

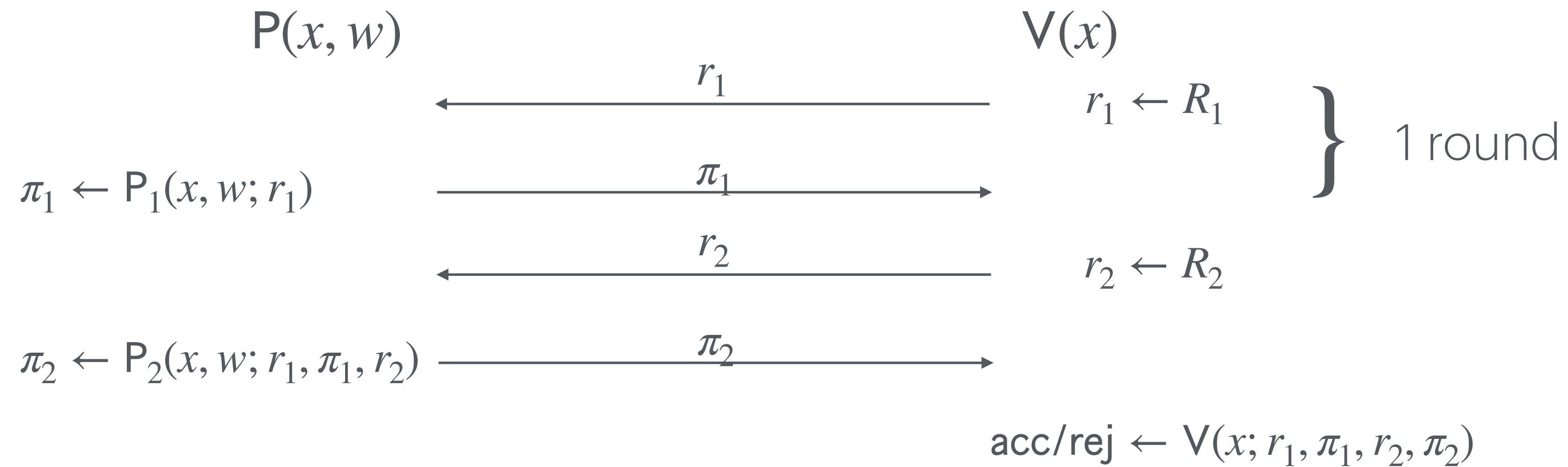
Weak Zero-Knowledge and One-Way Functions

Rohit Chatterjee, Yunqi Li and Prashant Nalini Vasudevan

National University of Singapore

Zero-Knowledge [GMR85]

$x \in \mathcal{L}?$



Zero-Knowledge [GMR85]

- Zero-Knowledge Proofs for **NP**

- Completeness

When $x \in \mathcal{L}$, $\Pr_{\langle P, V \rangle(x, w)} [V \text{ accepts}] \geq 1 - \epsilon_c$

- Soundness

When $x \notin \mathcal{L}$, $\Pr_{\langle P^*, V \rangle(x)} [V \text{ accepts}] \leq \epsilon_s$

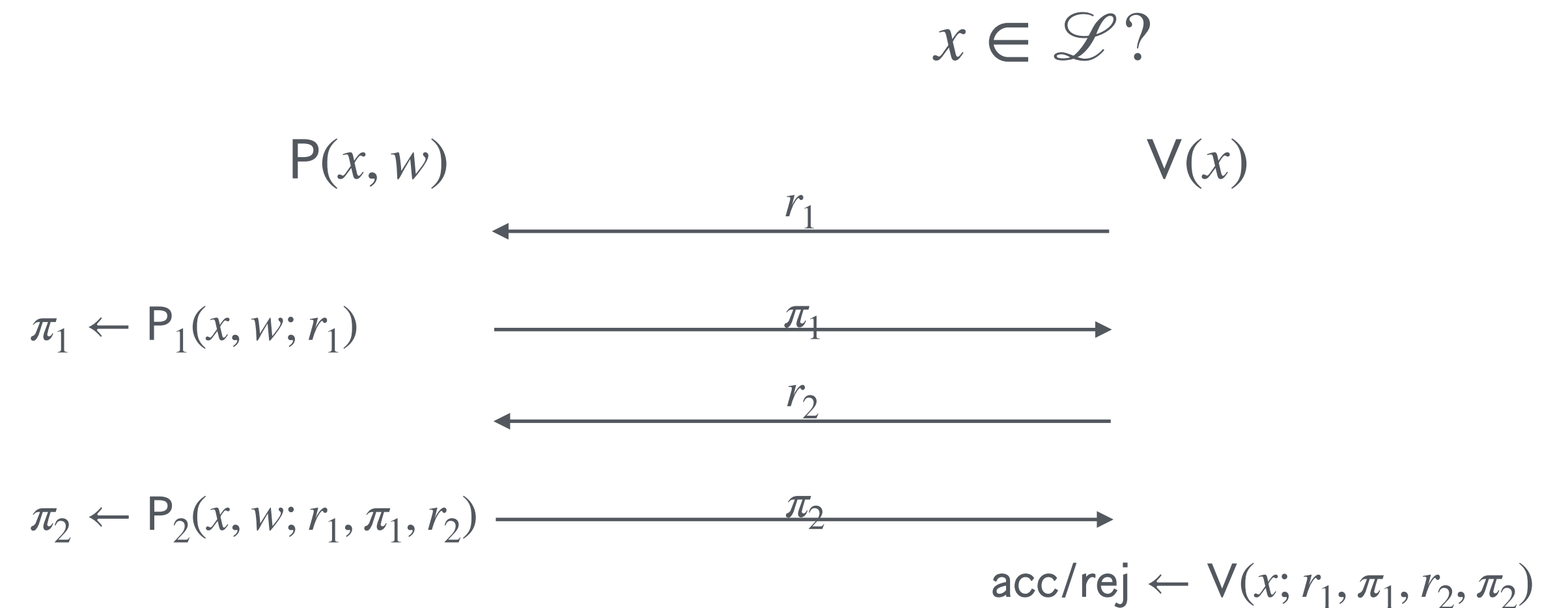
- Zero-knowledge: the verifier learns nothing beyond the validity of the statement

