A Parallel Repetition Theorem for **Any** Interactive Argument Or *On the Benefits of Cutting Your Argument Short*

Iftach Haitner

Microsoft Research New England

outline

- Motivating examples for the question:
  Does parallel repetition improve security?
- Our result
- Proof's sketch

Example #1 – CAPTCHAS

CAPTCHAS – Aim to distinguish human beings from a machines.
Used to fight spamming, denial of service,…

Basic task – 

$\stackrel{?}{=}$

Not hard enough (easy to guess with probability 1/36)

$\stackrel{?}{=}$

$\stackrel{?}{=}$

$\stackrel{?}{=}$

Amplification via “sequential repetition”
Improves security (to any degree)
Problem: impractical, too much time

$\stackrel{?}{=}$ – Amplification via “Parallel repetition”

By how much (if at all) does parallel repetition improve security?

Example #2 – Commitment Schemes

Commit stage

Reveal stage

Example #2 - Commitment Schemes

Reveal stage

S m

R

Security properties:

Hiding: **R** learns <u>nothing</u> about **m** during commit stage

Binding: **S** cannot decommit to two <u>different</u> values

- More "powerful" than encryption
  - Can have statistical hiding
- Extremely useful

Weakly binding: **S** cannot decommit to two different values with "too high" probability

Amplification idea: **S** commits to the same value many times (in parallel)

By how much (if at all) binding is improved?

Goal – Hardness Amplification

Starting point – A protocol/algorithm with "weak security" – security holds with some probability

Goal – Amplify to fully secure protocol/algorithm

Examples: one-way functions, PCP's, CAPTCHAS, identification schemes, interactive arguments, …

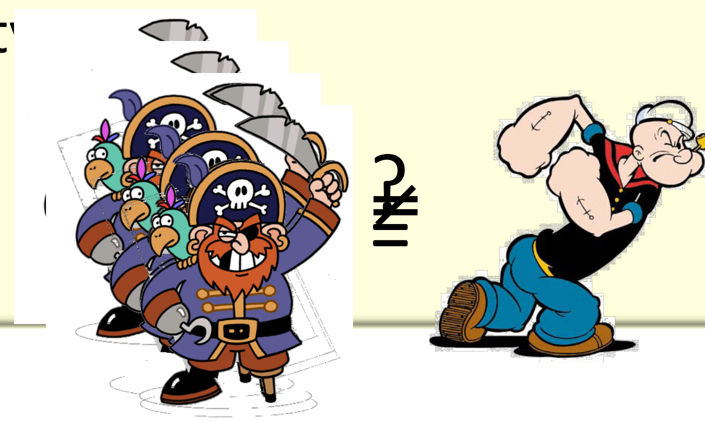
Real challenge – preserve other properties, in particular <u>efficiency</u>

Most natural approach is via parallel repetition

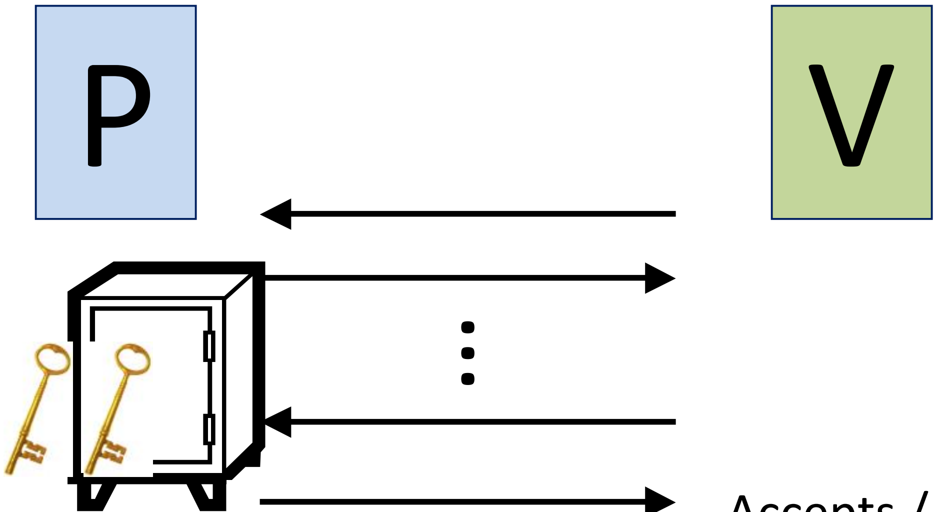
Does parallel repetition improve securit

Answer: (in general) No

Our result: Effectively, Yes



Interactive Arguments



Accepts / Rejects

The soundness of (P,V) equals the binding of (S,R)

V accepts if the keys open the safe into two different values

Soundness: For any P*
Pr[V accepts in (P*,V)] is negligible

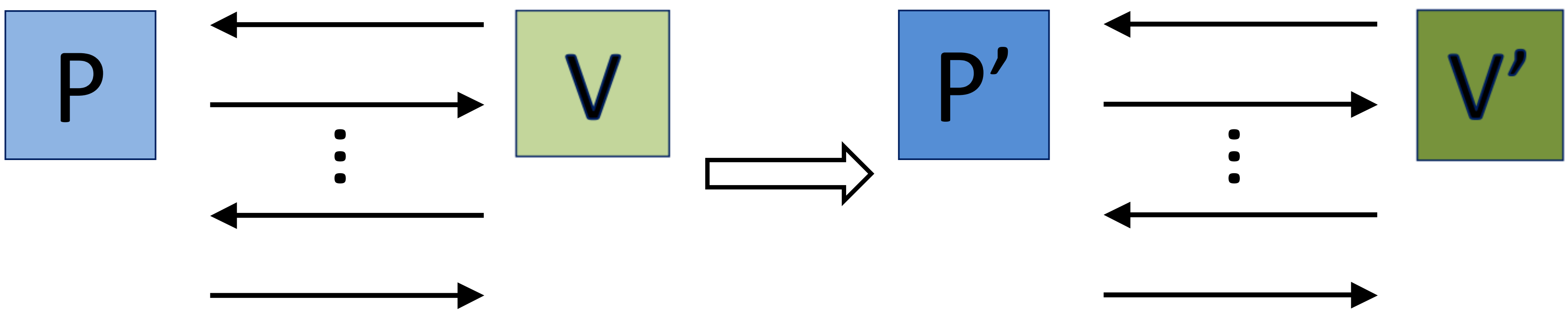
Soundness: For any efficient P*

Weak soundness: For any efficient P*
Pr[V accepts in (P*,V)] $\leq \varepsilon$ (e.g., ½)

