Universal Composability with Global Generic Groups

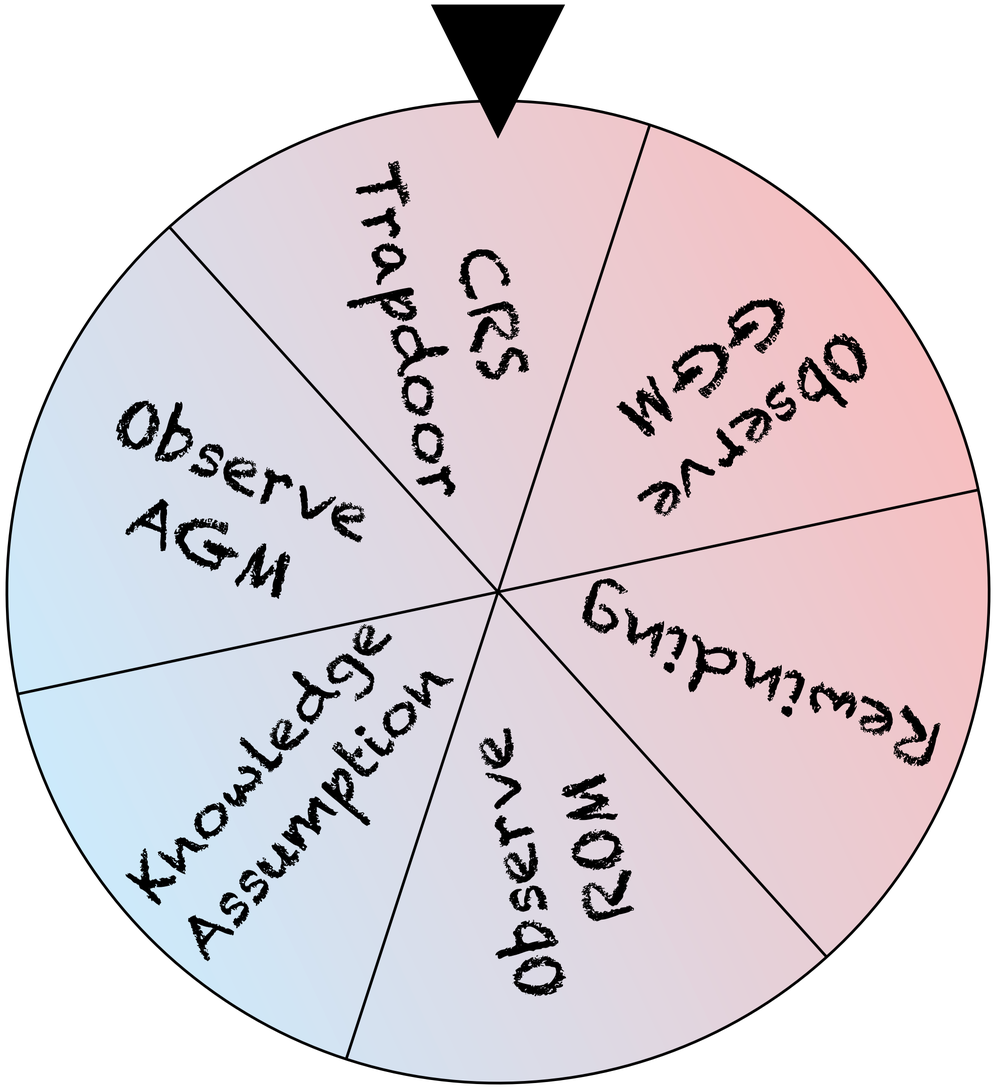
Jan Bobolz — THE UNIVERSITY of EDINBURGH

Pooya Farshim — INPUT | OUTPUT

Markulf Kohlweiss — THE UNIVERSITY of EDINBURGH / INPUT | OUTPUT

Akira Takahashi — J.P.Morgan — AlgoCRYPT CoE — AI Research

Let’s spin the PoK wheel 🎰



Pass's transform

Extractor

$(a,0,r_0),(a,1,r_1)$
$\leftarrow \Sigma.\mathrm{Prove}(x,w)$

a

$\mathscr{C}_0 = \mathsf{RO}(x,(a,0,r_0),nonce_0)$

$\mathscr{C}_1 = \mathsf{RO}(x,(a,1,r_1),nonce_1)$

$c \leftarrow \{0,1\}$

$r_c, nonce_c$

Check $\Sigma.\mathrm{Verify}(x,a,c,r_c)$

Check $\mathscr{C}_c = \mathsf{RO}(x,(a,c,r_c),nonce_c)$

⚠ (Inefficient; go for Fischlin variants instead)
ia.cr/2022/393

The UC RO hybrid model

eval(x):
If $Q(x) = \perp$:
$Q(x) \leftarrow \{0,1\}^n$
return $Q(x)$

Real world

Random oracle

Prover

Verifier

Adversary

Environment

Is this a good model?

Ideal world

$\mathcal{F}_{\text{PoK}}$

Ideal Adversary
Simulator

Environment

Is this a good model?

Let’s have two sessions of the protocol

Random Oracle

RO(123) = a03ab19b866fc

Adversary

What’s RO(123) ?

This does not model reality!

