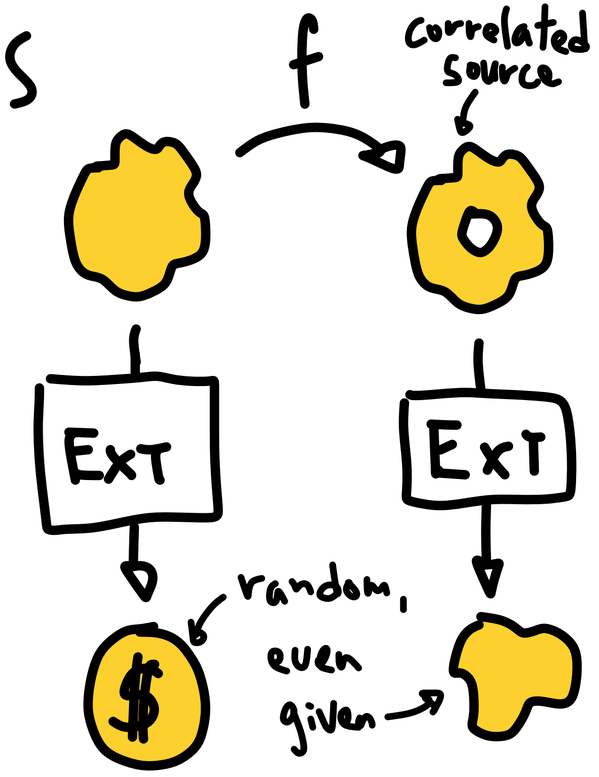
t-Non-Malleable Extractors

[Cheraghchi Guruswami '14, Chattopadhyay Goyal Li '16]

$S = \{X\}$ a class of sources

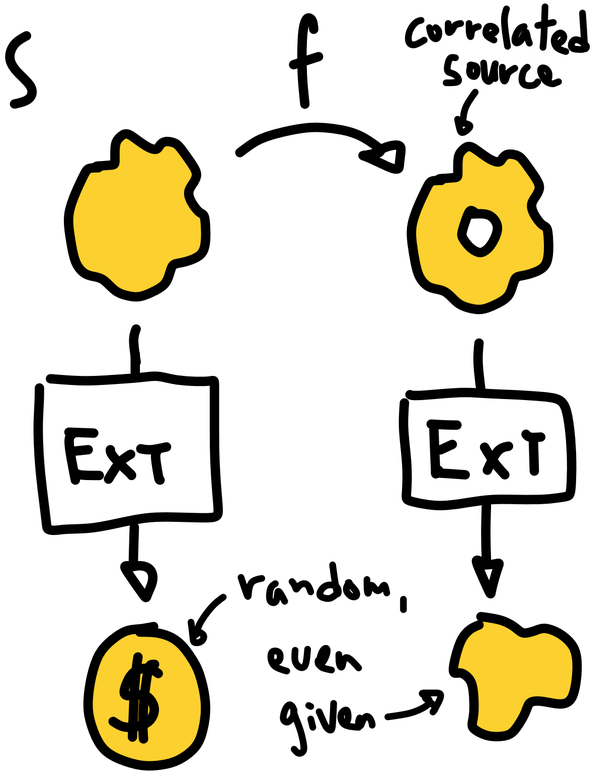
$T = \{f\}$ a class of tampering functions

(x with no fixed points)



t-Non-Malleable Extractors

[Cheraghchi Guruswami '14, Chattopadhyay Goyal Li '16]



$S = \{\mathbb{X}\}$ a class of sources

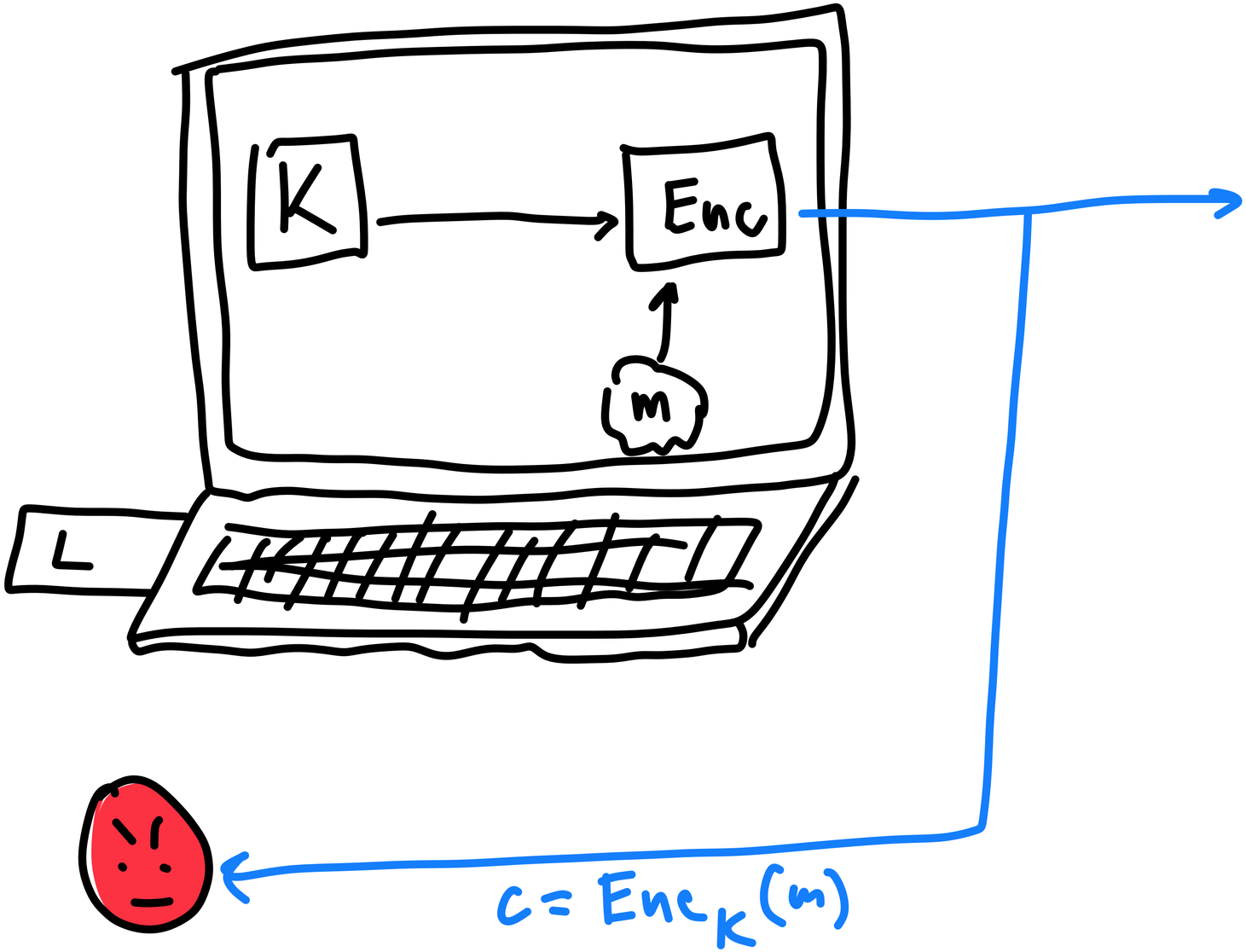
$T = \{f\}$ a class of tampering functions

($*$ with no fixed points)

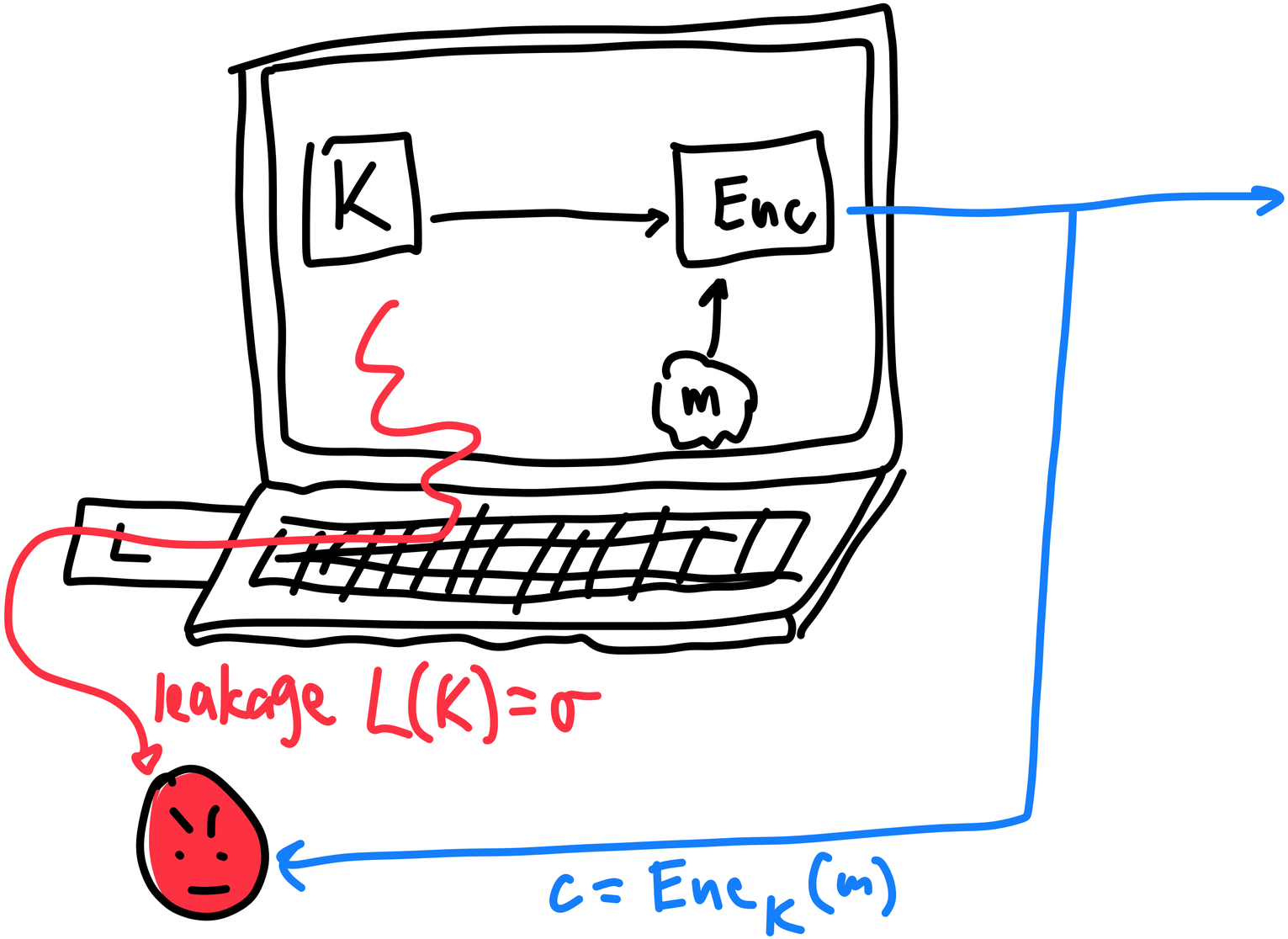
NMExt is a <u>t-non-malleable extractor</u> for S and T:

$$\forall \mathbb{X} \in S,\ \forall f_1, \dots, f_t \in T, \quad \left(\mathrm{NMExt}(\mathbb{X}), \mathrm{NMExt}(f_1(\mathbb{X})), \dots, \mathrm{NMExt}(f_t(\mathbb{X}))\right) \approx \left(U, \mathrm{NMExt}(f_1(\mathbb{X})), \dots, \mathrm{NMExt}(f_t(\mathbb{X}))\right)$$

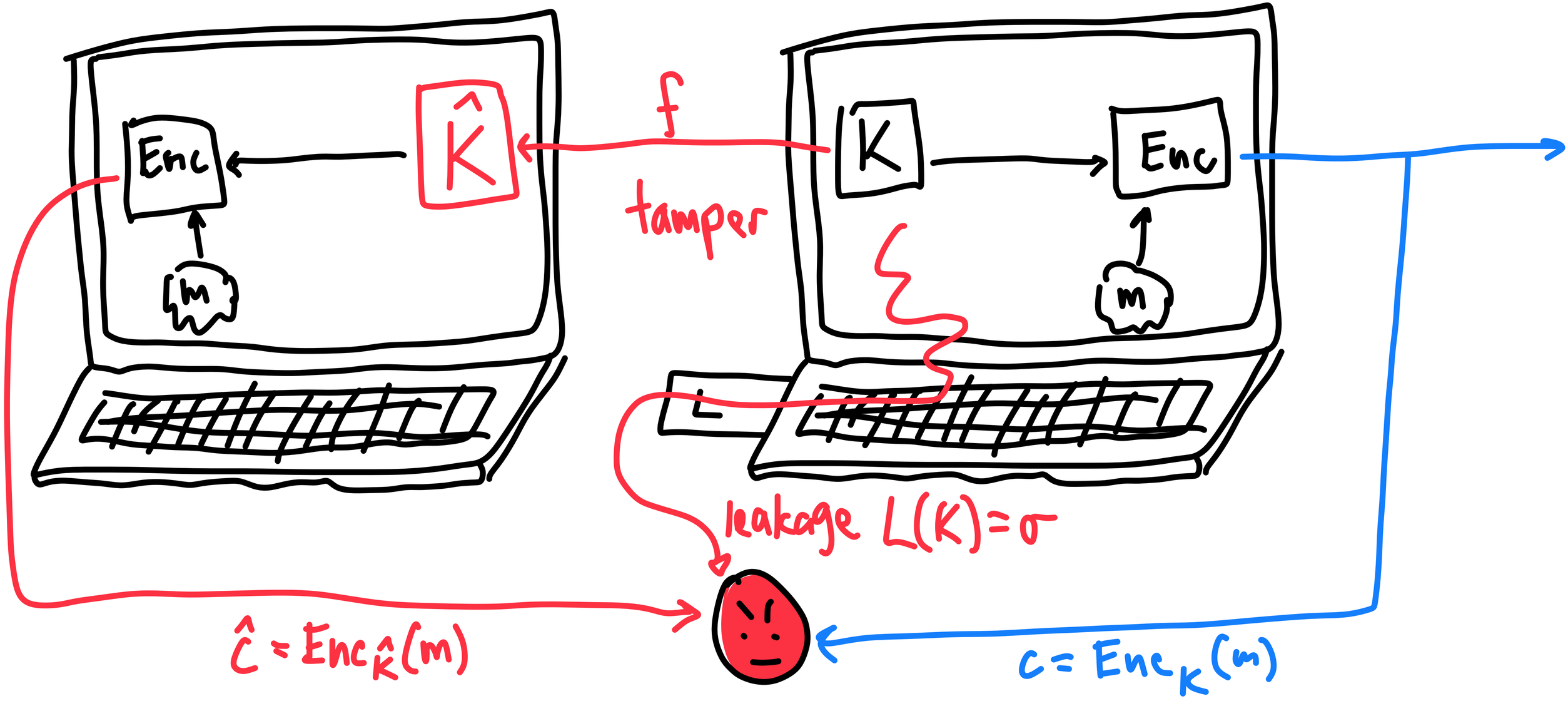
Application: tamper + leakage resilience