Soundness error

- Typically, (P,V) has additional functionality and other useful properties
- Realizes the security of significant types of systems

Amplification of Interactive Arguments



For any efficient P^*

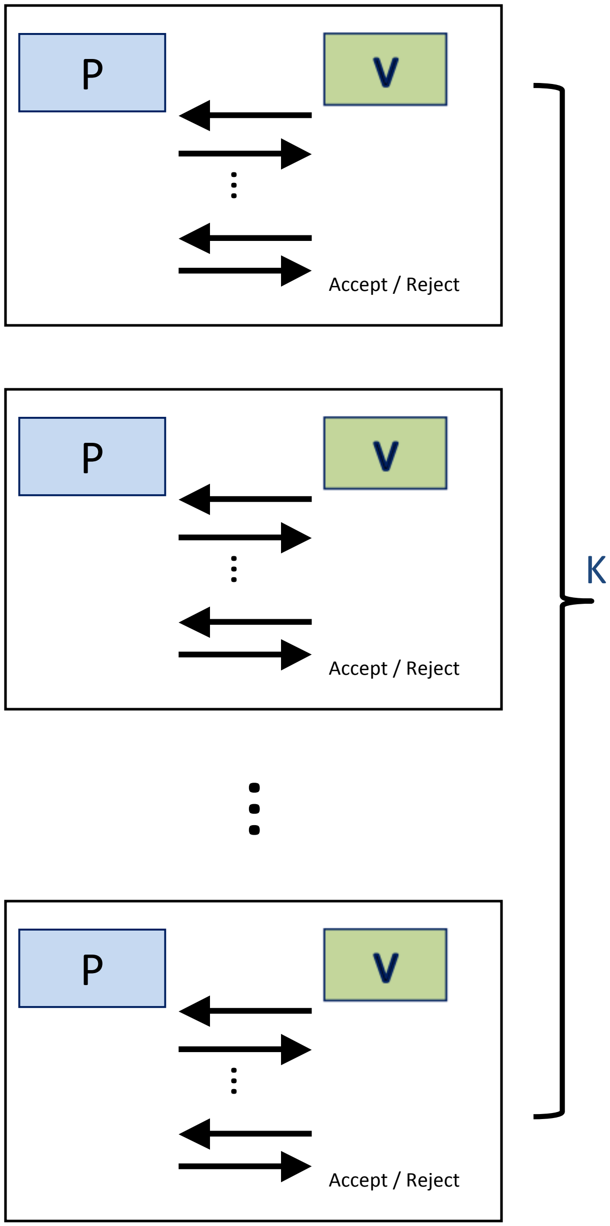
$\Pr[V \text{ accepts in } (P^*,V)] < \varepsilon$

For any efficient P^*

$\Pr[V' \text{ accepts in } (P^*,V')]$ is negligible

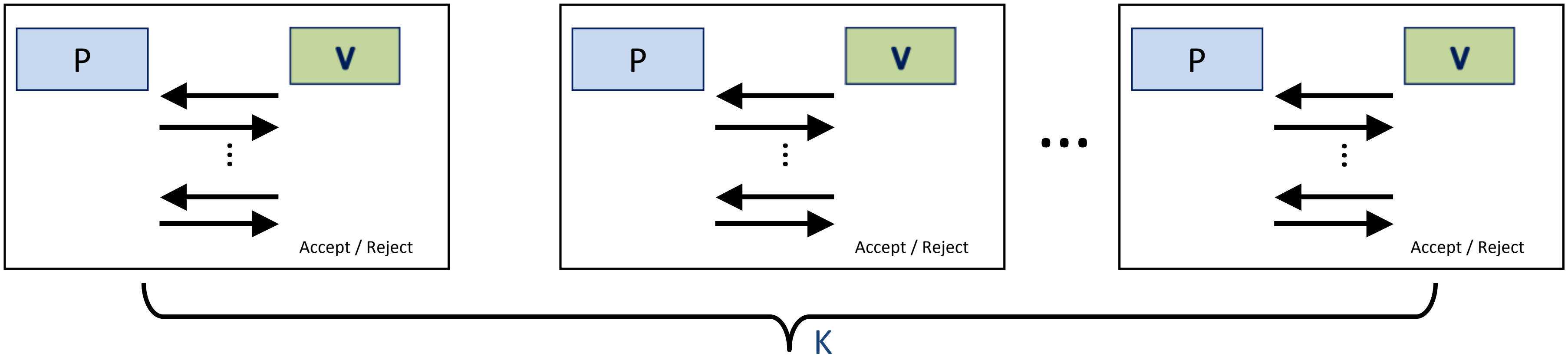
Goal – a generic transformation that preserves other properties of (P,V) (in particular efficiency), and can be applied to any protocol.

Sequential Repetition



- No overlap between executions
- Verifier accepts if all sub-verifiers do
- Known to reduce the soundness error (to any degree, i.e., ε^k)
  - Since repetitions are independent
- Preserves most properties of the original protocol
- Blows up round complexity
  (# of communication rounds)

Parallel Repetition



- Interactions are done in parallel
- Verifier accepts if all sub-verifiers do
- Preserves round complexity.