There is an efficient simulator, for all $x \in \mathcal{L}$, $\text{Sim}(x) \approx_{\epsilon_z} \langle P, V \rangle(x, w)$ ϵ_z -indistinguishable

- Standard ZK: $\epsilon_c, \epsilon_s, \epsilon_z = \text{negl}(n)$

$$|\Pr[\mathcal{D}(\text{Sim}(x)) = 1] - \Pr[\mathcal{D}(\langle P, V \rangle(x, w)) = 1]| \leq \epsilon_z$$

- Weak ZK: $\epsilon_c, \epsilon_s, \epsilon_z$ may be noticeable



One-Way Functions

One-way function f : any PPT algorithm $\mathcal{A}$

$$\Pr_{y \leftarrow f(\mathcal{U})} [\mathcal{A}(y) \in f^{-1}(y)] < \text{negl}(n)$$

Minimal assumption for MiniCrypt: PRGs [HILL99], symmetric-key encryption, commitments

Equivalent form

- Distributional OWF [IL89]

$(x, f(x))$ and $(\mathcal{A}(f(x)), f(x))$ are statistically far

- Inverter $f^{-1}(y)$: randomly sample from $\{x : f(x) = y\}$ (output $\perp$ if the set is empty)

Related Work

- OWF implies ZK for PSPACE
- From ZK to OWF
 - Worst-case hardness to ai-OWFs / average-case hardness to OWFs [Ost91, OW93]
 - Worst-case hardness to OWFs [HN24]
- What if ZK itself only has weak parameters?

Related Work

From (Weak) ZK to OWFs (Non-interactive)

	Error condition	Round	Hardness
[Ost91, OW93]	Negligible	Any	Average-case
[HN24]	Negligible	Any	Worst-case
[OW93]	$\epsilon_c + \epsilon_s + 2\epsilon_z < 1$	Non-interactive	Worst-case
[CHK25]	$\epsilon_c + \sqrt{\epsilon_s} + \epsilon_z < 1$	Non-interactive	Worst-case
This work	$\epsilon_c + \epsilon_s + \epsilon_z < 1$	Non-interactive	Worst-case
[CHKT26]	$\epsilon_c + \epsilon_s + \epsilon_z < 1$	Non-interactive	Worst-case

Related Work

From (Weak) ZK to OWFs (Interactive)

	Error condition	Round (k)	Hardness
[Ost91, OW93]	Negligible	Any	Average-case
[HN24]	Negligible	Any	Worst-case
[OW93]	$\epsilon_c + \epsilon_s + 2k\epsilon_z < 1$	Any	Worst-case
This work	$\epsilon_c + \epsilon_s + (2k - 1)\epsilon_z < 1$	Any	Worst-case
This work	$\epsilon_c + \epsilon_s + k\epsilon_z < 1$	Constant	Worst-case
[CHKT26]	$\epsilon_c + \epsilon_s + \epsilon_z < 1$	Constant	Worst-case