Random Oracle

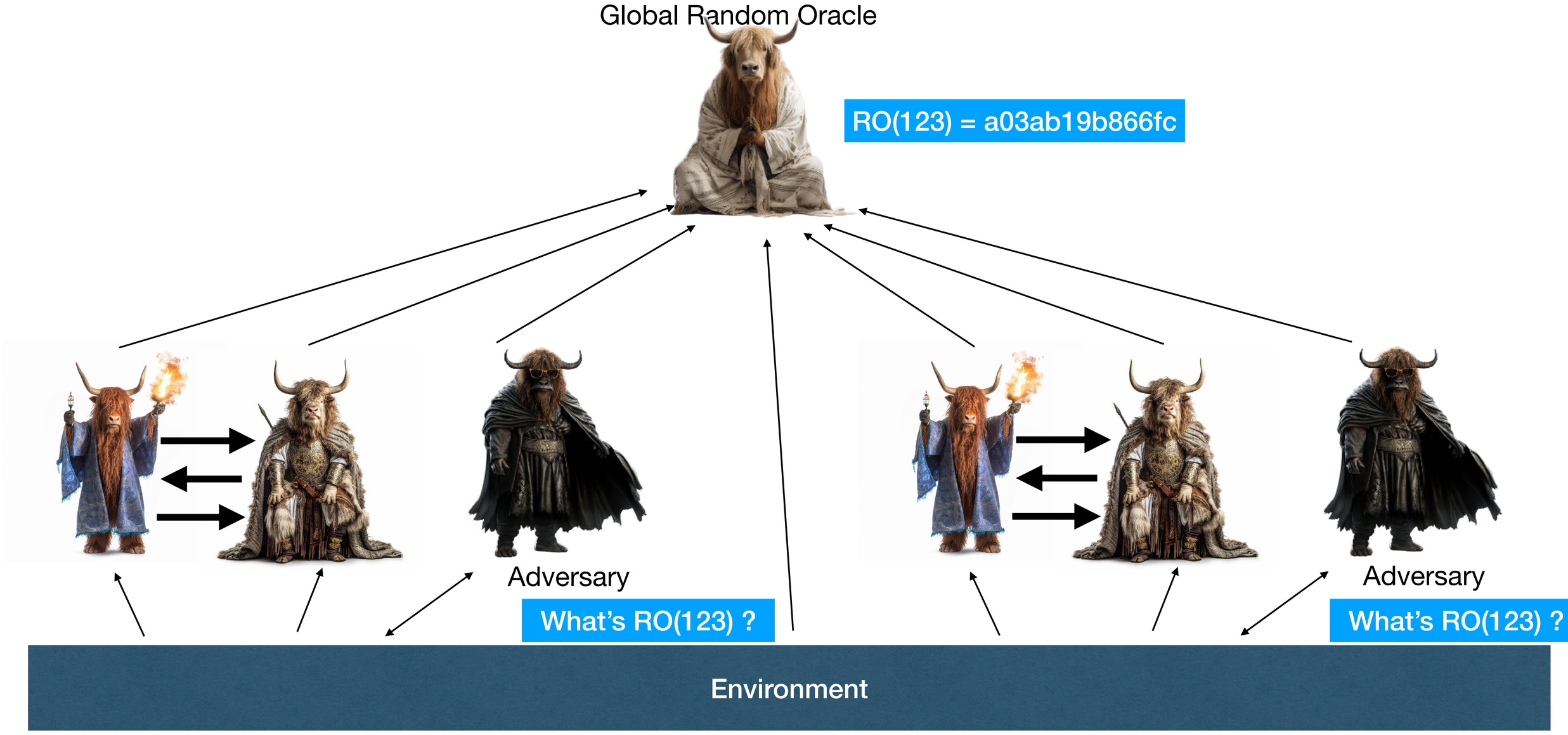
RO(123) = 7106b623725f

Adversary

What’s RO(123) ?

Environment

The better model: Global ROM

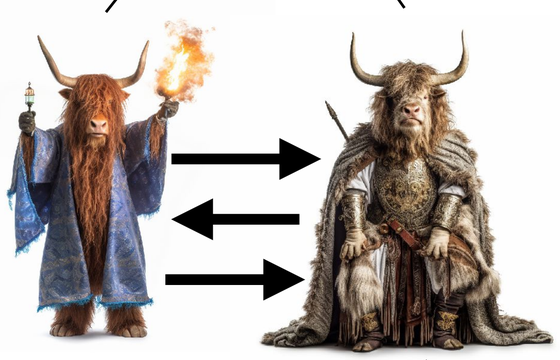


Two worlds with global ROM

Real world

Global random oracle

!



Adversary

Environment

Simulator does not get to observe all RO queries anymore ☹

Ideal world

Global random oracle

!

Ideal Adversary
Simulator

$\mathcal{F}_{\text{PoK}}$

Environment

The *observable* global ROM

💡 **Idea**: Add observation capabilities to the ROM functionality.

Fully observable global ROM

<u>eval(x):</u>
If $Q(x) = \perp$:
  $Q(x) \leftarrow \{0,1\}^n$
Add x to Ω

return $Q(x)$

<u>getObs():</u>
return Ω

Okay. Too much! **No chance for ZK.** Everyone sees everything always.

💡 **Idea**: Add observation capabilities to the ROM functionality.

📋 **Requirements**

- *For PoK*: Simulator **must be** able to observe environment's queries.
- *For ZK*: Environment **must not be** able to observe honest parties' queries.

💭 *Dream* observable global ROM

<u>eval(x):</u>
If $Q(x) = \perp$:
  $Q(x) \leftarrow \{0,1\}^n$

if caller is environment:
  Add x to Ω

return $Q(x)$

<u>getObs():</u>
return Ω

Cannot check this. *Doesn't make sense* from G-ROM PoV

💡 **Idea**: Add observation capabilities to the ROM functionality.

📋 **Requirements**

- *For PoK*: Simulator **must be** able to observe environment's queries.
- *For ZK*: Environment **must not be** able to observe honest parties' queries.

🦆 **Solution**: Domain separation

The *observable* global ROM

eval(x):
If $Q(x) = \perp$:
  $Q(x) \leftarrow \{0,1\}^n$

Parse $x = (s, x')$
if caller's session is not s:
  Add x to Ω

return $Q(x)$

getObs():
return Ω

💡 **Idea**: Add observation capabilities to the ROM functionality.

📋 **Requirements**

- *For PoK*: Simulator **must be** able to observe environment's queries.
- *For ZK*: Environment **must not be** able to observe honest parties' queries.