Does it improve security?

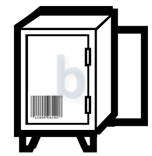
Does not work in general!

The Counterexample of [Bellare et al. '97]

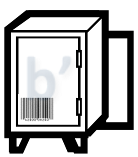
P

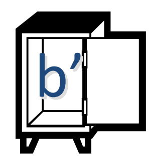
V

b ← {0,1}



b'





V accepts if b' = b,
and the safes are <u>different</u>

- Safes are realized as commitment schemes
- Soundness error ½

T ess

Both verifiers accept if $b_1 = b_2 \Rightarrow$ soundness error ½

Can be extended to any (# of repetitions) k

[Pietrzak-Wikstrom '07] There exists a single protocol whose soundness error remains ½ for any (poly) k

Can we improve security efficiently?

Parallel repetition does improve soundness in few special cases:

- 3-message protocols [Bellare-Impagliazzo-Naor '97]
- Public-coin protocols (i.e., verifier sends random coins as its messages) [Håstad-Pass-Pietrzak-Wikström '08] and [Chung-Liu '09]

❖ Also in Interactive proofs [Goldreich '97]
and two-prover Interactive proofs [Raz '95]

The above does not apply to many interesting cases

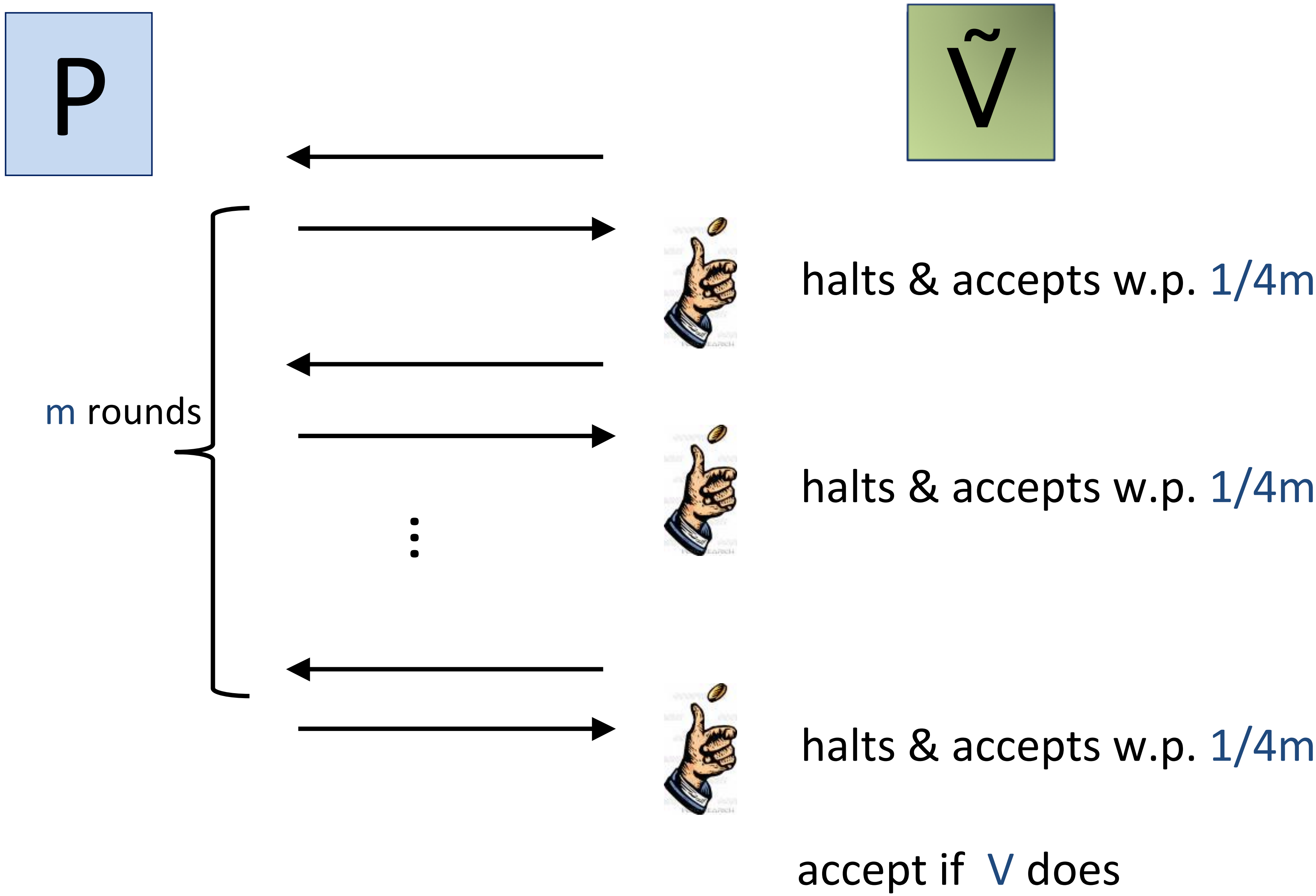
Can we efficiently improve the security of general interactive arguments?

