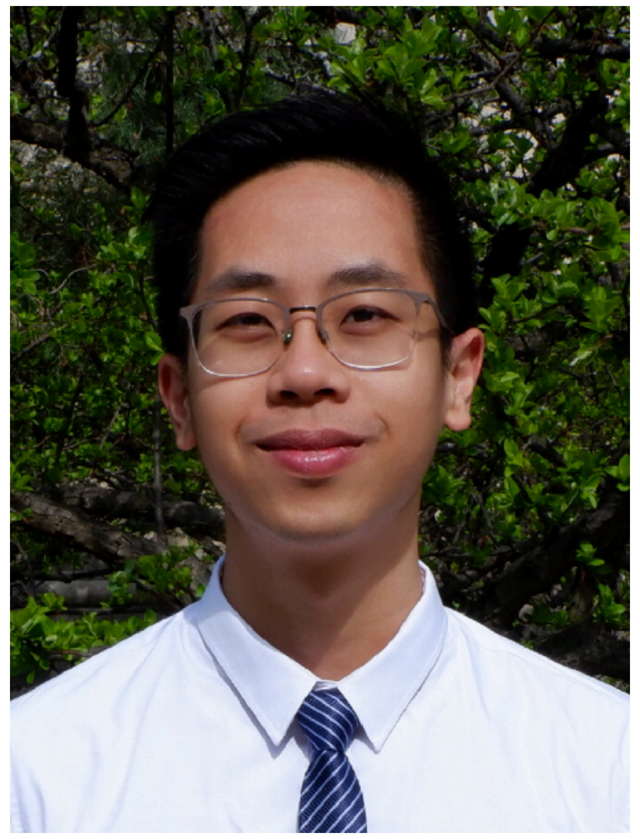
Spartan & Bulletproofs are simulation-extractable (for free!)

Quang Dao
CMU



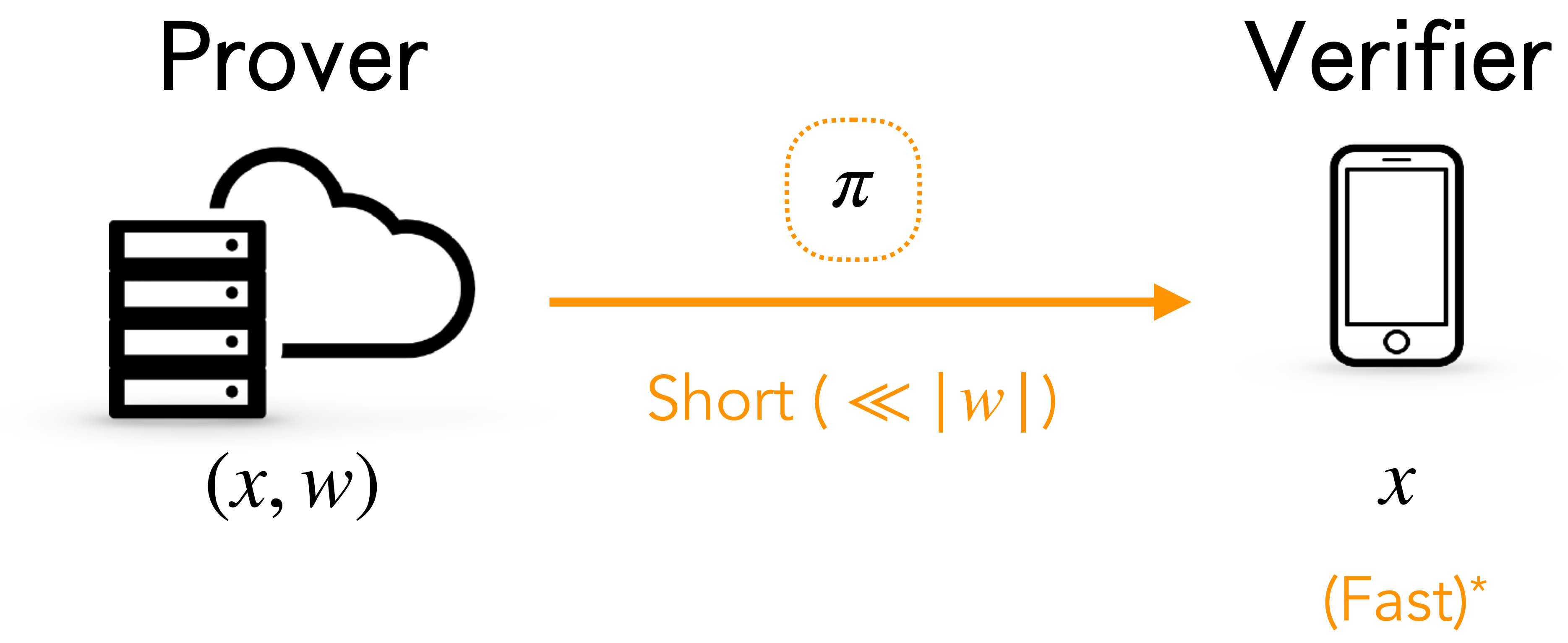
Paul Grubbs
Michigan

Lattices Meet Hashes, May 3rd 2023

zkSNARKs: Security & Use Cases

(Zero-knowledge Succinct Non-interactive ARguments of Knowledge)

short, non-interactive proofs



Knowledge Soundness: If V accepts, then P must "know" w.

Zero-Knowledge: π hides w.

Applications in blockchains:

- Private smart contracts
- Private transactions
- ZK-Rollups

Other applications:

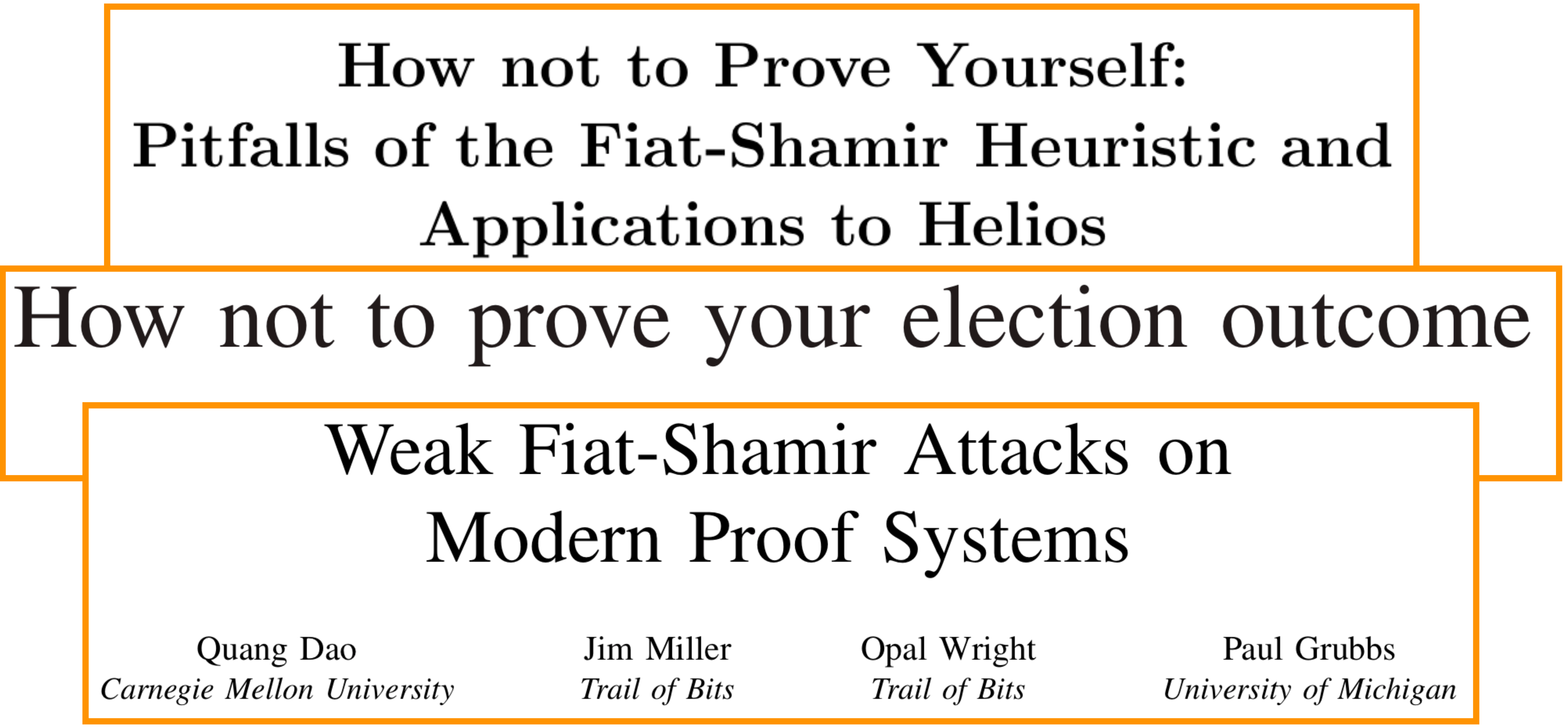
- Proof of solvency [DBBCB15]
- Image provenance [NT16], [BD22], [KHSS22]
- Content moderation [RMM22], [GAZBW22]
- And many more!

*For this talk, zkSNARKs may be without fast verification.

Standard ZKP security is not enough

<u>Adaptive attack:</u> choose the statement <u>adaptively</u> based on the proof

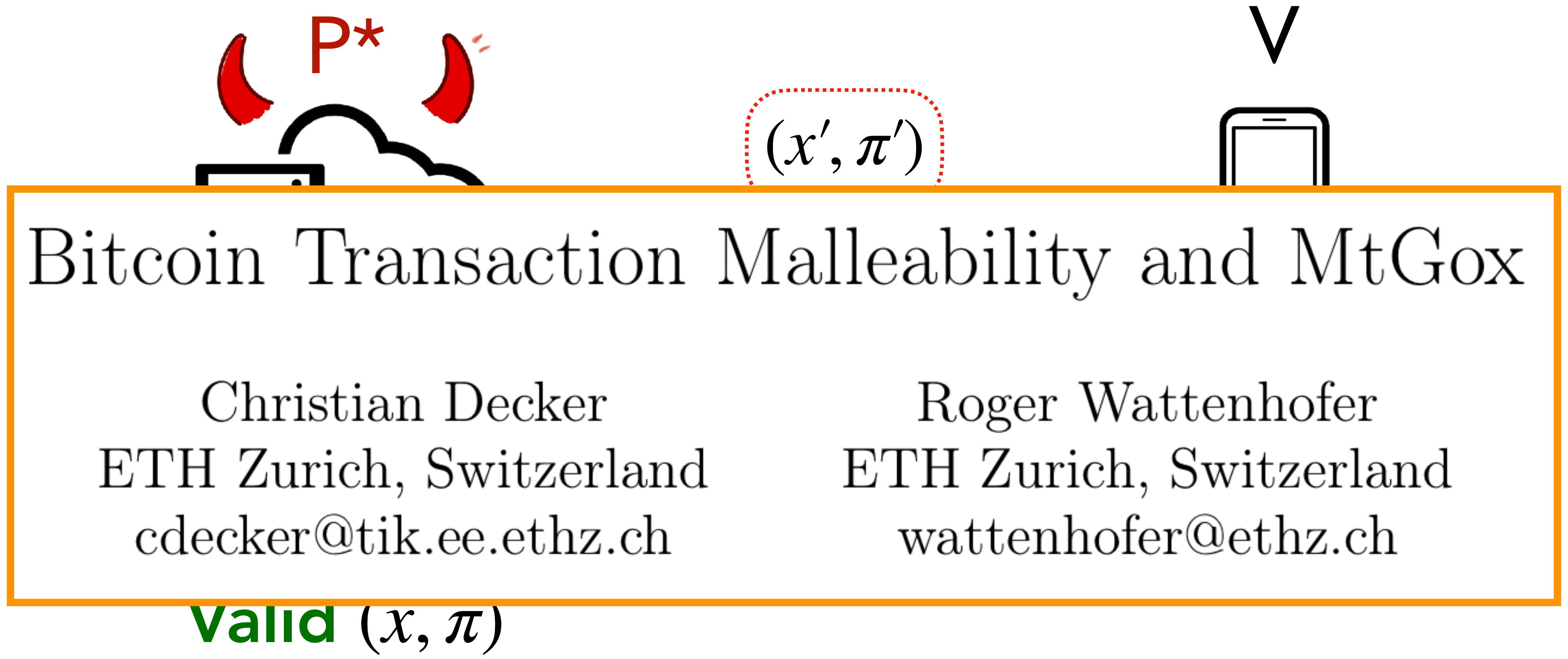
<u>Malleability attack:</u> modify an <u>existing</u> proof into a <u>new</u> proof <u>without</u> knowing the witness


How not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios

How not to prove your election outcome

Weak Fiat-Shamir Attacks on Modern Proof Systems

Quang Dao, Carnegie Mellon University
Jim Miller, Trail of Bits
Opal Wright, Trail of Bits
Paul Grubbs, University of Michigan



Bitcoin Transaction Malleability and MtGox

Christian Decker, ETH Zurich, Switzerland, cdecker@tik.ee.ethz.ch
Roger Wattenhofer, ETH Zurich, Switzerland, wattenhofer@ethz.ch

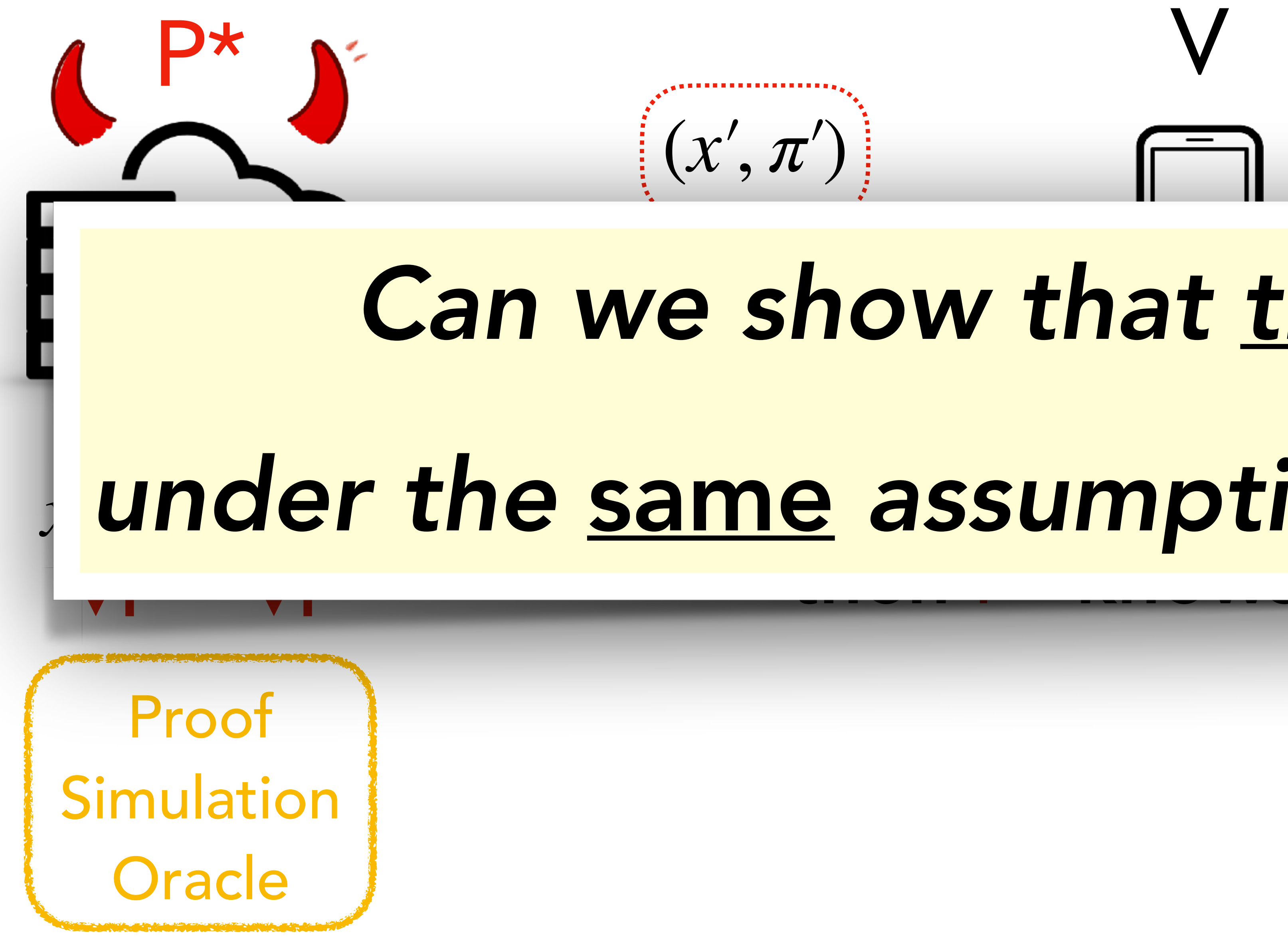
Not ruled out by (non-adaptive) knowledge soundness & zero-knowledge!

$\implies$ **We need *<u>stronger</u>* security properties for deployment**

Simulation Extractability

SIM-EXT (informal): [Sahai99], [DDOPS01]

Knowledge soundness holds *even when* P* gets *extra power*.



Can we show that transparent zkSNARKs satisfy SIM-EXT under the same assumptions used to prove (knowledge) soundness?

Rules out adaptive & malleability attacks.

Required for many applications. [KMSWP16], [BCG+20]

Prior works:

- Constructing SIM-EXT zkSNARKs directly. [GM17], [Lipmaa20]
- Achieving SIM-EXT via generic transformations.

out-of-

  - Sonic, Plonk, Marlin [GKKNZ22] $\Longleftarrow$ *not transparent*
  - Bulletproofs [GOPTT22] $\Longleftarrow$ *require stronger-than-necessary assumption (AGM)*

Our Results

We show that **Spartan** and **Bulletproofs**, two *transparent* zkSNARKs, satisfy SIM-EXT in the random oracle model (ROM) assuming the discrete log assumption (DLOG) holds.

- Bulletproofs [BBBPWM18] has seen deployment in Monero, MimbleWimble, etc.
- Spartan [Setty20] is a state-of-the-art zkSNARK for prover time.

These assumptions (DLOG + ROM) are the ***minimal*** ones used to prove their soundness.