🦆 **Solution**: Domain separation

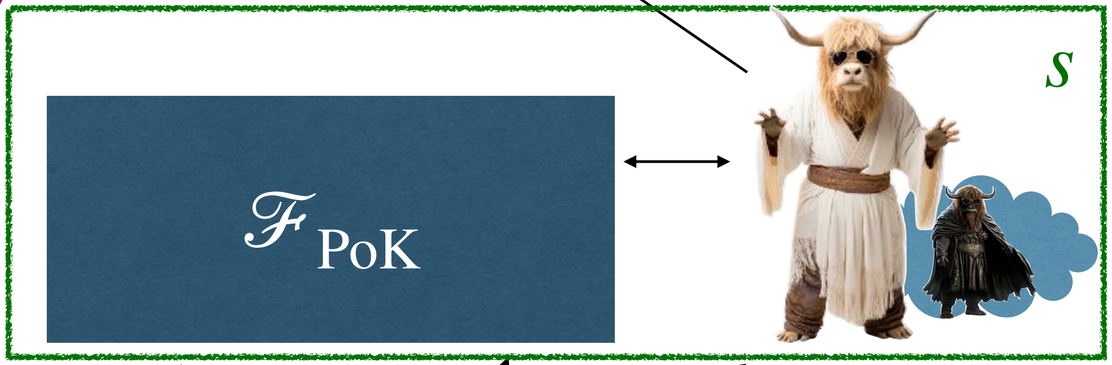
Two worlds with global *observable* ROM

Parties in session s can query $\mathrm{RO}(s, x')$ unobservably

Environment's (sessions $s' \neq s$) queries $\mathrm{RO}(s, x')$ are observable

Environment

Adversary's queries $\mathrm{RO}(s, x')$ can be seen by simulator anyway

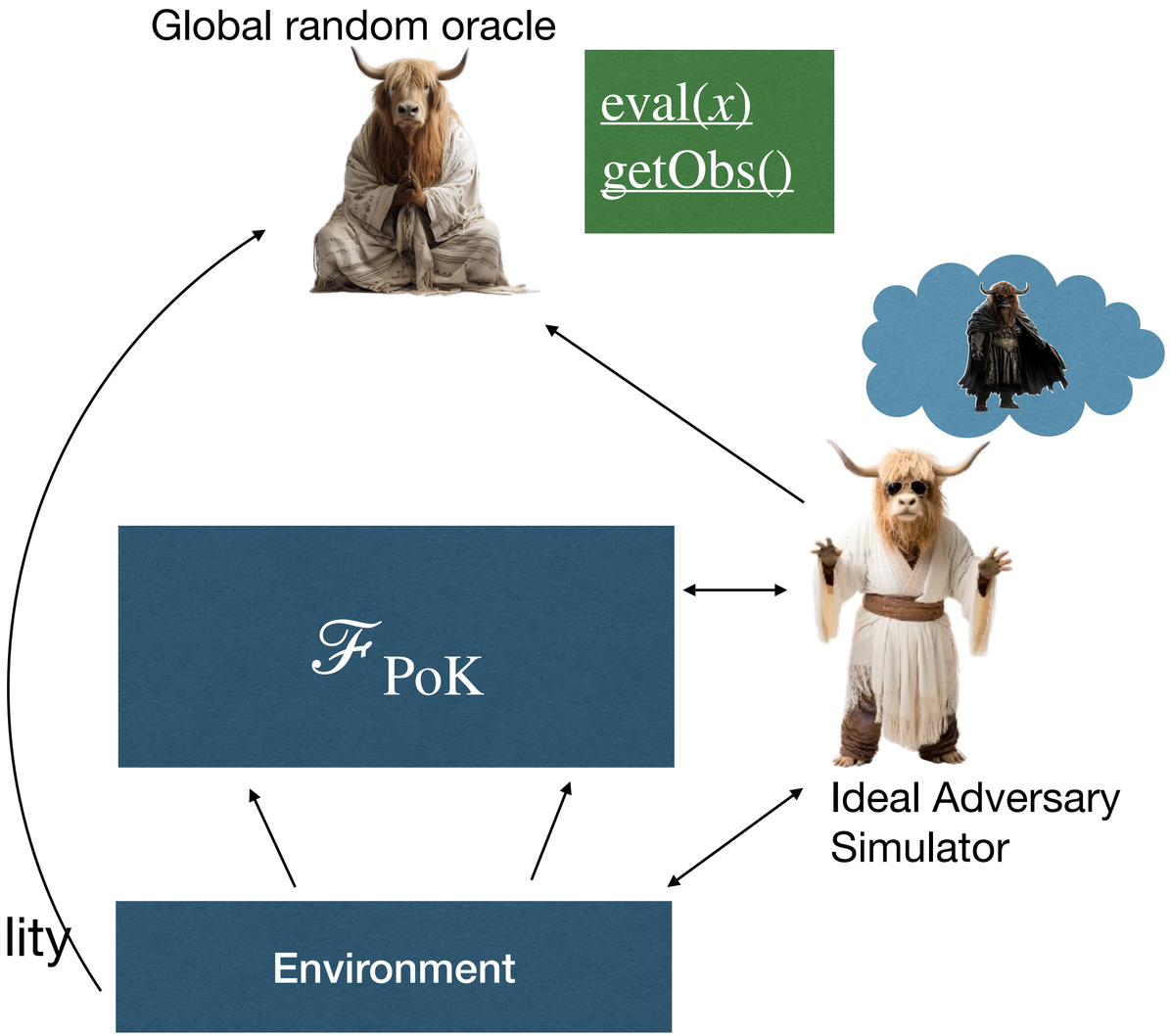


Environment

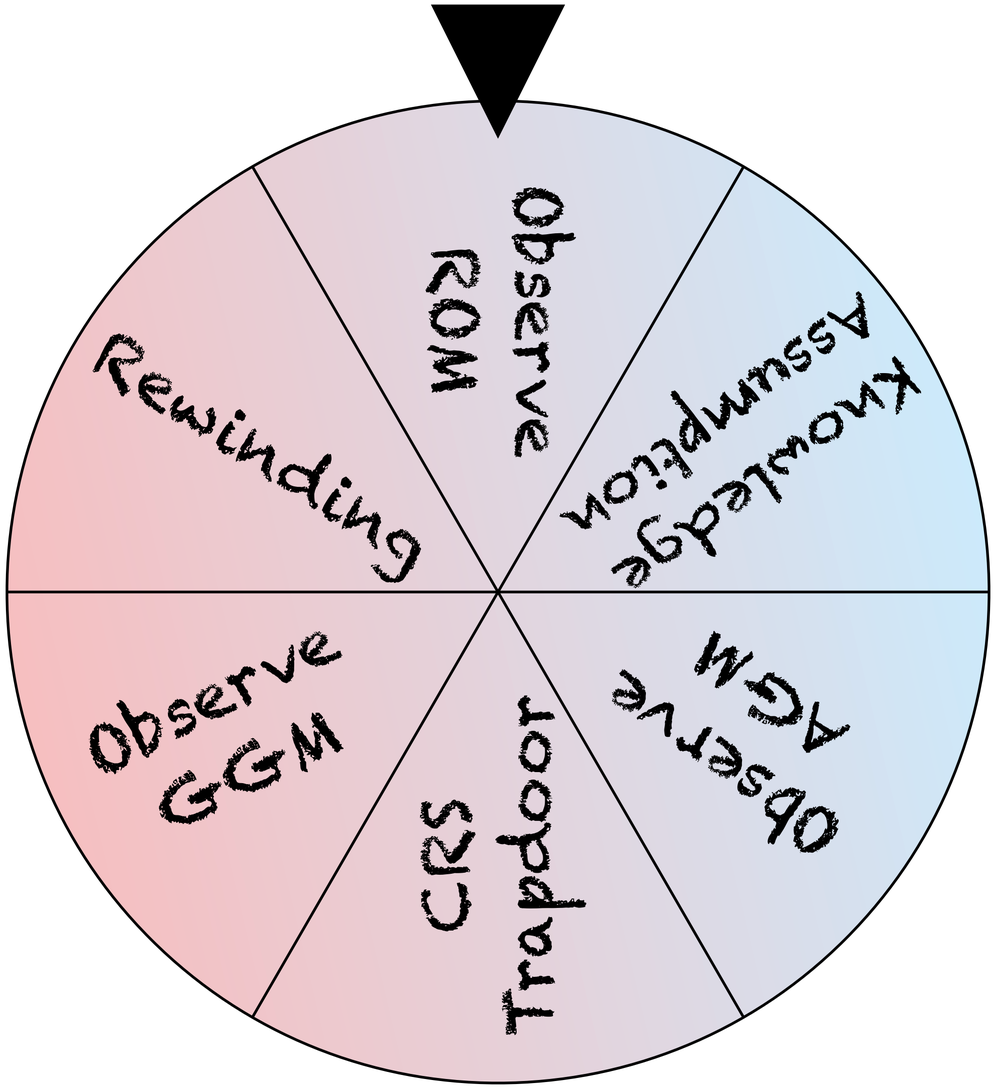
Summary part 1

- 🪞 **Global ROM** better reflects reality
- 👀 Need to explicitly **model observability**
- 😇 **Good**: Session s calls $\mathrm{RO}(s, x')$
- 😈 **Bad**: Session $s' \neq s$ calls $\mathrm{RO}(s, x')$

  $\Rightarrow$ Observable
- 🥳 Get NIZKPoK in UC via Fischlin/Pass (+crs for simulation)
  Lysyanskaya, Rosenbloom; ia.cr/2022/290
- 👩‍🏫 Learned: need domain-separation for composability (“implicit” in one-shot games/hybrid UC)



Let’s spin the PoK wheel 🎰



The generic group model

- 🎯 Goal: model “**idealized**” group with no “extra” structure (just group operations[, pairing]).
  - Similar to random oracles, which model “idealized” hash function with no structure.
- 💡 Idea: group elements get ***random*** **encoding** (= no structure), but **oracle enables group ops**.
  - 👀 Corollary: oracle *sees* all group ops.

The generic group model

`private` random injective $\tau : \mathbb{G} \to S$

`public` generator g

$\underline{\text{op}(g_1, g_2)}$:

return $\tau(\tau^{-1}(g_1) + \tau^{-1}(g_2))$

Groth16

CRS: group elements $u_i, \alpha, \beta, \delta$

Witness: wire values $a_i \in \mathbb{Z}_p$

$$A = \sum_{i=0}^{m} a_i u_i + \alpha + r\delta$$

$$B = \sum_{i=0}^{m} a_i v_i + \beta + r'\delta$$

$$C = \ldots$$

Extractor

[Check some pairing equations on A,B,C]

Fast forwarding through the story

Local ROM: bad model 😕 Both sessions use SHA-3, why am I getting different hashes?	**Local GGM**: bad model 😕 Both sessions use BLS12-381, why are elements incompatible?
Global ROM: **lose observability**. Remodel. Environment/other protocols can access global ROM without going through the simulator.	*Global* GGM: **lose observability**. Remodel. Environment/other protocols can access global GGM without going through the simulator.