Main Results

One-way functions from NIZK

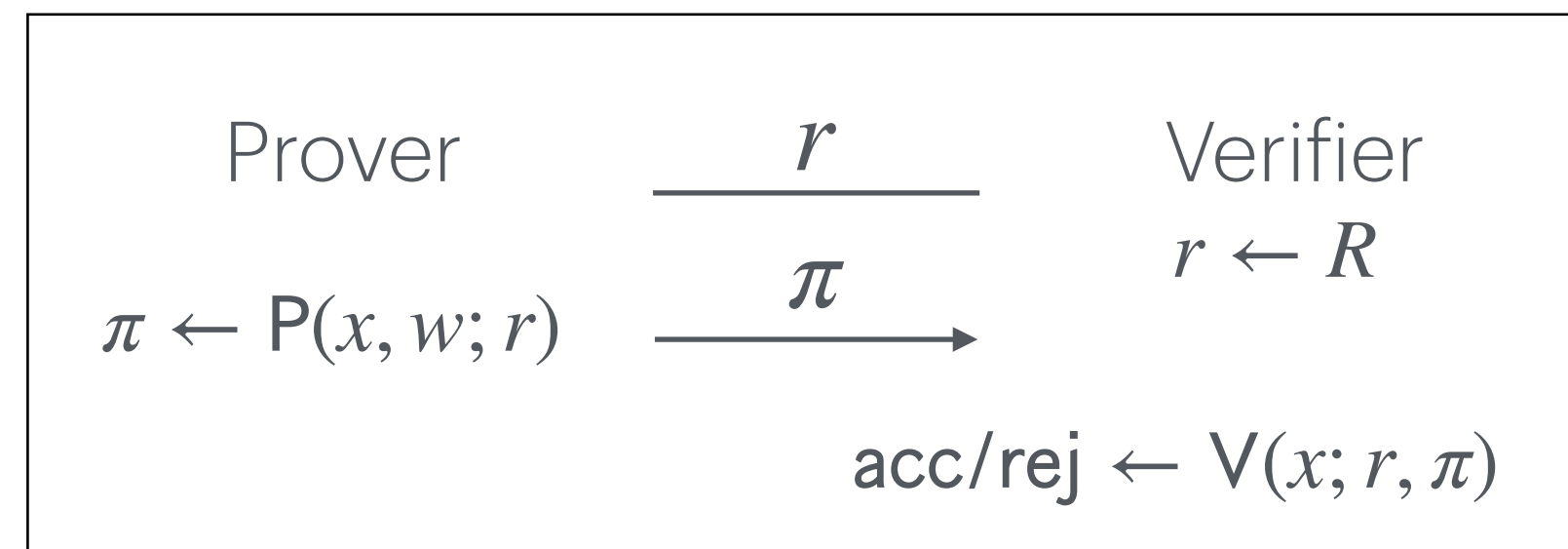
Theorem 1. If $\text{NP} \not\subseteq \text{P/poly}$ and every NP language has weak NIZK proofs/arguments, satisfying

$$\epsilon_c + \epsilon_s + \epsilon_z < 1$$

Then OWFs exist.

- Nontrivial parameters: NP language has weak NIZK with $\epsilon_c + \epsilon_s + \epsilon_z = 1$
- Corollary: Weak NIZK proof with $\epsilon_c + \epsilon_s + \epsilon_z < 1$ and negligible $\epsilon_c \implies$ (standard) NIZK proof
(NIZK amplification [BG24])

Proof Overview

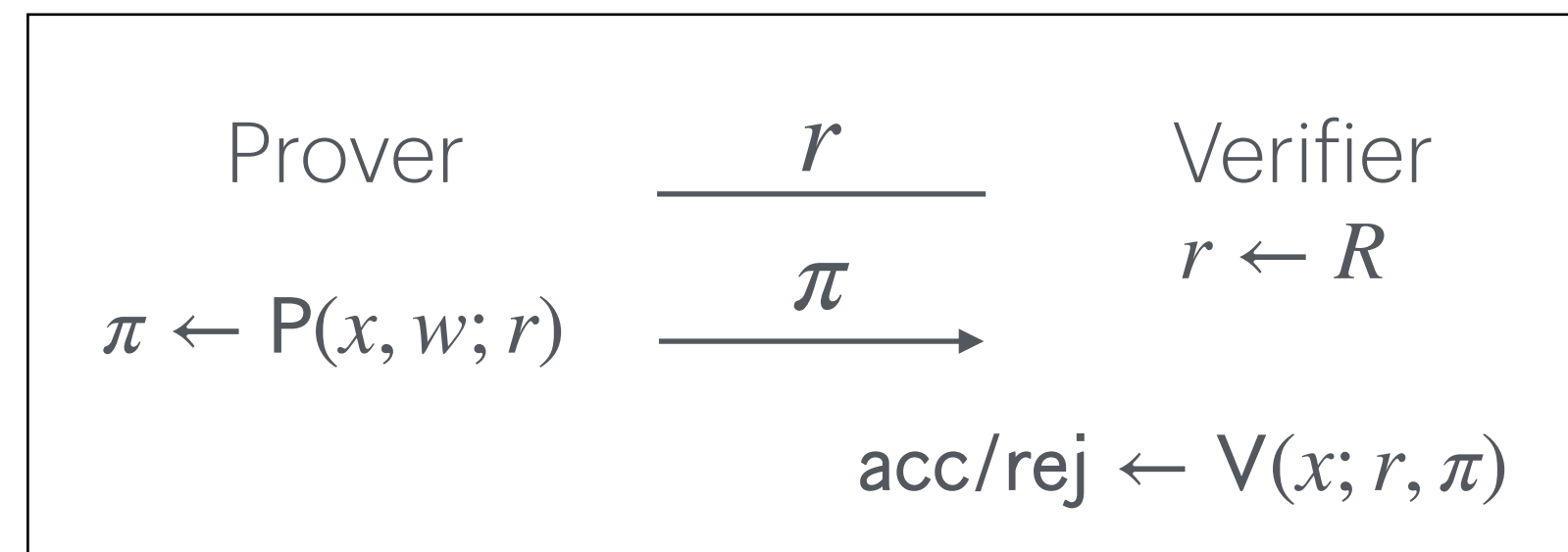


1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{P}(x; r)$
4. Run the protocol $\langle \hat{P}, V \rangle$ as the distinguisher

Use ZK simulator **Sim**($x; \rho$)

Without knowing NP witness w

[OW93] Construction



1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{P}(x; r)$
4. Run the protocol $\langle \hat{P}, V \rangle$ as the distinguisher

$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho)$, output r

Auxiliary-input OWFs

$\hat{\rho} \leftarrow f_x^{-1}(r), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho})$, send $\hat{\pi}$

What is the probability of acceptance for
 $x \in \mathcal{L}$ and $x \notin \mathcal{L}$?