To prove our results, we develop a few tools that might be of independent interest:

- A **template** for proving SIM-EXT from smaller properties
  (building on the work of Ganesh, Khoshakhlagh, Kohlweiss, Nitulescu & Zajac [GKKNZ22])
- A more general **tree extraction** lemma for proving knowledge soundness
  (building on the work of Attema, Fehr & Klooß [AFK22])

Agenda

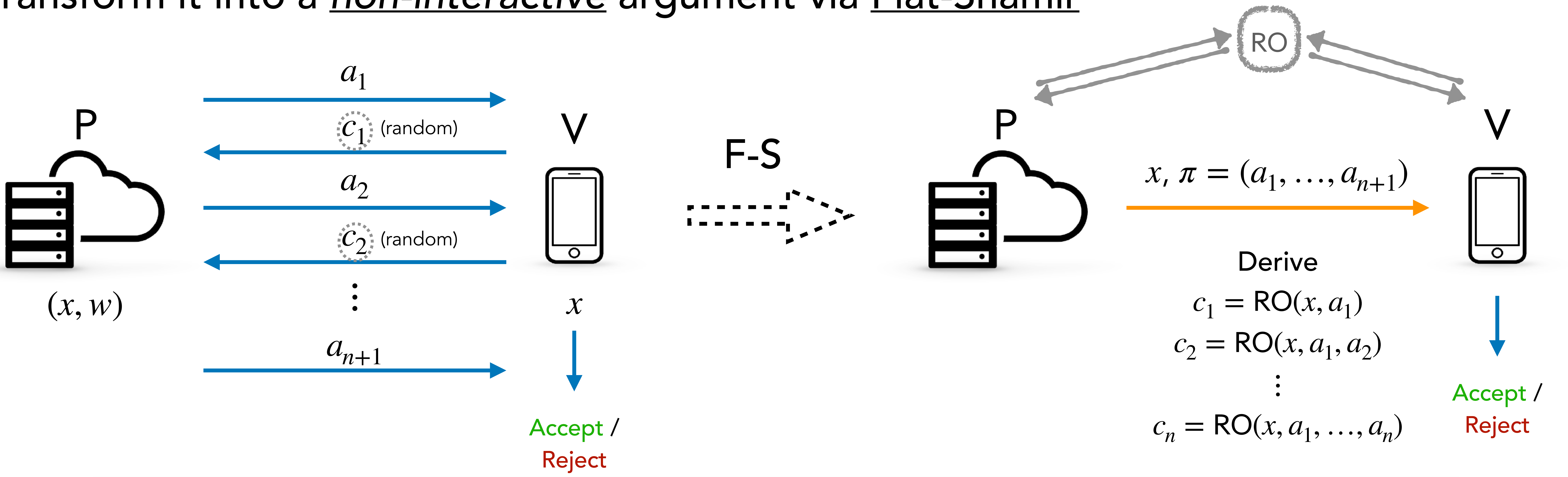
1. Breaking SIM-EXT into smaller properties
2. Instantiating SIM-EXT template for Bulletproofs & Spartan
3. Knowledge Soundness via Generalized Tree Builder

Agenda

1. Breaking SIM-EXT into smaller properties
2. Instantiating SIM-EXT template for Bulletproofs & Spartan
3. Knowledge Soundness via Generalized Tree Builder

The Fiat-Shamir Transform & SIM-EXT Insight

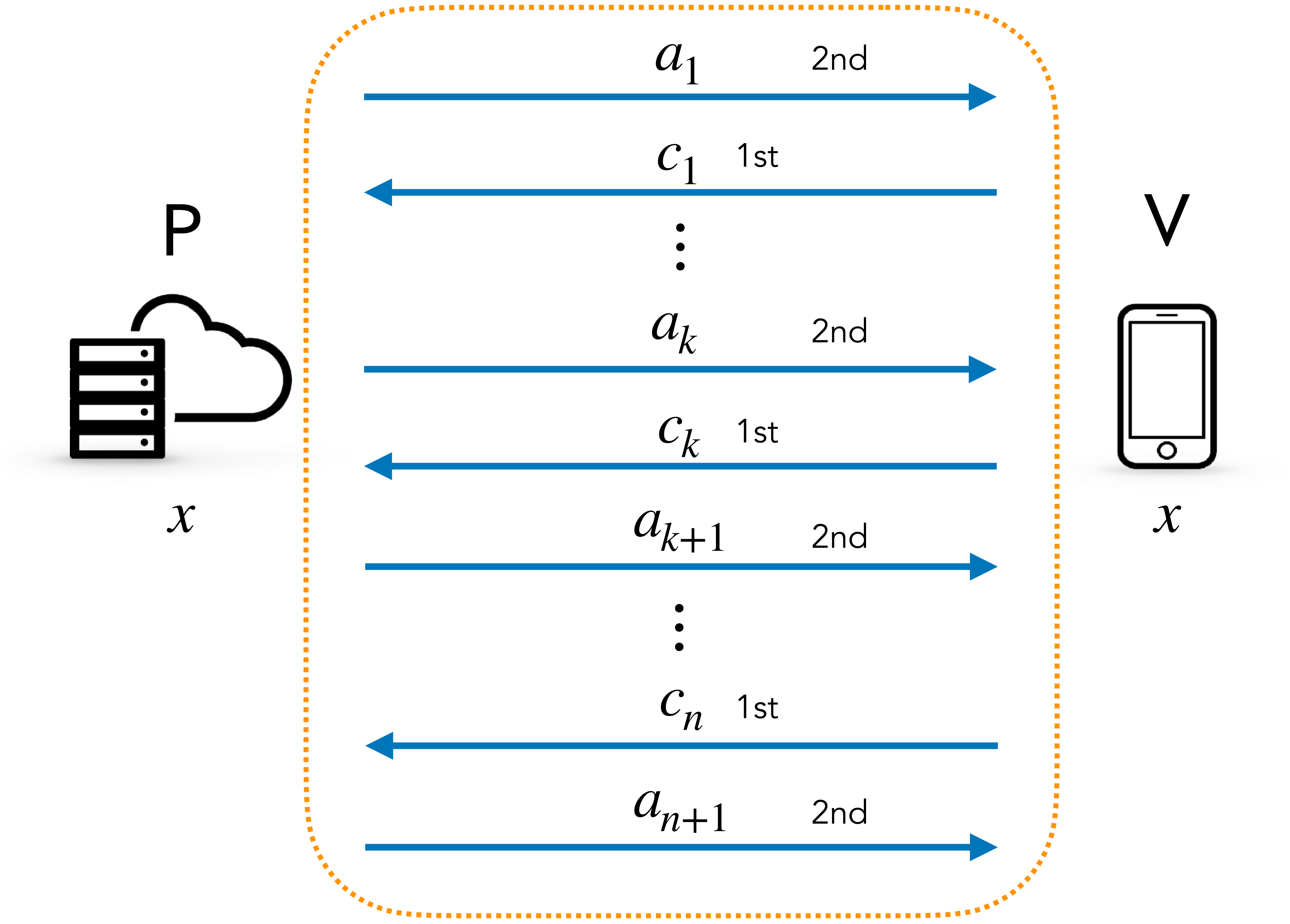
- Construct an *interactive*, *public-coin* argument
- Transform it into a *non-interactive* argument via Fiat-Shamir



Insight: [GKKNZ22] ***Assuming 2 smaller properties, SIM-EXT of F-S argument may be reduced to its knowledge soundness (KS).***

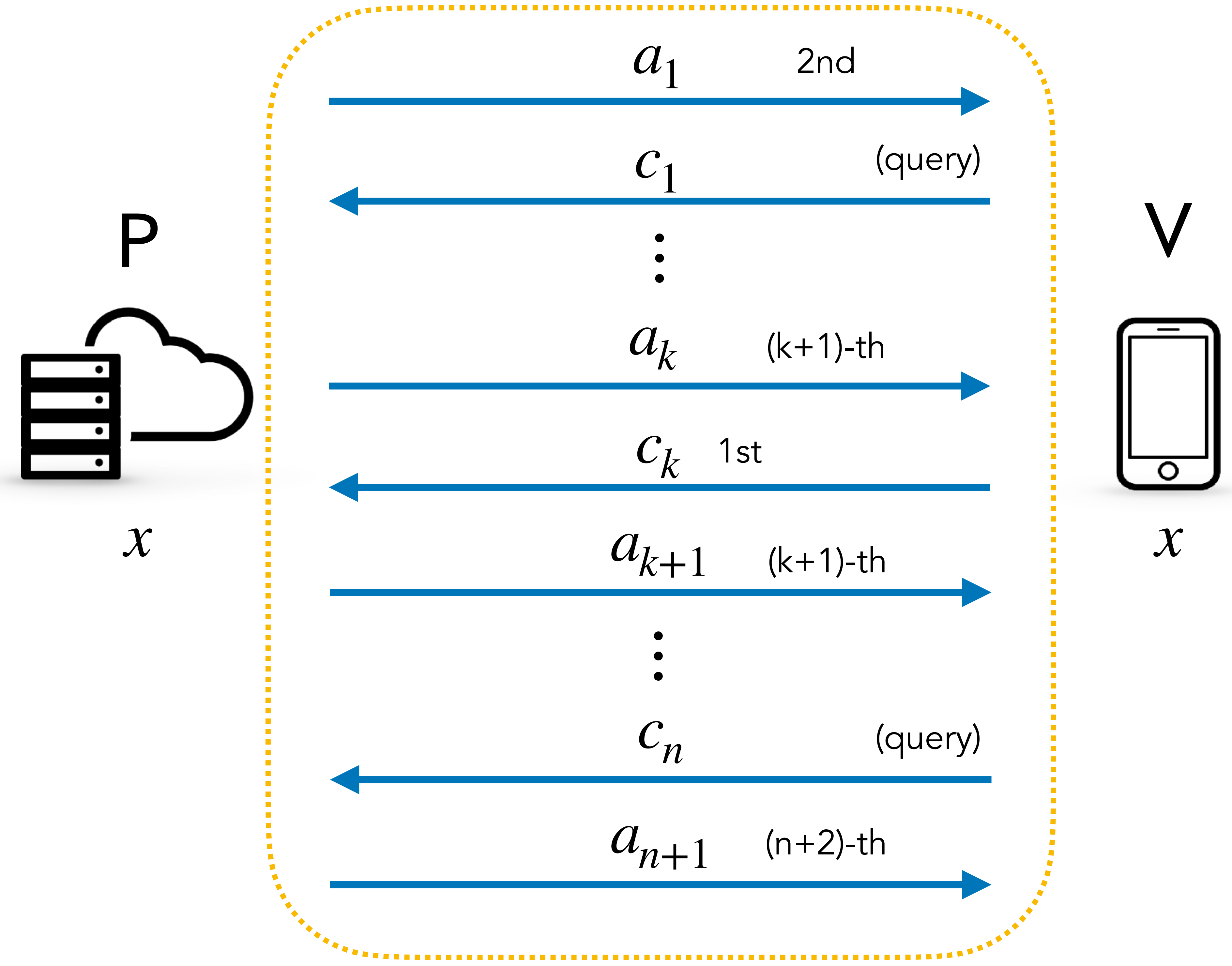
k-Zero-Knowledge and k-Unique Response

<u>Zero-Knowledge (ZK):</u> The simulator Sim may choose all challenges *<u>before</u>* computing P's messages.

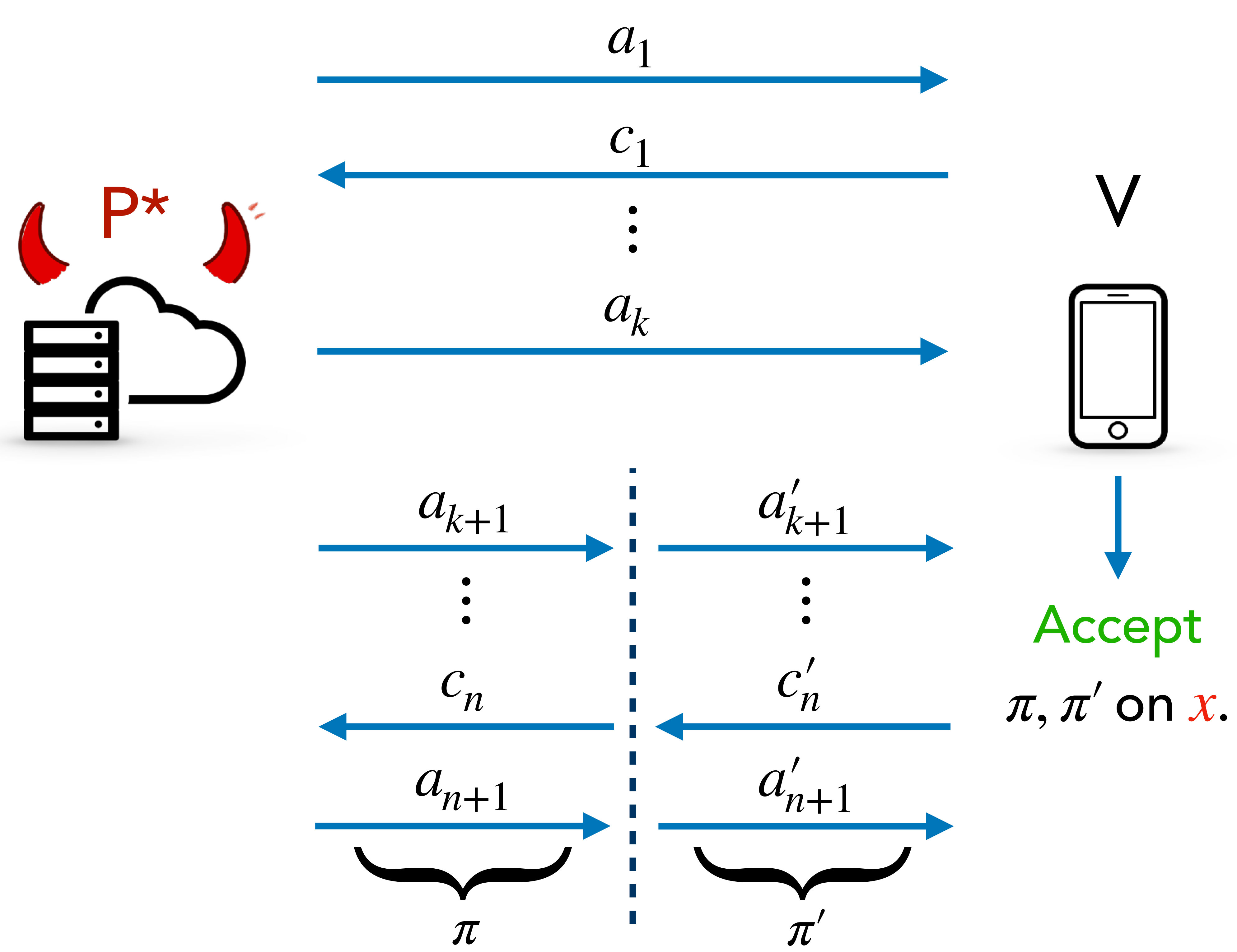


k-Zero-Knowledge and k-Unique Response

k-Zero-Knowledge (k-ZK): The simulator Sim_k may *only* choose k^{th} challenge, and compute other messages in order.

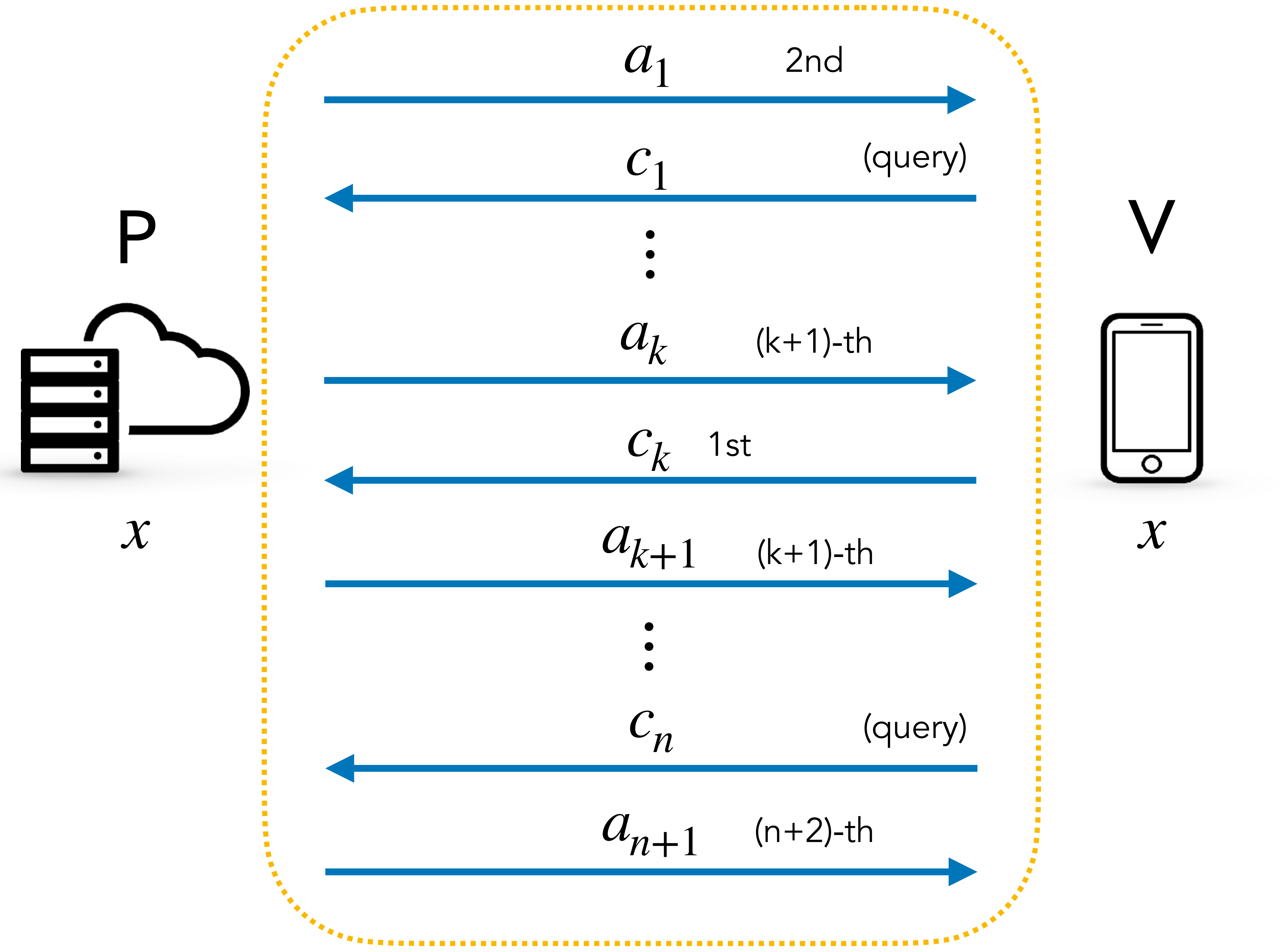


k-Unique Response (k-UR): P* cannot output accepting proofs $\pi \neq \pi'$ that agree up to round k, *even* given power to choose statement x and k^{th} challenge c_k.

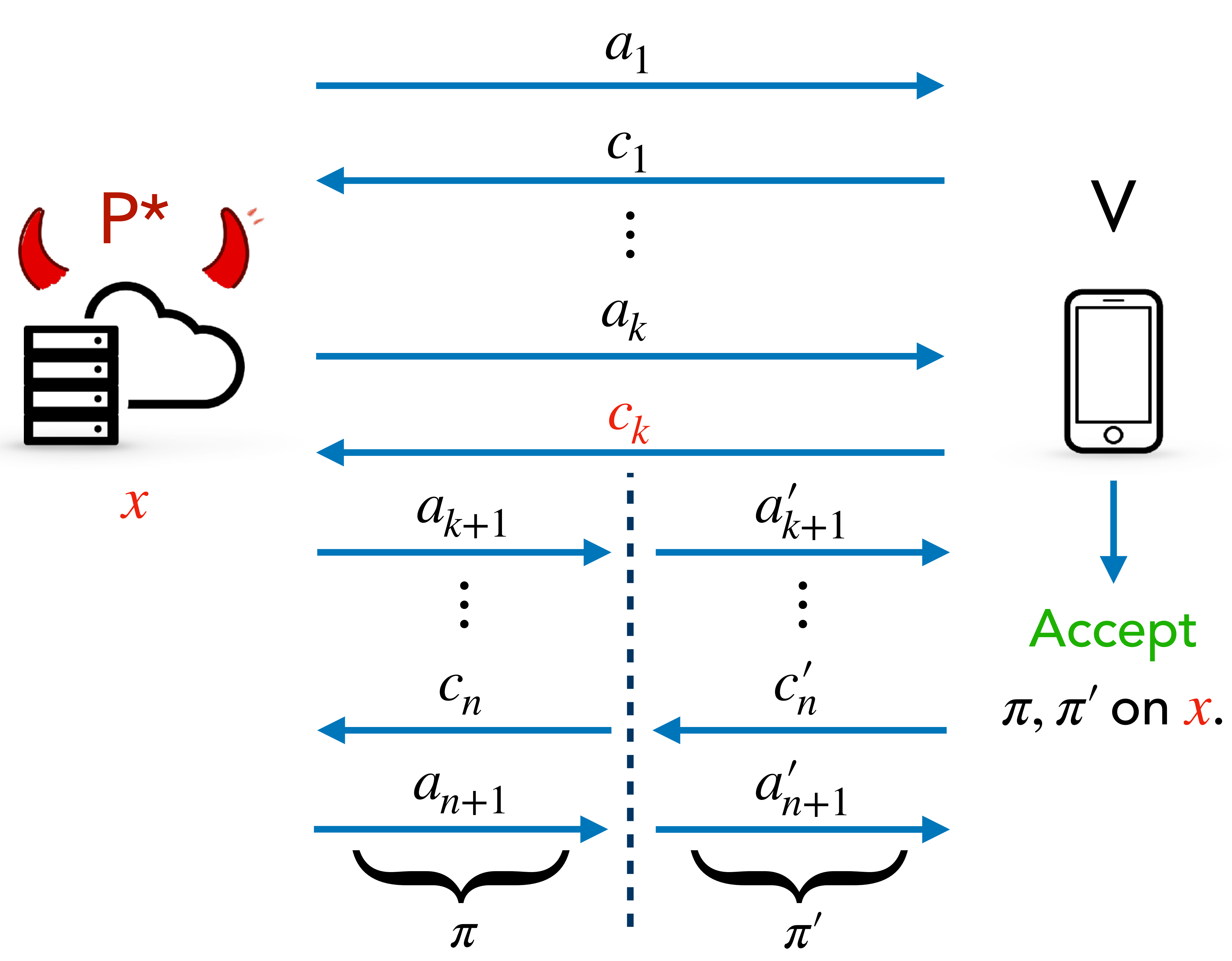


k-Zero-Knowledge and k-Unique Response

<u>k-Zero-Knowledge (k-ZK):</u> The simulator Sim_k may *<u>only</u>* choose k^{th} challenge, and compute other messages in order.