Domain separation: $\mathrm{RO}(s, x')$ is "valid/in-session" iff caller is in session s. Invalid queries are observable.	**Domain separation**: $\mathrm{op}(g_1, g_2)$ is "valid/in-session" iff g_1, g_2 are based on caller session's generator h_s

ZK: honest parties only make "valid" unobservable queries within their domain.

PoK: when environment / protocol in session $s' \neq s$ queries related to domain s, it's observable.

Simplified G-oGG

`private` random inj. $\tau : \mathbb{G} \to S$

`public` rnd generator h_s for each session s

`public` poly variable X_s for gen of each session s

`private` representation $R[e]$ for each $e \in S$, initially $R[h_s] = X_s$

$\underline{\text{op}(g_1, g_2)}$:

$s =$ caller session

$result = \tau(\tau^{-1}(g_1) + \tau^{-1}(g_2))$

$R[result] = R[g_1] + R[g_2]$ //bookkeep sum of polynomials

if $R[result] \notin \mathbb{Z}_p[X_s]$: //invalid in caller session

  Add $(g_1, g_2, result)$ to public observation list

return $result$

Simplified G-oGG

`private` random inj. $\tau : \mathbb{G} \to S$

`public` rnd generator h_s for each session s

`public` poly variable X_s for gen of each session s

`private` representation $R[e]$ for each $e \in S$, initially $R[h_s] = X_s$

$\underline{\text{op}(g_1, g_2)}$:

$s =$ caller session

$result = \tau(\tau^{-1}(g_1) + \tau^{-1}(g_2))$

$R[result] = R[g_1] + R[g_2]$ //bookkeep sum of polynomials

if $R[result] \notin \mathbb{Z}_p[X_s]$: //invalid in caller session

  Add $(g_1, g_2, result)$ to public observation list

return $result$

Example ops with caller session s

- $17X_s$ op $X_{s'}$ observable
- $(17X_s + 3X_{s'})$ op X_s observable
- $17X_s$ op $4X_s$ unobservable
- $(17X_s + 0X_{s'})$ op X_s unobservable

Actual G-oGG

- 👯 Multiple generators per session
- 🤷‍♂️ Oblivious Sampling
- 💫 Pairing operations

Groth16 proof challenges

Idea
Extract dlog representation of proof elements

Challenge
Cannot observe *everything* (only *my* session's generator(s))

💡 Solution
Argue that valid proofs cannot contain foreign generators

Extraction

Simulation

Idea
Use CRS trapdoor to generate proofs without witness

Challenge
Prover/Simulator GGM ops must not be observable

💡 Solution
Prover/simulator only operates on CRS elements

ElGamal

sk

Can contain foreign $g_{s'}$

(c_0, c_1)