Our Result [H '09]

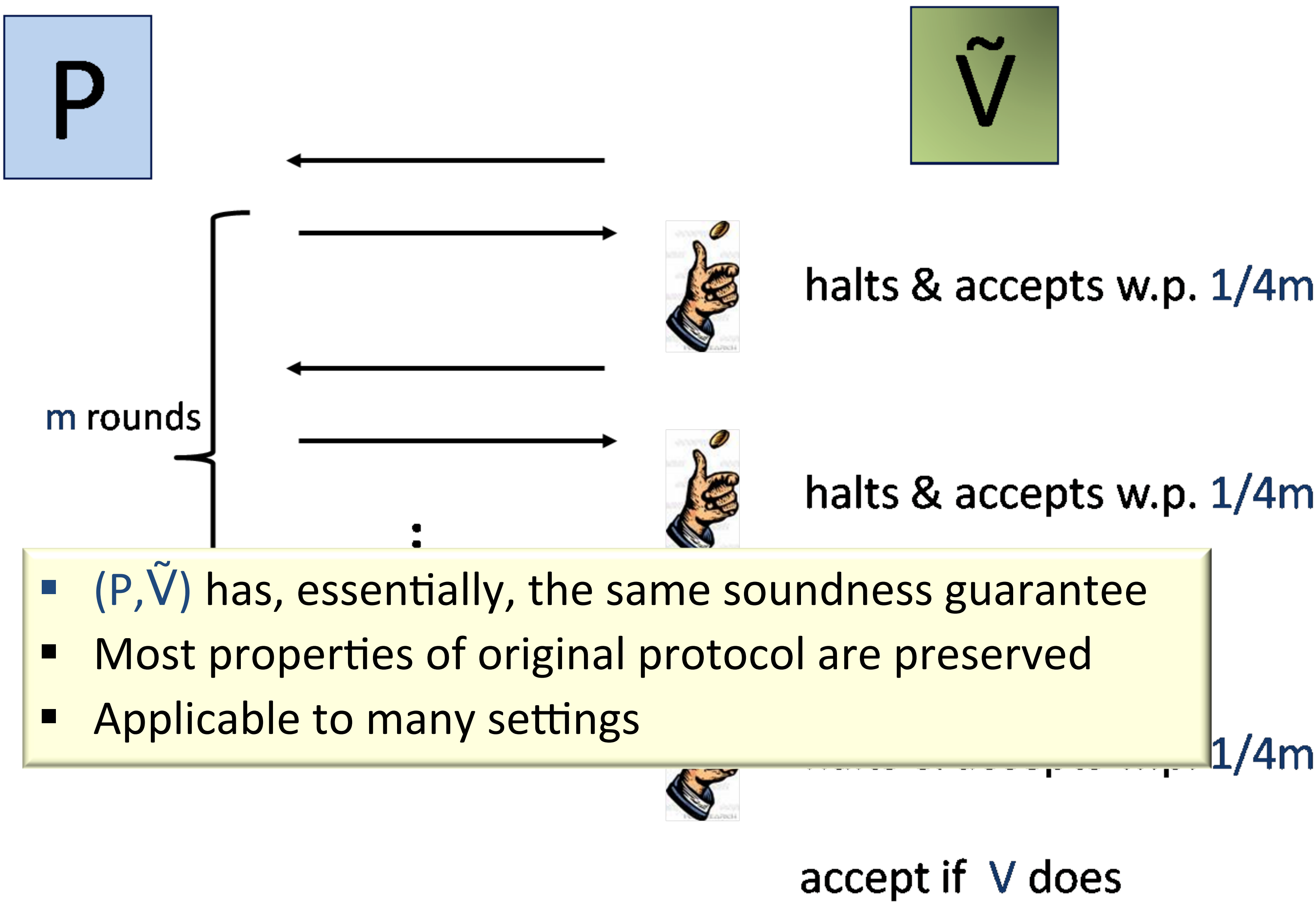
A simple modification of the verifier of any interactive argument, yields a protocol whose security is improved (to any degree) by parallel repetition

In fact, we are going to "cripple" the original protocol, in a way that, paradoxically, enables repetition to improve security

The Random Terminating Verifier



The Random Terminating Verifier



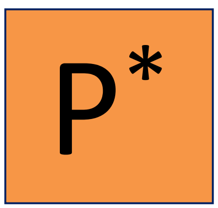
- (P,Ṽ) has, essentially, the same soundness guarantee
- Most properties of original protocol are preserved
- Applicable to many settings

Why Does Random-Termination Help?

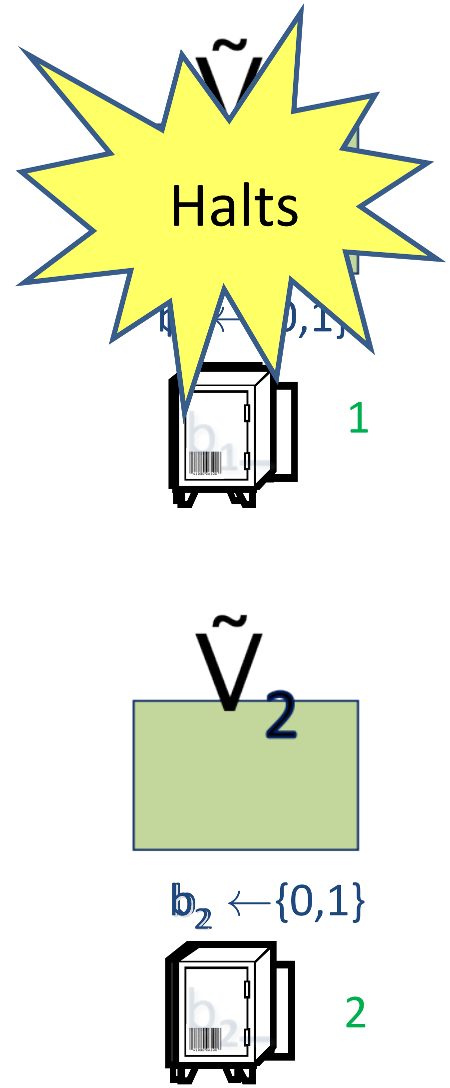
The transformation makes the verifier less predictable
Prevents cheating prover from using one verifier against the other

Beats the Counterexample

$\Pr[\tilde{V} \text{ accepts in } (P^*, \tilde{V})] = 9/32 < ½$



?