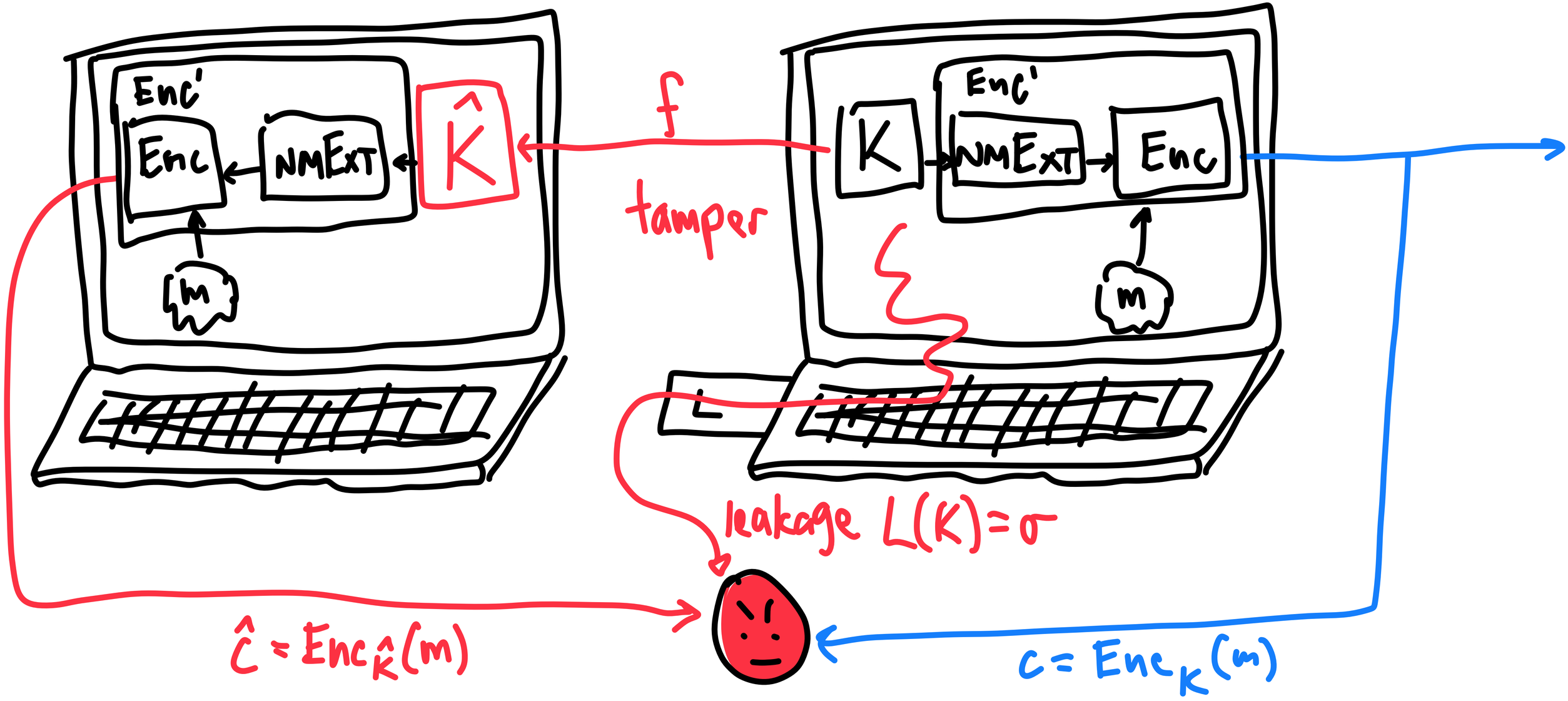
Application: tamper + leakage resilience



Application: tamper + leakage resilience

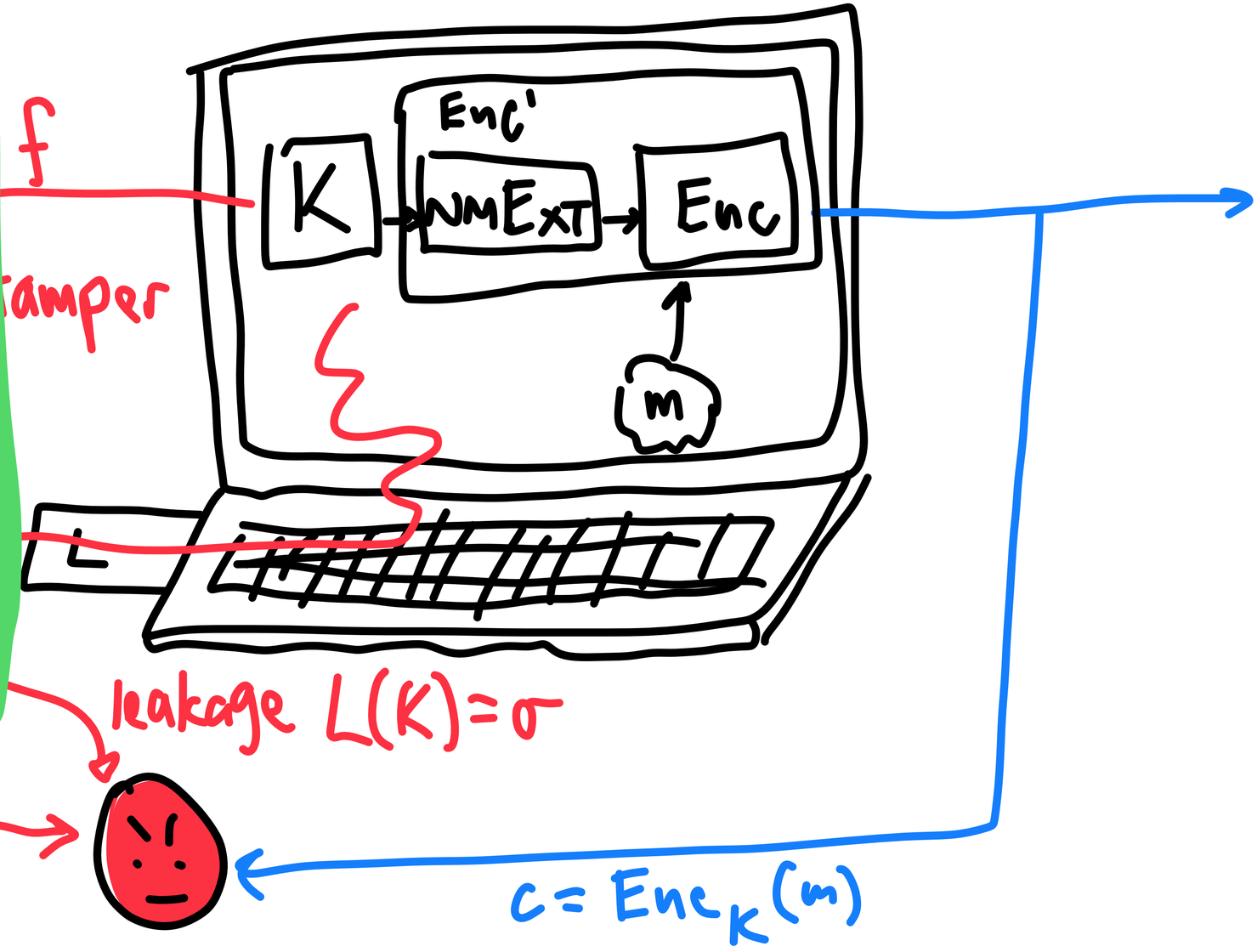


Application: tamper + leakage resilience



Application: tamper + leakage resilience

- K conditioned on short leakage = high entropy source!
- extracted key, NMExt(K), is random, even given $\hat{c}$!



2-SOURCE t-NON-MALLEABLE EXTRACTORS

source class

$$\boxed{X} \quad \boxed{Y}$$

independent

tampering class $\{(f,g)\}$

$$\begin{tikzcd} \boxed{X} \arrow[d, "f"] & \boxed{Y} \arrow[d, "g"] \\ \boxed{X'} & \boxed{Y'} \end{tikzcd}$$

explicit constructions!

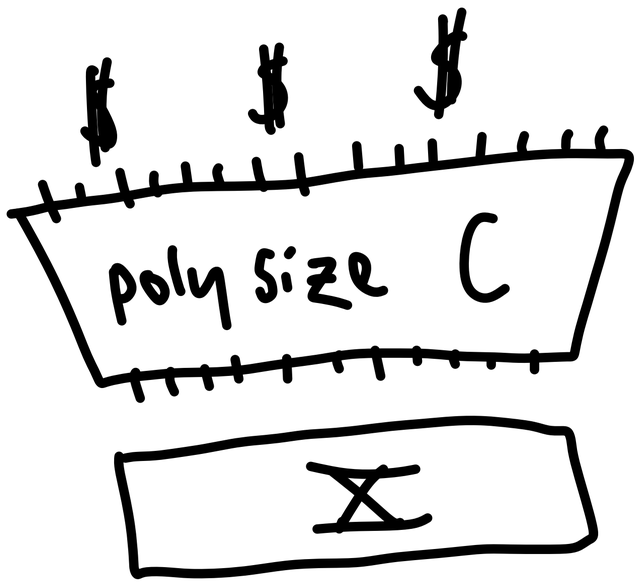
[Chattopadhyay Goyal Li '16]

applications:

- split-state non-malleable codes
- low entropy 2-source extractors (and explicit ramsey graphs)
- other non-malleable extractors

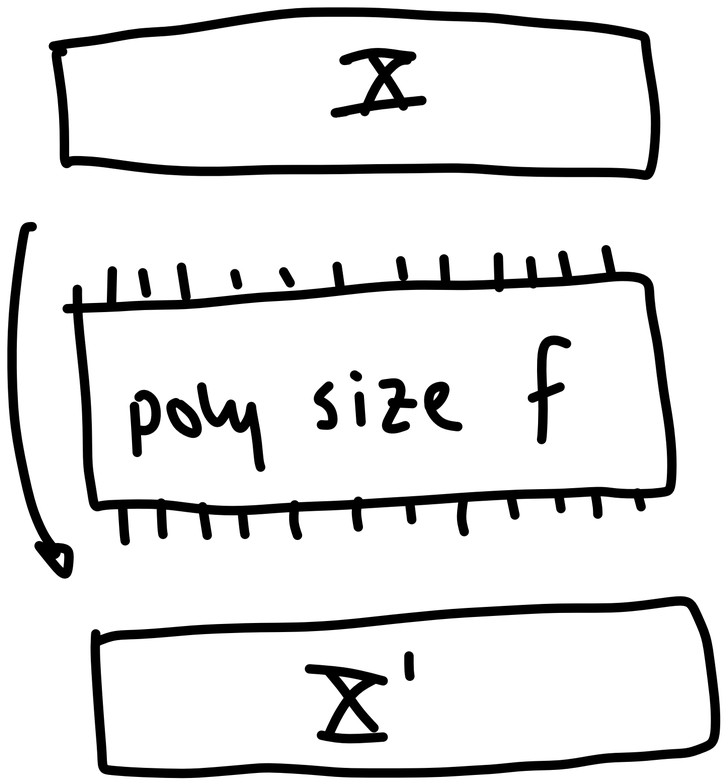
This Work: (bounded) poly size

Sources: poly size samplable



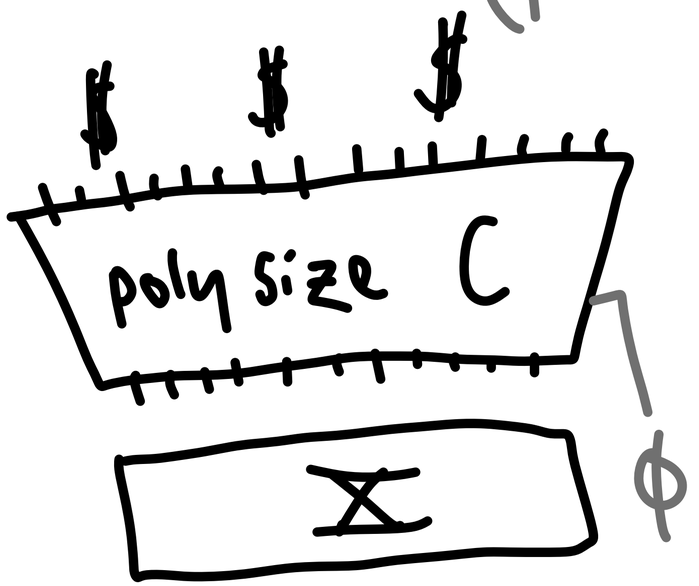
$X = C(\$)$

tampering: poly size computable



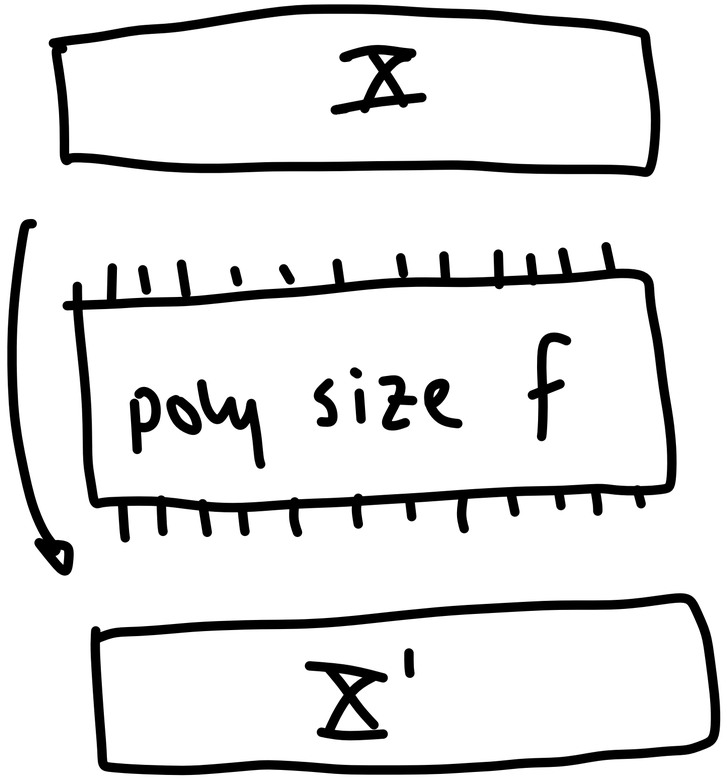
This Work: (bounded) poly size

Sources: poly size samplable (post-selecting)



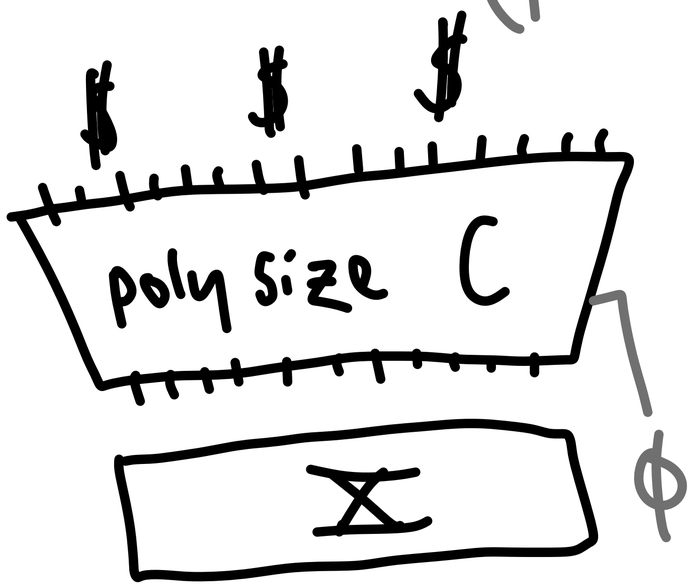
$X = C(\$)$ conditioned on $\phi = 1$

tampering: poly size computable



This Work: (bounded) poly size

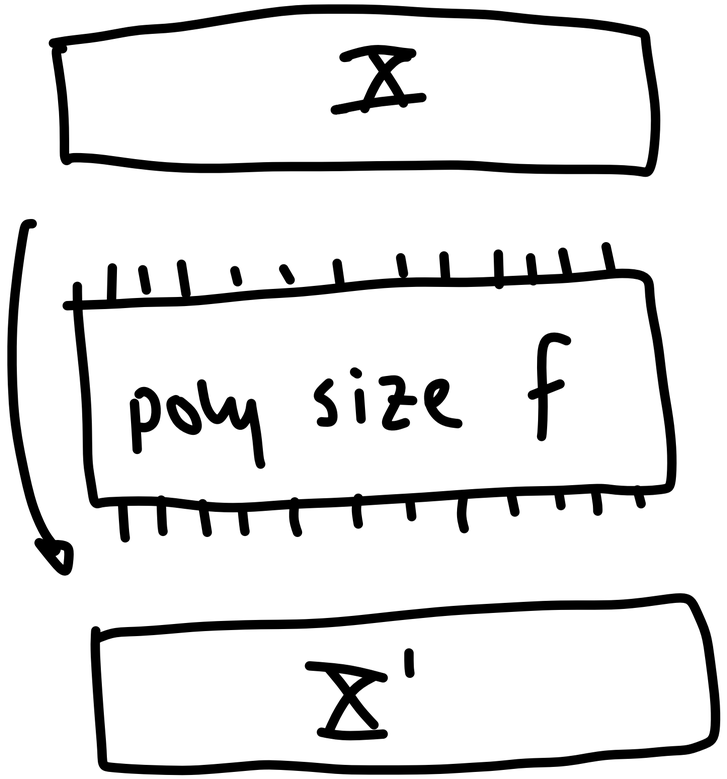
Sources: poly size samplable (post-selecting)



captures leakage on samplable sources

$X = C(\$)$ conditioned on $\phi = 1$

tampering: poly size computable



Informal Theorem

assume E requires exp size nondeterministic circuits.

for any $s(n) = \mathrm{poly}(n)$,

$\exists$ constant c & efficient $\tilde{O}(n)$-non-malleable extractor

for n-bit sources w/ min-entropy $k \geq c \cdot n$

samplable by size $s(n)$ post selecting circuits

and size $s(n)$ tampering

with statistical error $\frac{1}{n^{\Omega(1)}}$.

Informal Theorem

assume E requires exp size nondeterministic circuits.

for any $s(n) = \mathrm{poly}(n)$,

$\exists$ constant c & efficient $\tilde{O}(n)$-non-malleable extractor

for n-bit sources w/ min-entropy $k \geq c \cdot n$

samplable by size $s(n)$ post selecting circuits

and size $s(n)$ tampering

with statistical error $\frac{1}{n^{\Omega(1)}}$.