<u>k-Unique Response (k-UR):</u> P* cannot output accepting proofs $\pi \neq \pi'$ that agree up to round k, *<u>even</u>* given power to choose statement x and k^{th} challenge c_k.



k-Zero-Knowledge and k-Unique Response

<u>k-Zero-Knowledge (k-ZK):</u> The simulator FS-Sim$_k$ may *<u>only</u>* program the k^{th} challenge.

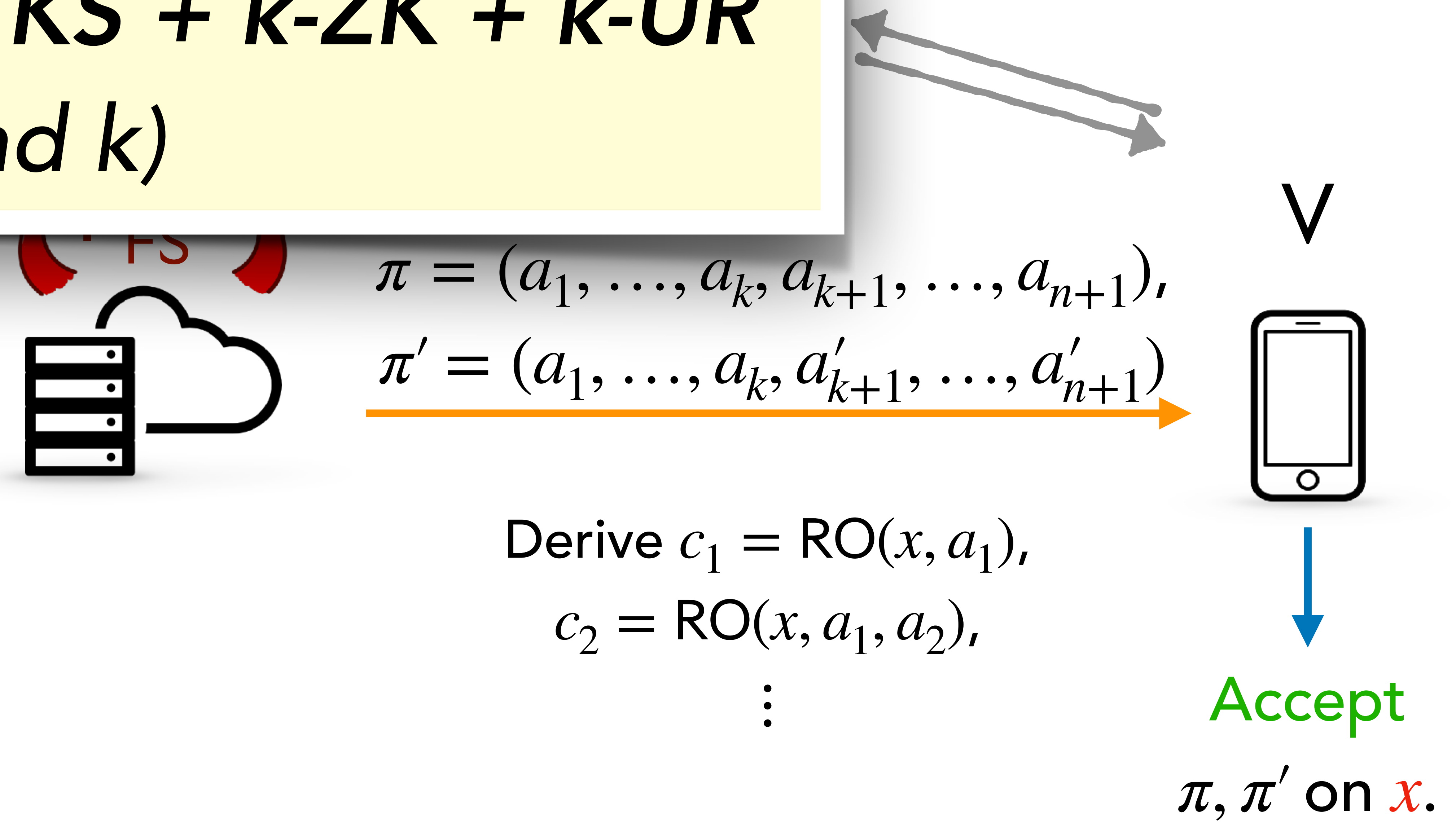
<u>k-Unique Response (k-UR):</u> P$_{FS}$* cannot output accepting proofs $\pi \neq \pi'$ that agree up to round k, *<u>even</u>* given power to choose statement x and <u>program</u> k^{th} challenge c_k.

FS-Sim

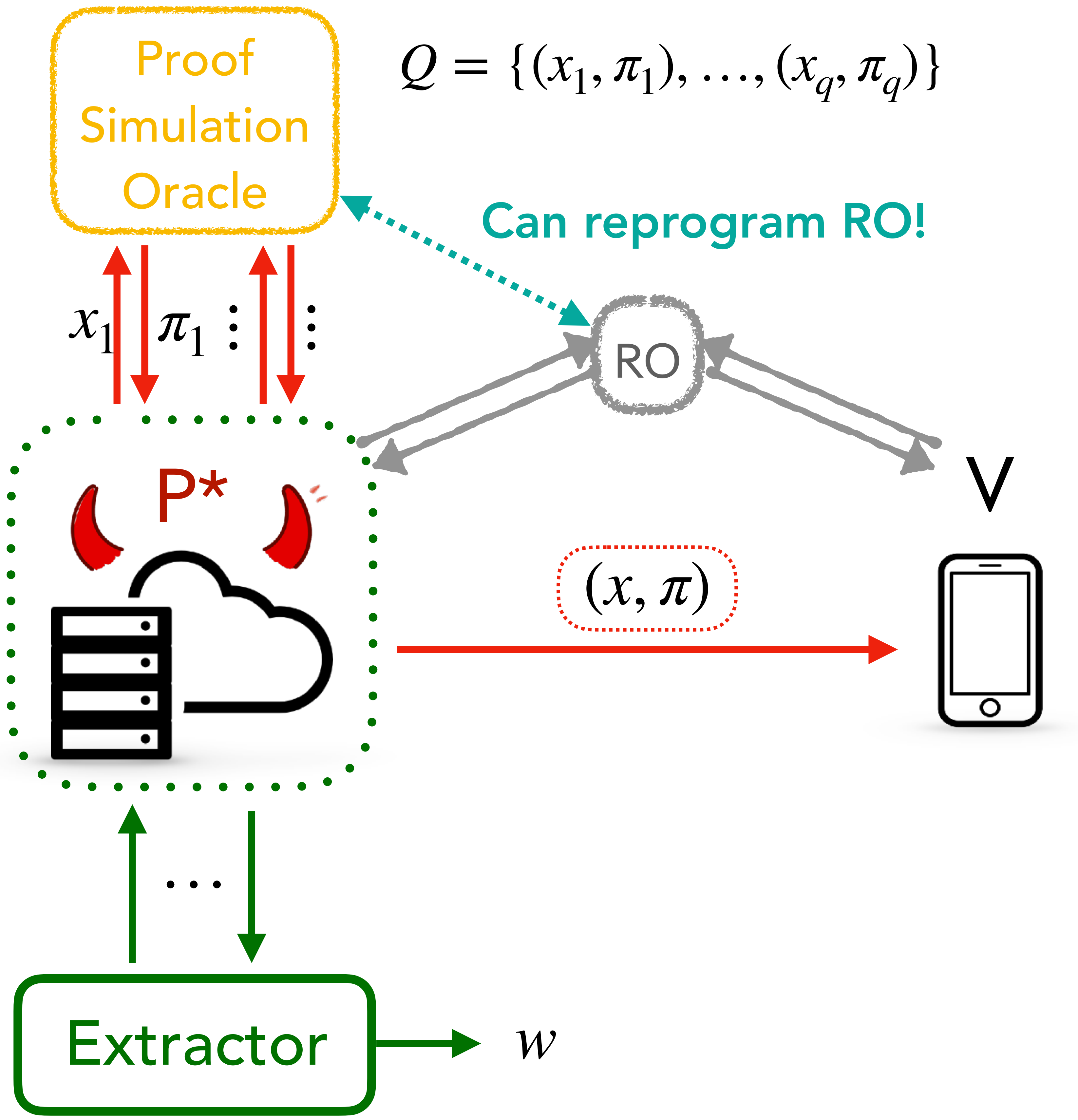
On statement x':
1. Sample random c'_k.
2. Simulate $a'_1, \ldots, a'_{n+1}$, querying for other challenges $c'_i = \mathrm{RO}(x', a'_1, \ldots, a'_i)$ for $i \neq k$.
3. Reprogram $\mathrm{RO}(x', a'_1, \ldots, a'_k) := c'_k$.

<u>Theorem (informal):</u> SIM-EXT = KS + k-ZK + k-UR

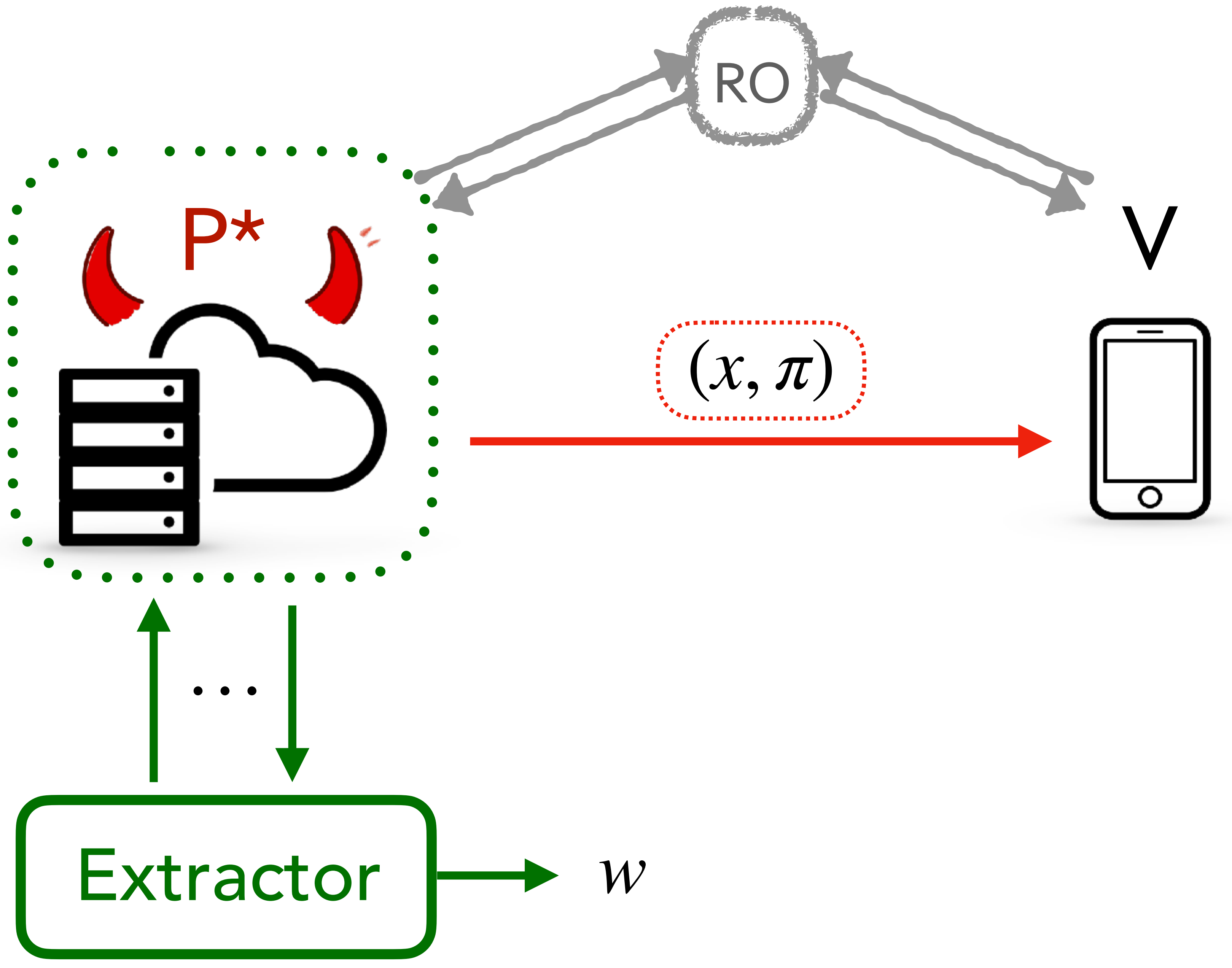
(for the <u>same</u> round k)



Reducing SIM-EXT to Knowledge Soundness



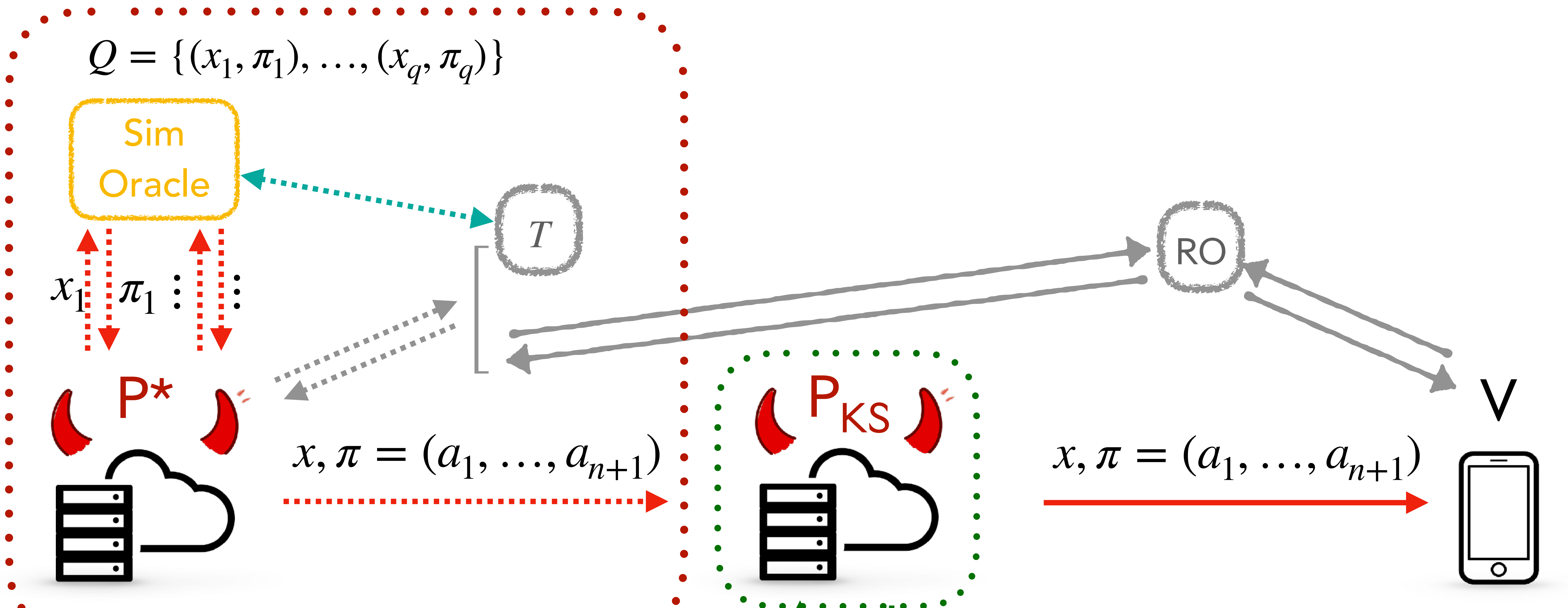
1. Build a KS prover from a SIM-EXT prover.
2. Use k-ZK and k-UR to remove proof simulation oracle.

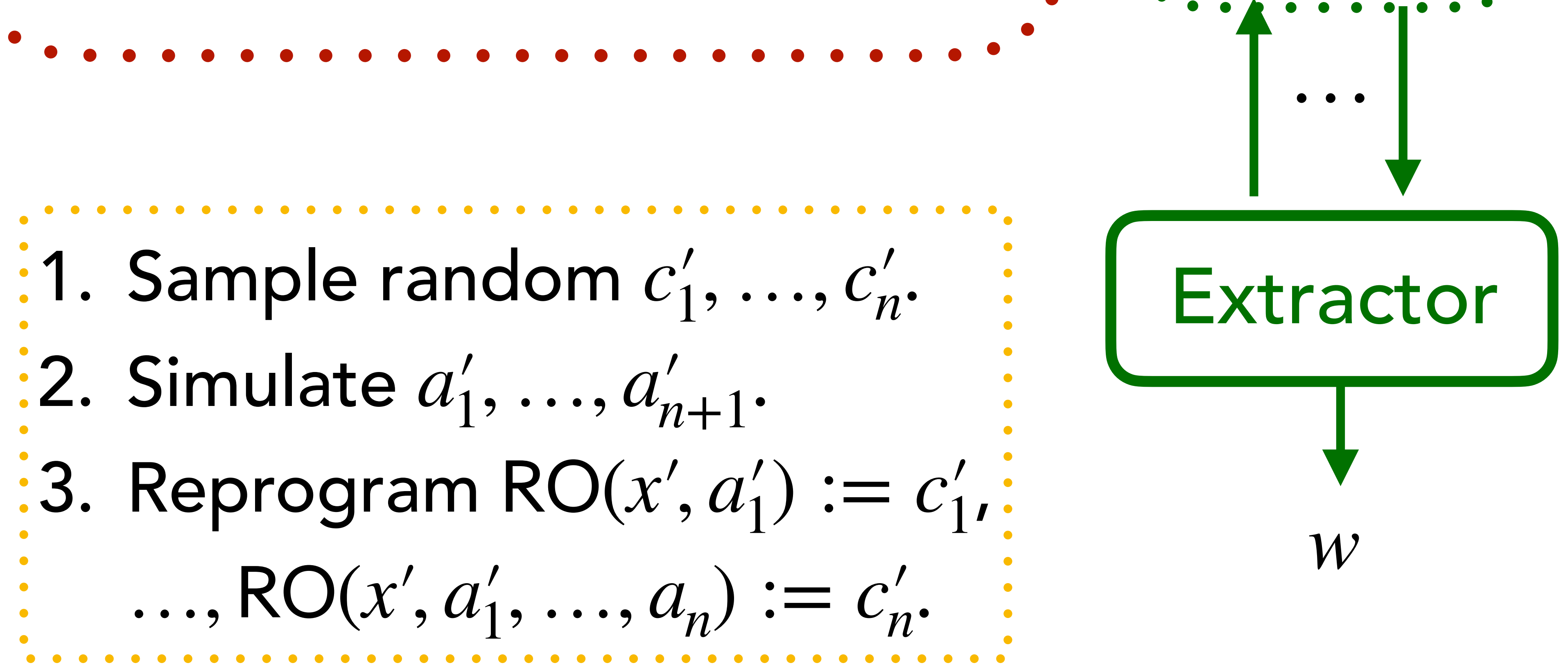


SIM-EXT: $\exists$ E such that if V$(x, \pi) = 1$ and $(x, \pi) \notin Q$, then $(x, w) \in R$.