Skip

Proof's Overview

Assume for any efficient P^*

(1) $\Pr[V \text{ accepts in } (P^*,V)] < \varepsilon$

Prove for any efficient $P^{(k)^*}$

(2) $\Pr[V \text{ accepts in } (P^*,V)] < \varepsilon^{(k)} \backsimeq \varepsilon^k$

Proof by **reduction** –

Assuming $P^{(k)^*}$ contradicts (2)

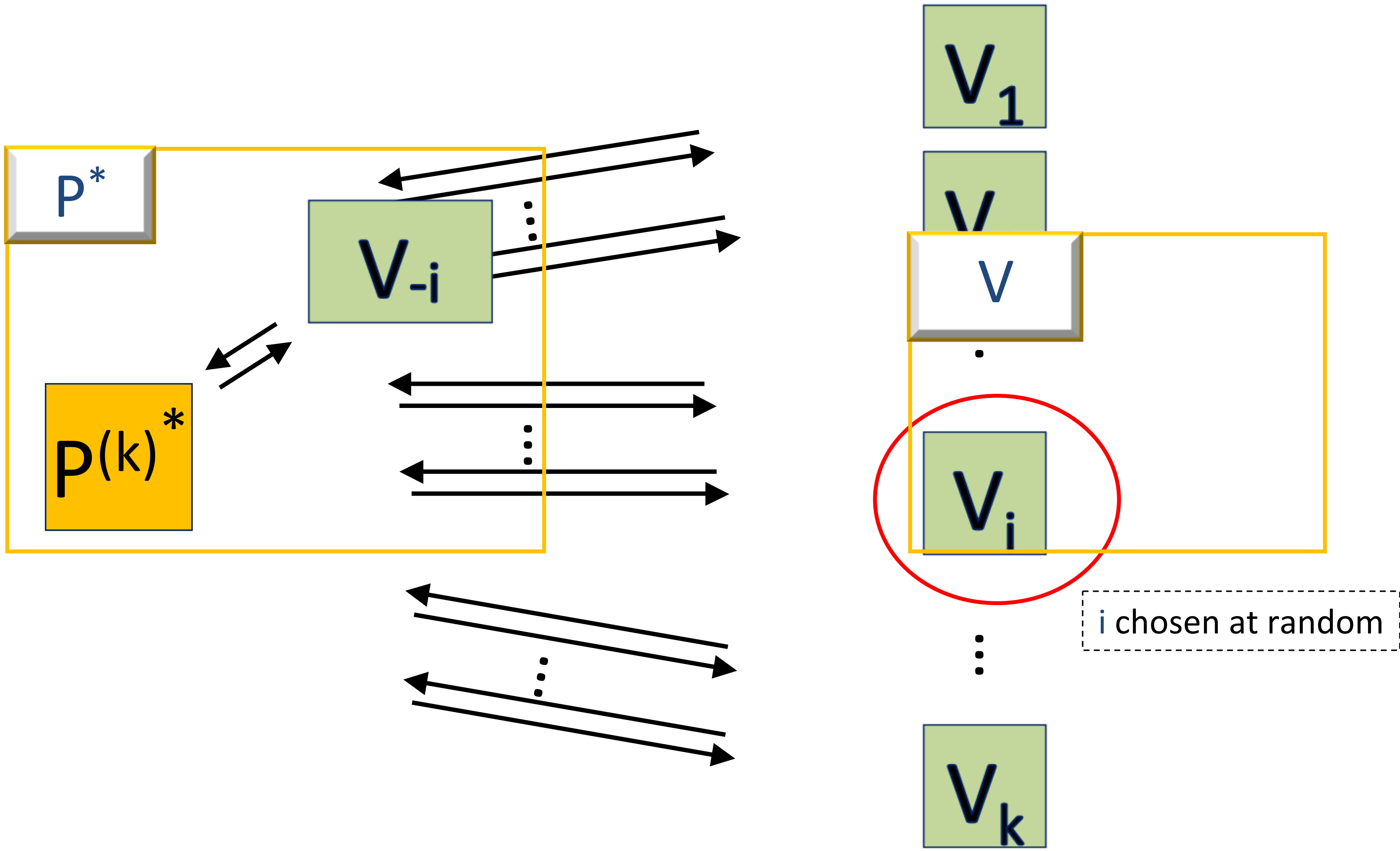
build P^* that contradicts (1)

ε is much larger than ε^k, thus an averaging argument would not be enough

The proof "almost" works for any interactive argument

V accepts in (P^*,V) $\leftrightarrow$ P^* "wins"

Defining P^*



If succeeded, do the same for the second round

Does such $\mathbf{q}_{-i}$ always exist?

W.h.p, over $\mathbf{q}_i$, a noticeable fraction of the $\mathbf{q}_{-i}$ are "good"

Proposition (follows [Raz '95] or [Talagrand '96]):

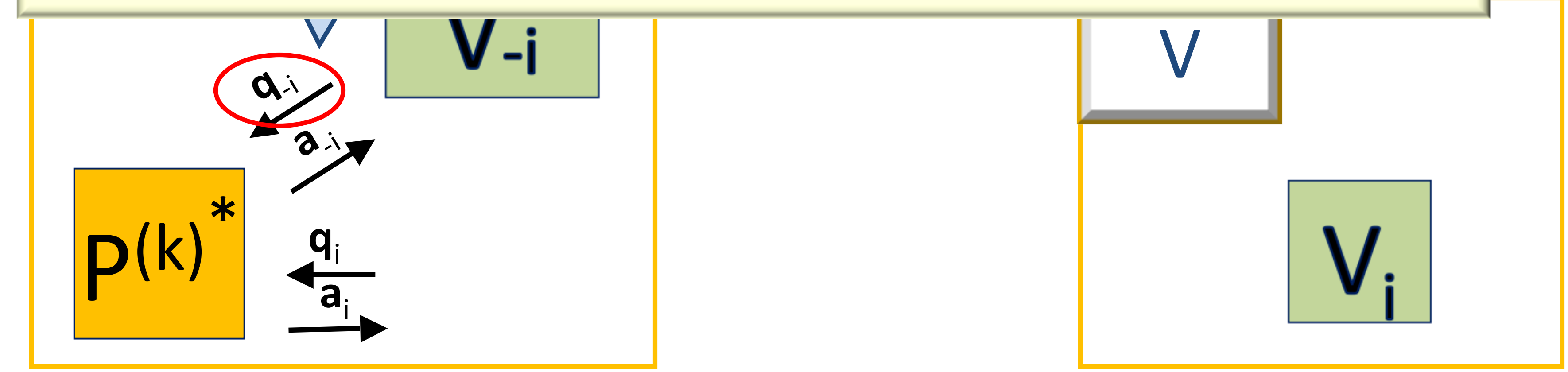
Let W be an event over $\mathbf{X} = (X_1,\ldots,X_k)$, then (for large enough k)