☹ bad for building crypto

☺ fine for ruling it out!

(part II)

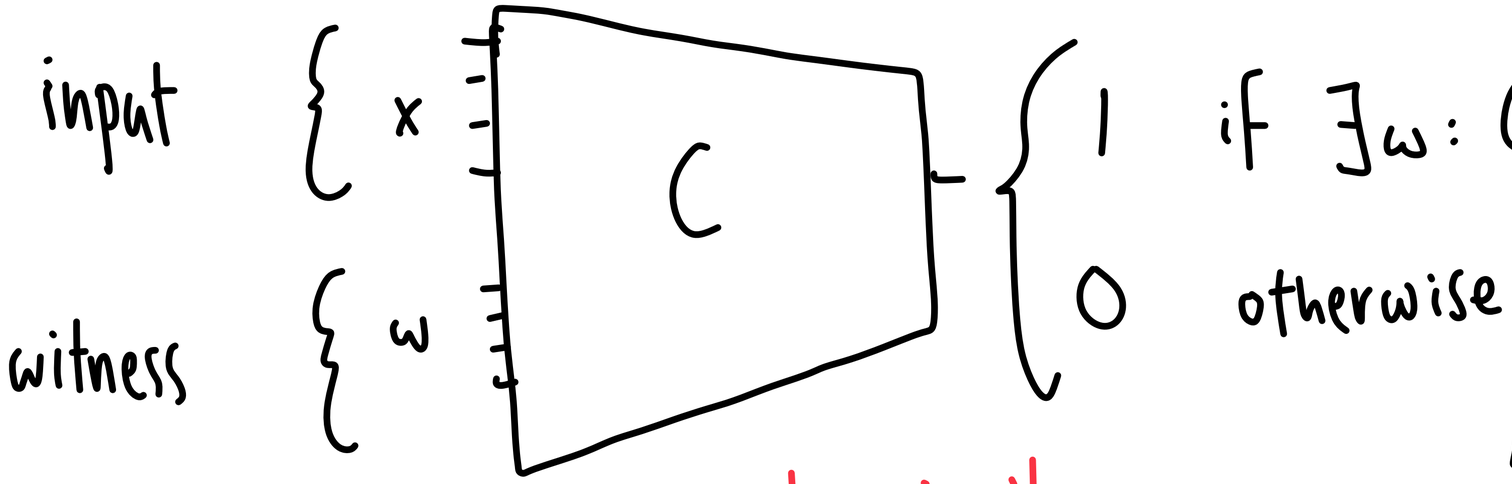
matches state of art for regular extractors

ASSUMPTION

can use to show AM=NP

$\exists\, L \in E = DTIME[2^{O(n)}]$ s.t. L requires nondeterministic circuits of size $2^{\Omega(n)}$

"nondeterminism + nonuniformity doesn't speedup all computations"



nondeterministic circuit

essentially minimal for source class

Improvement over State of Art

[Ball Dachman-Soled Loss '22]

1. many ($t = \tilde{O}(n)$) tamperings

   (prior: $t = 1$)

2. wider source class: post selecting sources

   (prior: samplable and recognizable)

3. better assumptions: nondeterministic circuit lower bound

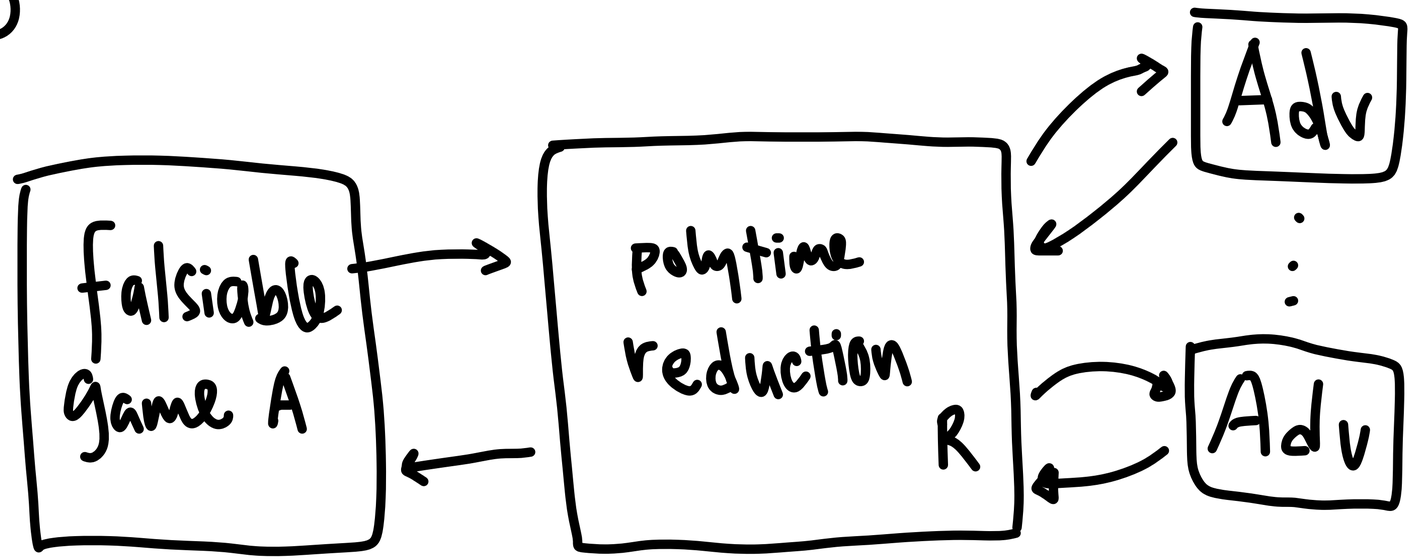
   (prior: Σ_2-circuit lower bound)

II. a new APPLICATION:

impossibility of <u>uniform</u> black-box reductions

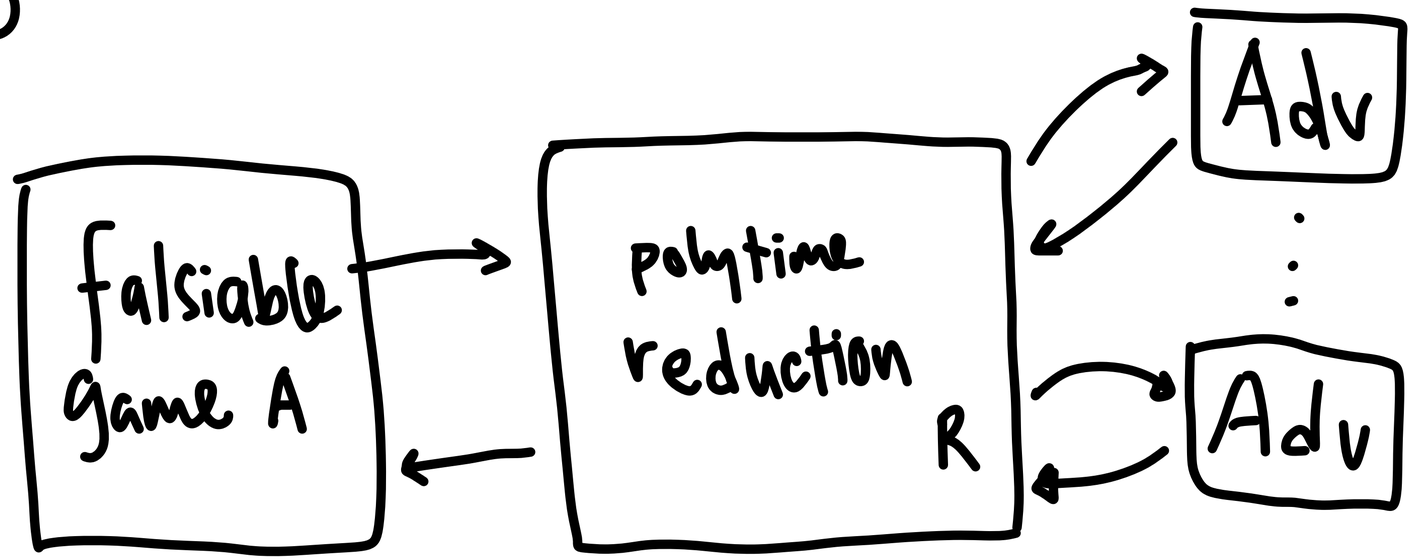
Black-Box Reductions

"$A \overset{BB}{\Longrightarrow} B$"



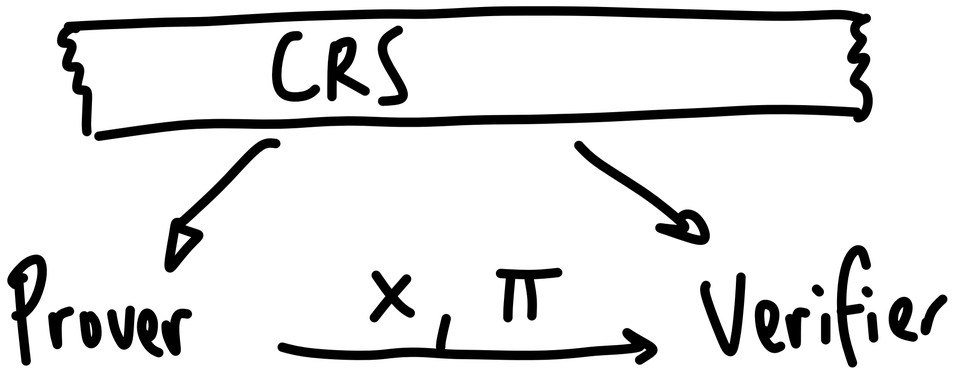
Black-Box Reductions

"$A \overset{BB}{\Rightarrow} B$"



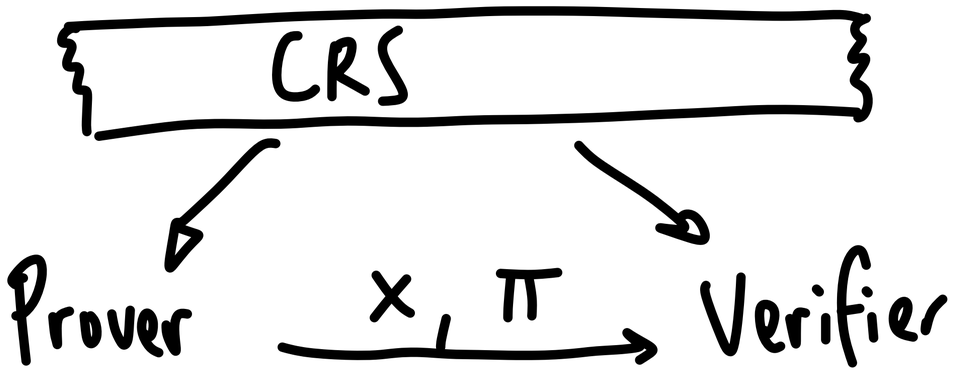
Promise: if Adv breaks B, then R^{Adv} breaks A.
(of Reduction)

SNARGs for L



π very short

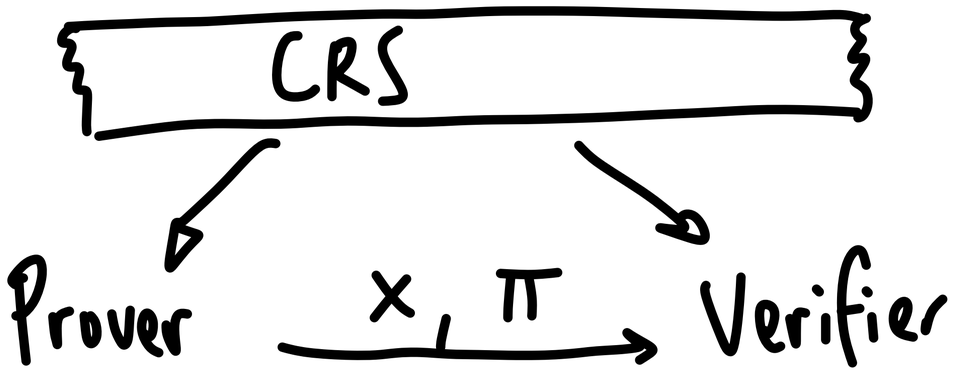
SNARGs for L



<u>completeness</u>: $x \in L \Rightarrow$ Verifier accepts

π very short

SNARGs for L



π very short

<u>completeness</u>: $x \in L \Rightarrow$ Verifier accepts

<u>soundness</u>: hard to find $(\hat{x}, \hat{\pi})$ given CRS

s.t. $\hat{x} \notin L \wedge$ Verifier accepts

Gentry Wichs Barrier (2011)

Let L be hard NP-language and A falsifiable assumption.

if $A \xRightarrow{BB}$ SNARG for L, then A broken

Gentry Wichs Barrier (2011)

Let L be hard NP-language and A falsifiable assumption.

if $A \xRightarrow{BB}$ SNARG for L, then A broken

CAVEATS:

1. A is broken by non-uniform polytime.

Gentry Wichs Barrier (2011)

Let L be hard NP-language and A falsifiable assumption.

if $A \xRightarrow{BB}$ SNARG for L, then A broken

CAVEATS:

1. A is broken by non-uniform polytime.
2. BB reduction does NOT control Adv's random tape.

Gentry Wichs Barrier (2011)

Let L be hard NP-language and A falsifiable assumption.

if $A \xRightarrow{BB}$ SNARG for L, then A broken

CAVEATS:

1. A is broken by non-uniform polytime. uniform simulation: [Vadhan Zheng '13]
2. BB reduction does NOT control Adv's random tape.

[Chung et al.] → use nonuniformity

Gentry Wichs Barrier (2011)

Let L be hard NP-language and A falsifiable assumption.

if $A \xRightarrow{BB}$ SNARG for L, then A broken

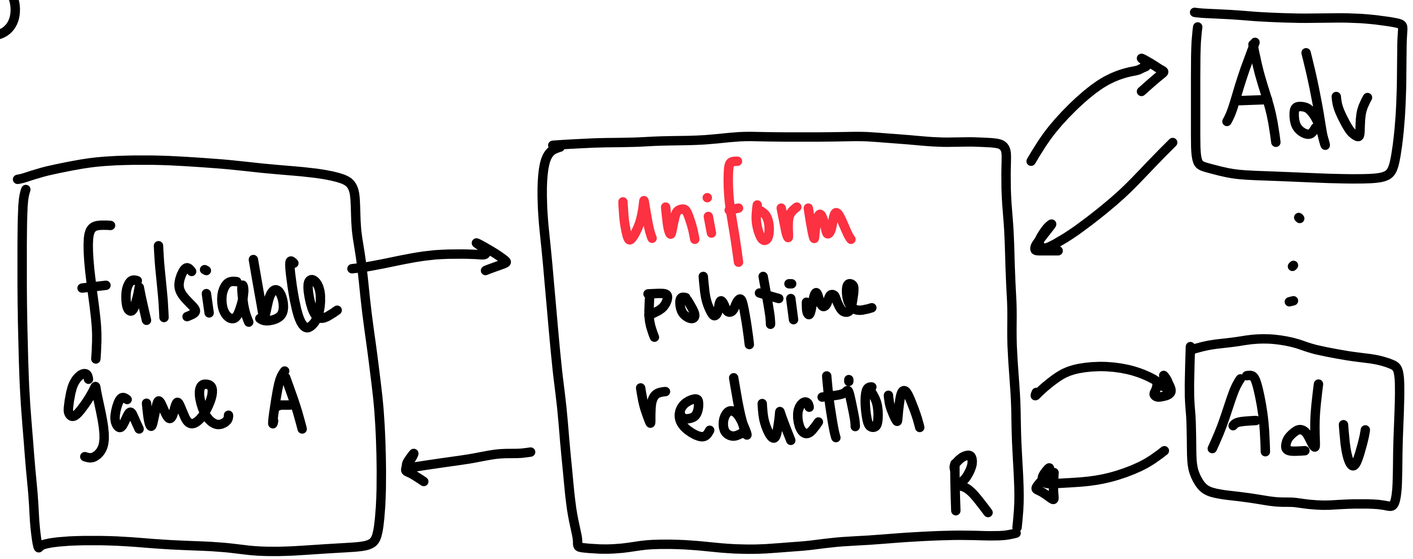
CAVEATS:

1. A is broken by non-uniform polytime. uniform simulation: [Vadhan Zheng '13]
2. BB reduction does NOT control Adv's random tape. [Chung et al.] → use nonuniformity

→ Possible to get uniform soundness by exploiting 1 AND 2?