[OW93] Construction

1. Propose a candidate function f

$$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), \text{ output } r$$

2. Assume an inverter f^{-1}

3. Construct an efficient prover's algorithm $\hat{\mathbf{P}}(x; r)$

$$\hat{\rho} \leftarrow f_x^{-1}(r), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$$

4. Run the protocol $\langle \hat{\mathbf{P}}, \mathbf{V} \rangle$ as the distinguisher

If $x \notin \mathcal{L}$, soundness ensures

$$\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \leq \epsilon_s$$

If $x \in \mathcal{L}$, transcript $(r, \hat{\pi})$ follows $\text{Sim}(x; f_x^{-1}(R))$

Compare it with $\langle \mathbf{P}, \mathbf{V} \rangle(x)$: $\langle \mathbf{P}, \mathbf{V} \rangle(x) \approx_{\epsilon_z} \text{Sim}(x; U)$

$$\text{Sim}(x; f_x^{-1}(R)) \approx_{\epsilon_z} \text{Sim}(x; U)$$

Then $\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \geq \Pr[\mathbf{V}(x; r, \pi) = 1] - 2\epsilon_z$

[OW93] Construction

1. Propose a candidate function f

$$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), \text{ output } r$$

2. Assume an inverter f^{-1}

3. Construct an efficient prover's algorithm $\hat{\mathbf{P}}(x; r)$

$$\hat{\rho} \leftarrow f_x^{-1}(r), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$$

4. Run the protocol $\langle \hat{\mathbf{P}}, \mathbf{V} \rangle$ as the distinguisher

$$\text{Require } \epsilon_c + \epsilon_s + 2\epsilon_z < 1$$

If $x \notin \mathcal{L}$, soundness ensures

$$\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \leq \epsilon_s$$

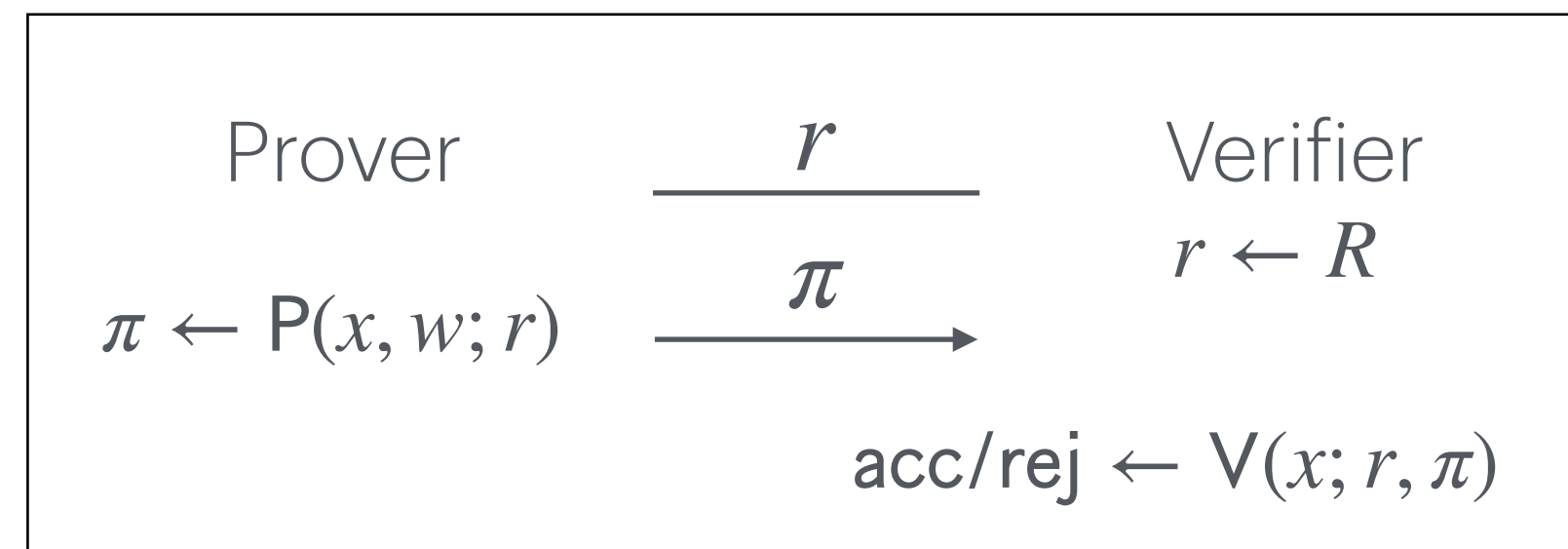
If $x \in \mathcal{L}$, transcript $(r, \hat{\pi})$ follows $\text{Sim}(x; f_x^{-1}(R))$

Compare it with $\langle \mathbf{P}, \mathbf{V} \rangle(x)$: $\langle \mathbf{P}, \mathbf{V} \rangle(x) \approx_{\epsilon_z} \text{Sim}(x; U)$

$$\text{Sim}(x; f_x^{-1}(R)) \approx_{\epsilon_z} \text{Sim}(x; U)$$

Then $\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \geq 1 - \epsilon_c - 2\epsilon_z$