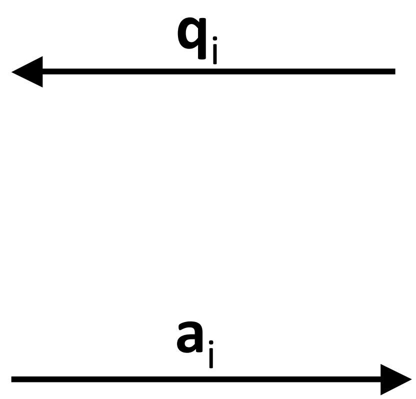
$\Pr[W \mid X_i = x] \approx \Pr[W]$, w.h.p. over $i \leftarrow [k]$ and $x \leftarrow X_i$

How to find $\mathbf{q}_{-i}$?

Sample (at random) many candidates, and for each estimate

$\alpha = \Pr[P^{(k)*} \text{ wins} \mid \mathbf{q}_i, \mathbf{q}_{-i}]$

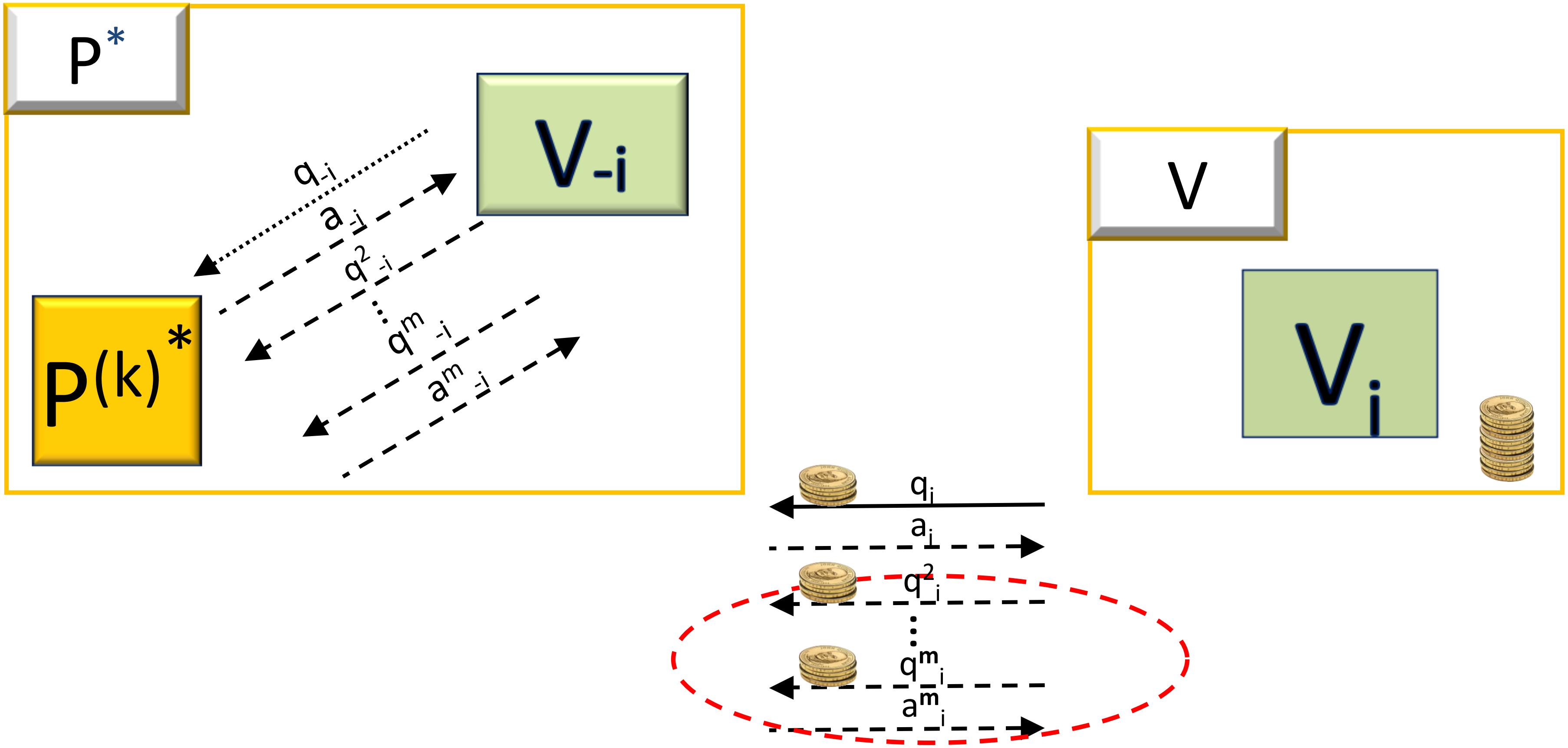




Given $\mathbf{q}_i$, find $\mathbf{q}_{-i}$ such that

$\Pr[P^{(k)*} \text{ wins}] \geq (1 - 1/2m) \cdot \boldsymbol{\varepsilon}^{(k)}$