(Adaptive) KS: $\exists$ E such that if V$(x, \pi) = 1$, then $(x, w) \in R$.

Proof Sketch: SIM-EXT = KS + k-ZK + k-UR





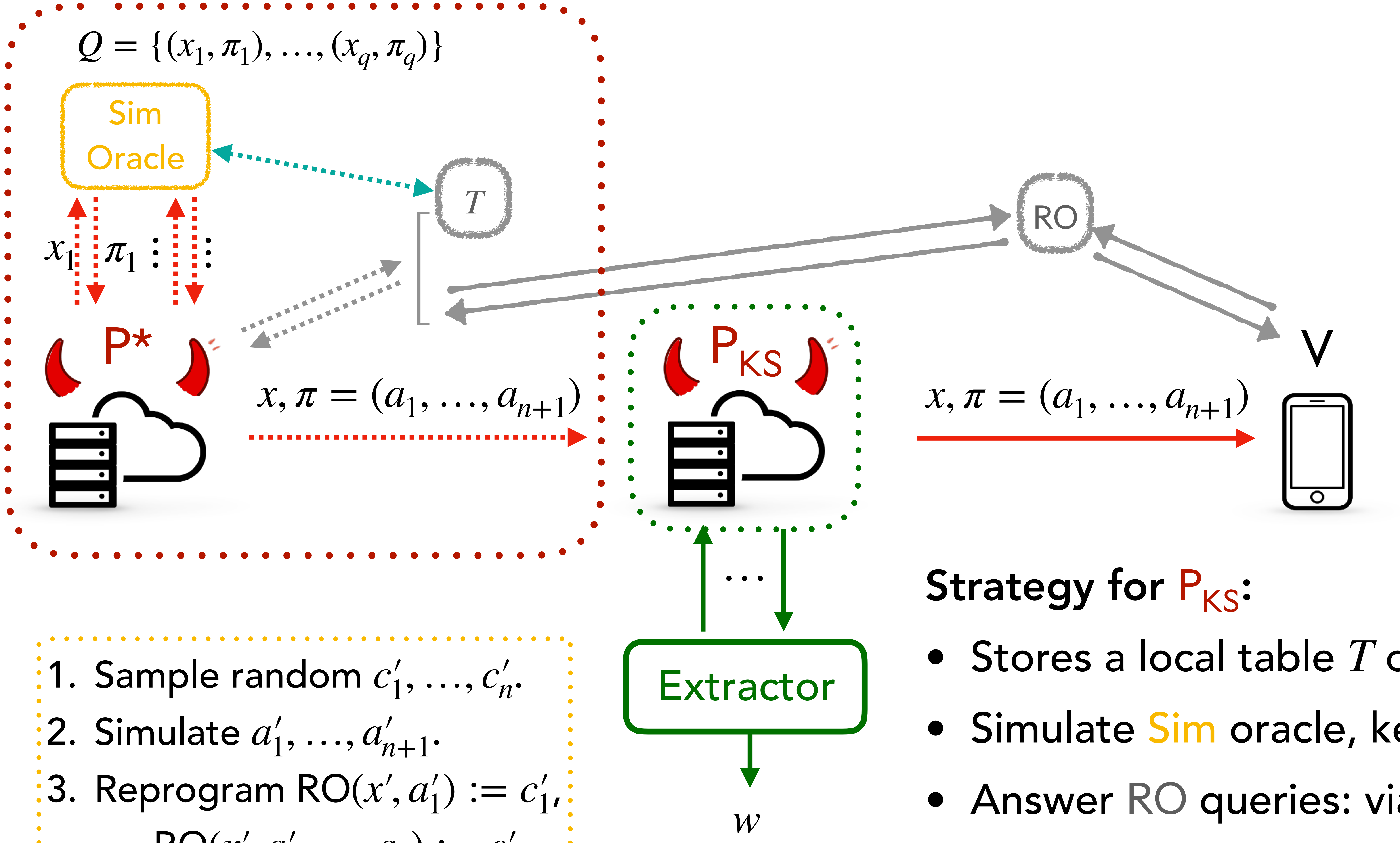
Proof Idea:

- If P_{KS} wins whenever P* wins, then we can use E to extract w.
- Differ when V queries RO on programmed queries in T.

Strategy for P_{KS}:

- Stores a local table T of RO reprogramming.
- Simulate Sim oracle, keeping all programmed queries in T.
- Answer RO queries: via T if programmed, via P_{KS}'s own RO if not.
- Receive (x, π) and pass on if π is not simulated.

Proof Sketch: SIM-EXT = KS + k-ZK + k-UR

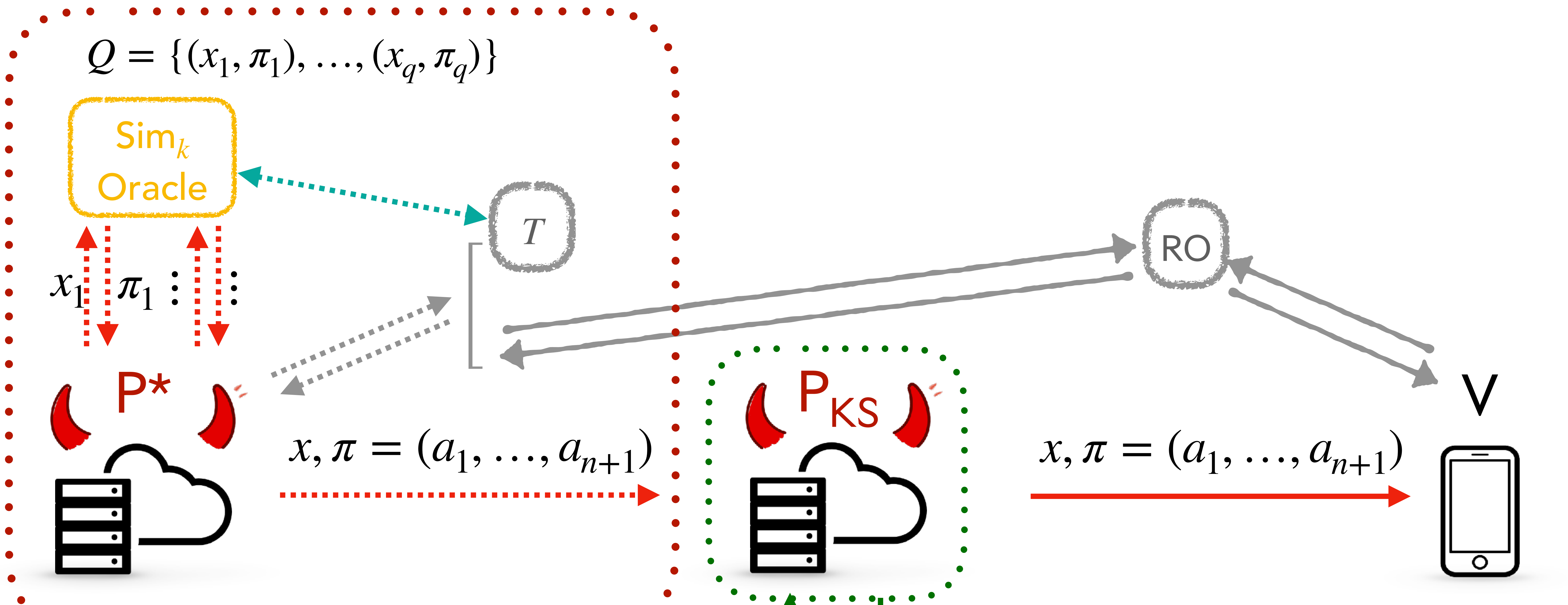


1. Sample random $c'_1, \ldots, c'_n$.
2. Simulate $a'_1, \ldots, a'_{n+1}$.
3. Reprogram $\mathsf{RO}(x', a'_1) := c'_1$, $\ldots, \mathsf{RO}(x', a'_1, \ldots, a_n) := c'_n$.

Strategy for P_{KS}:

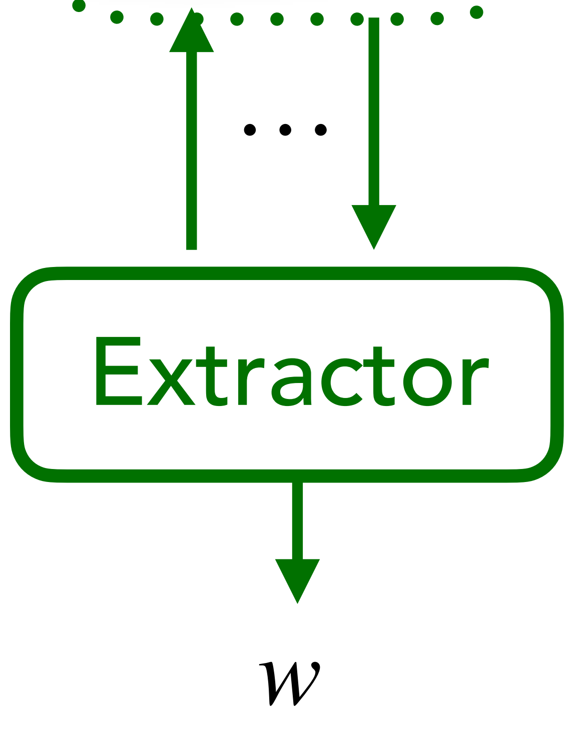
- Stores a local table T of RO reprogramming.
- Simulate Sim oracle, keeping all programmed queries in T.
- Answer RO queries: via T if programmed, via P_{KS}'s own RO if not.
- Receive (x, π) and pass on if π is not simulated.

Proof Sketch: SIM-EXT = KS + k-ZK + k-UR



Hyb_1: switch to k-ZK simulator Sim$_k$.

- $Hyb_1 \approx SIM\text{-}EXT$ via k-ZK.

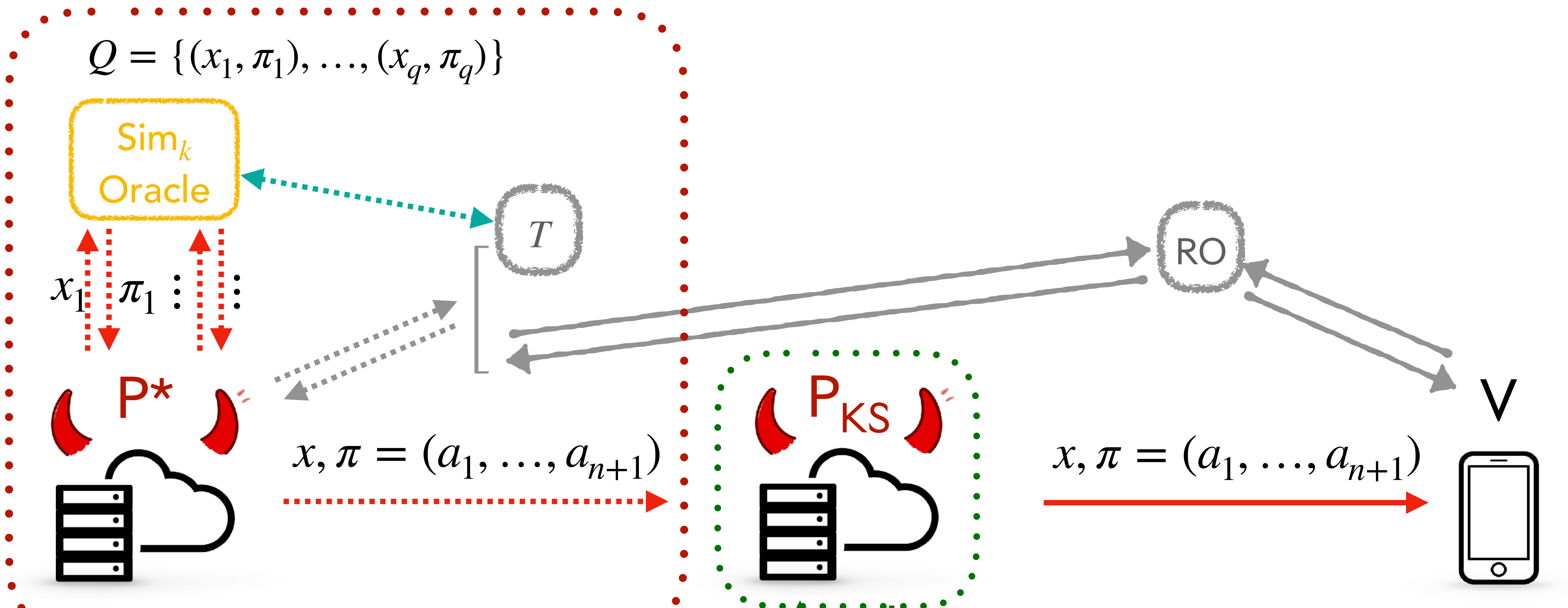


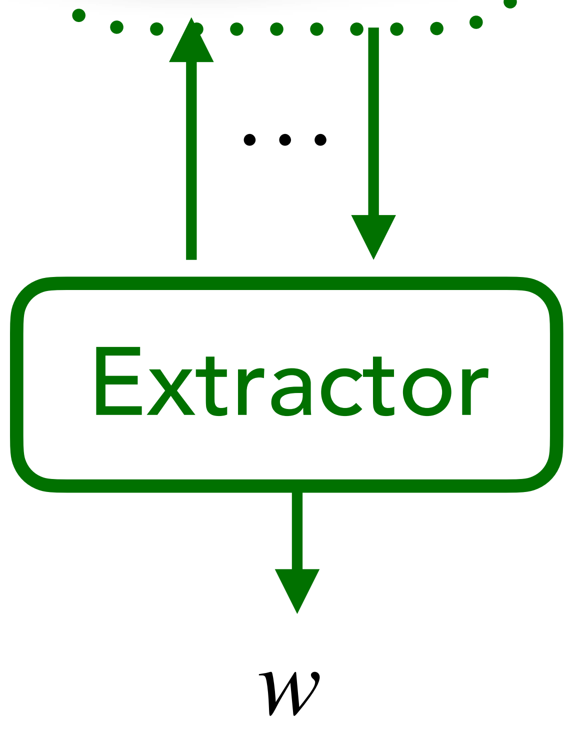
1. Sample random c'_k.
2. Simulate $a'_1, \ldots, a'_{n+1}$, querying for other challenges $c'_i = \mathsf{RO}(x', a'_1, \ldots, a'_i)$ for $i \neq k$.
3. Reprogram $\mathsf{RO}(x', a'_1, \ldots, a'_k) := c'_k$.

Strategy for P$_{\mathsf{KS}}$:

- Stores a local table T of RO reprogramming.
- Simulate Sim$_k$ oracle, keeping all programmed queries in T.
- Answer RO queries: via T if programmed, via P$_{\mathsf{KS}}$'s own RO if not.
- Receive (x, π) and pass on if π is not simulated.

Proof Sketch: SIM-EXT = KS + k-ZK + k-UR





1. Sample random c_k'.
2. Simulate $a_1', \ldots, a_{n+1}'$, querying for other challenges $c_i' = \mathsf{RO}(x', a_1', \ldots, a_i')$ for $i \neq k$.
3. Reprogram $\mathsf{RO}(x', a_1', \ldots, a_k') := c_k'$.

Hyb_1: switch to k-ZK simulator Sim_k.

- $Hyb_1 \approx SIM\text{-}EXT$ via k-ZK.

Hyb_2: abort if ***bad*** happens, where ***bad*** $= \exists\ (x, \pi') \in Q$ with $\pi|_k = \pi'|_k$.

- $Hyb_2 \approx Hyb_1$ via k-UR and up-to-bad reasoning.

In Hyb_2, P_{KS} wins whenever P* wins.

Strategy for P_{KS}:

- Stores a local table T of RO reprogramming.
- Simulate Sim_k oracle, keeping all programmed queries in T.
- Answer RO queries: via T if programmed,
  via P_{KS}'s own RO if not.
- Receive (x, π) and pass on if ***bad*** does not happen.

Agenda

1. SIM-EXT = KS + k-ZK + k-UR (for same k)
2. Instantiating SIM-EXT template for Bulletproofs & Spartan
3. Knowledge Soundness via Generalized Tree Builder

Bulletproofs Range Proof

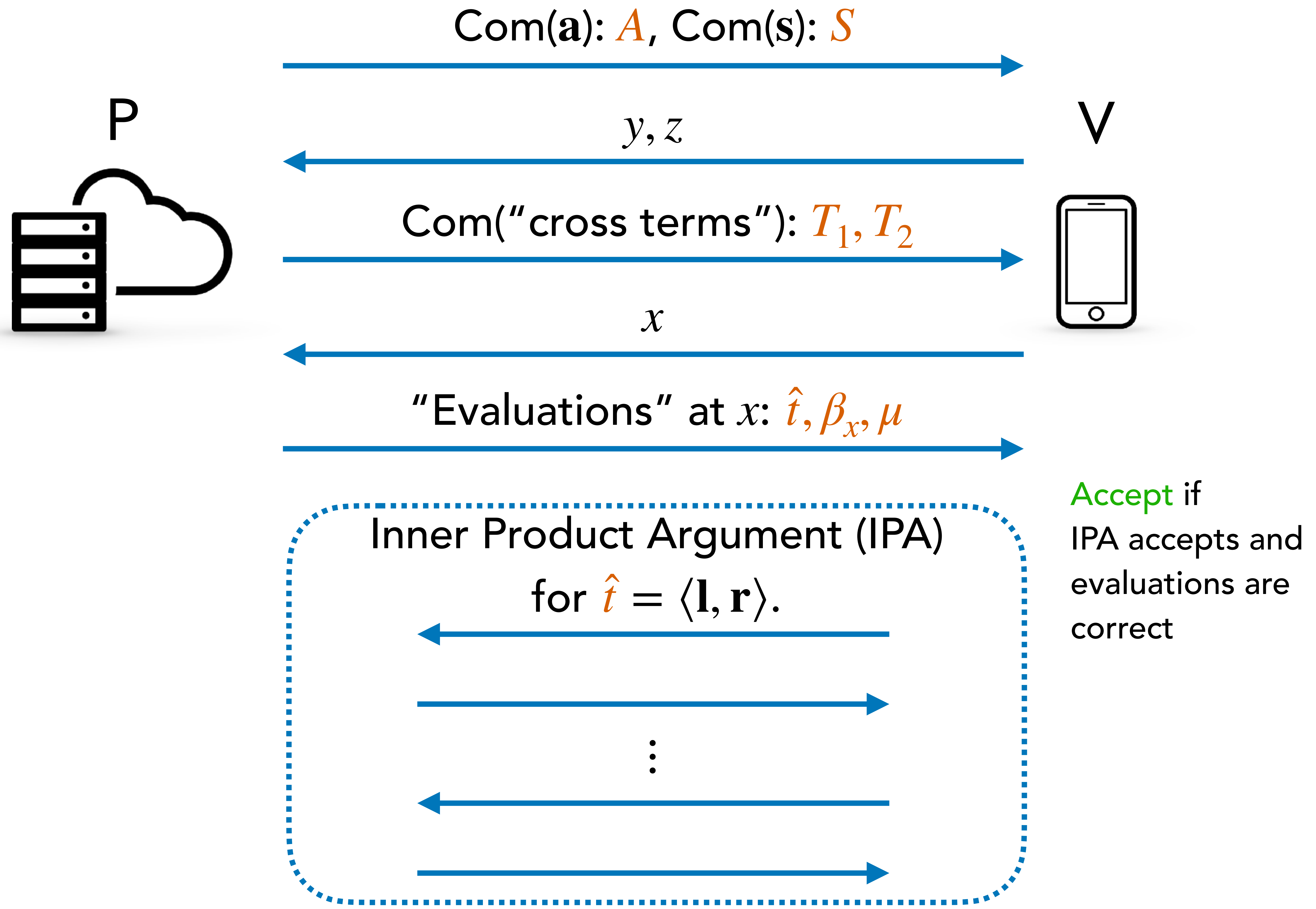
Let $\mathbf{a} = \text{BinDecomp}(v)$.