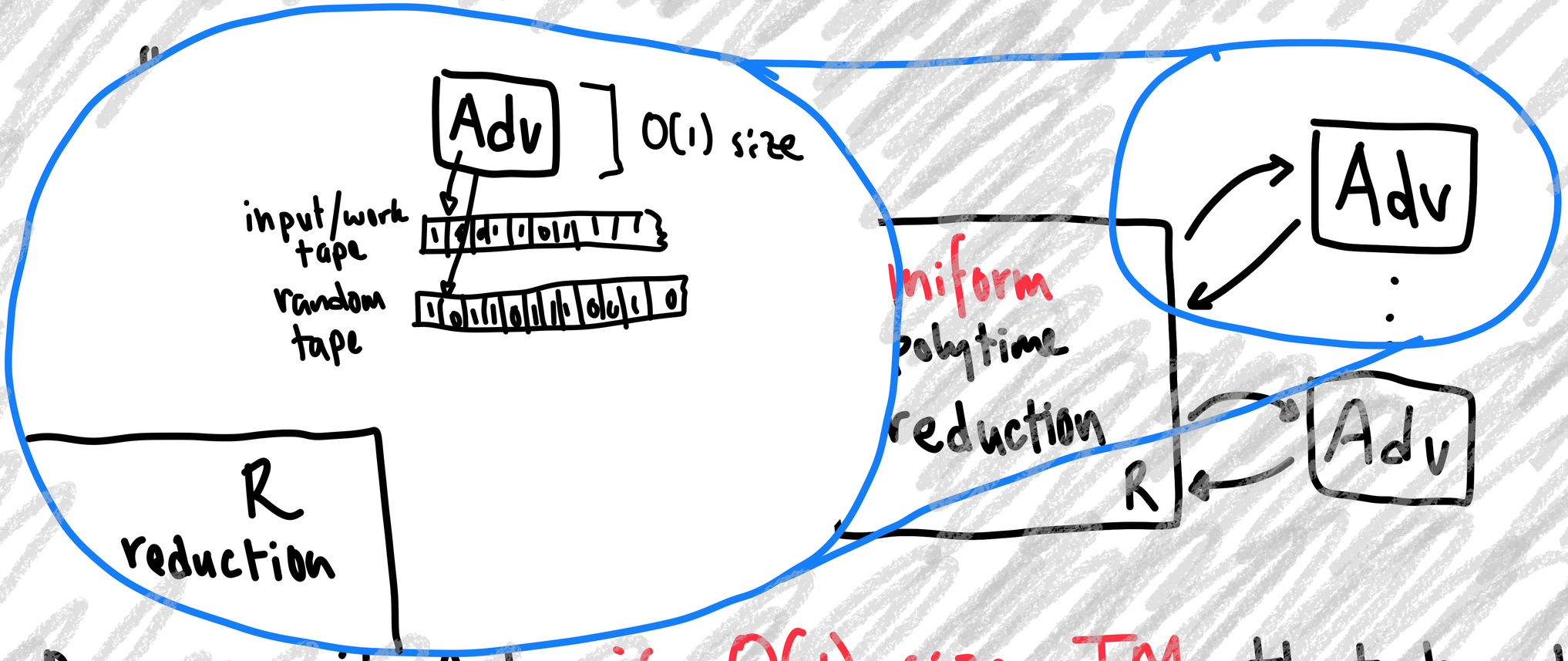
Uniform Black-Box Reductions

"$A \overset{BB}{\Rightarrow} B$"



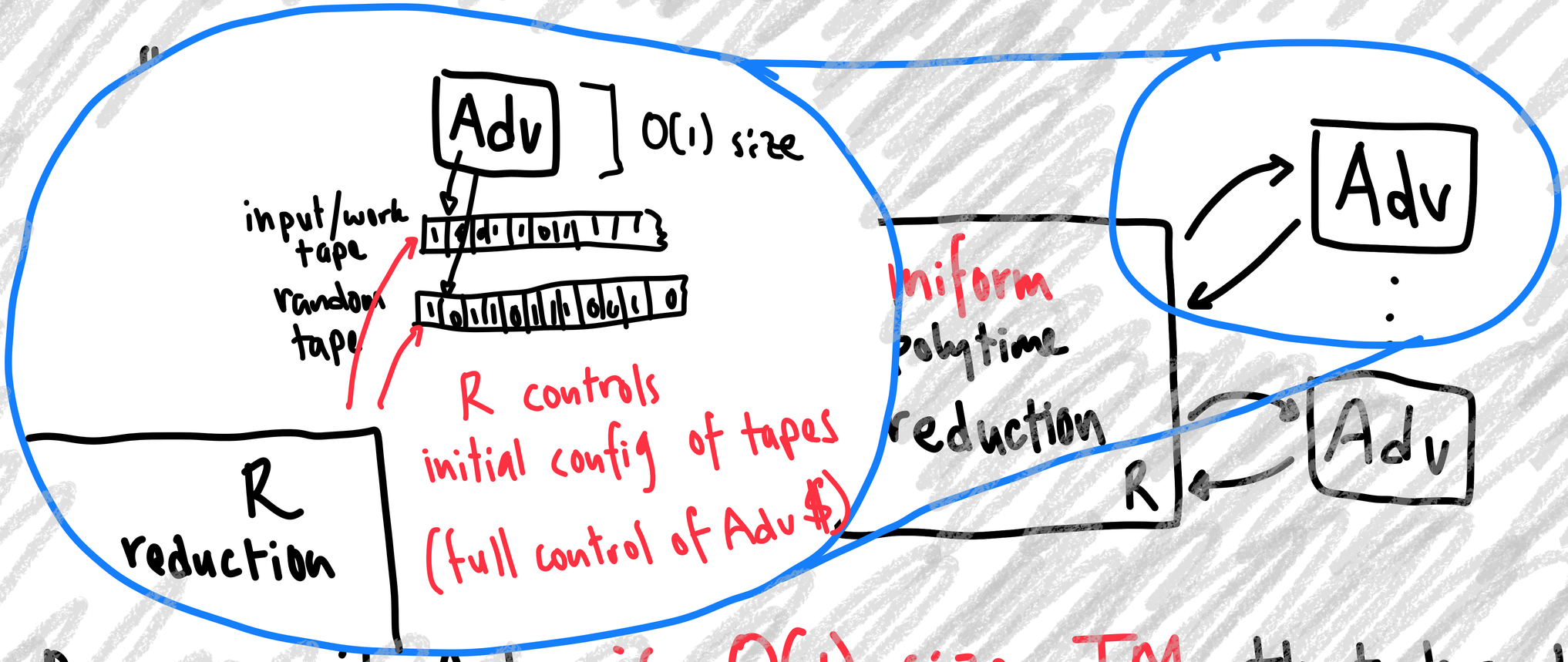
Promise: if Adv is $O(1)$ size TM that breaks B,
(of Reduction)
then R^{Adv} breaks A.

Uniform Black-Box Reductions



Promise: if Adv is O(1) size TM that breaks B,
(of Reduction)
then R^{Adv} breaks A.

Uniform Black-Box Reductions



Promise: if Adv is O(1) size TM that breaks B,
(of Reduction)
then R^{Adv} breaks A.

This Work

assume L is hard NP-language and E requires exp size Σ_2-circuits.

if candidate SNARG for L has a

- security preserving
- non-adaptive

uniform black-box reduction to falsifiable assumption A,

then A broken in uniform poly time.

This Work

assume L is hard NP-language and E requires exp size Σ_2-circuits.

if candidate SNARG for L has a

can only query on poly-related security params → • security preserving

• non-adaptive

uniform black-box reduction to falsifiable assumption A,

then A broken in uniform poly time.

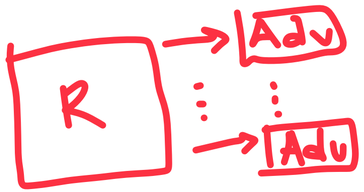
This Work

assume L is hard NP-language and E requires exp size Σ_2-circuits.

if candidate SNARG for L has a

can only query on poly-related security params → • security preserving

• non-adaptive ← all reduction queries made in parallel.



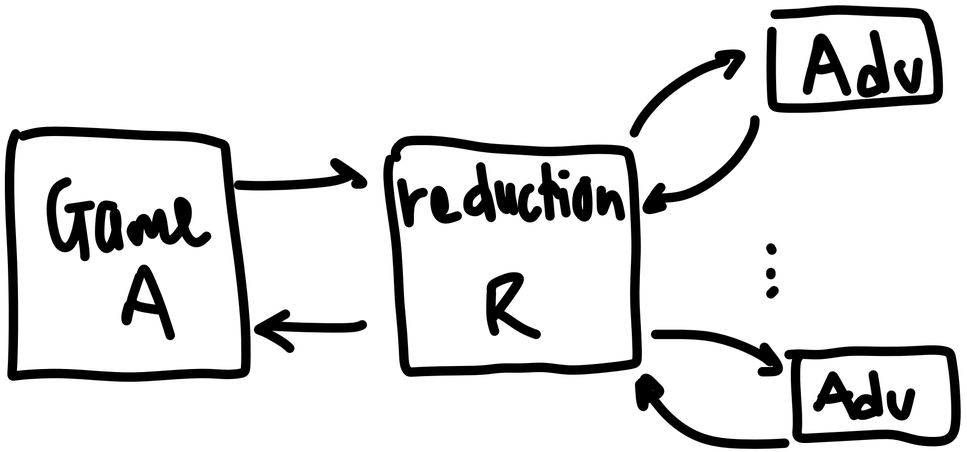
uniform black-box reduction to falsifiable assumption A,

then A broken in uniform poly time.

a (very) brief overview
of
Gentry Wichs

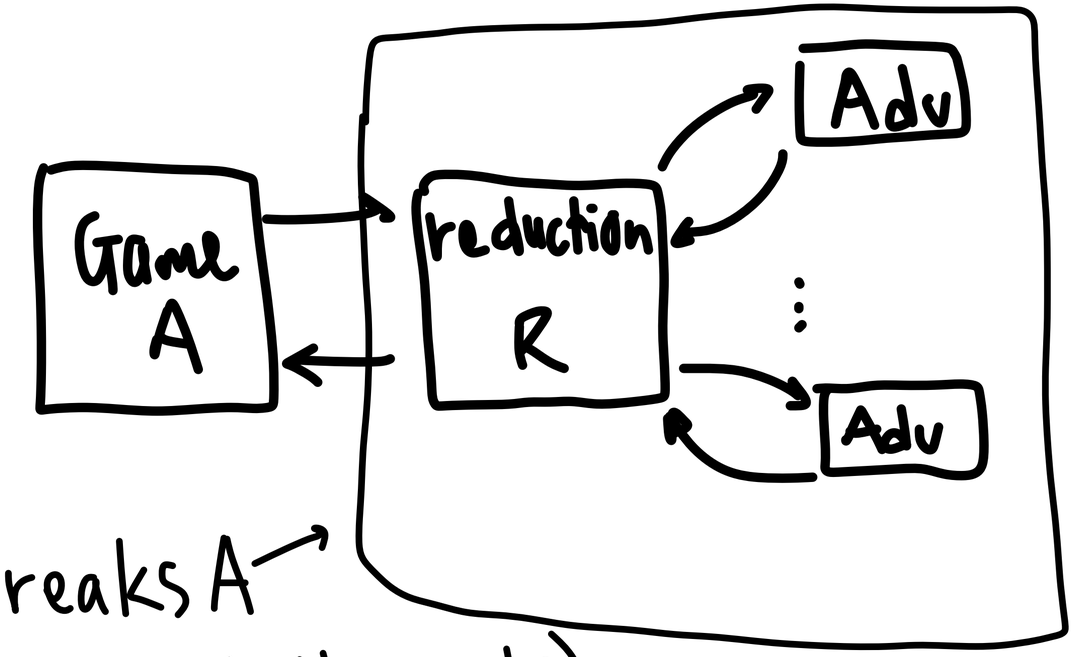
Metareduction Framework: A $\overset{\text{PPT}}{\not\Rightarrow}$ SNARG

1. construct inefficient Adv that breaks SNARG



Metareduction Framework: A $\overset{BB}{\not\Rightarrow}$ SNARG

1. construct inefficient Adv that breaks SNARG

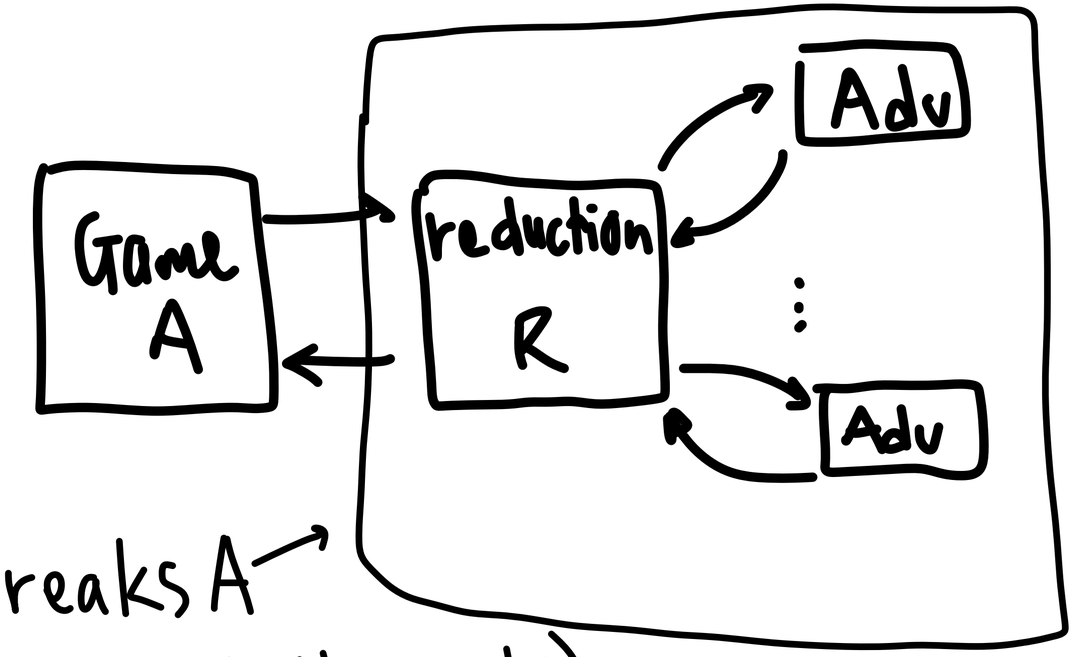


breaks A
(b/c BB-reduction works)
but not efficient.

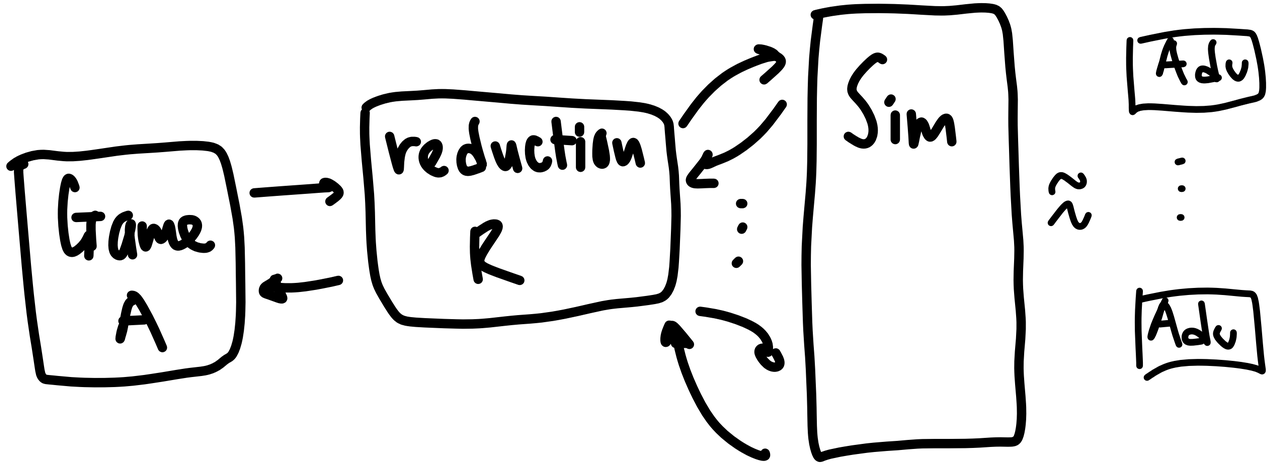
Metareduction Framework: A $\overset{BB}{\not\Rightarrow}$ SNARG

1. construct inefficient Adv that breaks SNARG

2. simulate Adv efficiently to fool R+A

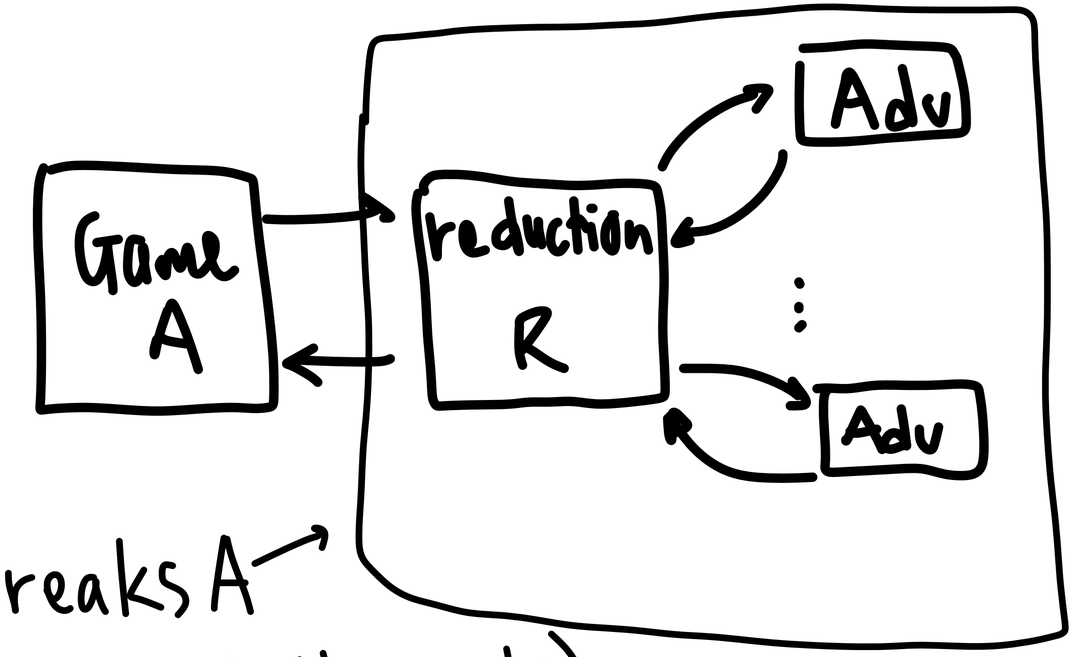


breaks A
(b/c BB-reduction works)
but not efficient.