Our Approach

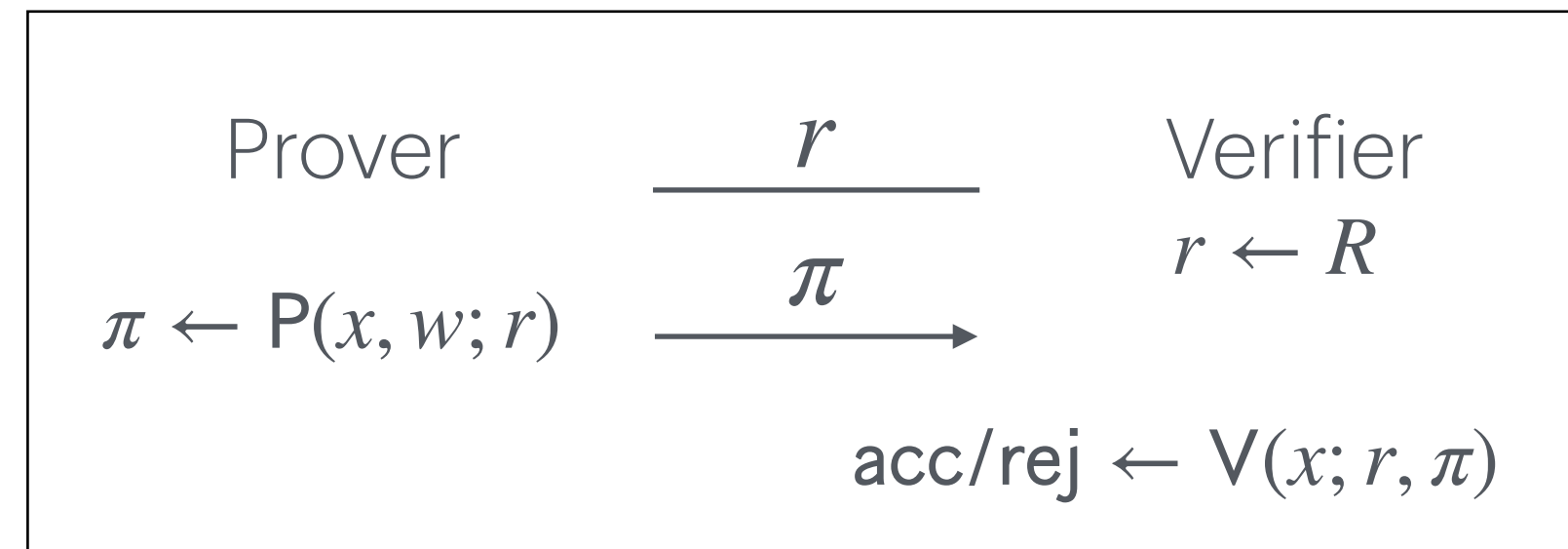


1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{P}(x; r)$
4. Run the protocol $\langle \hat{P}, V \rangle$ as the distinguisher

$$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), \text{ output } r$$

$$\hat{\rho} \leftarrow f_x^{-1}(r), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$$

Our Approach

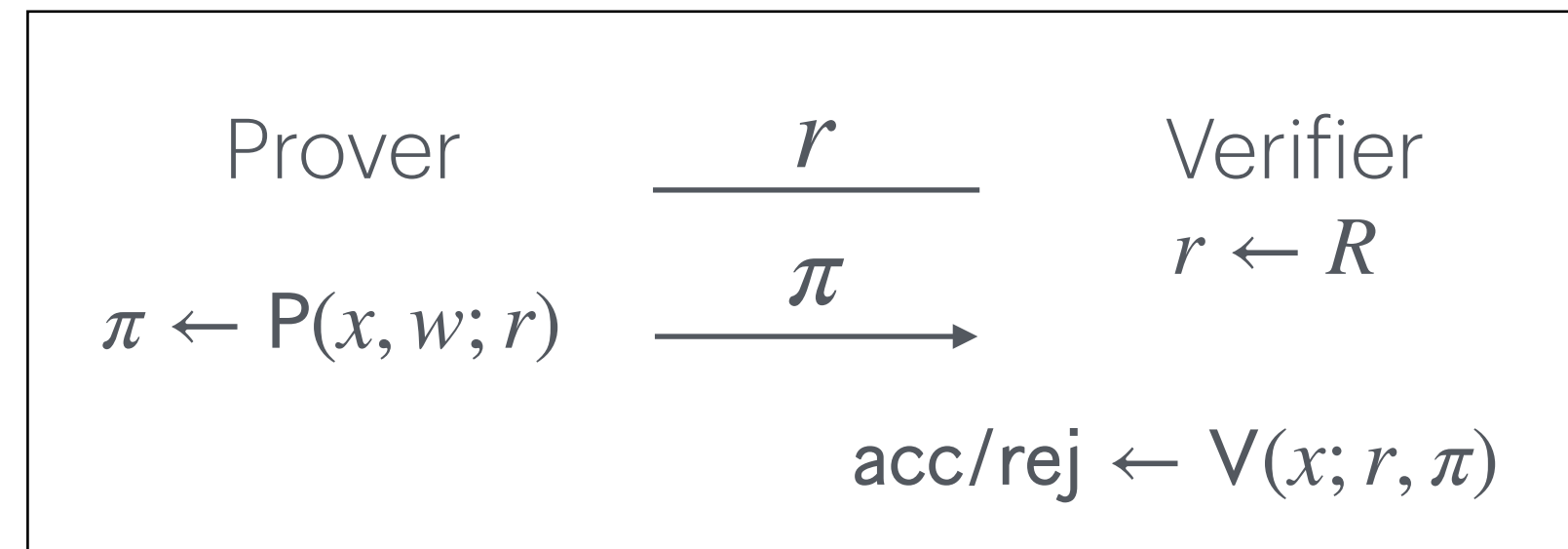


1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{P}(x; r)$
4. Run the protocol $\langle \hat{P}, V \rangle$ as the distinguisher