Environment

$-sk \cdot c_1 + c_0$

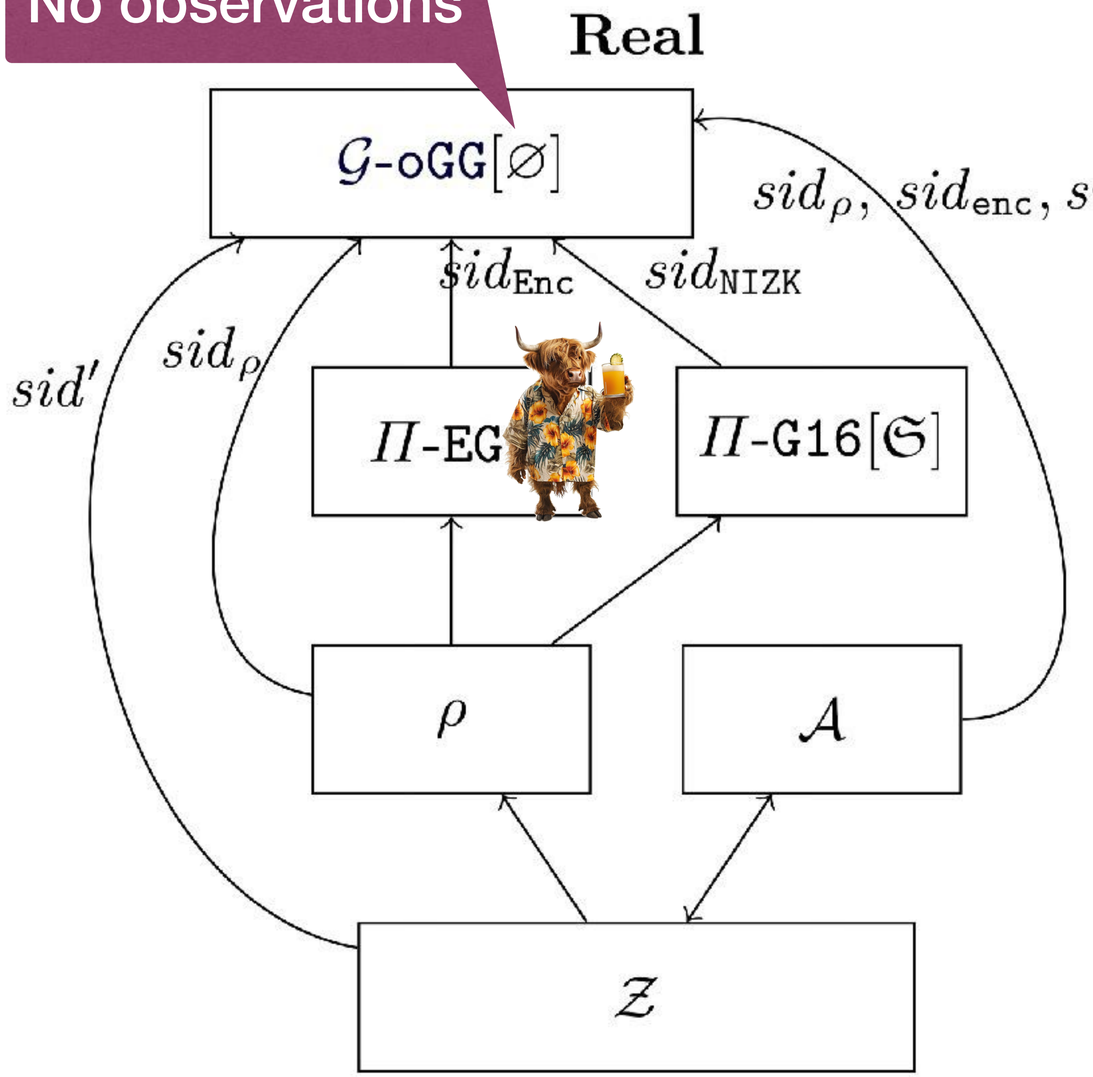
Observable 🚨

Challenge
Decrypting foreign ciphertext makes secret key observable

💡 Solution
Observations only needed in Groth16 ideal world, not in real.
Can *switch them off* in real world