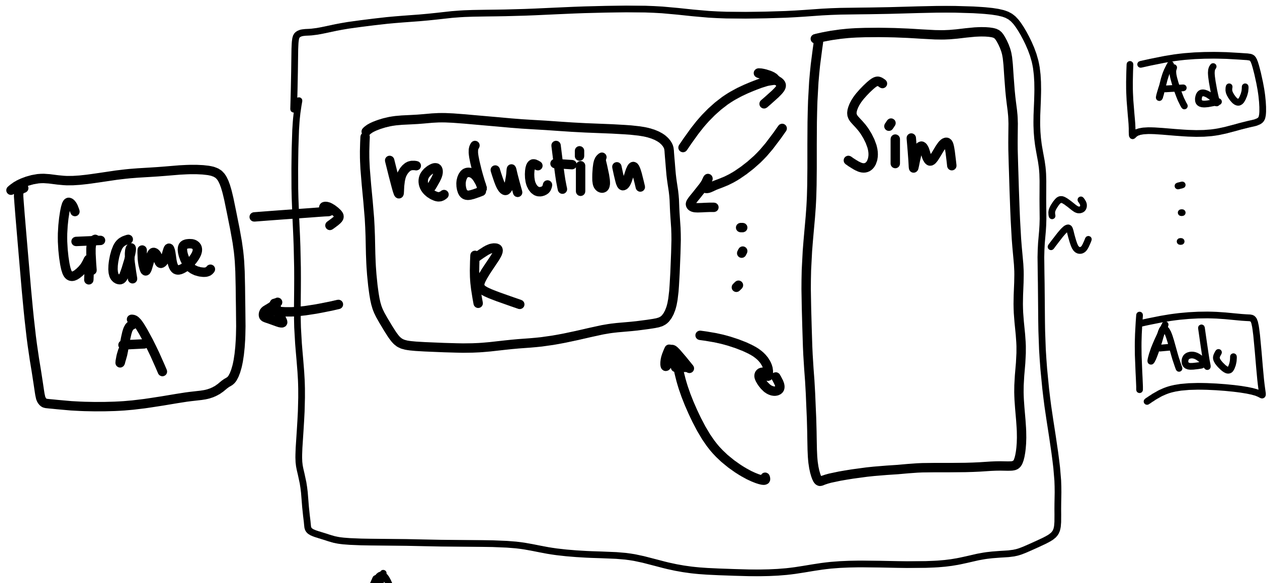
Metareduction Framework: A $\overset{BB}{\not\Rightarrow}$ SNARG

1. construct inefficient Adv that breaks SNARG



breaks A
(b/c BB-reduction works)
but not efficient.

2. simulate Adv efficiently to fool R+A

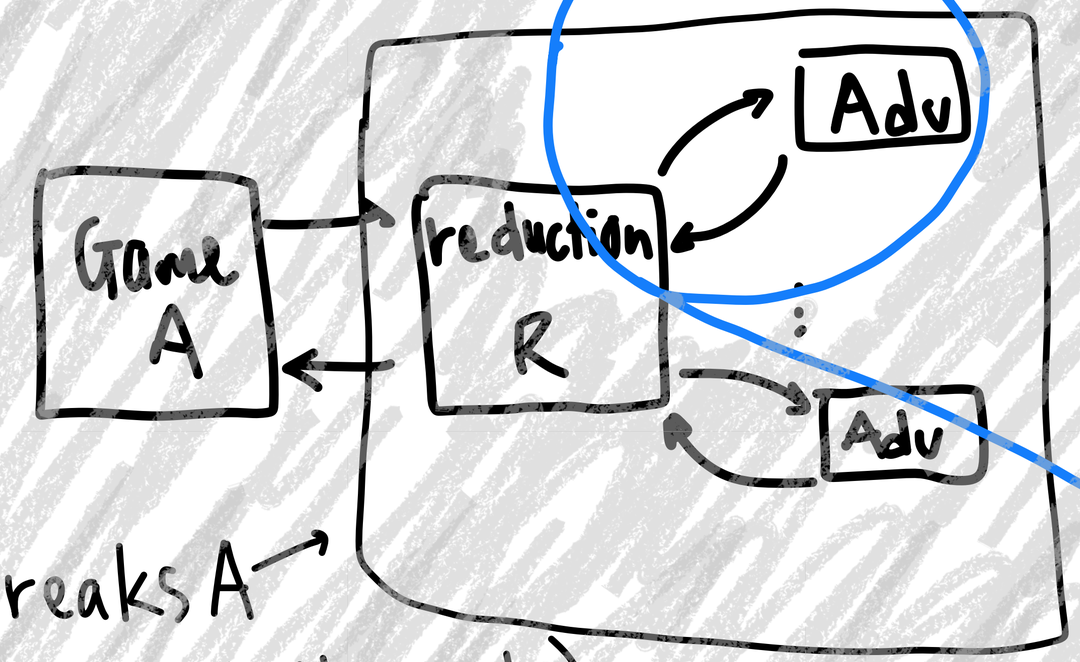


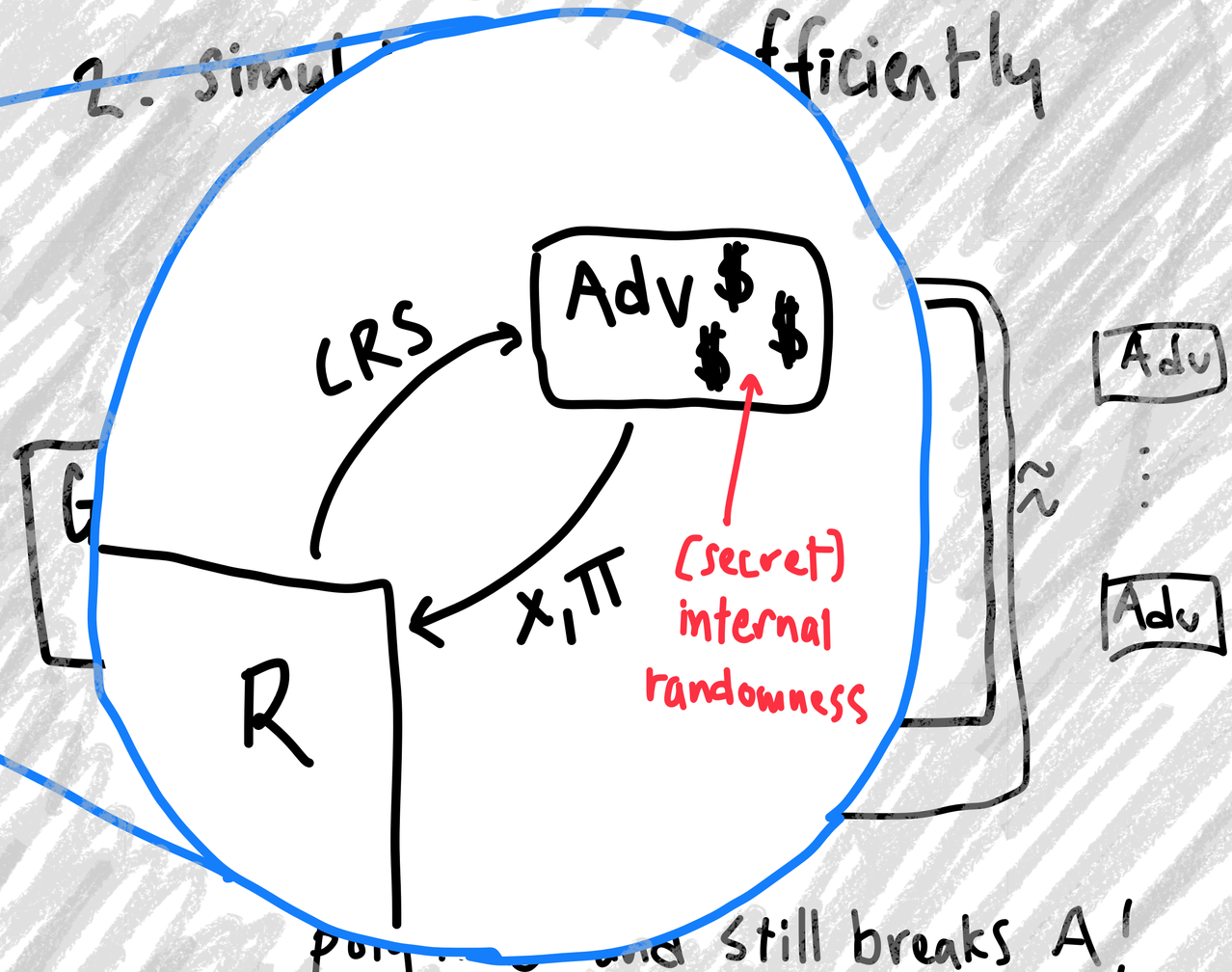
poly time and still breaks A!

Metareduction Framework: A $\overset{BB}{\not\Rightarrow}$ SNARG

1. construct inefficient Adv that breaks SNARG

2. simul[illegible]fficiently





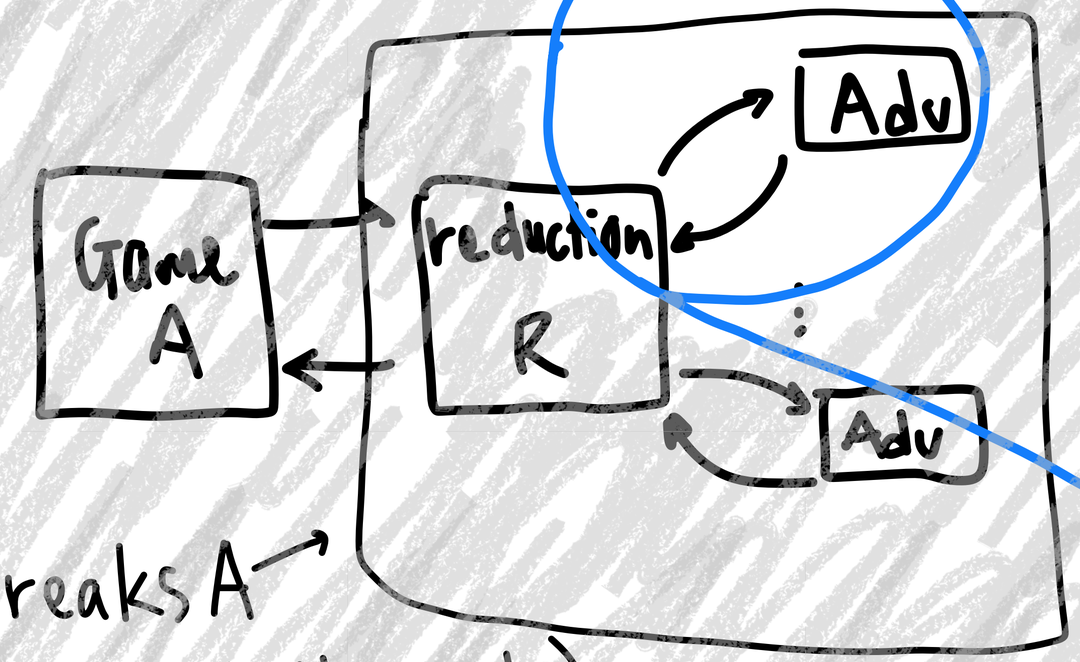
breaks A
(b/c BB-reduction works)
but not efficient.

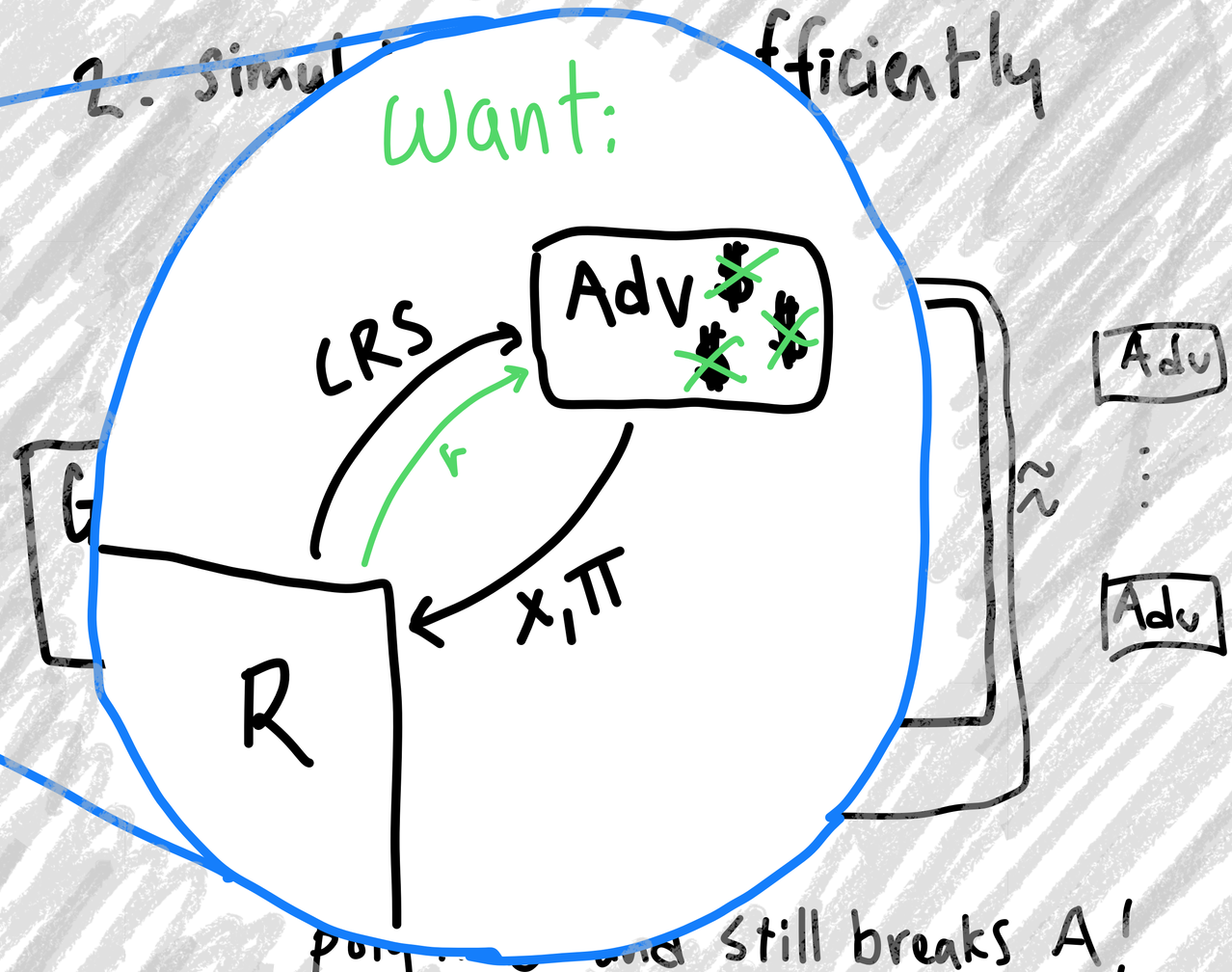
[illegible] and still breaks A!