$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), \textcolor{red}{a} \leftarrow \textcolor{red}{V}(x; r, \pi)$
 output $(r, \textcolor{red}{a})$

$\hat{\rho} \leftarrow f_x^{-1}(r), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$

Our Approach



1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{P}(x; r)$
4. Run the protocol $\langle \hat{P}, V \rangle$ as the distinguisher

$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), \textcolor{red}{a} \leftarrow \textcolor{red}{V}(x; r, \pi)$
 output $(r, \textcolor{red}{a})$

$\hat{\rho} \leftarrow f_x^{-1}(r, \textcolor{red}{1}), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$

Our Approach

1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{\mathbf{P}}(x; r)$
4. Run the protocol $\langle \hat{\mathbf{P}}, \mathbf{V} \rangle$ as the distinguisher

If $x \notin \mathcal{L}$, soundness ensures

$$\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \leq \epsilon_s$$

$$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), a \leftarrow \mathbf{V}(x; r, \pi) \text{ output } (r, a)$$

$$\hat{\rho} \leftarrow f_x^{-1}(r, 1), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$$

$\hat{\pi}$ is accepting if and only if f_x^{-1} finds a valid preimage

If $x \in \mathcal{L}$, compare the input $(R, 1)$ with $f_x(U)$

$$(R, 1) \approx_{\epsilon_z + \epsilon_c} f_x(U)$$

Our Approach

1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{\mathbf{P}}(x; r)$
4. Run the protocol $\langle \hat{\mathbf{P}}, \mathbf{V} \rangle$ as the distinguisher

If $x \notin \mathcal{L}$, soundness ensures

$$\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \leq \epsilon_s$$

$$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), a \leftarrow \mathbf{V}(x; r, \pi) \text{ output } (r, a)$$

$$(r, \pi) \leftarrow \langle \mathbf{P}, \mathbf{V} \rangle(x), a \leftarrow \mathbf{V}(x; r, \pi) \text{ output } (r, a)$$

$$\hat{\rho} \leftarrow f_x^{-1}(r, 1), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$$

$\hat{\pi}$ is accepting if and only if f_x^{-1} finds a valid preimage

If $x \in \mathcal{L}$, compare the input $(R, 1)$ with $f_x(U)$

$$(R, 1) \approx_{\epsilon_z + \epsilon_c} f_x(U)$$

Our Approach

1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{\mathbf{P}}(x; r)$
4. Run the protocol $\langle \hat{\mathbf{P}}, \mathbf{V} \rangle$ as the distinguisher

If $x \notin \mathcal{L}$, soundness ensures

$$\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \leq \epsilon_s$$

$$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), a \leftarrow \mathbf{V}(x; r, \pi) \text{ output } (r, a)$$

$$(r, \pi) \leftarrow \langle \mathbf{P}, \mathbf{V} \rangle(x), a \leftarrow \mathbf{V}(x; r, \pi) \text{ output } (r, a)$$

$$(r, \pi) \leftarrow \langle \mathbf{P}, \mathbf{V} \rangle(x) \text{ output } (r, 1)$$

$$\hat{\rho} \leftarrow f_x^{-1}(r, 1), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$$

$\hat{\pi}$ is accepting if and only if f_x^{-1} finds a valid preimage

If $x \in \mathcal{L}$, compare the input $(R, 1)$ with $f_x(U)$

$$(R, 1) \approx_{\epsilon_z + \epsilon_c} f_x(U)$$

Our Approach

1. Propose a candidate function f
2. Assume an inverter f^{-1}
3. Construct an efficient prover's algorithm $\hat{\mathbf{P}}(x; r)$
4. Run the protocol $\langle \hat{\mathbf{P}}, \mathbf{V} \rangle$ as the distinguisher

If $x \notin \mathcal{L}$, soundness ensures

$$\Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] \leq \epsilon_s$$

$$f_x(\rho): (r, \pi) \leftarrow \text{Sim}(x; \rho), a \leftarrow \mathbf{V}(x; r, \pi) \text{ output } (r, a)$$

$$(r, \pi) \leftarrow \langle \mathbf{P}, \mathbf{V} \rangle(x), a \leftarrow \mathbf{V}(x; r, \pi) \text{ output } (r, a)$$

$$(r, \pi) \leftarrow \langle \mathbf{P}, \mathbf{V} \rangle(x) \text{ output } (r, 1)$$

$$\hat{\rho} \leftarrow f_x^{-1}(r, 1), (r, \hat{\pi}) \leftarrow \text{Sim}(x; \hat{\rho}), \text{ send } \hat{\pi}$$

$$\text{Require } \epsilon_c + \epsilon_s + \epsilon_z < 1$$

$\hat{\pi}$ is accepting if and only if f_x^{-1} finds a valid preimage

If $x \in \mathcal{L}$, compare the input $(R, 1)$ with $f_x(U)$

$$(R, 1) \approx_{\epsilon_z + \epsilon_c} f_x(U)$$

$$\text{Then } \Pr[\mathbf{V}(x; r, \hat{\pi}) = 1] = \Pr[f_x^{-1} \text{ inverts } (r, 1)] \geq 1 - \epsilon_c - \epsilon_z$$