Public

Private

<u>Relation:</u> $V = g^v h^r$ and $0 \leq v \leq 2^n - 1$

<u>Recall:</u> We need to show Bulletproofs satisfy KS, k-ZK, and k-UR for the same round k.



Bulletproofs Range Proof

Let $\mathbf{a} = \text{BinDecomp}(v)$.

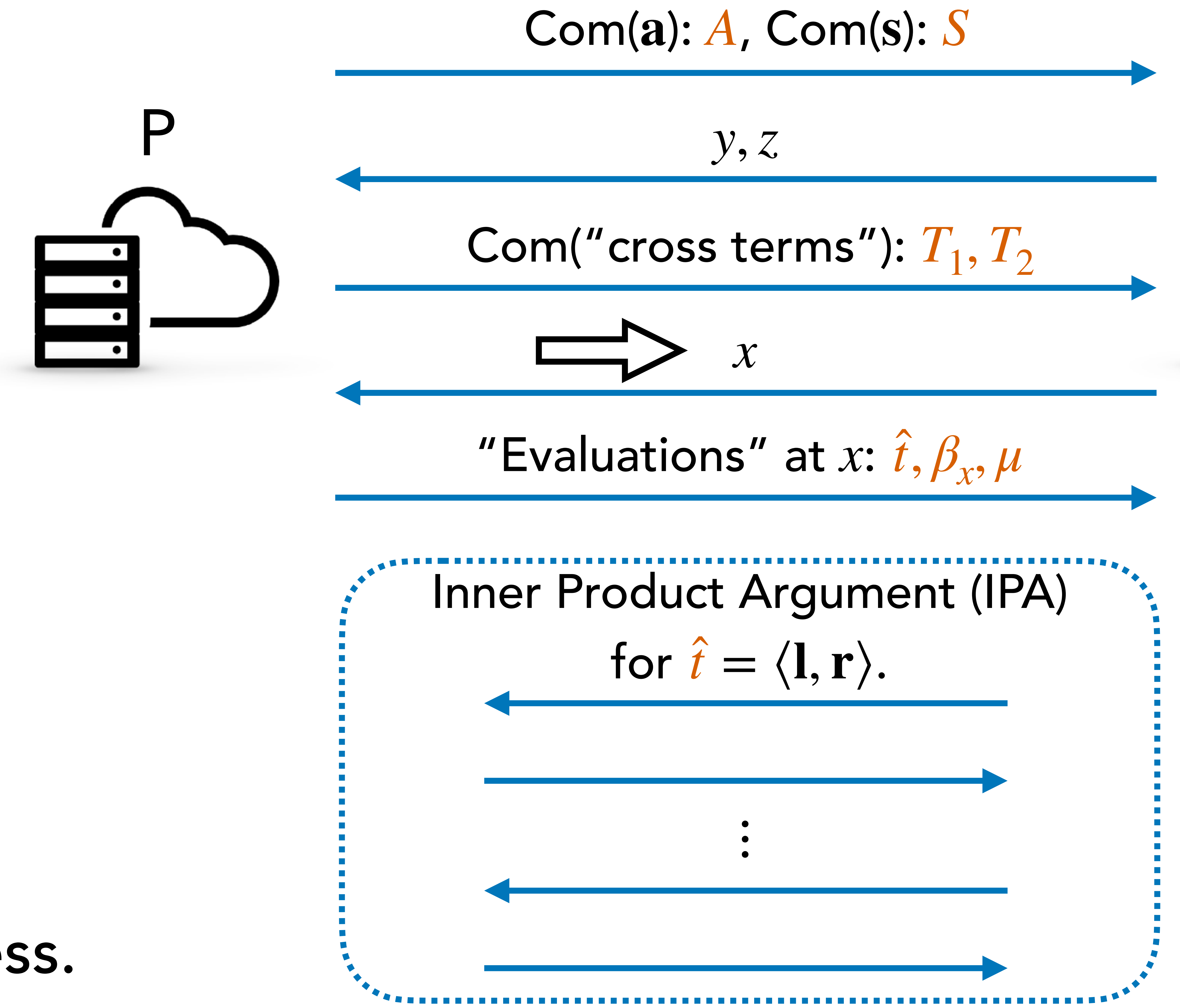
Public Private

Relation: $V = g^{v}h^{r}$ and $0 \le v \le 2^n - 1$

Recall: We need to show Bulletproofs satisfy KS, ***k*-ZK**, and ***k*-UR** for the same round k.

Q: Which round k to prove k-ZK and k-UR?

A: Choose the *last* round with P's randomness. ($k = 2$ in this case)



Accept if IPA accepts and evaluations are correct

Bulletproofs Range Proof

Let $\mathbf{a} = \text{BinDecomp}(v)$.

Public Private

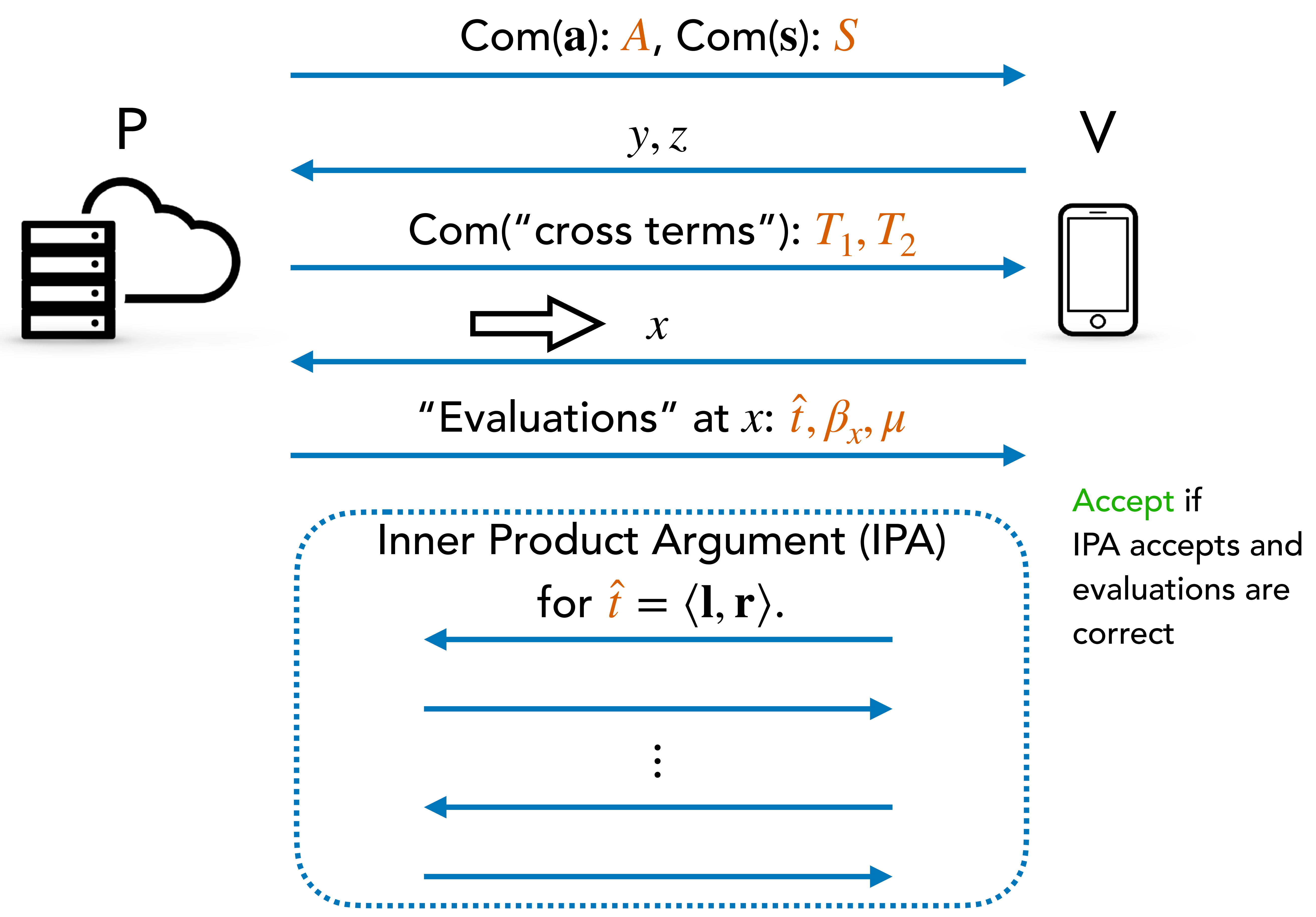
<u>Relation:</u> $V = g^v h^r$ and $0 \leq v \leq 2^n - 1$

<u>2-ZK:</u> Simulator can only choose x first.

<u>Problem:</u> How to simulate IPA?

<u>Idea:</u>

1. Run the honest prover's algorithm with a "*<u>fake</u>*" witness.
2. Resolve contradiction via choosing k^{th} and $(k+1)^{th}$ message *<u>at the same time</u>*.



Bulletproofs Range Proof

Let $\mathbf{a}$ = BinDecomp(v).

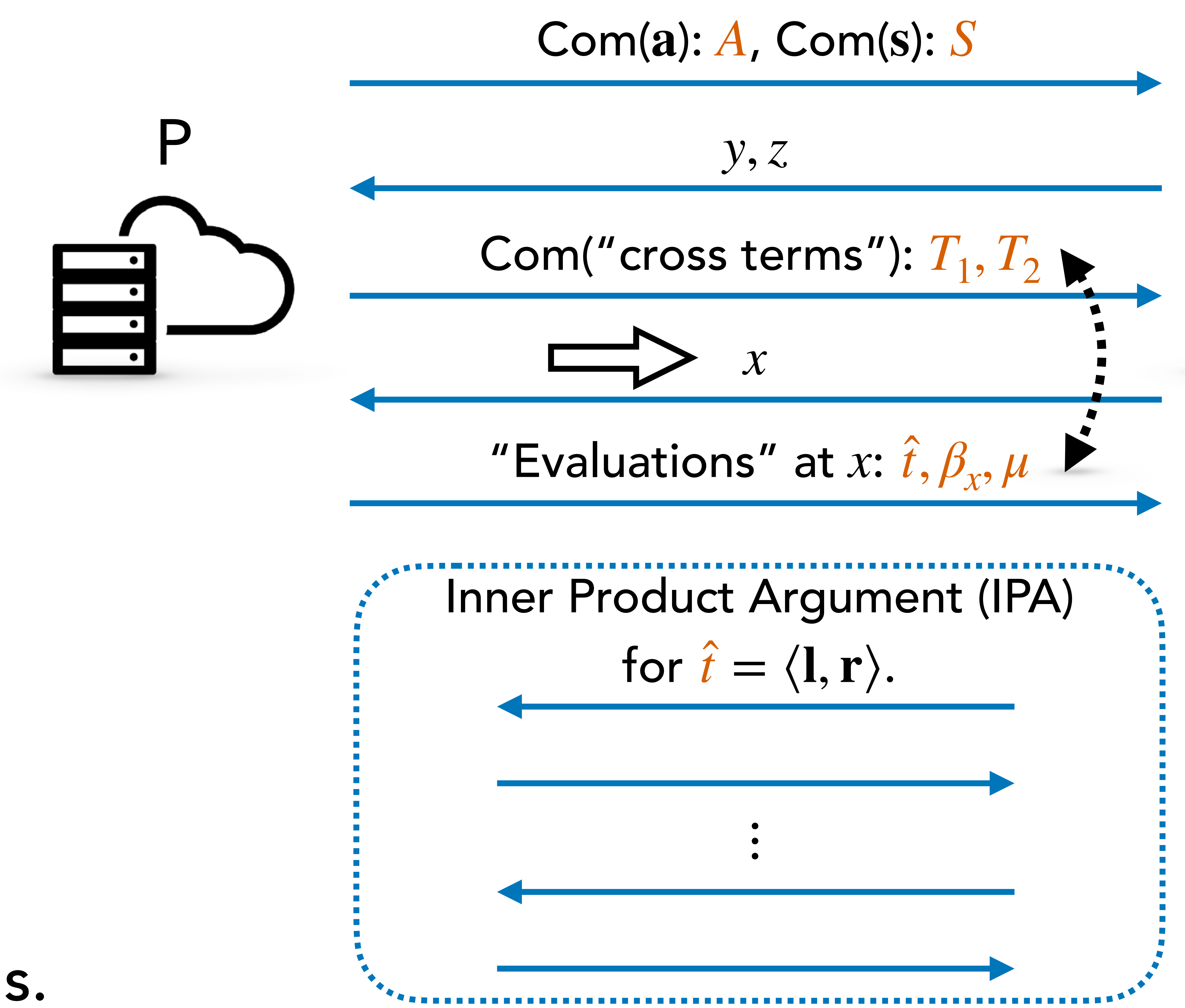
Public Private

Relation: $V = g^{v}h^{r}$ and $0 \leq v \leq 2^n - 1$

2-ZK: Simulator can only choose x first.

1. Pick random 2^{nd} challenge x.
2. Pick *arbitrary* witness $\mathbf{a}$, *random* blind $\mathbf{s}$.
   Compute A, S.
3. Pick *random* evaluations $\hat{t}, \beta_x, \mu$.
   Choose T_1, T_2 *consistent* with evaluations.

$$g^{\hat{t}} \cdot h^{\beta_x} = V^{z^2} \cdot g^{\delta(y,z)} \cdot T_1{}^{x} \cdot T_2{}^{x^2}$$
(eval check)



Accept if
IPA accepts and
evaluations are
correct

Bulletproofs Range Proof

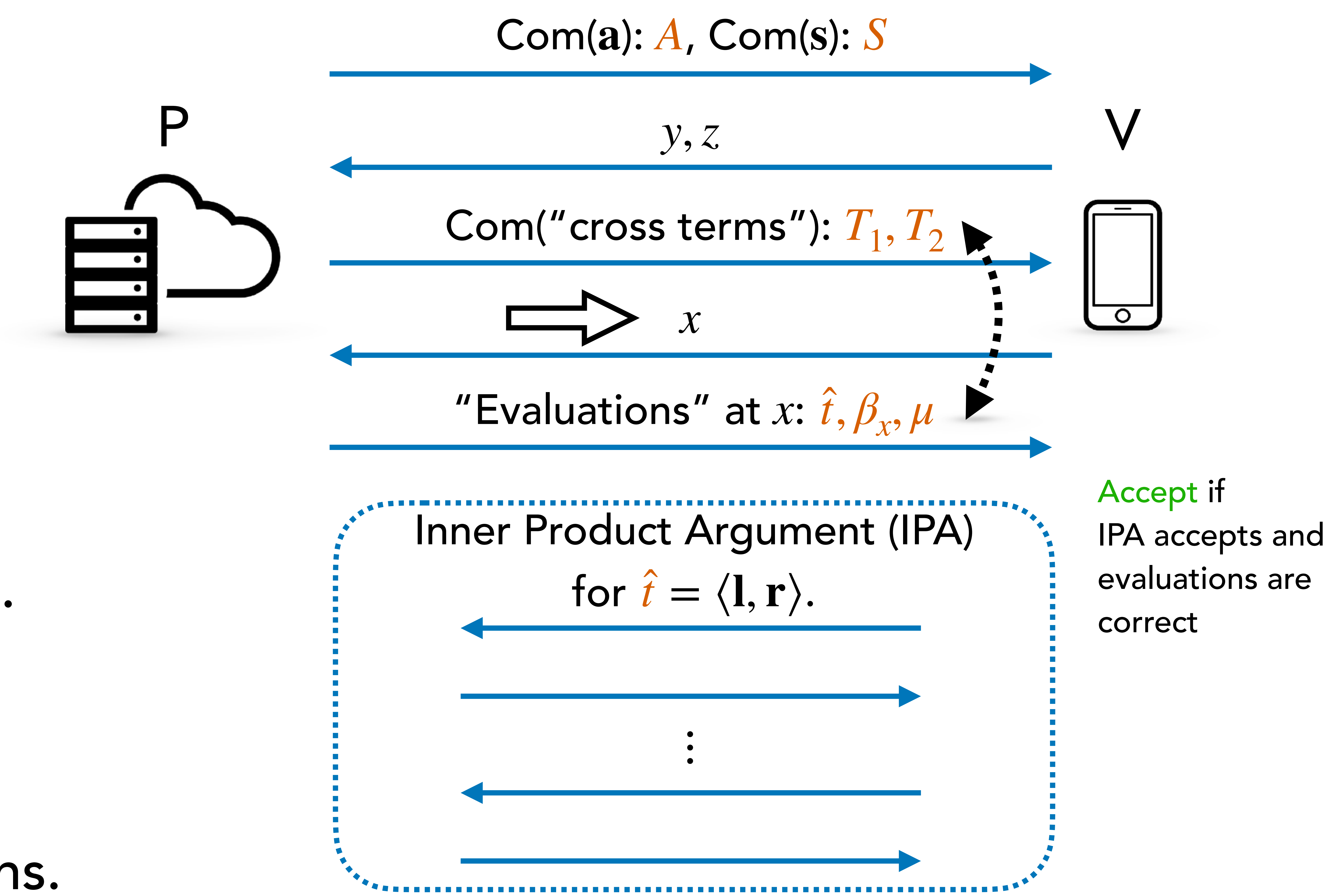
Let $\mathbf{a} = \mathrm{BinDecomp}(v)$.

Public Private

Relation: $V = g^{v}h^{r}$ and $0 \leq v \leq 2^n - 1$

2-ZK: Simulator can only choose x first.

1. Pick random 2^{nd} challenge x.
2. Pick *arbitrary* witness $\mathbf{a}$, *random* blind $\mathbf{s}$. Compute A, S.
3. Pick *random* evaluations $\hat{t}, \beta_x, \mu$. Choose T_1, T_2 *consistent* with evaluations.
4. Execute IPA with satisfying witness $\mathbf{l}, \mathbf{r}$ (derived from $\mathbf{a}, \mathbf{s}$).