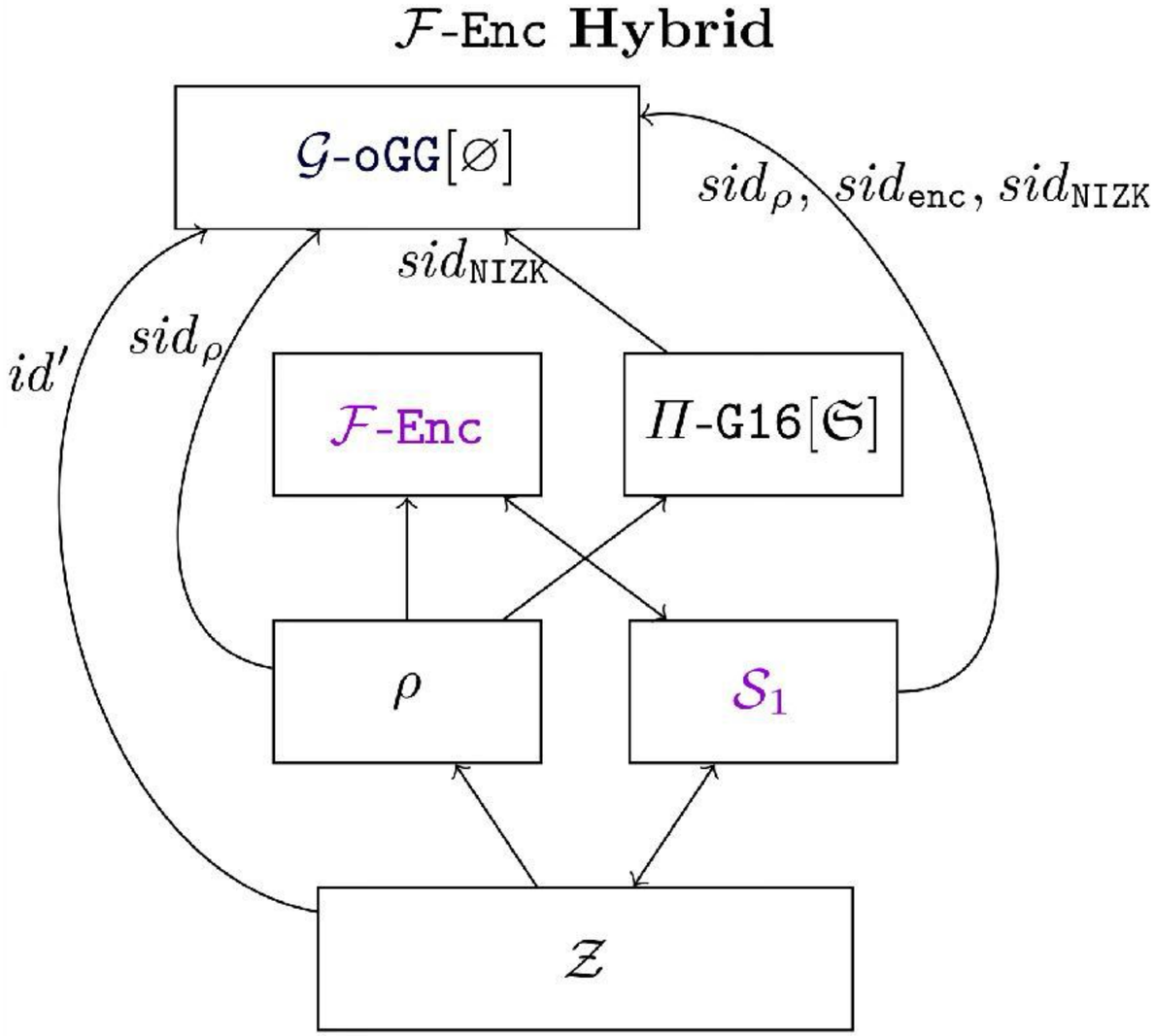
Composing ElGamal with Groth16

No observations

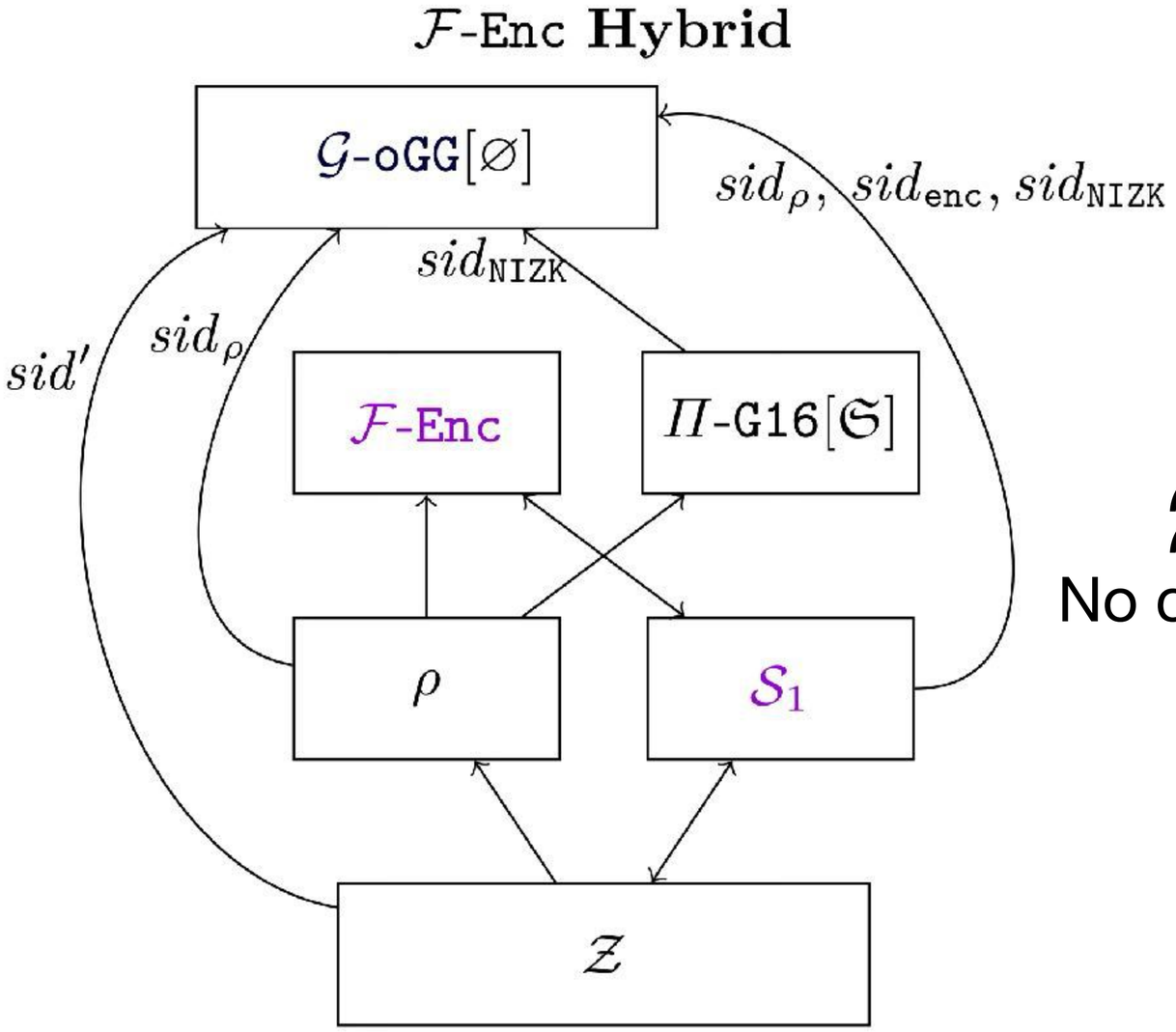


$\approx$

UCGS

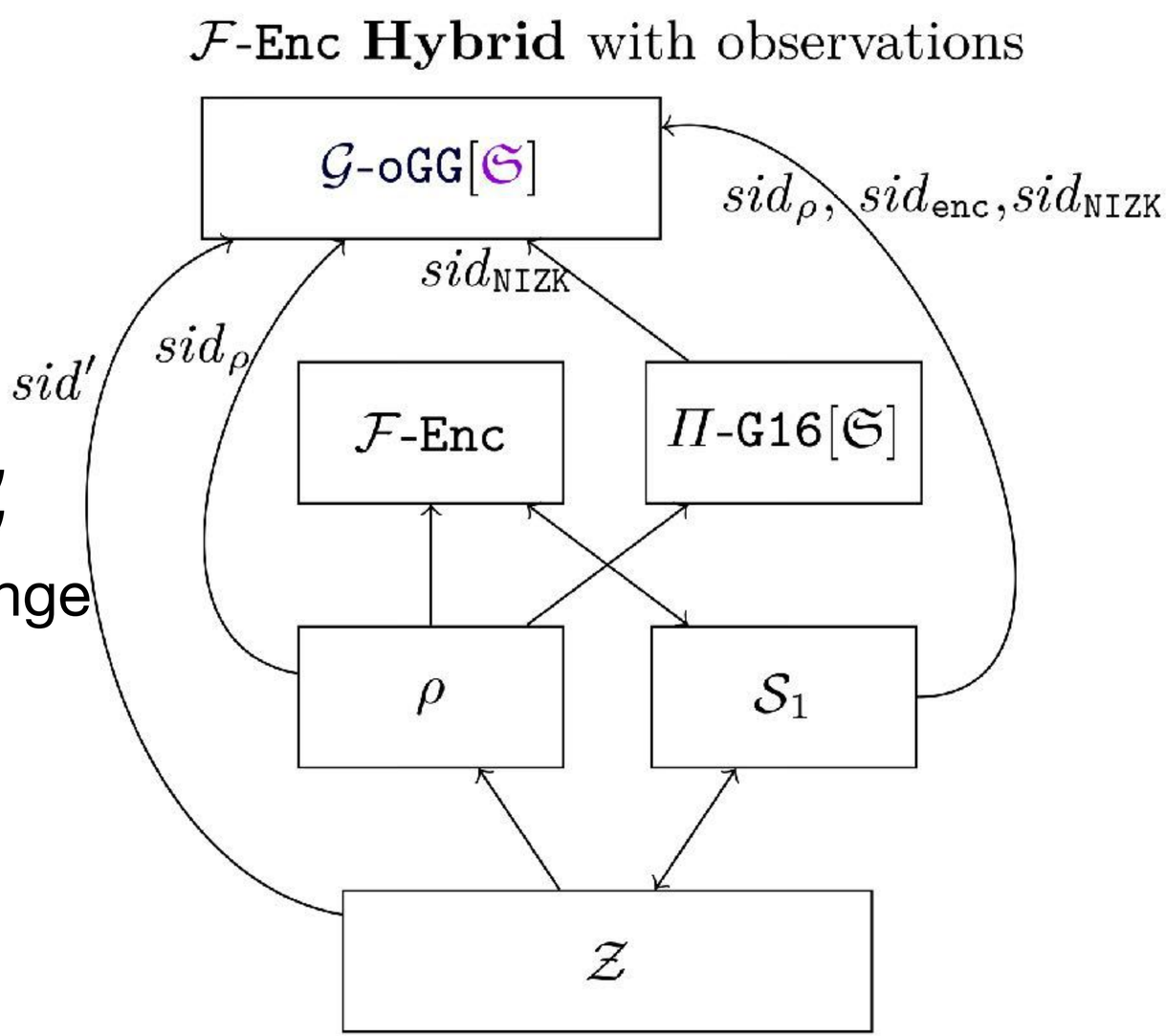


Composing ElGamal with Groth16

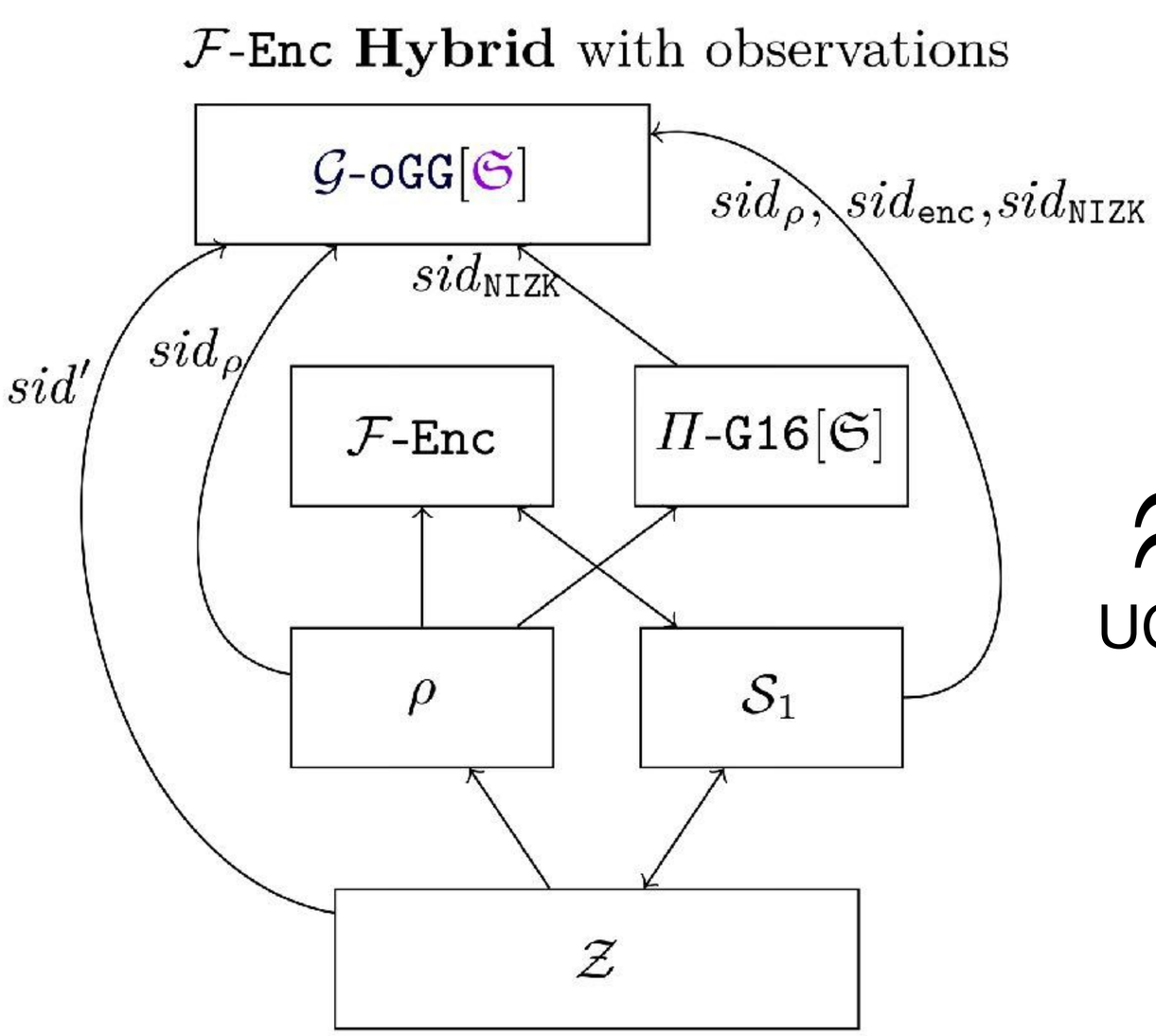


$\approx$

No change

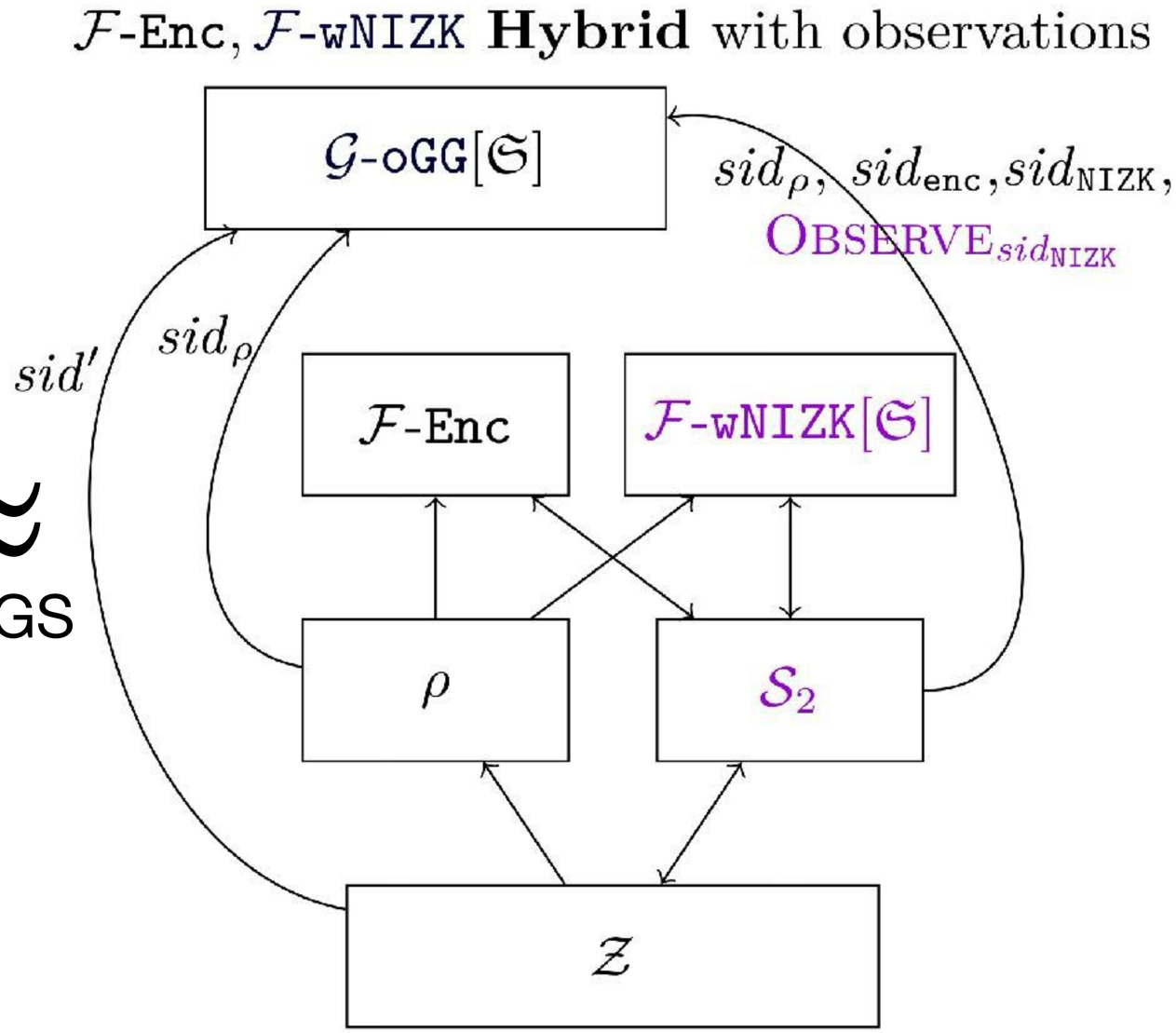


Composing ElGamal with Groth16



$\approx$

UCGS



Summary

- 👀 Need to explicitly **model observability**