Metareduction Framework: A $\overset{BB}{\not\Rightarrow}$ SNARG

1. construct inefficient Adv that breaks SNARG

2. simul[illegible]fficiently





breaks A
(b/c BB-reduction works)
but not efficient.

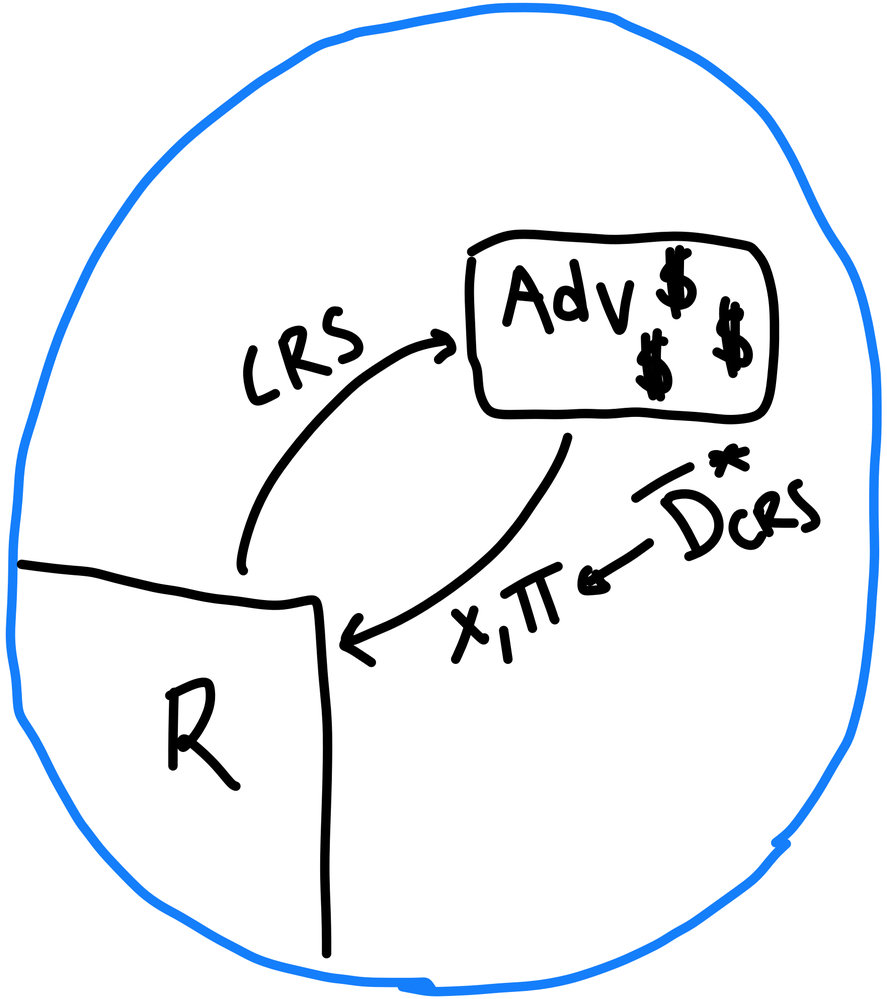
[illegible] and still breaks A!

Leakage Simulation

L hard $\Rightarrow$ D supported on L

$\overline{D}$ supported on $\overline{L}$

s.t. $D \approx \overline{D}$.

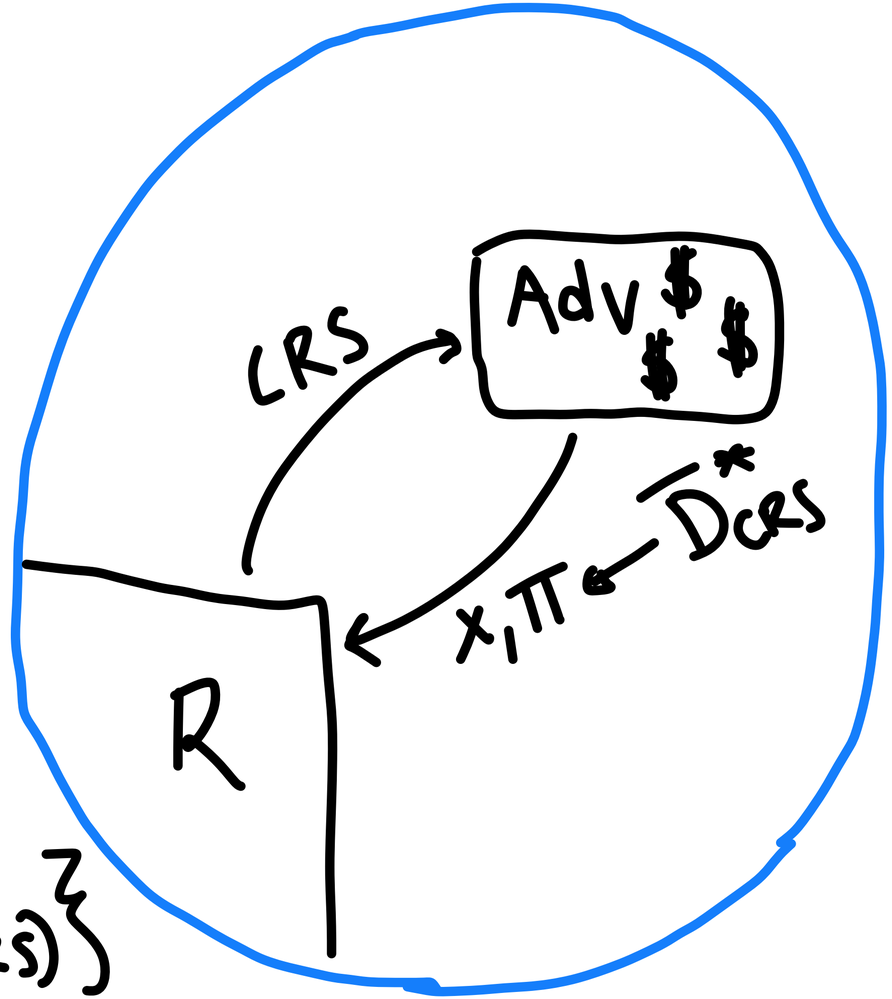


Leakage Simulation

L hard $\Rightarrow$ D supported on L

$\overline{D}$ supported on $\overline{L}$

s.t. $D \approx \overline{D}$.



if π short, then

$\exists\ \overline{D}^*_{CRS} = \{(\bar{x}, \bar{\pi}) : \bar{x} \leftarrow \overline{D}\}$

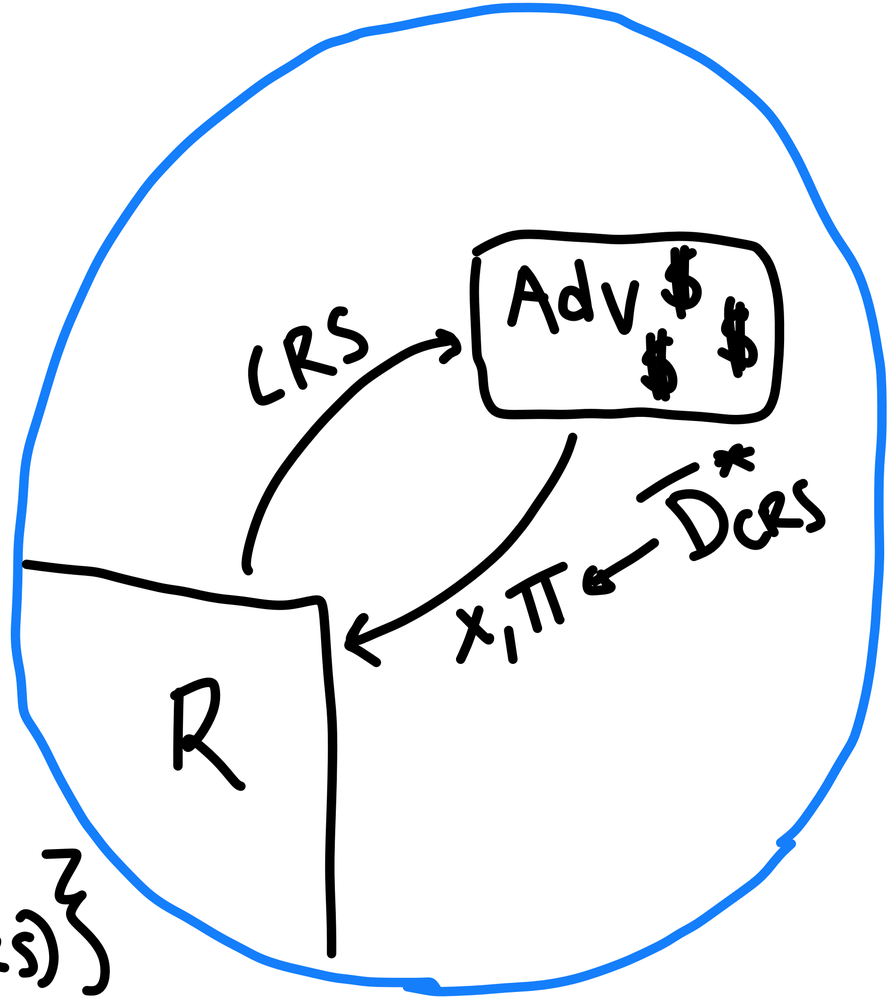
s.t. $\overline{D}^*_{CRS} \approx D^*_{CRS} = \left\{(x, \pi) : \begin{array}{l} x \leftarrow D \\ \pi \leftarrow P(x, CRS) \end{array}\right\}$

Leakage Simulation

L hard $\Rightarrow$ D supported on L

$\overline{D}$ supported on $\overline{L}$

s.t. $D \approx \overline{D}$.



if π short, then

$$\exists\ \overline{D}^*_{CRS} = \{(\bar{x}, \bar{\pi}) : \bar{x} \leftarrow \overline{D}\}$$

s.t. $\overline{D}^*_{CRS} \approx D^*_{CRS} = \left\{(x, \pi) : \begin{array}{l} x \leftarrow D \\ \pi \leftarrow P(x, CRS) \end{array}\right\}$

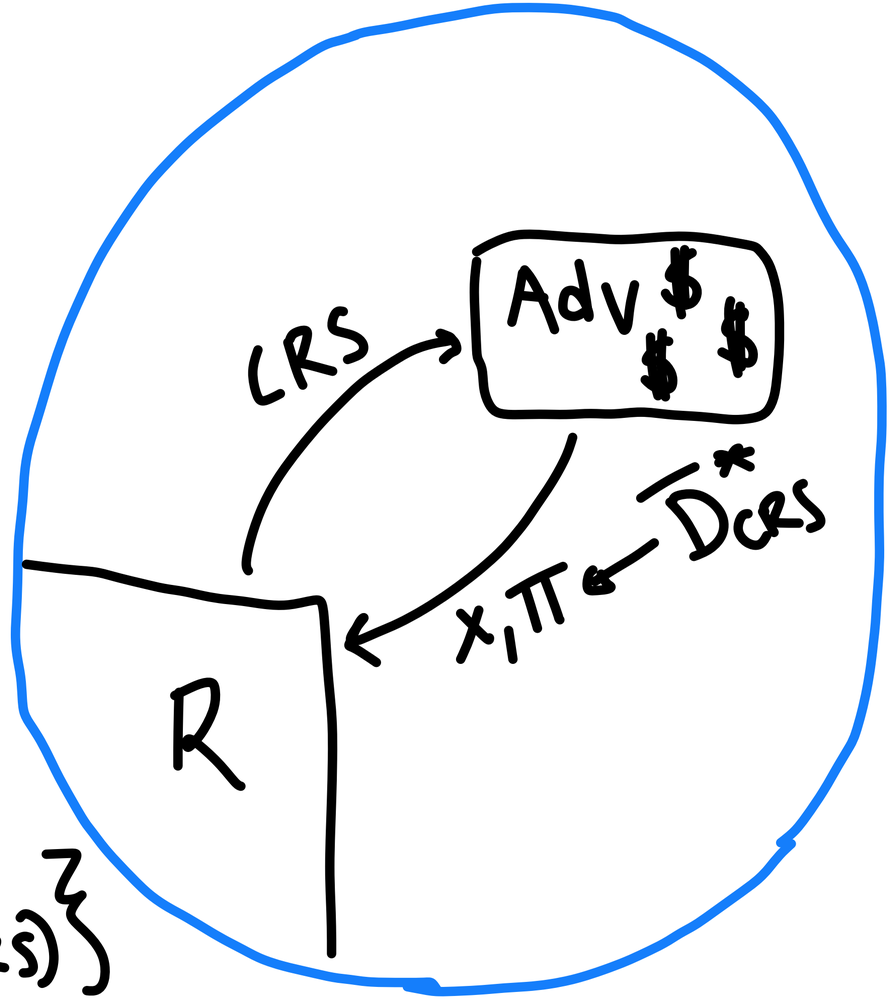
breaks soundness (inefficiently)

Leakage Simulation

L hard $\Rightarrow$ D supported on L

$\overline{D}$ supported on $\overline{L}$

s.t. $D \approx \overline{D}$.



if π short, then

$$\exists\ \overline{D}^*_{CRS} = \{(\bar{x}, \bar{\pi}) : \bar{x} \leftarrow \overline{D}\}$$

s.t. $\overline{D}^*_{CRS} \approx D^*_{CRS} = \left\{(x, \pi) : \begin{array}{l} x \leftarrow D \\ \pi \leftarrow P(x, CRS) \end{array}\right\}$

will use to simulate (efficiently)

breaks soundness (inefficiently)

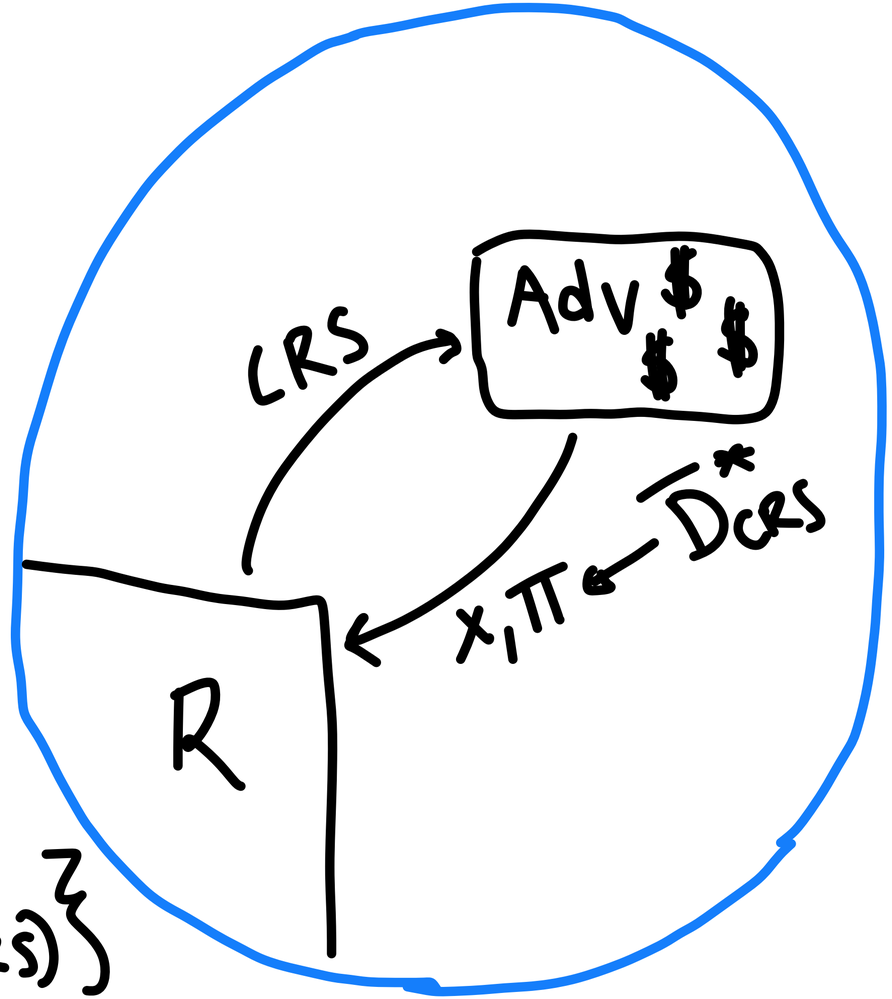
Leakage Simulation

[Jetchev Pietrzak '14]: constructive $\overline{D}^*_{CRS}$

L hard $\Rightarrow$ D supported on L

$\overline{D}$ supported on $\overline{L}$

s.t. $D \approx \overline{D}$.



if π short, then

$\exists\ \overline{D}^*_{CRS} = \{(\bar{x}, \bar{\pi}) : \bar{x} \leftarrow \overline{D}\}$

s.t. $\overline{D}^*_{CRS} \approx D^*_{CRS} = \left\{(x, \pi) : \begin{array}{l} x \leftarrow D \\ \pi \leftarrow P(x, CRS) \end{array}\right\}$