Bulletproofs Range Proof

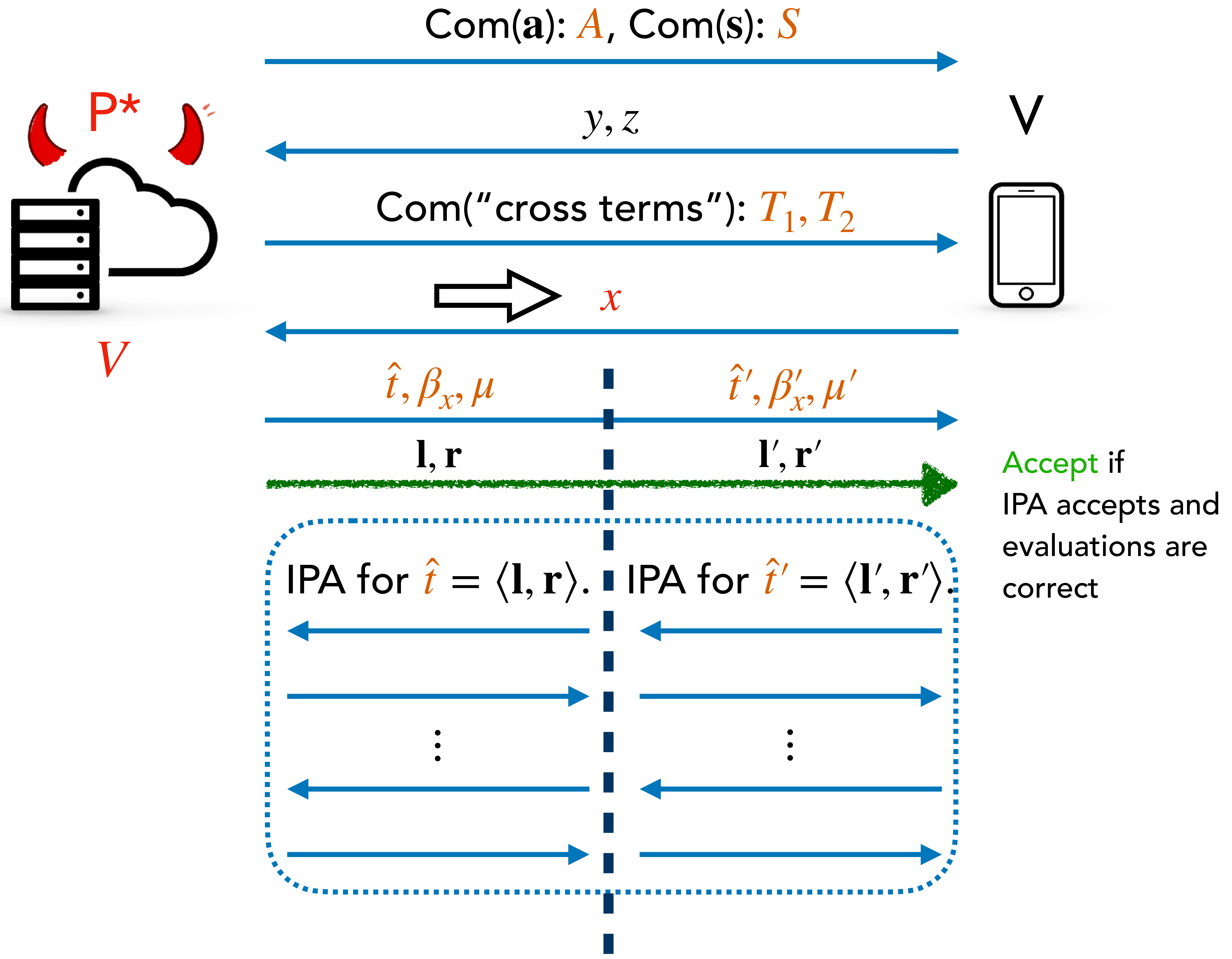
Let $\mathbf{a} = \mathrm{BinDecomp}(v)$.

2-UR: P* cannot produce two accepting proofs $\pi \neq \pi'$ that agree on A, S, T_1, T_2 (even if it can choose V and x).

1. Use KS extractor for IPA to extract $(\mathbf{l}, \mathbf{r})$ from π_{IPA}, $(\mathbf{l}', \mathbf{r}')$ from π'_{IPA}.
2. If $(\hat{t}, \beta_x) \neq (\hat{t}', \beta'_x)$, we have a non-trivial DLOG relation $\Longrightarrow$ P* breaks DLOG.

$$g^{\hat{t}} \cdot h^{\beta_x} = V^{z^2} \cdot g^{\delta(y,z)} \cdot T_1^{x} \cdot T_2^{x^2} = g^{\hat{t}'} \cdot h^{\beta'_x}$$

(eval check)

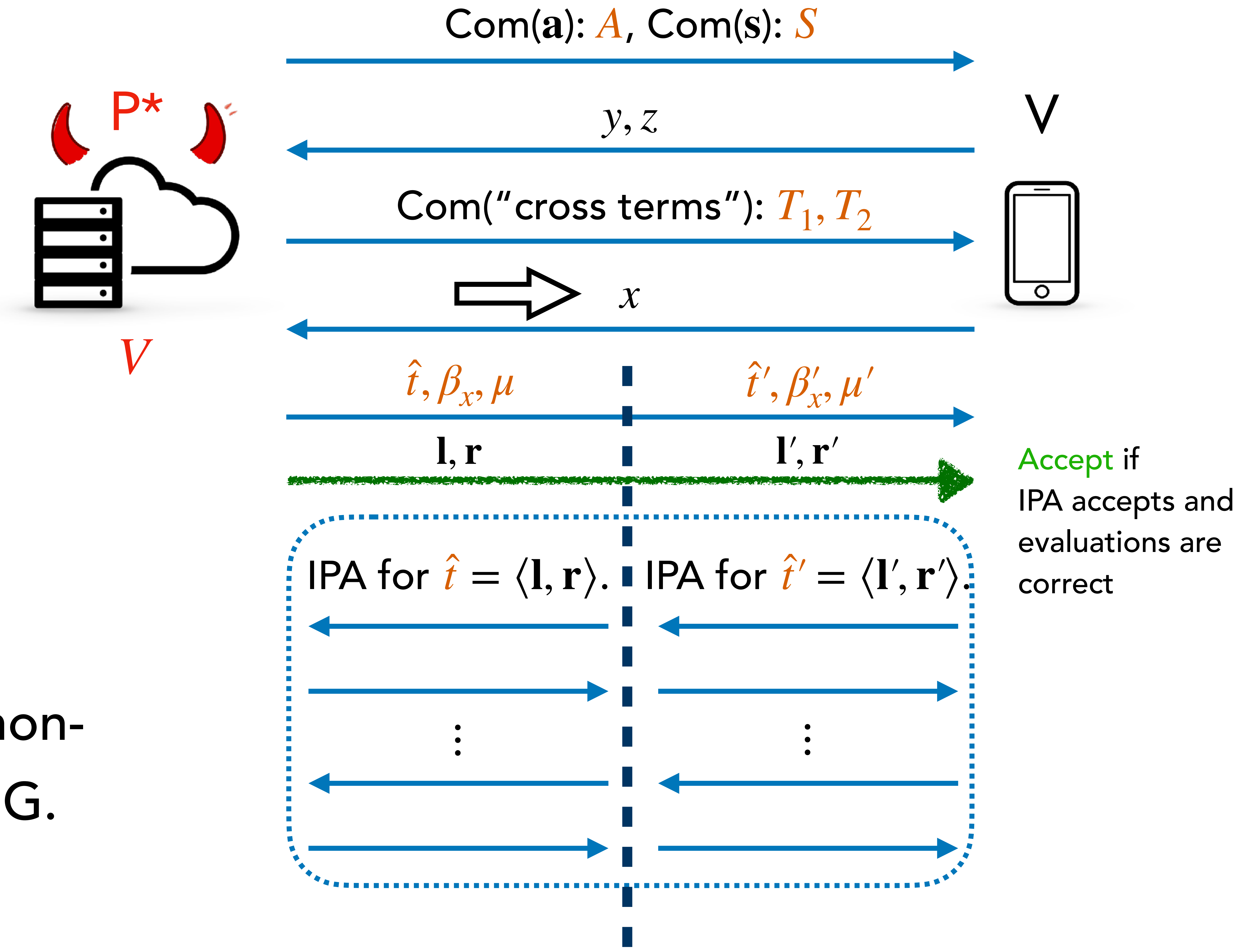


Bulletproofs Range Proof

Let $\mathbf{a} = \text{BinDecomp}(v)$.

<u>**2-UR:**</u> P* cannot produce two accepting proofs $\pi \neq \pi'$ that agree on A, S, T_1, T_2 (even if it can choose V and x).

1. Use KS extractor for IPA to extract $(\mathbf{l}, \mathbf{r})$ from π_{IPA}, $(\mathbf{l}', \mathbf{r}')$ from π'_{IPA}.
2. If $(\hat{t}, \beta_x) \neq (\hat{t}', \beta'_x)$, we have a non-trivial DLOG relation $\Longrightarrow$ P* breaks DLOG.
3. Else if $(\mathbf{l}, \mathbf{r}, \mu) \neq (\mathbf{l}', \mathbf{r}', \mu')$, we also get a non-trivial DLOG relation $\Longrightarrow$ P* breaks DLOG.
4. Else $(\mathbf{l}, \mathbf{r}) = (\mathbf{l}', \mathbf{r}')$ but $\pi_{\text{IPA}} \neq \pi'_{\text{IPA}} \Longrightarrow$ P* breaks DLOG.

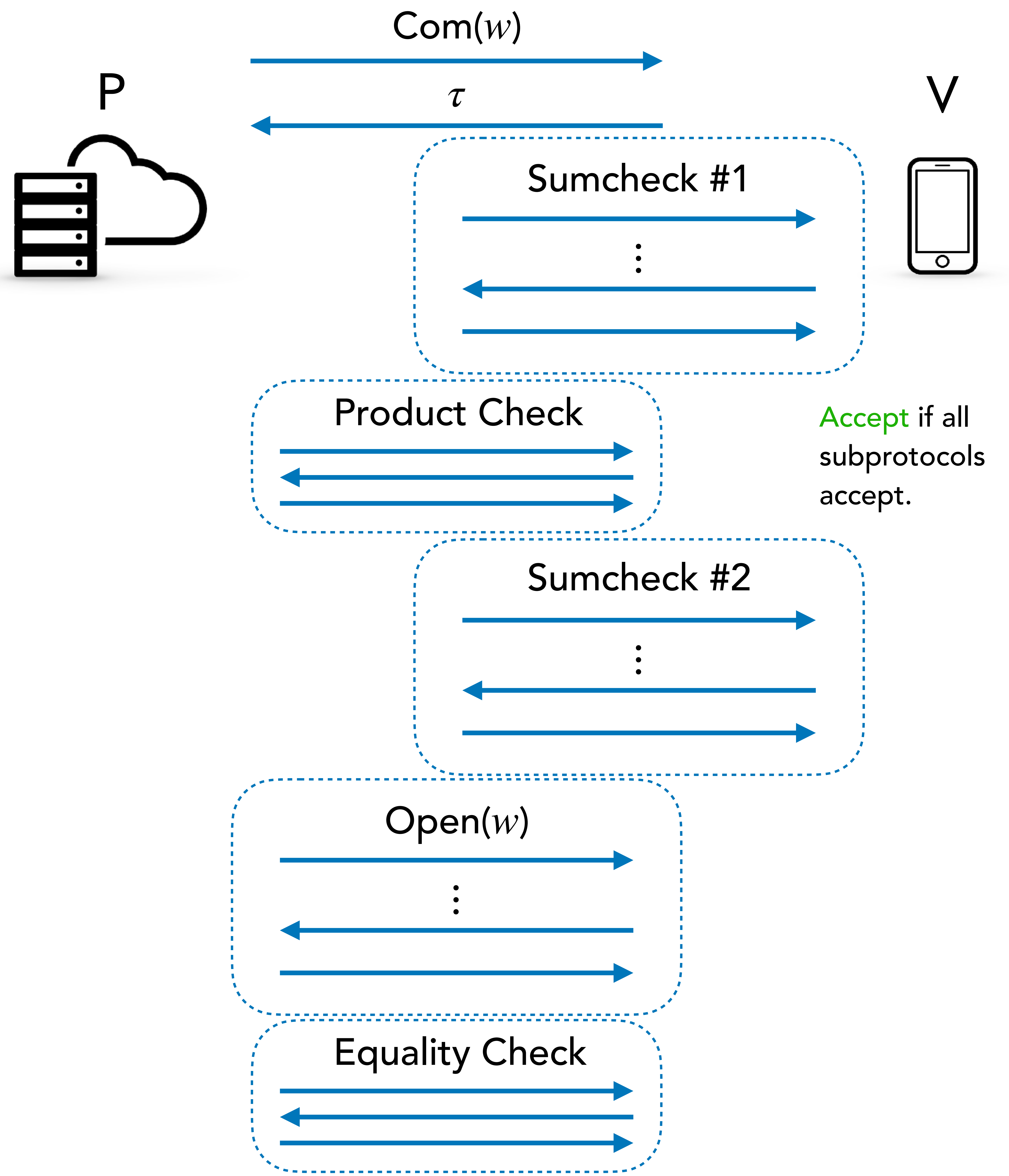


Spartan

A, B, C, x = Public w = Private

Relation (R1CS): $(A \cdot Z) \circ (B \cdot Z) = C \cdot Z,$

where $Z = (x, w, 1)$.



Spartan

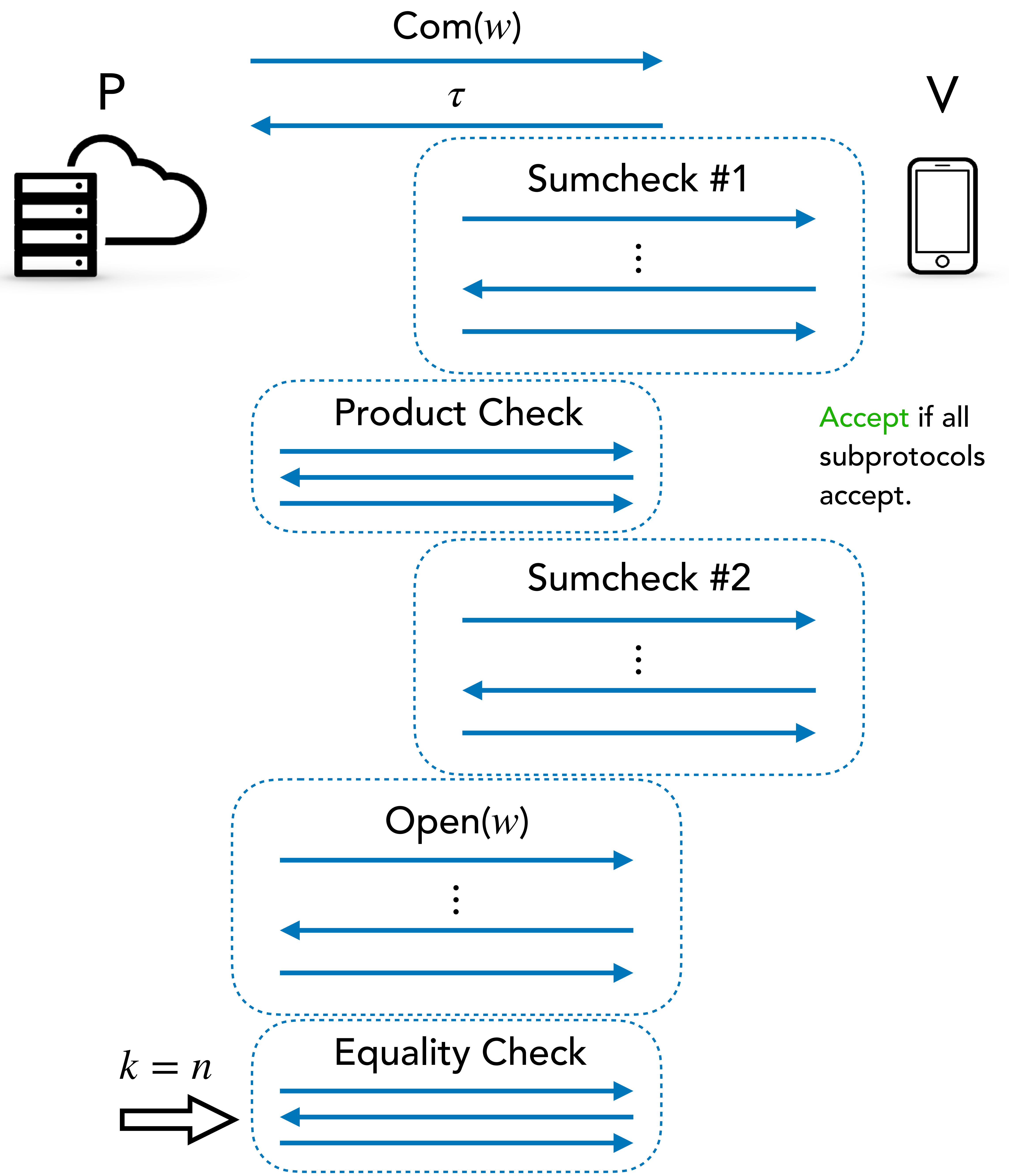
$A, B, C, x =$ Public

$w =$ Private

<u>Relation (R1CS):</u> $(A \cdot Z) \circ (B \cdot Z) = C \cdot Z$,

where $Z = (x, w, 1)$.

<u>Q:</u> Which round k has the last randomness?



Spartan

$A, B, C, x =$ Public $w =$ Private

Relation (R1CS): $(A \cdot Z) \circ (B \cdot Z) = C \cdot Z$,

where $Z = (x, w, 1)$.

Q: Which round k has the last randomness?

n-UR: Equality check is a Σ protocol.

Equality Check: $C_1 = C_2 \cdot h^r$

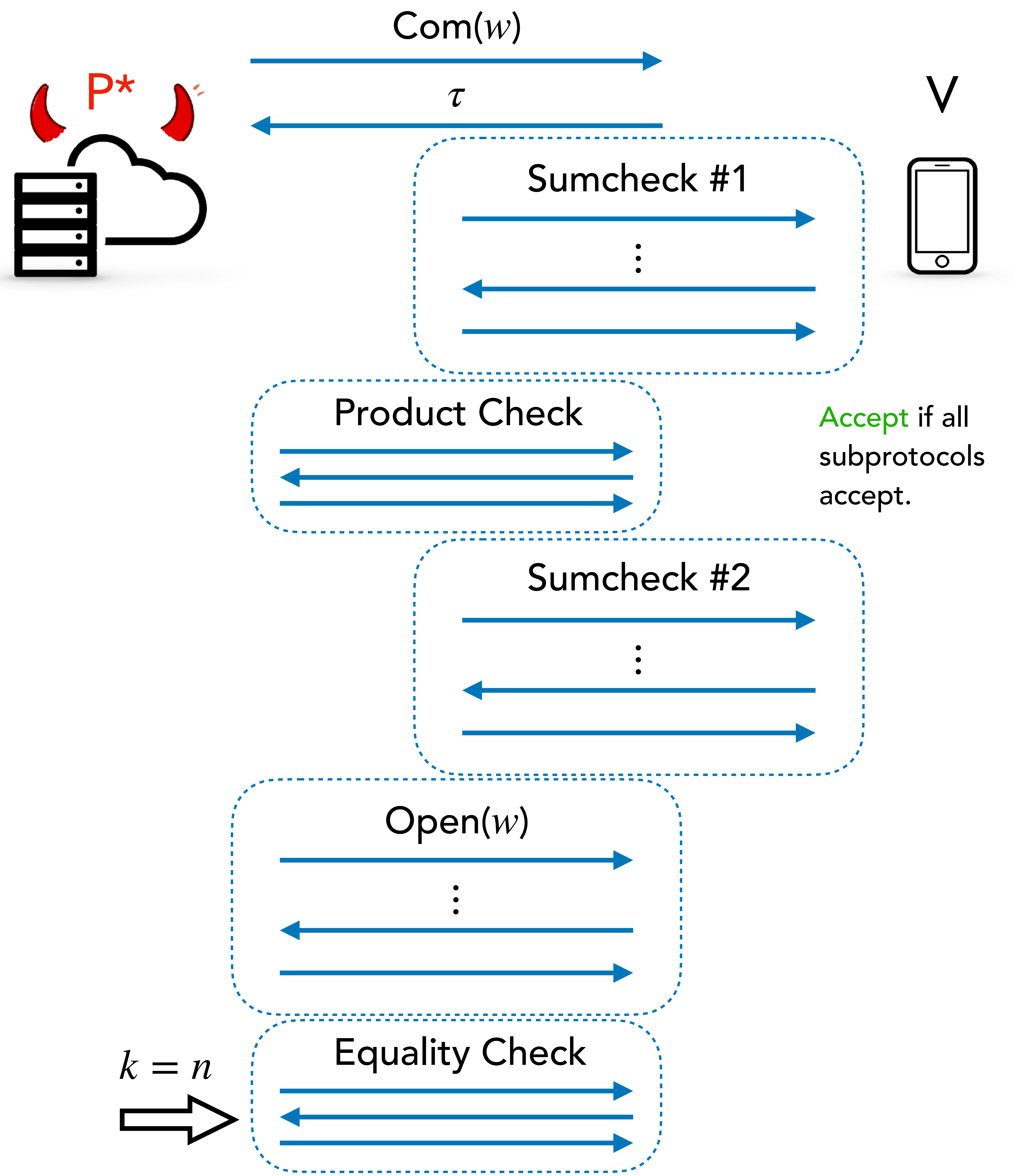
$A = h^t$ $\longrightarrow$

c $\longleftarrow$

$z = c \cdot r + t$ $\longrightarrow$

Verify $h^z \stackrel{?}{=} (C_1/C_2)^c \cdot A$

$\Longrightarrow$ z is *uniquely* determined given any c.