Estimating α

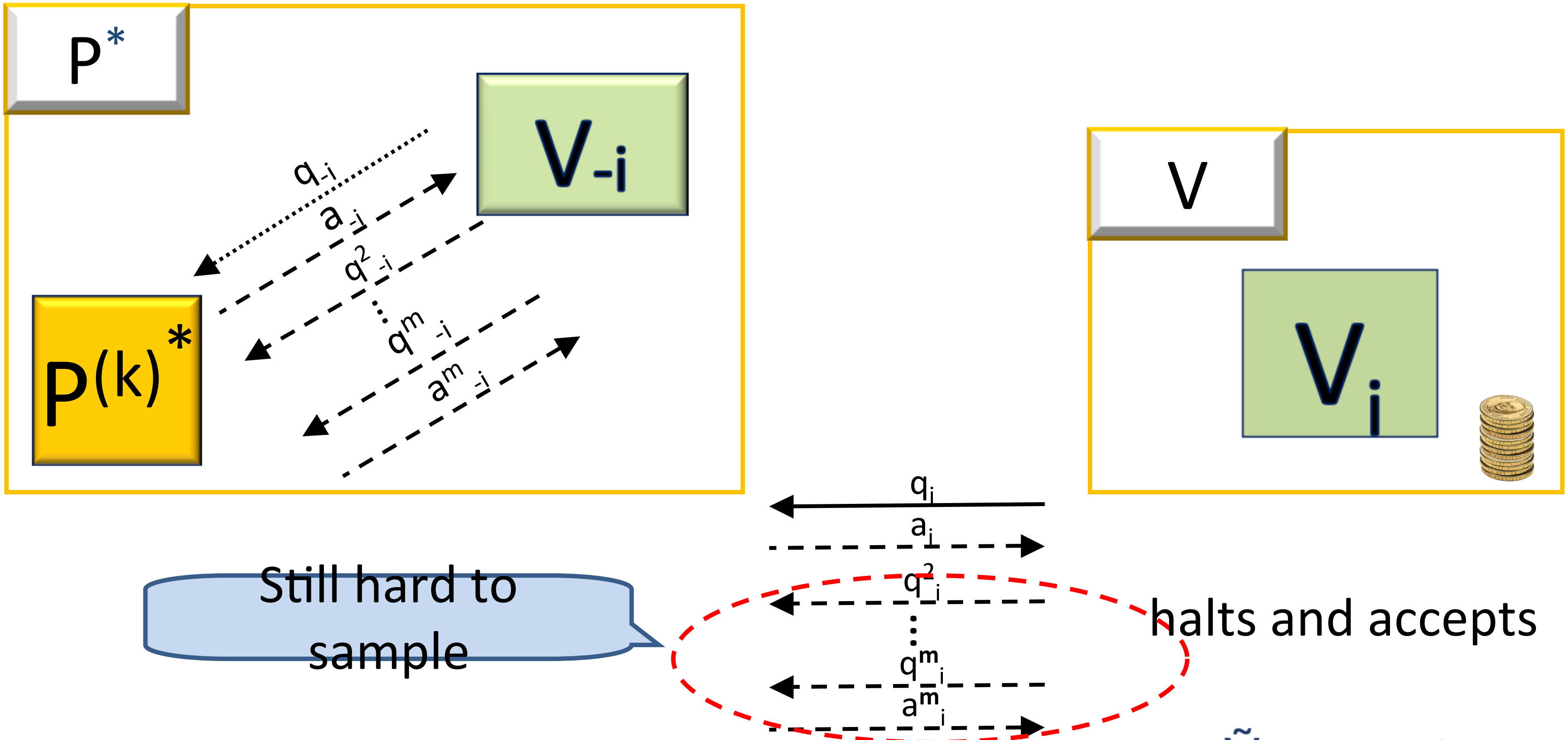


Estimate α (= $\Pr[P^{(k)*} \text{ wins} \mid \mathbf{q}_i, \mathbf{q}_{-i}]$) as the fraction of successful, random, continuations (i.e., $P^{(k)*}$ wins – all sub-verifiers accept)

If V is public coin, sampling random continuations is easy

Sampling might be infeasible for arbitrary V – As hard as finding a random preimage of an arbitrary (efficient) function. This is why parallel repetition fails

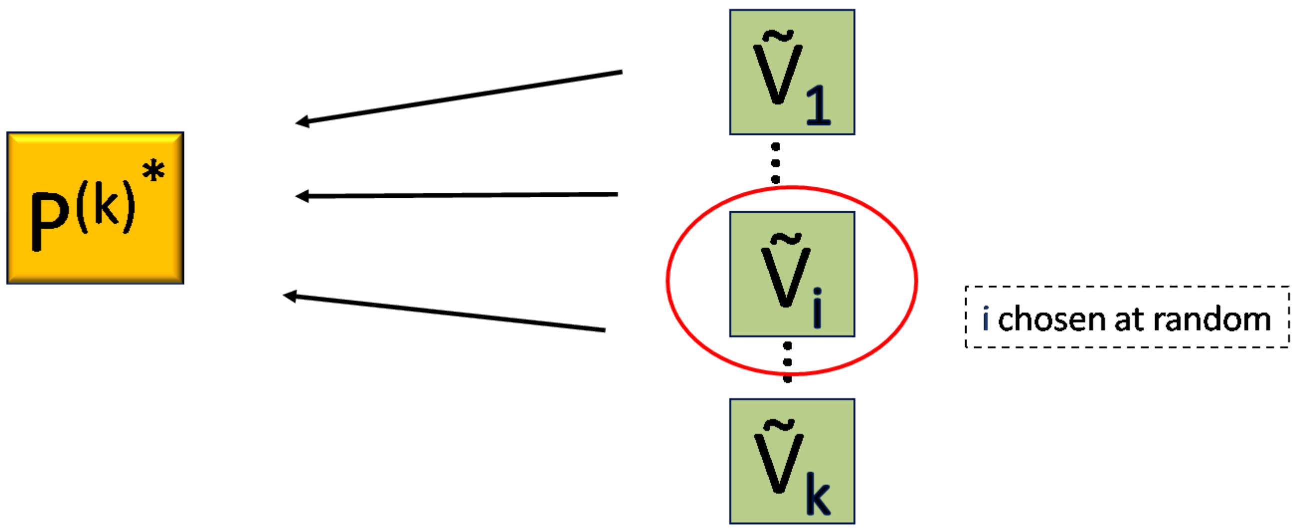
The Random Terminating Case



We sample random continuations, conditioned that $\tilde{V}_i$ halts after
f
i
r
s
t