  - **Not trivial!**
- 🥳 Get Groth16 SNARK in UC
- 🥳 Still make ElGamal composition work
  - UC is (unnecessarily?) strict.

Backup slides

Resource conflicts

PoK

$B^{E.Extr, E.O}$

$Pr[extr\ error] \leq negl$

$A^{O} \leftrightarrow G^{O}$

$A^{E.O} \leftrightarrow G^{E.O} \leftrightarrow E.Extr$

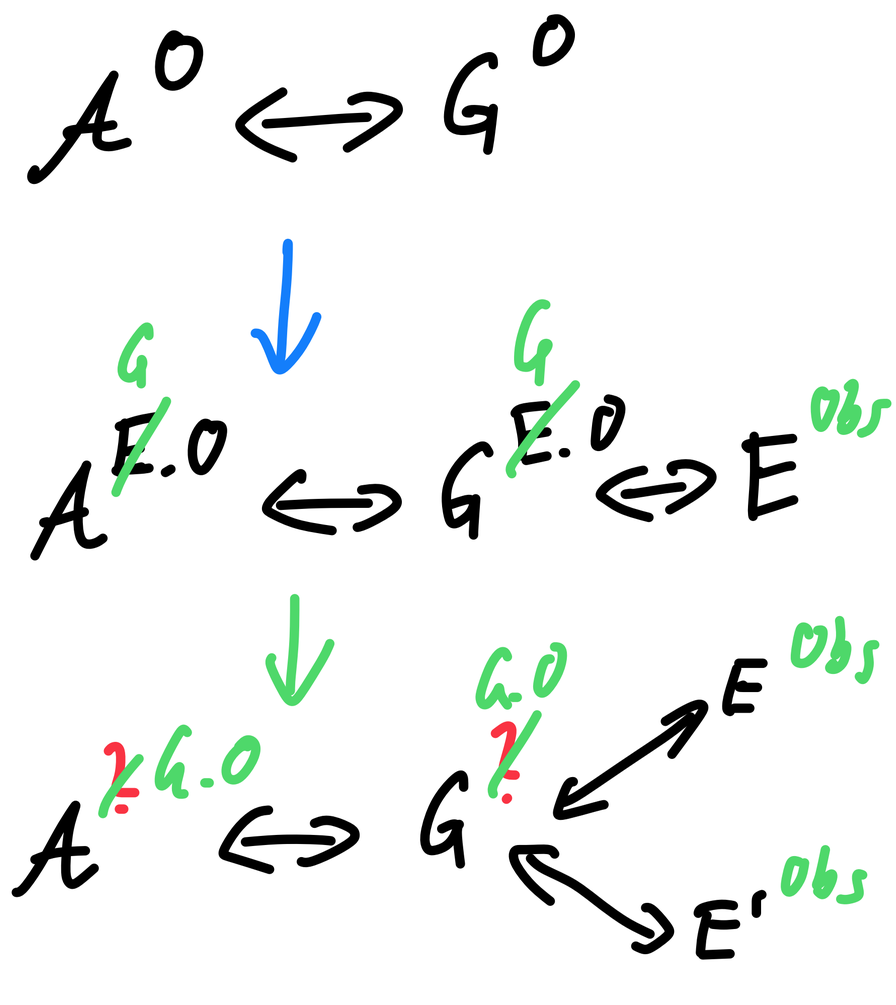
$A^{?} \leftrightarrow G^{?}$ → $E.Extr$, $E'.Extr$

Resource conflicts: Restrict to Obs

PoK

$B^{E^{obs}}$, ~~$E.O$~~

$Pr[extr\ error] \leq negl$



Resource conflicts: UC

PoK

$B^{E^{obs}}$, ~~$E.\Phi$~~

$Pr[\text{extr error}] \leq negl$

$A^{\emptyset} \leftrightarrow G^{\emptyset}$

$A^{F.\emptyset} \leftrightarrow G^{F.\emptyset} \leftrightarrow E^{obs}$

UC: need to model observation power in-framework

(GB: just thought-experiment)

$A^{\neq G.\emptyset} \leftrightarrow G^{?}$

E^{obs}

E'^{obs}

Know each other's secrets.

GB: Ok ✓

UC: Other Sim evil