Spartan

A, B, C, x = Public w = Private

Relation (R1CS): $(A \cdot Z) \circ (B \cdot Z) = C \cdot Z$,

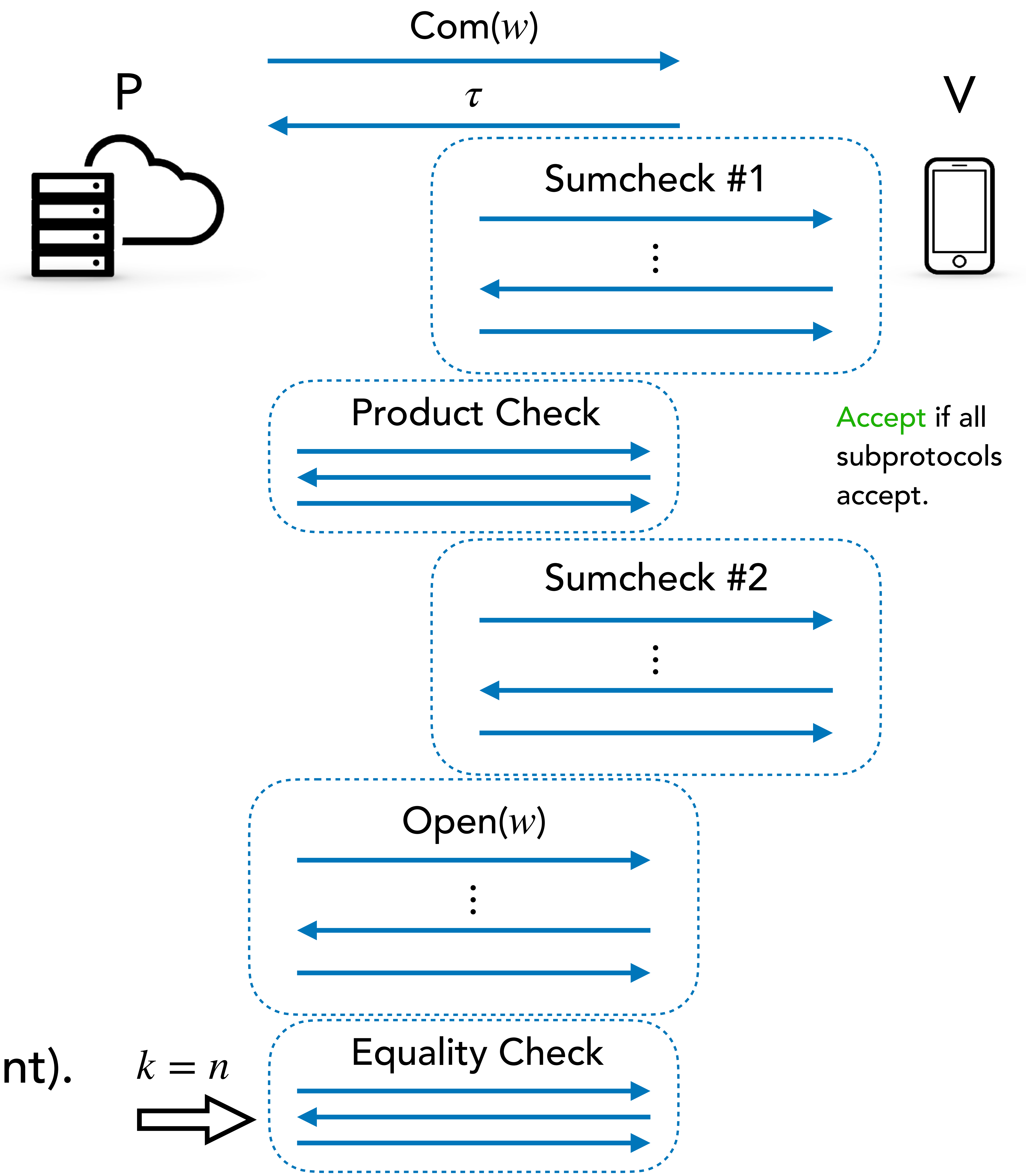
where $Z = (x, w, 1)$.

***n*-ZK:** Simulator can only reprogram c.

Problem: How to simulate all prior subprotocols?

Idea:

1. Generate *real* proofs of subprotocols using a "*fake*" witness w.
2. Delay contradiction until equality check.
3. Simulate equality check (possible for *all* statement).

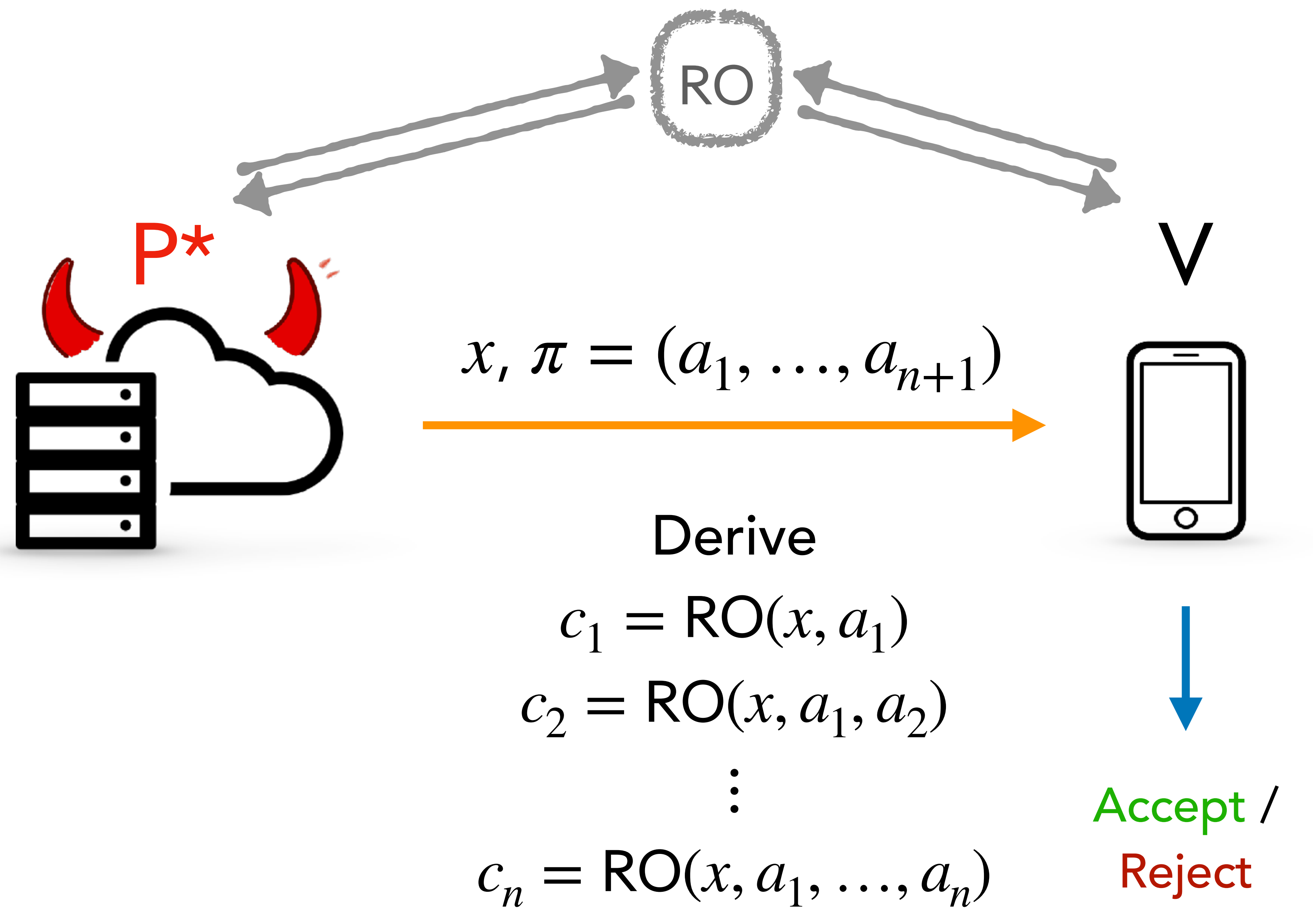


Agenda

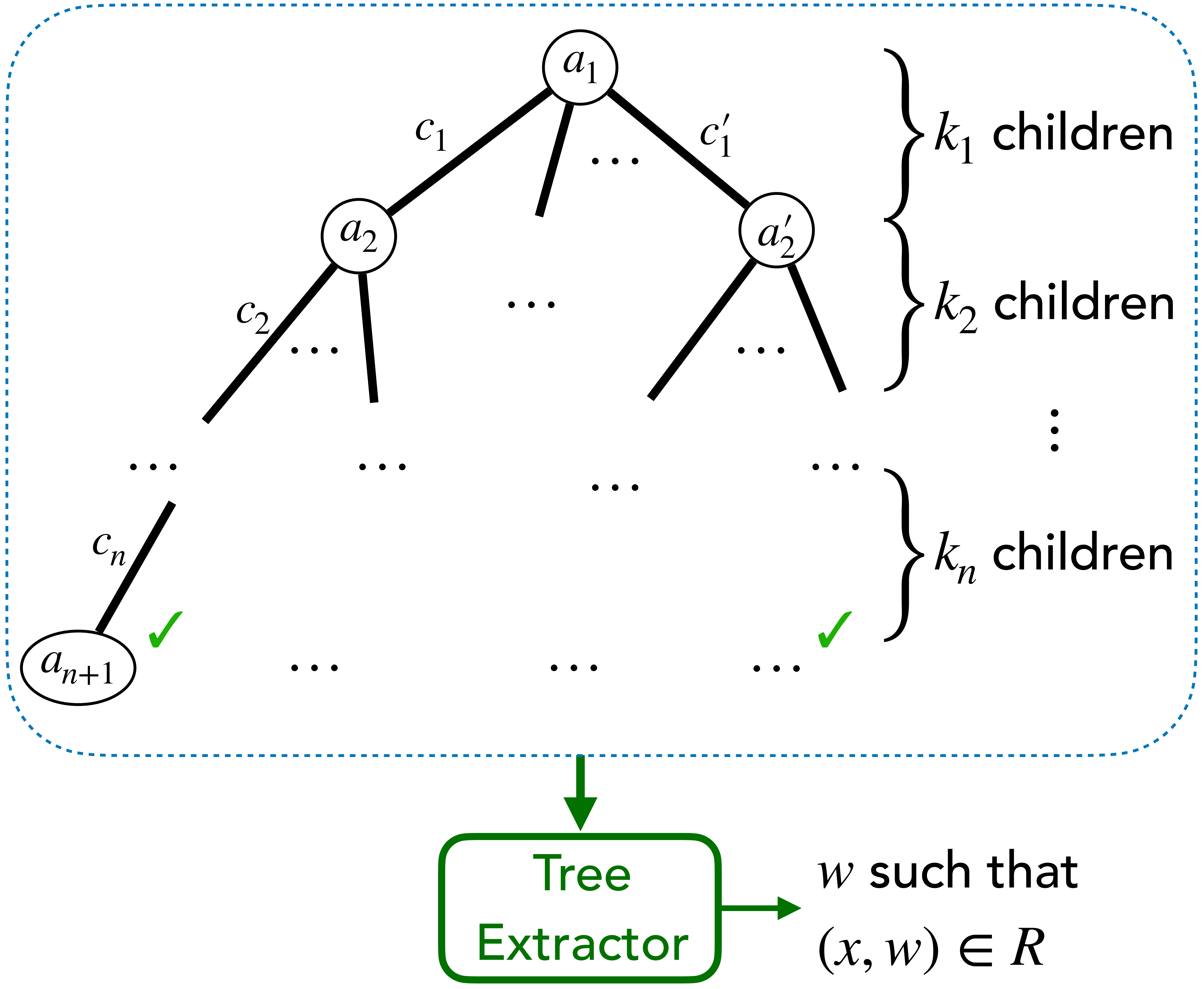
1. SIM-EXT = KS + k-ZK + k-UR (for same k)
2. k-ZK and k-UR for Bulletproofs & Spartan
3. Knowledge Soundness via Generalized Tree Builder

Knowledge Soundness from Special Soundness

F-S Argument:



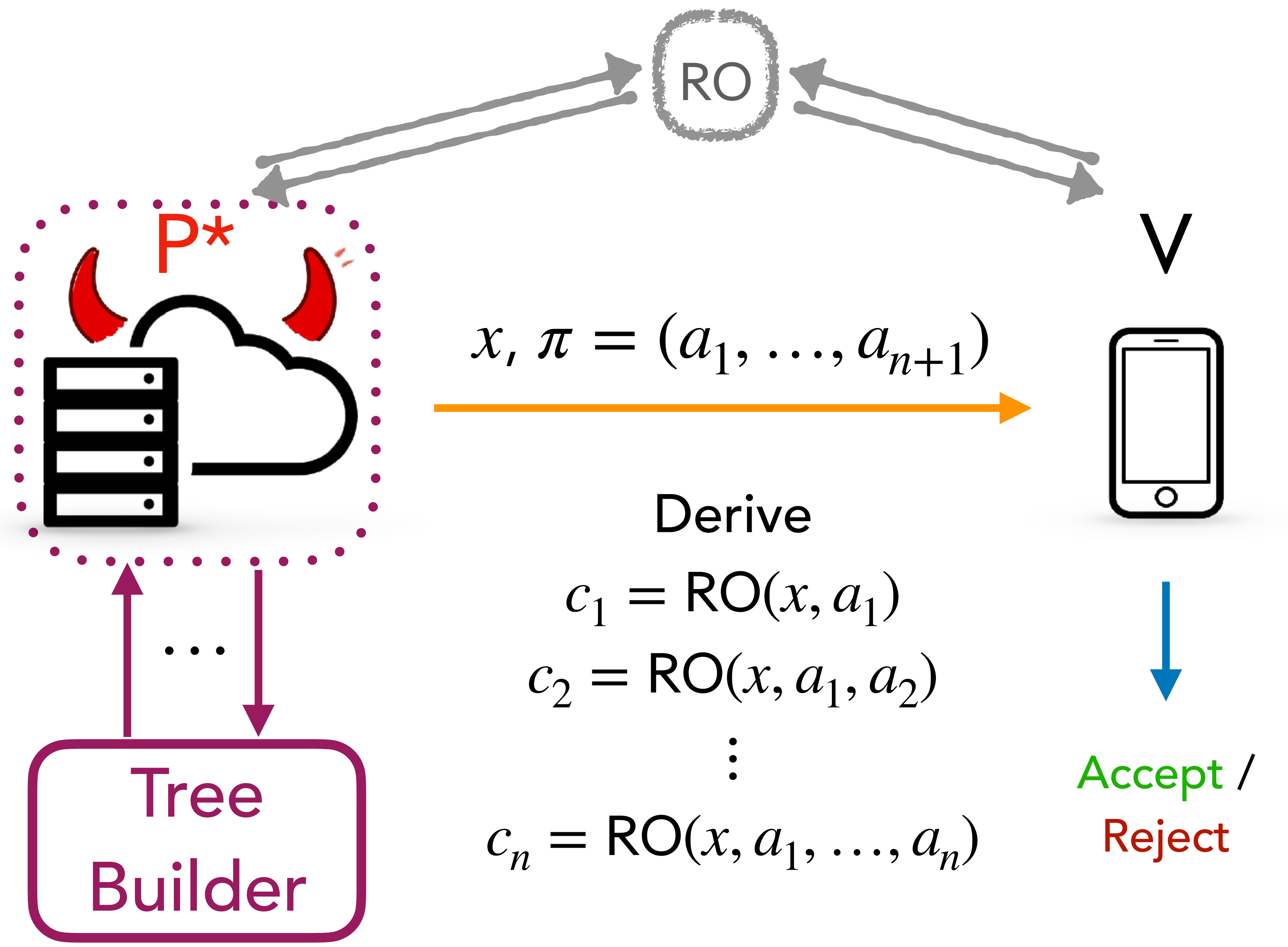
$(k_1, \ldots, k_n)$-Tree of Accepting Transcripts



Special Soundness: There exists $k_1, \ldots, k_n$ such that a witness w can be extracted from any $(k_1, \ldots, k_n)$-tree of accepting transcripts.

Knowledge Soundness from Special Soundness

F-S Argument:



Special Soundness: There exists $k_1, \ldots, k_n$ such that a witness w can be extracted from any $(k_1, \ldots, k_n)$-tree of accepting transcripts.

Attema et al. (TCC '22): There exists a *tree-builder* AFK-TB that builds a $(k_1, \ldots, k_n)$-tree of accepting transcripts in *expected* time $O(Q \cdot K \cdot t(\mathsf{P}^*))$.

$$\left(Q = \#\ \text{RO queries},\ K = \prod_{i=1}^{n} k_i\right)$$

Combine TB with TE

Corollary: Special soundness implies knowledge soundness.

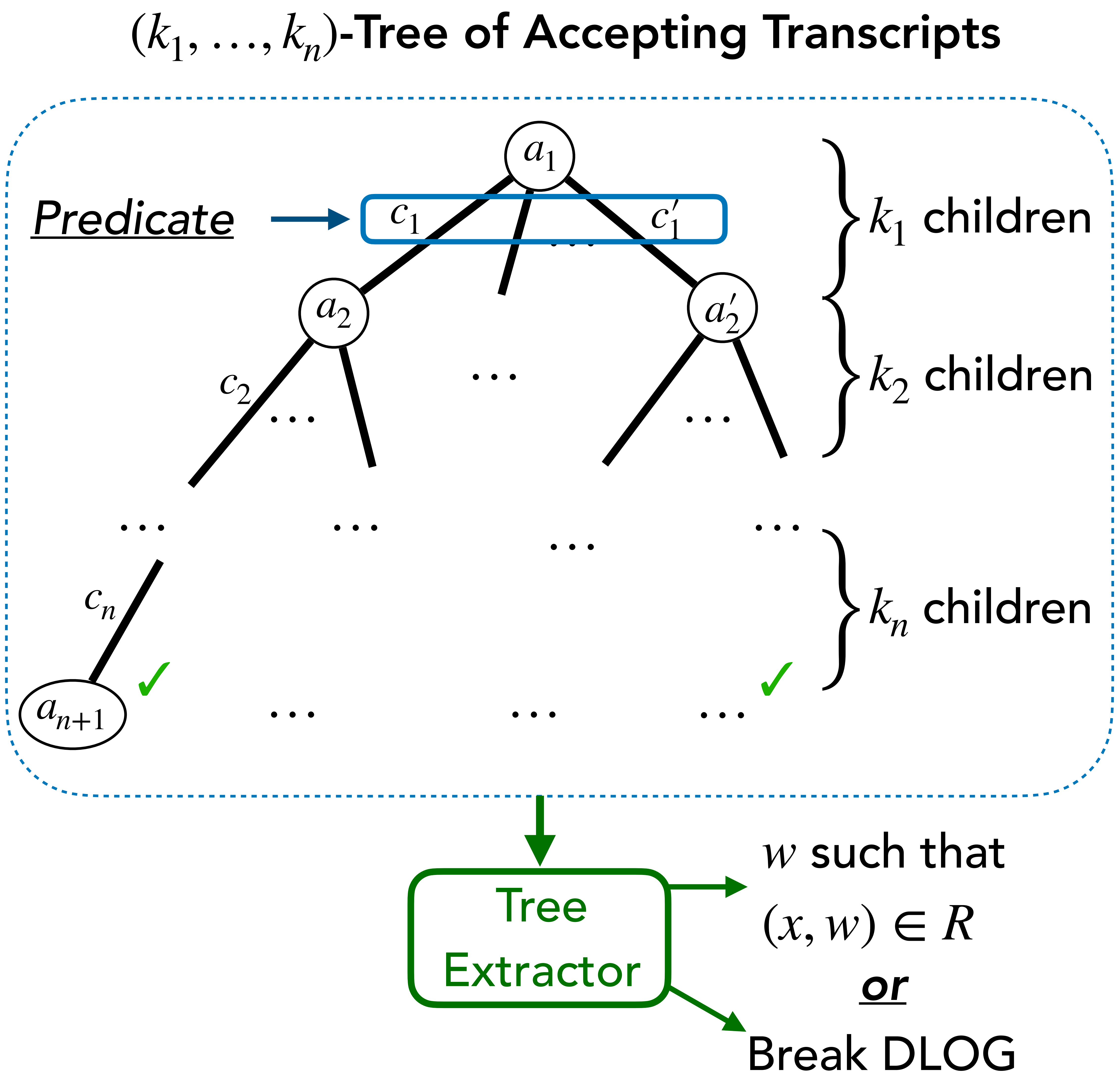
Generalized Special Soundness & Tree Building

Observation: Spartan and Bulletproofs do *not* satisfy special soundness.