Main Results

OWFs from interactive ZK

Theorem 2. If NP language has k -round public-coin ZK proofs/arguments, satisfying

$$\epsilon_c + \epsilon_s + (2k - 1) \cdot \epsilon_z < 1$$

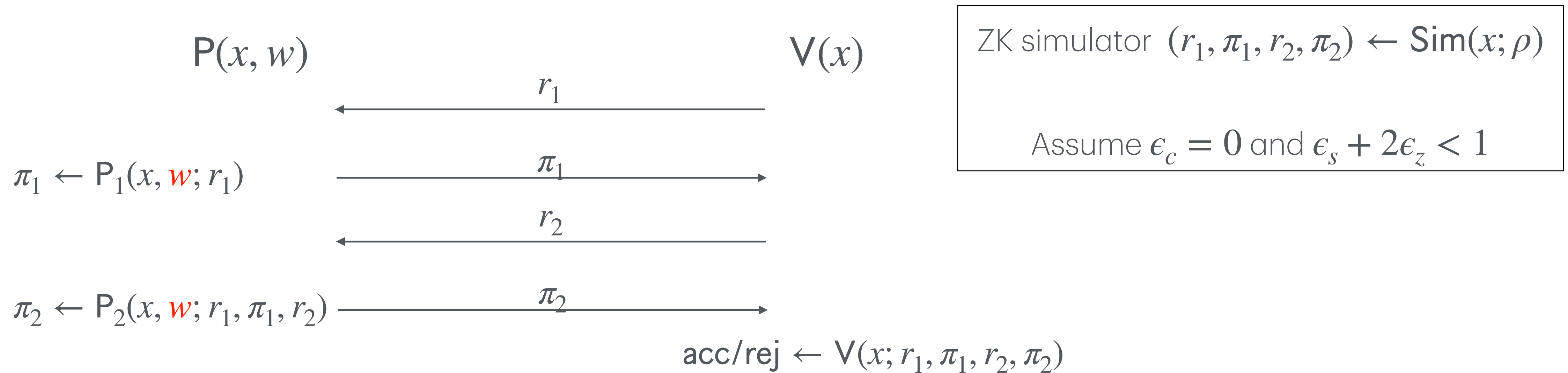
Then OWFs exist.

Theorem 3. If NP language has k -round public-coin ZK proofs/arguments, satisfying

$$\epsilon_c + \epsilon_s + k \cdot \epsilon_z < 1 \text{ and } k \text{ is a constant}$$

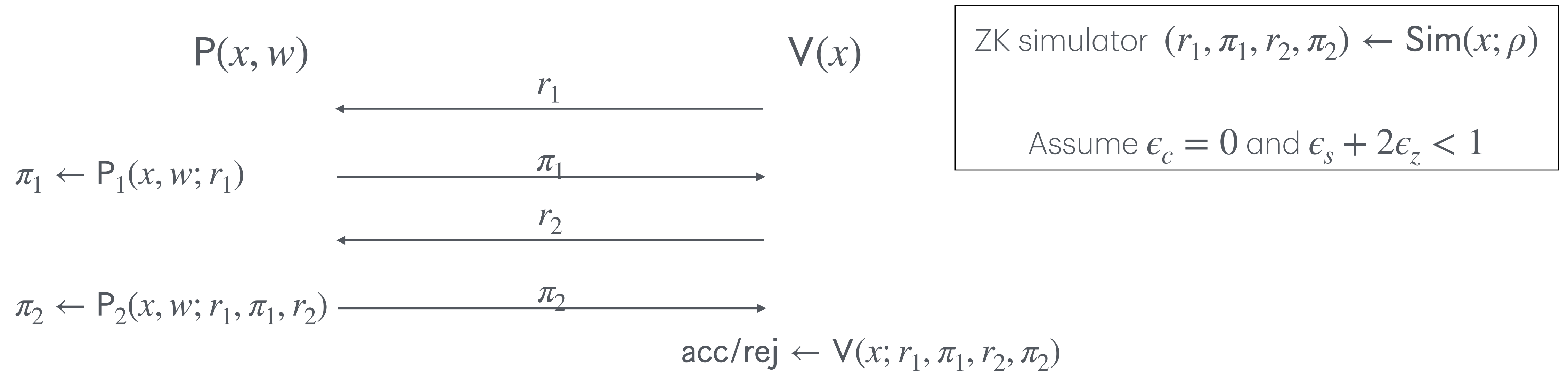
Then (infinitely-often) OWFs exist.