α' Approximates α Well

$\alpha' = \Pr[P^{(k)^*}$ wins | $(\mathbf{q}_i, \mathbf{q}_{-i})$ & $\tilde{V}_i$ halts after first round]



Since many of the $\tilde{V}_j$'s are expected to halt after the first round
$\Rightarrow$ $\alpha' \simeq \alpha$ for a random i

Proposition: Let W be an event over **X**= $(X_1,\ldots,X_k)$, then
$\Pr[W \mid X_i = x] \simeq \Pr[W]$ w.h.p. over $i \leftarrow [k]$ and $x \leftarrow X_i$

Skip

More Details

Estimate $\alpha = \Pr[P^{(k)^*} \text{ wins} \mid \mathbf{q}_i, \mathbf{q}_{-i}]$

– sample random continuations of **all** verifiers

$\mathbf{r}_j$ - random coins of the j'th verifier drawn at first round

Easy

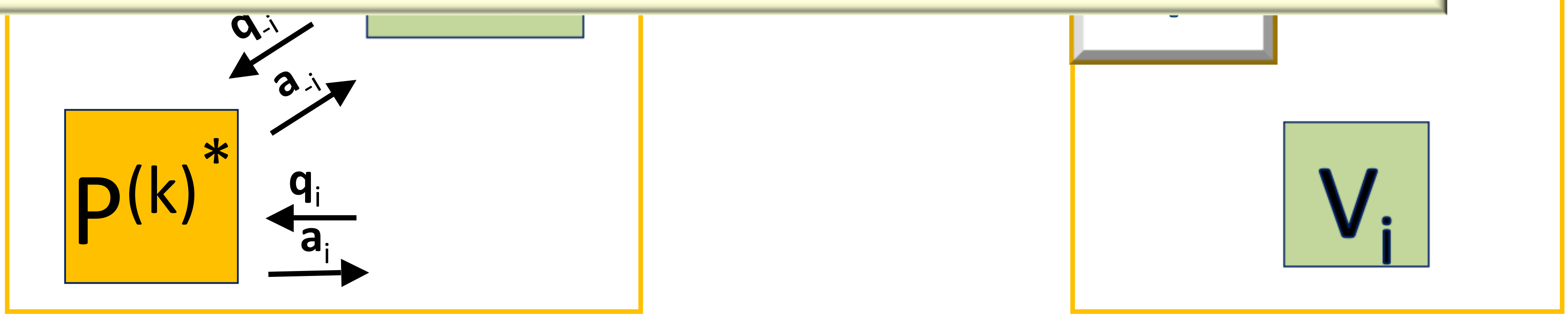
- For the emulated verifiers, as hard as finding a random second pre-image of a function!

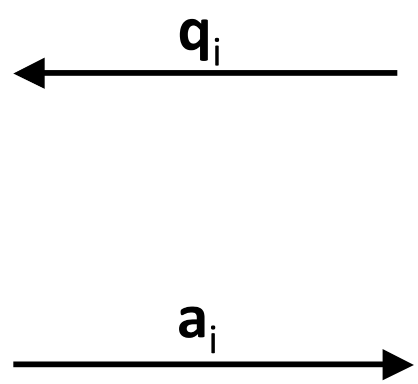
Sample random continuations is:

– feasible for (arbitrary) emulated verifiers

– impossible for (arbitrary) real verifier (even for unbounded sampler)

– feasible for the real random termination verifier





Find $\mathbf{q}_{-i}$ such that

$\Pr[P^{(k)^*} \text{ wins} \mid \mathbf{q}_i, \mathbf{q}_{-i}] \geq (1 - 1/2m)\varepsilon^{(k)}$

Defining P^*(revisited #2)

P^* picks the first $\mathbf{r}_{-i}$ s.t.
$\alpha'(r_{-i}) = \Pr[P^{(k)^*}$ wins$| (\mathbf{r}_i,\mathbf{r}_{-i})$ & V_i halts after first round$] > (1-1/2m)\epsilon^{(k)}$,
where $\alpha'(r_{-i})$ is estimation for $\alpha(r_{-i}) = \Pr[P^{(k)^*}$ wins$| \mathbf{r}_i,\mathbf{r}_{-i}]$

Problem: threshold sensitivity

Solution: follows "Smooth sampling" approach of Håstad et al.:

P^* Samples many $(\mathbf{r}_{-i},\mathbf{r}^2,\dots,\mathbf{r}^m)$ (all protocol's random coins), and chooses $\mathbf{r}_{-i}$ as the prefix of first successful execution (P^* wins).

- Proof w.r.t. α still goes through
- The probability that $\mathbf{r}_{-i}$ is picked, is proportional to $\alpha(\mathbf{r}_{-i})$
- Hence, proof still go through w.r.t. α'

❖ The original proof can be fixed, using soft thresholds

Summary

- Parallel repetition may not improve security
- Does improve security of a slight variant of any protocol
- Main reason, the modified verifier is unpredictable
- Useful for many settings

Main open question:

- Can this proof technique be applied to other settings