However, they satisfy a *generalized* notion:

- Tree extraction can either output a witness or a break of some *computational* assumption (DLOG).
- The tree of transcripts needs to satisfy *extra* predicates on the challenges at certain levels.

$\implies$ we construct a *generalized tree builder* that can handle ***partition predicates***



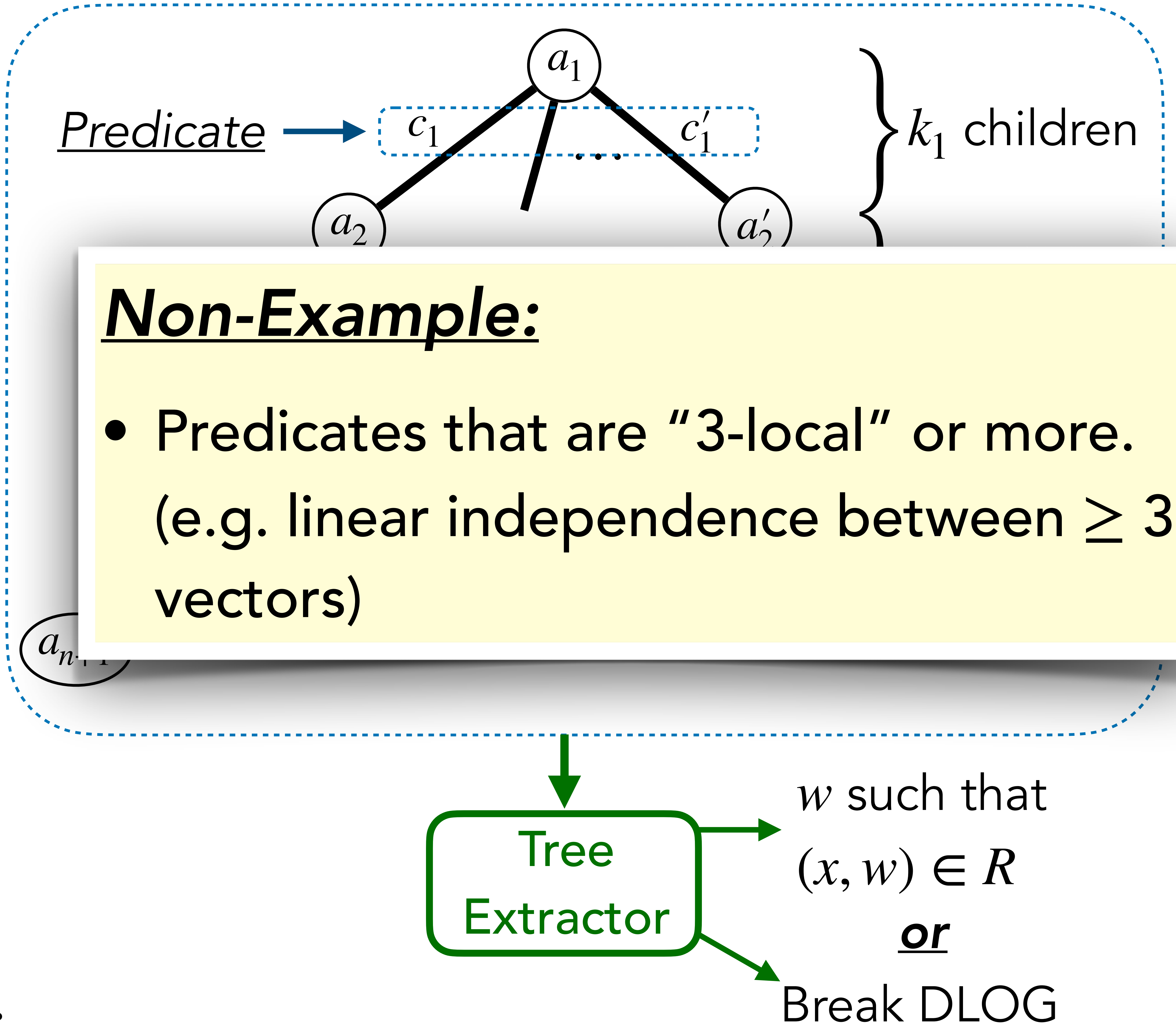
Generalized Tree Builder for Partition Predicates

Partition Predicate: Let $Ch_i = Ch_{i,1} \sqcup \ldots \sqcup Ch_{i,p}$. Then the k_i challenges $c_{i,1}, \ldots, c_{i,k_i}$ from any node a_i must belong to different partitions.

- Bulletproofs: $c_{i,j} \neq \pm c_{i,j'}$ for all $j \neq j' \in [1,4]$ (partitions are $\{x, -x\}$)
- Spartan: $(c_1, c_2) \neq \lambda \cdot (c_1', c_2')$ for all $\lambda \neq 0$ (partitions are lines $\{\lambda \cdot x \mid \lambda \neq 0\}$)

AFK-TB handles *distinctness* predicate ($c_{i,j} \neq c_{i,j'}$ for all $j \neq j'$, or partitions are singletons).

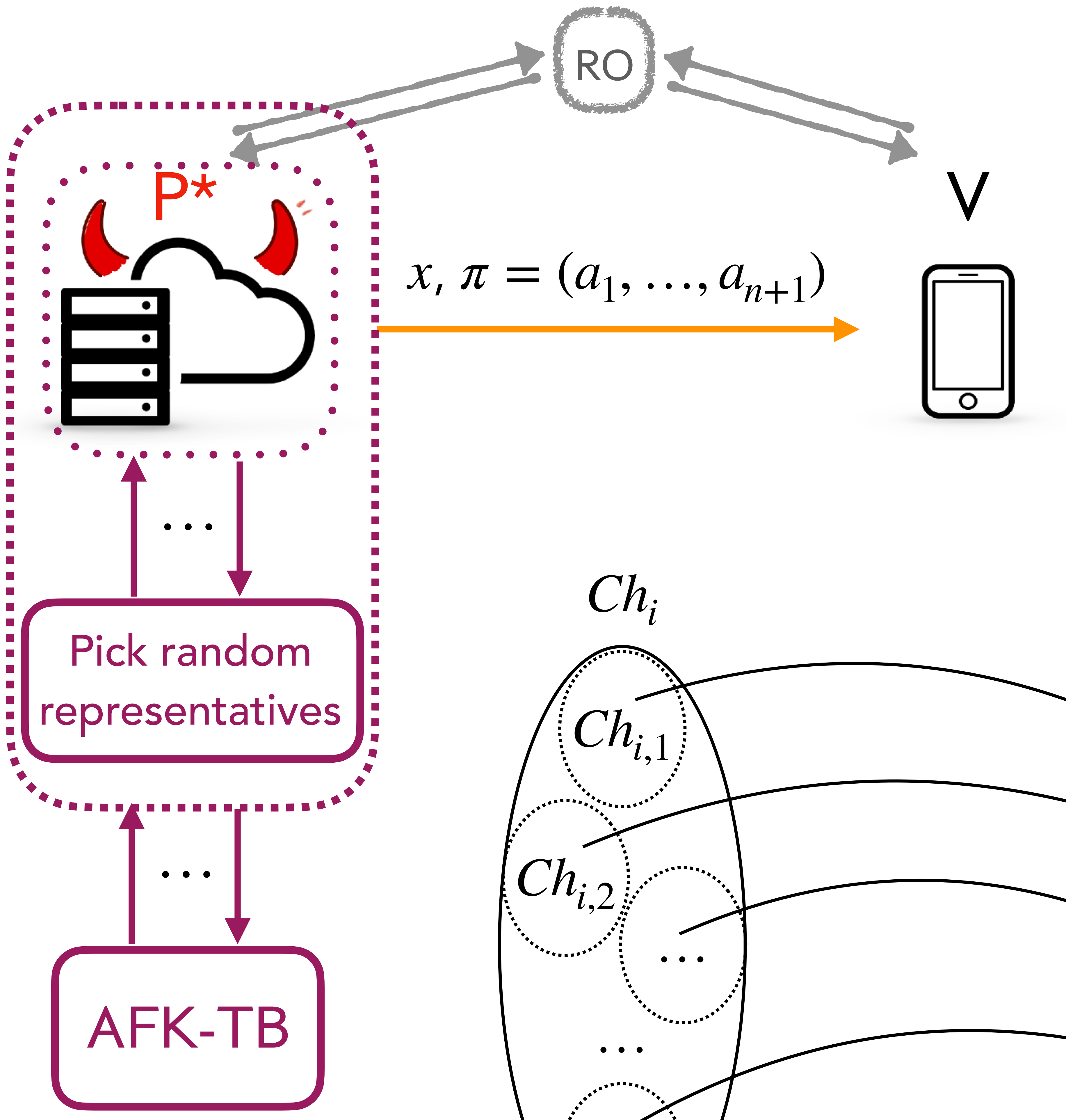
$(k_1, \ldots, k_n)$-Tree of Accepting Transcripts



Non-Example:

- Predicates that are "3-local" or more. (e.g. linear independence between ≥ 3 vectors)

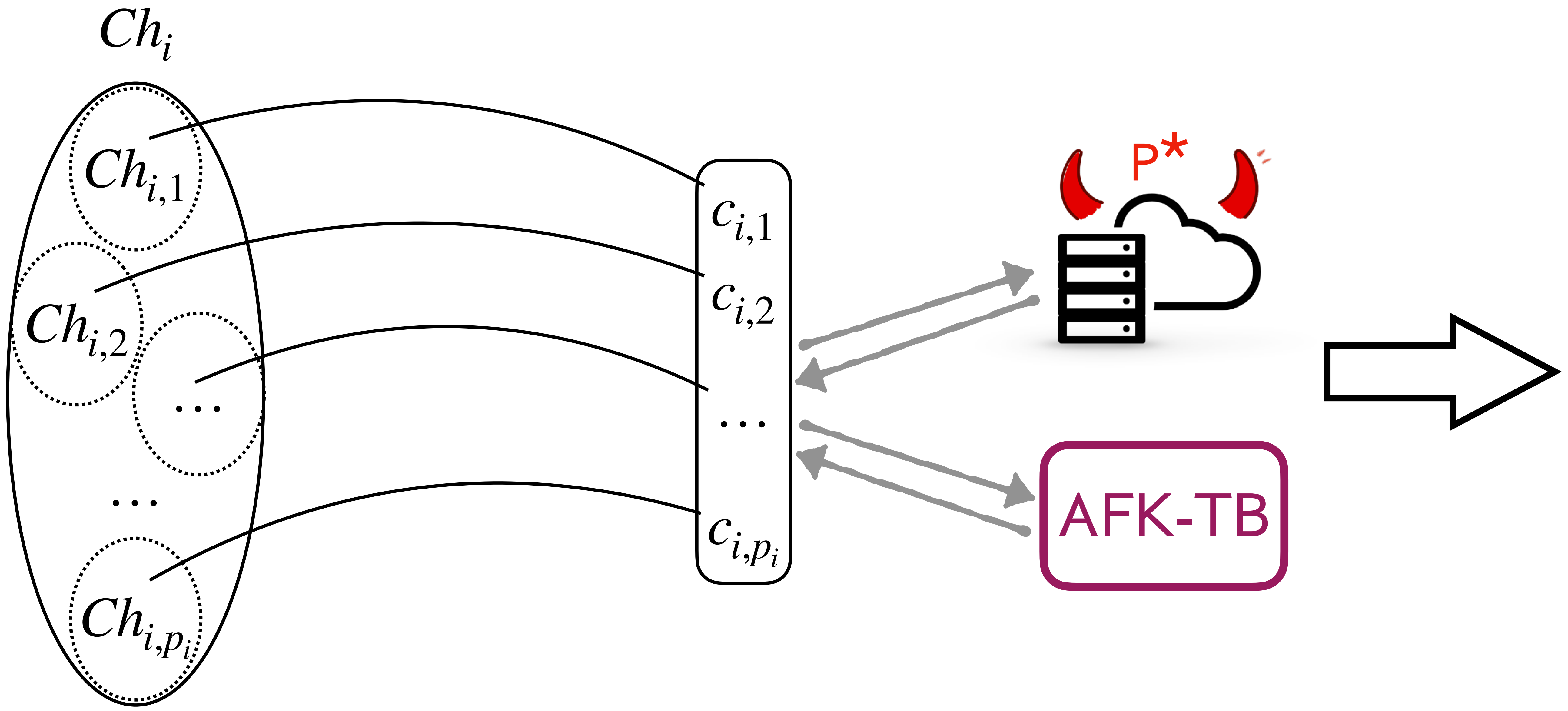
Tree Builder for Partition Predicates - Construction



Idea:

1. Provide a wrapper that ***restricts*** the challenge space, by picking a *random* representative for each partition $Ch_i = Ch_{i,1} \sqcup \ldots \sqcup Ch_{i,p_i}$.

2. Invoke AFK-TB on the restricted challenge space.

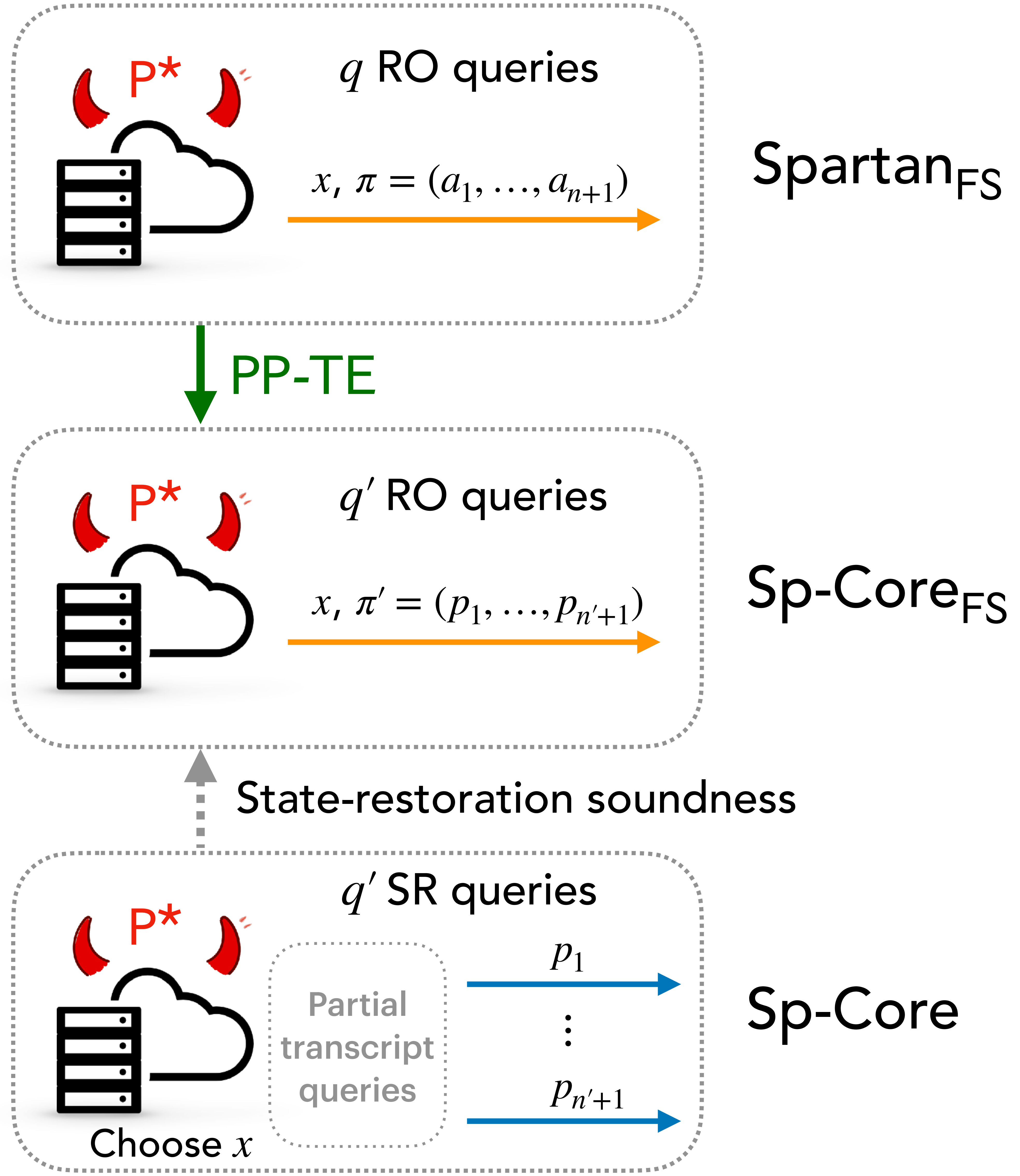


Since AFK-TB guarantees ***distinctness***, the resulting challenges belong to ***different*** partitions.

Knowledge Soundness - Proof for Spartan

Proof Strategy:

1. Use partition-predicate tree builder to extract underlying polynomials from Spartan$_{FS}$. (one such polynomial is witness w)
2. Conditioned on success (no DLOG break), get P* for Sp-Core$_{FS}$.
3. Define ***bad*** = (x, π') accepted in Sp-Core$_{FS}$, yet w not a valid witness.
4. Bound Pr[***bad***] by the ***state-restoration soundness*** of Sp-Core.



Summary

We show that Bulletproofs and Spartan satisfies SIM-EXT, a **strong** security notion for zkSNARKs that rules out most attacks in practice.

Limitation: bounds for knowledge soundness are *non-tight* due to rewinding

Open Questions:

- SIM-EXT for other classes of protocols:
  - Lattice-based / Hash-based
  - Post-quantum analysis in the QROM
  - Recursive SNARKs
- Tighter rewinding bounds
- UC security

	Lemma 6.3 ($m=1$)	[36, **Theorem 4**]
Asymptotic	$O\left(\frac{Q^2+Qn}{\lvert\mathbb{F}\rvert}\right) + \mathbf{Adv}^{\mathsf{DL\text{-}REL}}_{\mathbb{G},2n+3}(\mathcal{A})$ where $\mathbb{E}[t(\mathcal{A})] = O(Q \cdot n^3 \cdot t(\mathcal{P}^*))$	$O\left(\frac{Qn}{\lvert\mathbb{F}\rvert}\right) + \mathbf{Adv}^{\mathsf{DL\text{-}REL}}_{\mathbb{G},2n+3}(\mathcal{A}')$ where $t(\mathcal{A}') = O(Q \cdot n)$
Concrete	$\approx$ 22 bits of security	$\approx$ 164 bits of security

Concrete: $\lvert\mathbb{F}\rvert \approx 2^{256}$, $n = 64$, $t(\mathscr{P}^*) = 2^{48}$, $Q = 2^{40}$.

Problem: $\mathbf{Adv}^{\mathsf{DL}}_{\mathbb{G}}(A) \leq \sqrt{t(A)^2/\lvert\mathbb{F}\rvert}$ for *expected* time A.

Thank You!