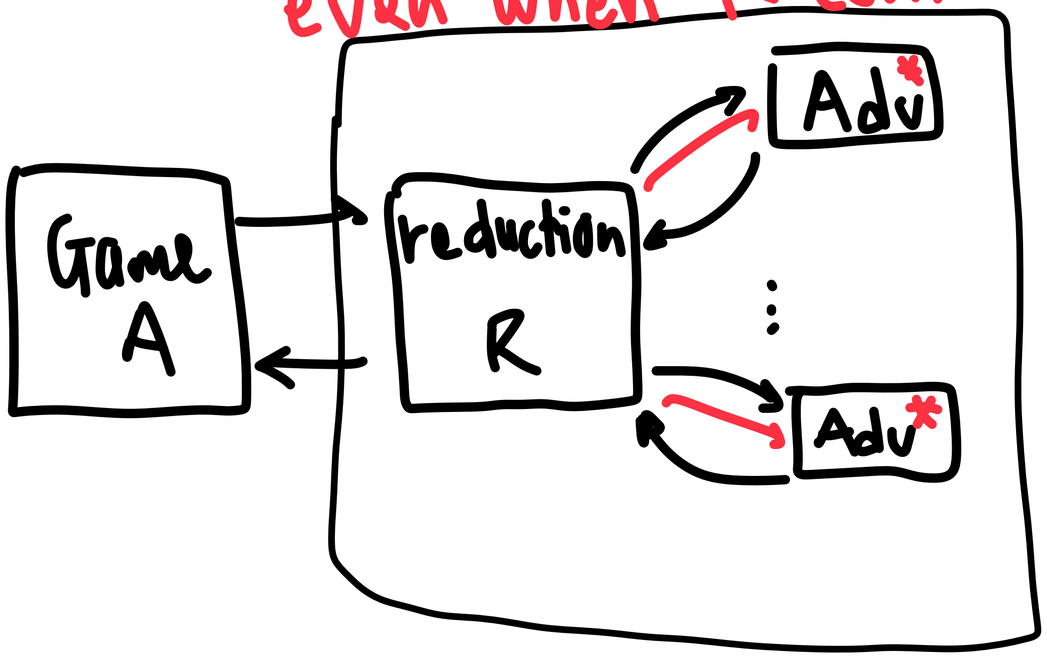
will use to simulate (efficiently)

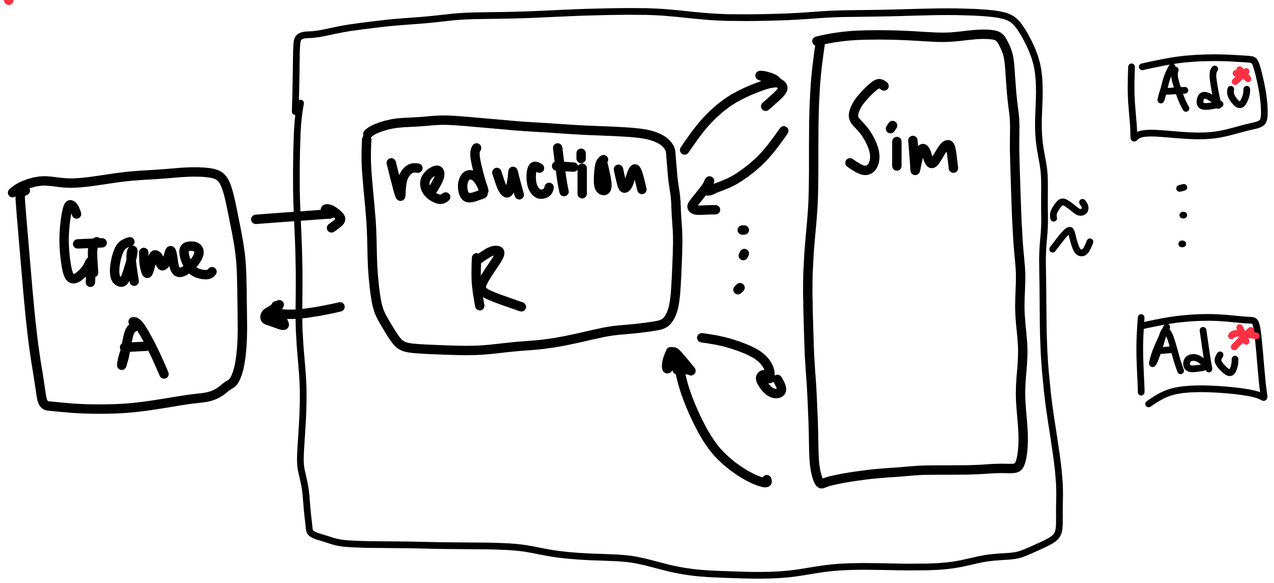
breaks soundness (inefficiently)

How to rule out UNIFORM BB-reductions?

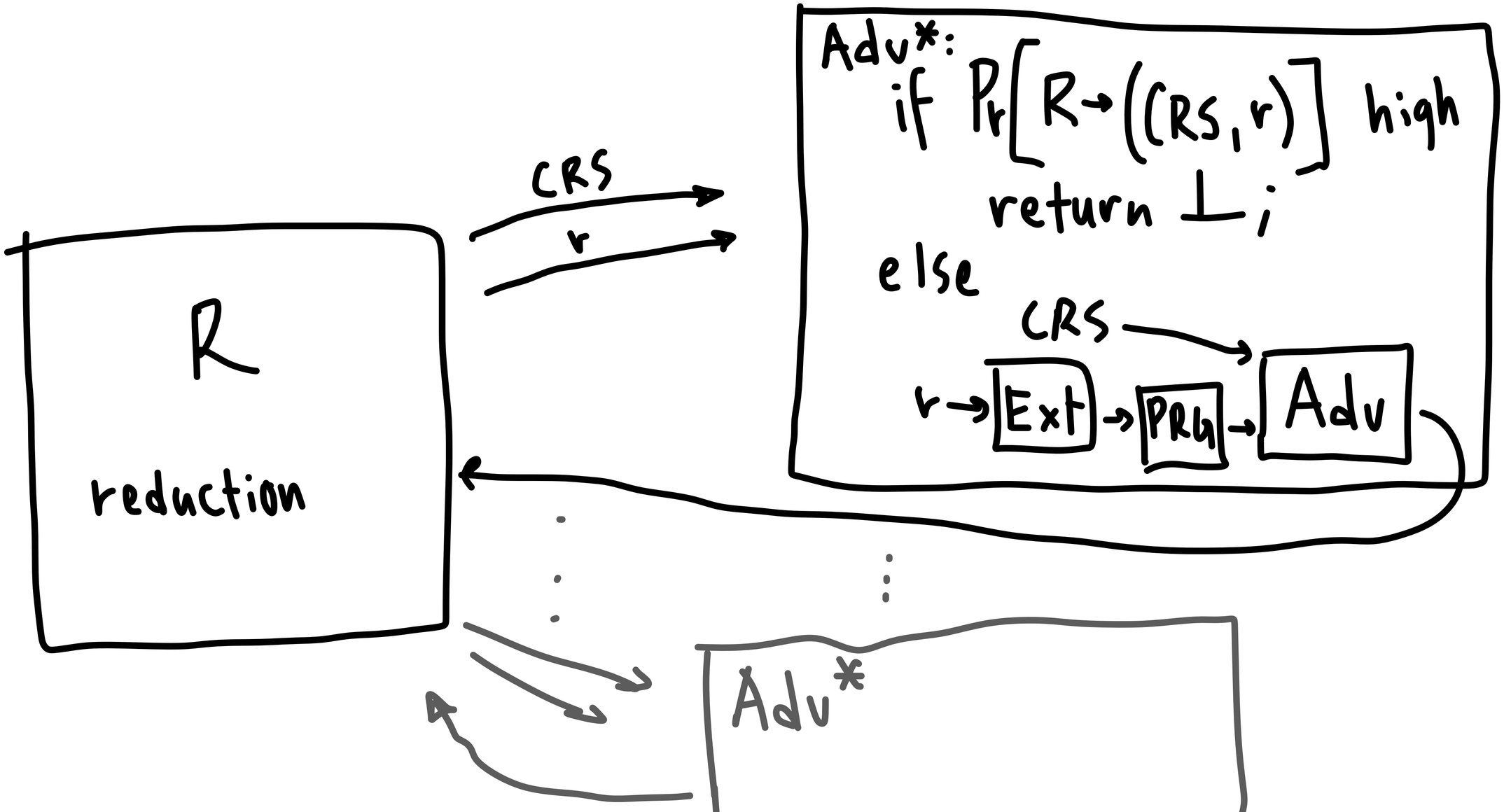
1. construct inefficient O(1) sized Adv* that breaks SNARG even when R controls #

2. simulate Adv* efficiently to fool R+A



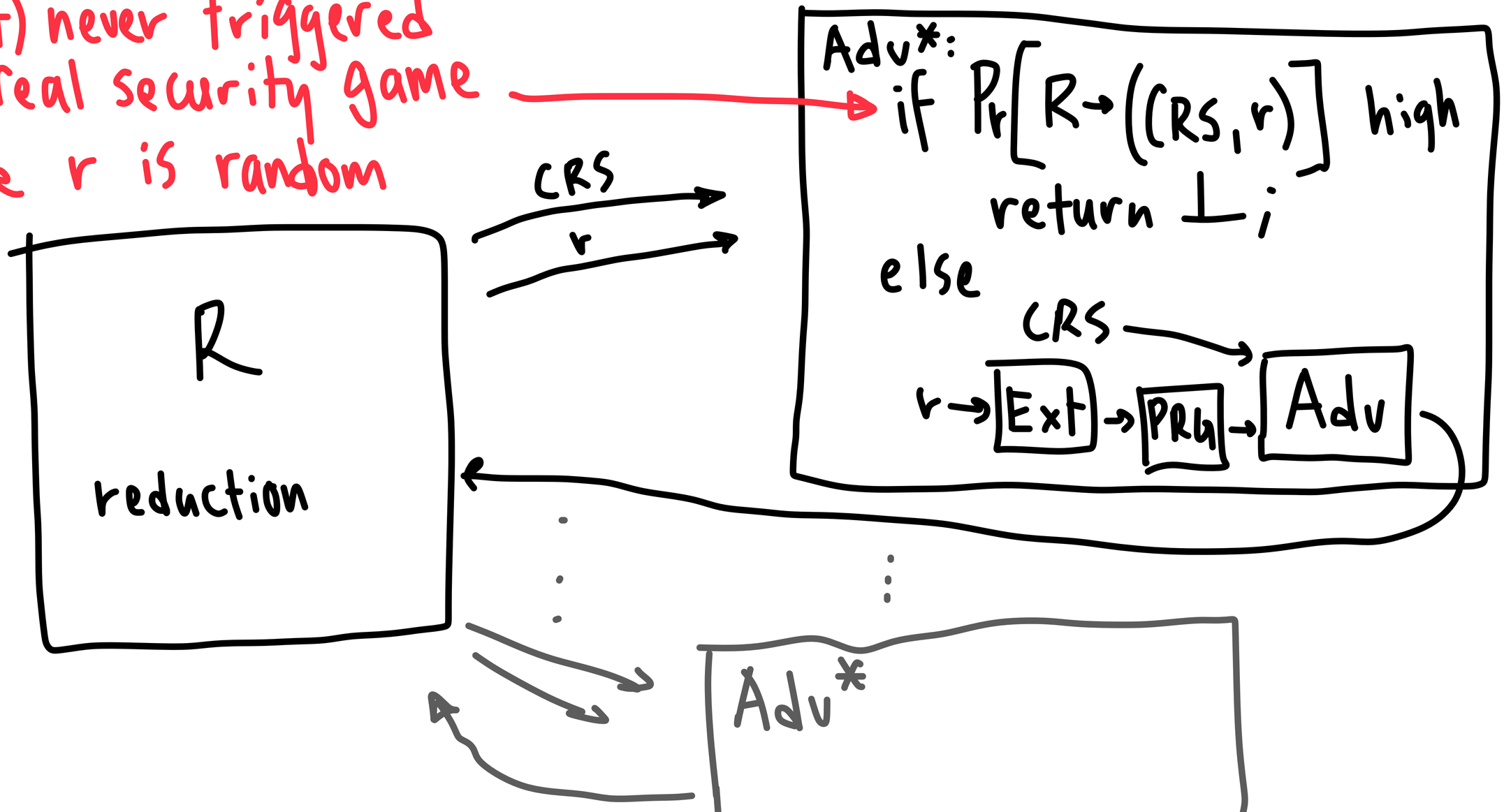


Initial Attempt

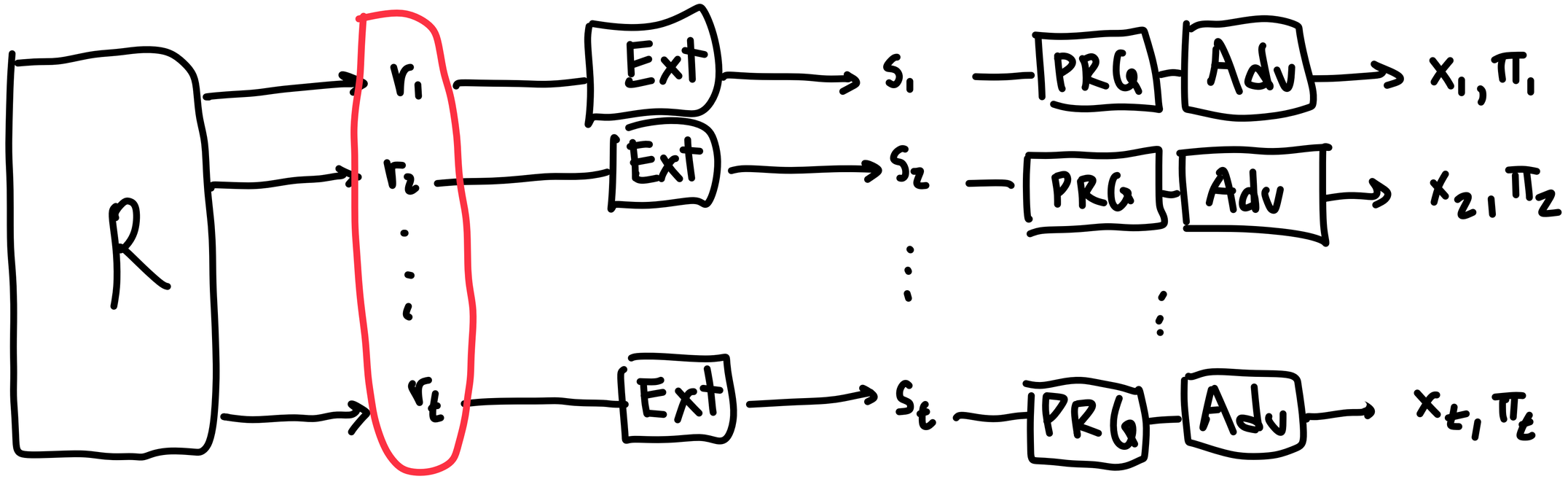


Initial Attempt

(almost) never triggered in real security game where r is random

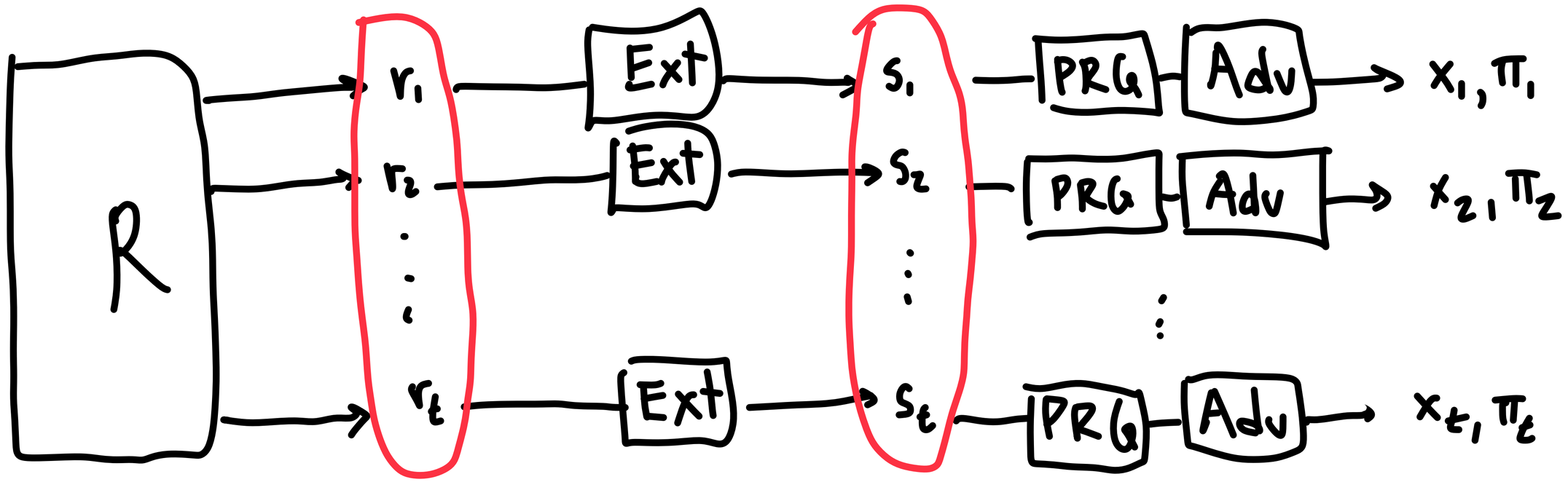


Problem 1: correlated queries



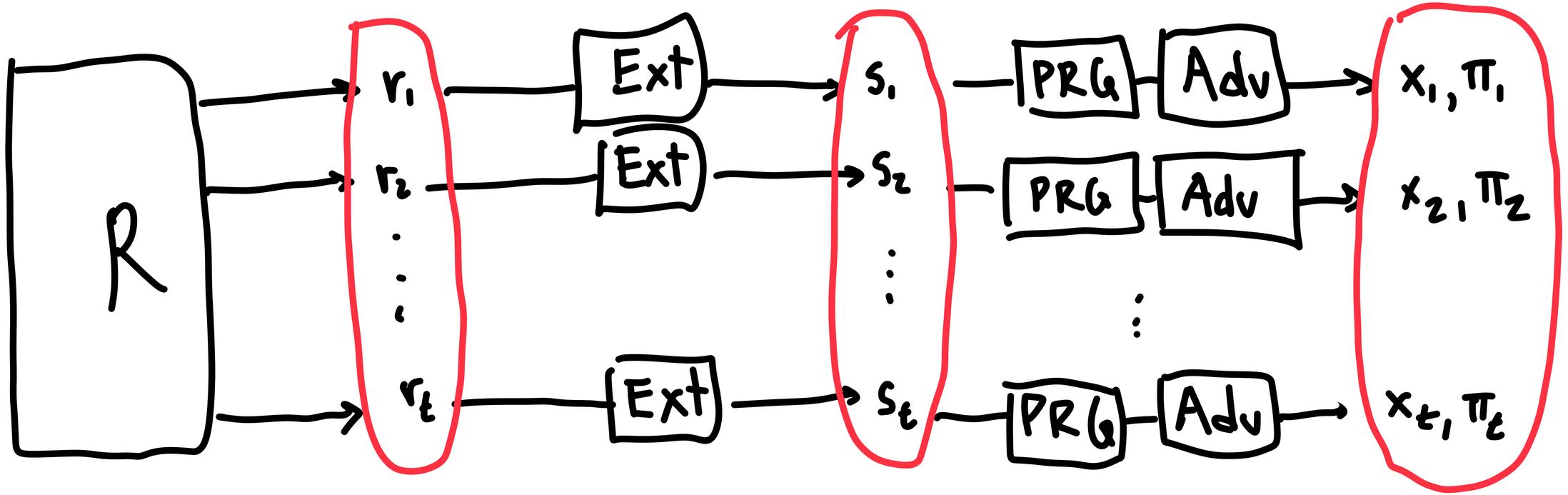
Correlated

Problem 1: correlated queries



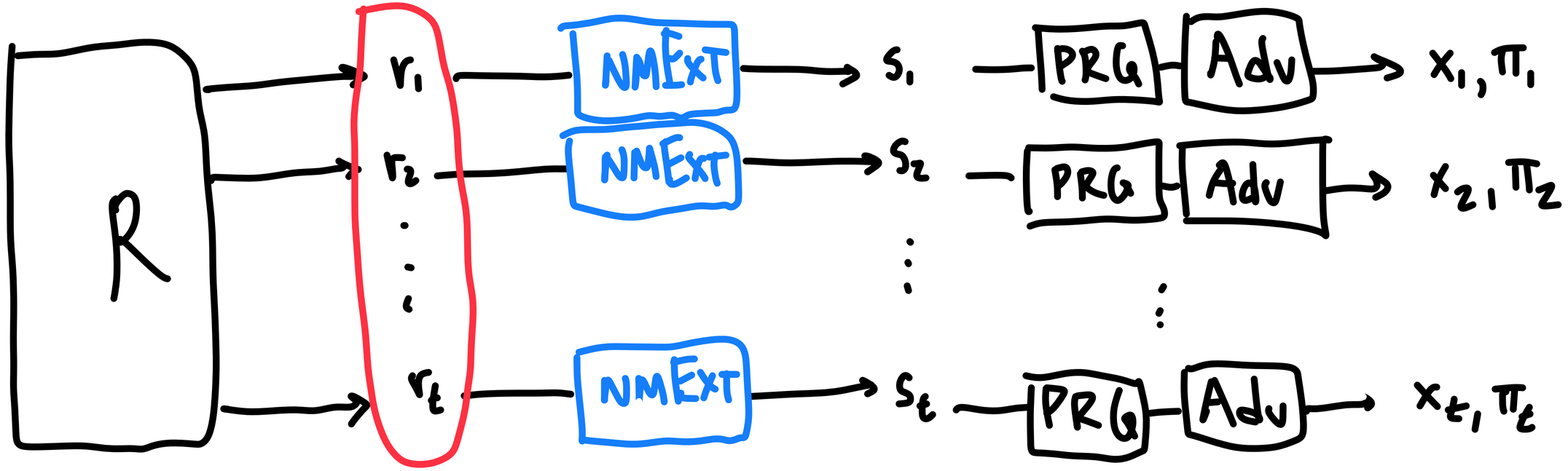
Correlated $\Longrightarrow$ correlated

Problem 1: correlated queries



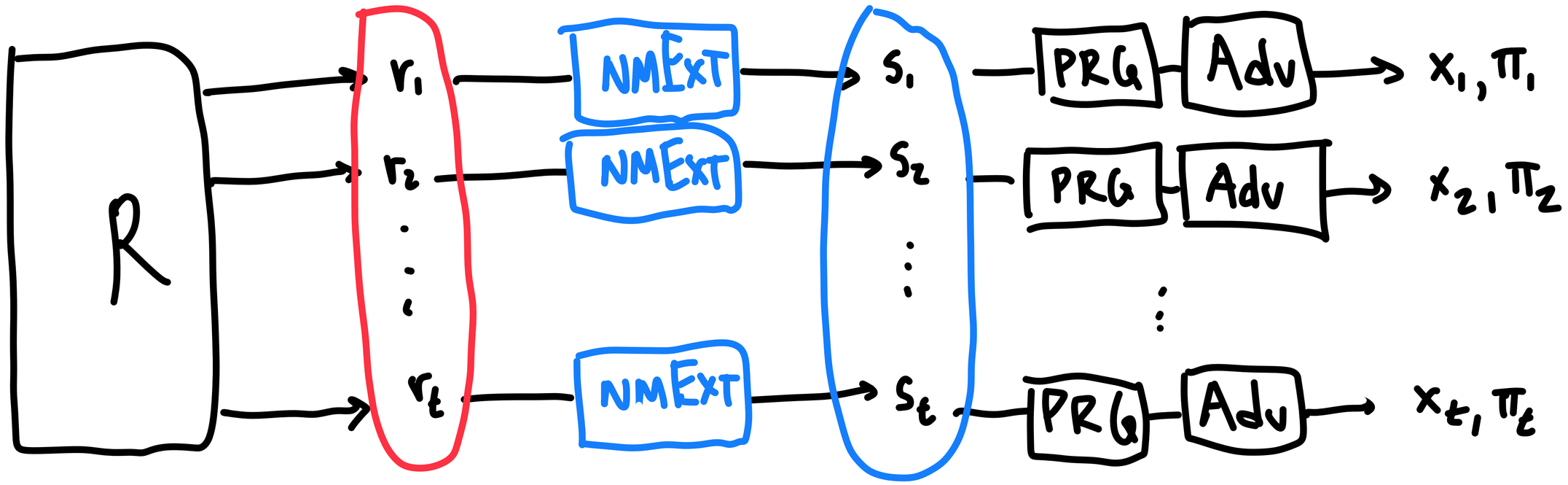
Correlated $\Longrightarrow$ correlated $\Longrightarrow$ no guarantee

Problem 1: correlated queries



Correlated

Problem 1: correlated queries

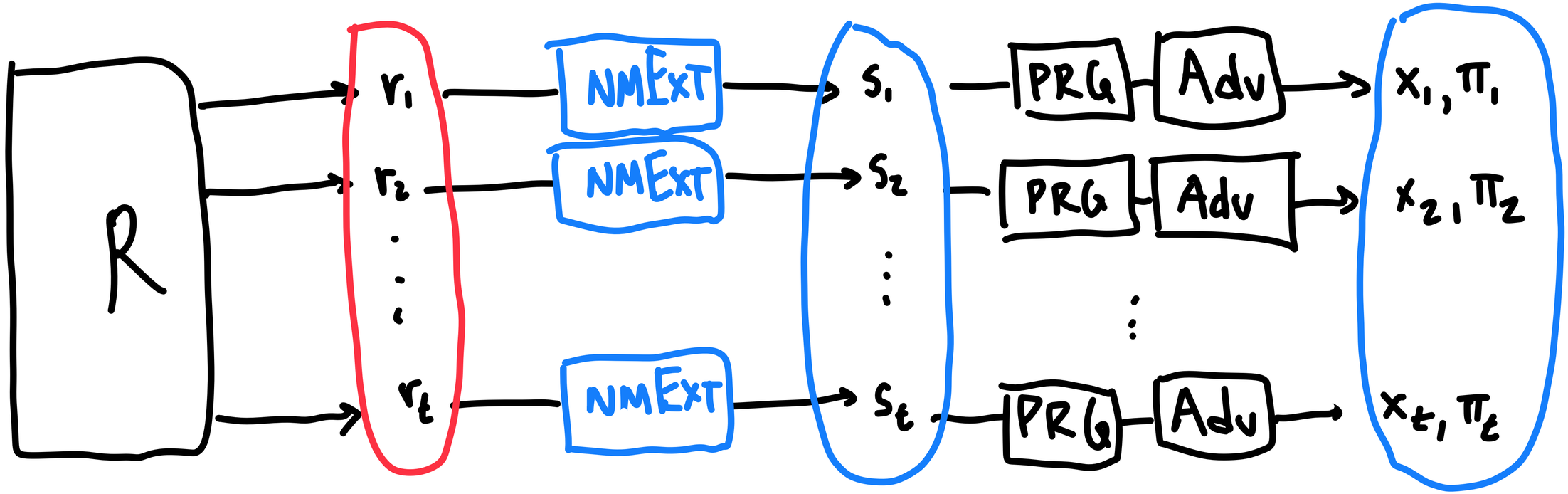


Correlated

each s_i random, even given other s_j

<u>Tampering</u>: fix r_i & nondeterministically sample consistent $\$$, use to sample r_j $\longrightarrow$

Problem 1: correlated queries



Correlated

each s_i random, even given other s_j $\Rightarrow$ leakage simulation works (in hybrid argument)

<u>Tampering</u>: fix r_i & nondeterministically sample consistent $\$$, use to sample r_j $\longrightarrow$

Problem 2:

efficient Sim can't check $\Pr[R \to (CRS, r)]$ high to know when to output $\perp$ ...

Problem 2:

efficient Sim can't check $\Pr[R \to (CRS, r)]$ high to know when to output $\perp$ ...

$\longrightarrow$ Sim will ultimately use D^* (not $\overline{D}^*$ like Adv) so just sample that (instead of $\perp$)

Problem 2:

efficient Sim can't check $\Pr[R \to (CRS, r)]$ high to know when to output $\perp$ ...

$\longrightarrow$ Sim will ultimately use D^* (not $\overline{D}^*$ like Adv) so just sample that (instead of $\perp$)

Problem 3: in PRG hybrids, need to simulate $\overline{D}^*$ which is not Poly time

$\Rightarrow$ this PRG is not poly time...

Problem 2:

efficient Sim can't check $\Pr[R \to (CRS, r)]$ high to know when to output $\perp$ ...

$\longrightarrow$ Sim will ultimately use D^* (not $\overline{D}^*$ like Adv) so just sample that (instead of $\perp$)

Problem 3: in PRG hybrids, need to simulate $\overline{D}^*$ which is not Poly time

$\Rightarrow$ this PRG is not poly time...

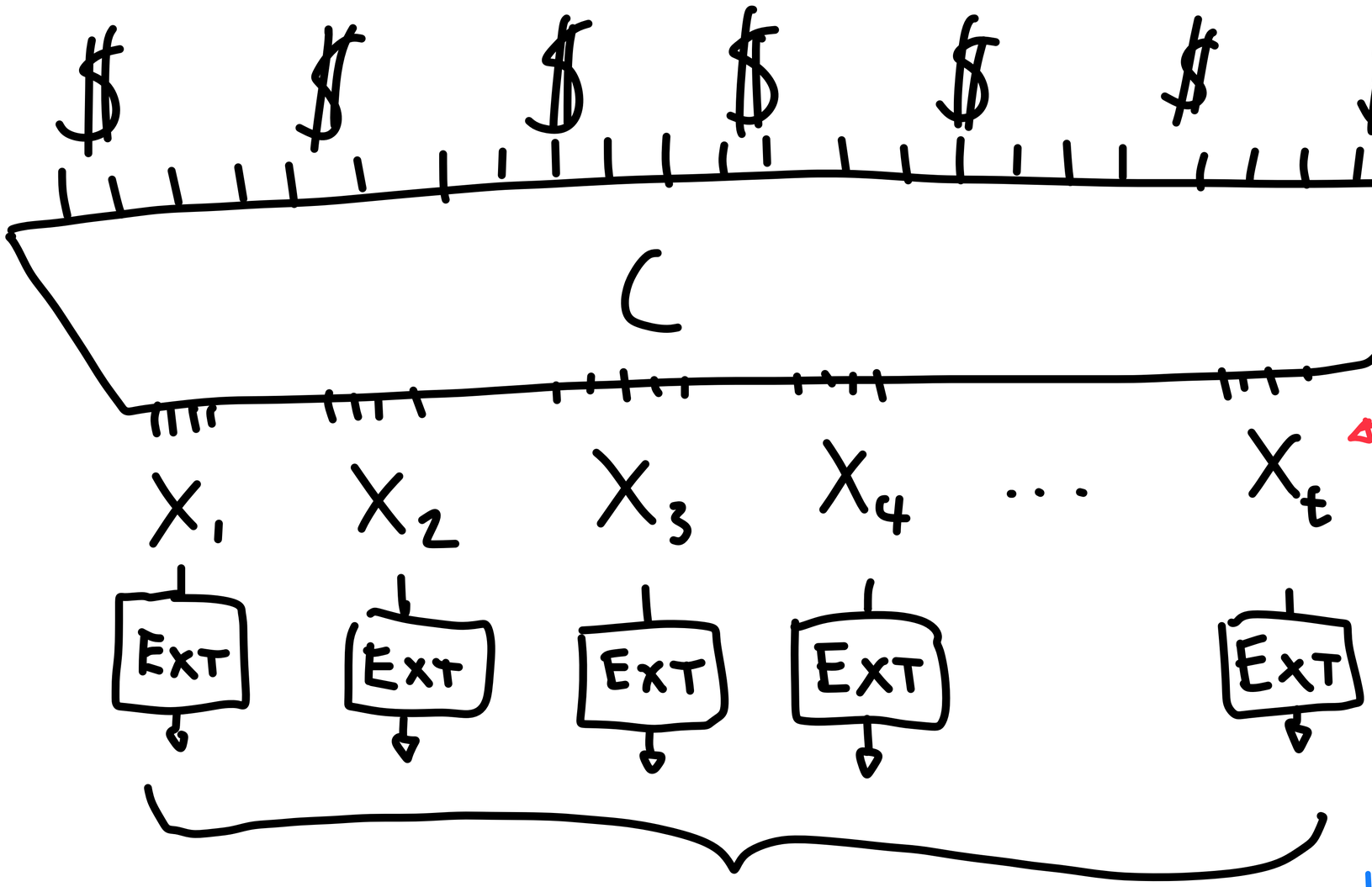
$\longrightarrow$ use 2 PRGs, easy PRG AND hard PRG, initially and carefully switch all to easy in Sim.

Recap

I. (seedless) t-NON-MALLEABLE EXTRACTORS
for poly size sources and tampering.

II. how to use these extractors
to prove black-box separations.

Open Problem: Local Extraction from MANY correlated sources



(jointly) Pseudorandom

promise:

① $H_\infty(X_i) \geq k$

② $X_i \neq X_j$

never take same value

this paper: $t < k$.

want: $t \gg k$.