Example: 2-round public-coin ZK



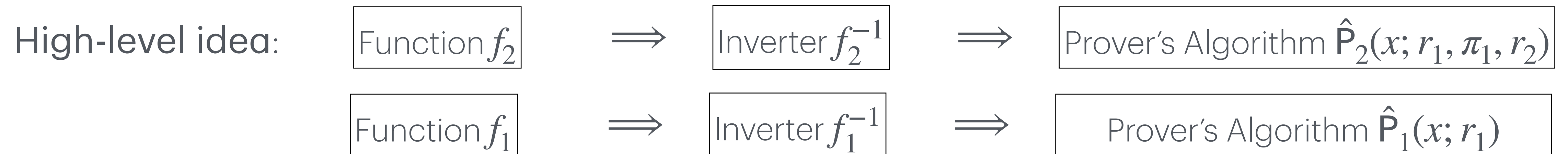
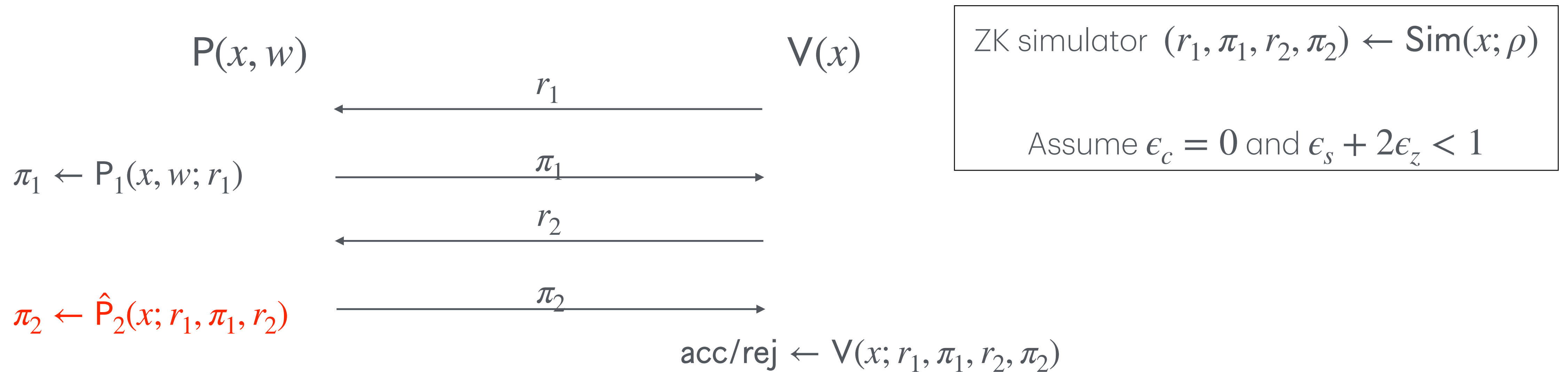
High-level idea: use inverters to construct efficient prover's algorithms (without NP **witness**), run the protocol to decide the language

Example: 2-round public-coin ZK

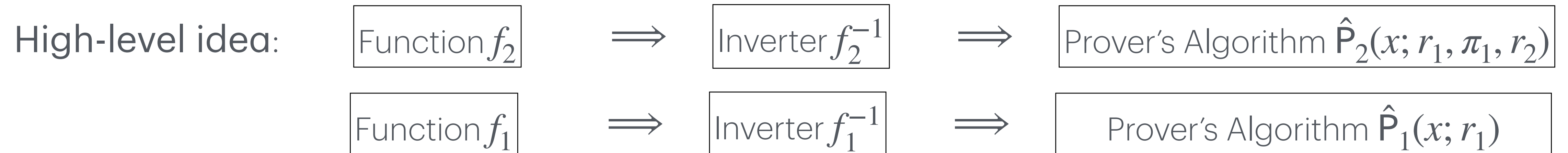
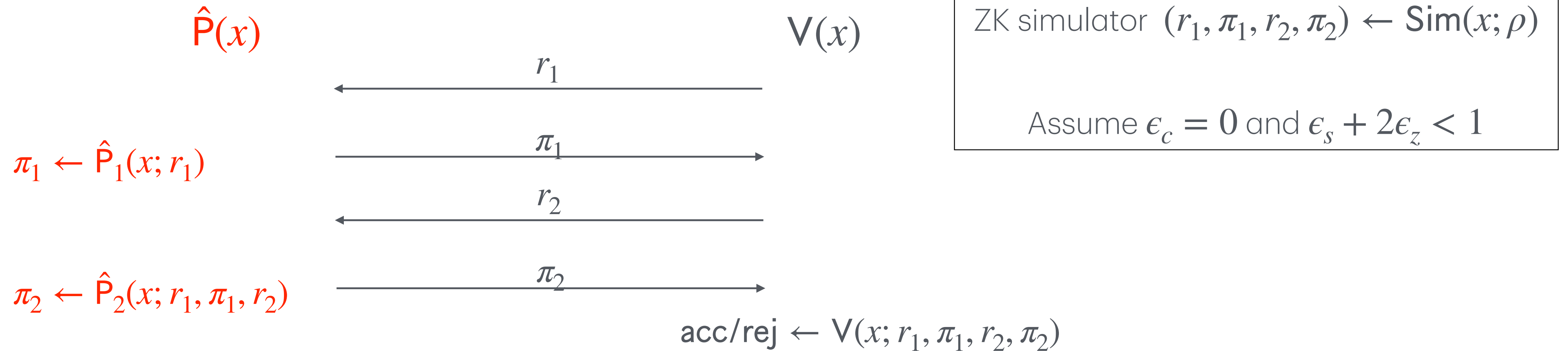


High-level idea: Function f_2 $\implies$ Inverter f_2^{-1} $\implies$ Prover's Algorithm $\hat{P}_2(x; r_1, \pi_1, r_2)$

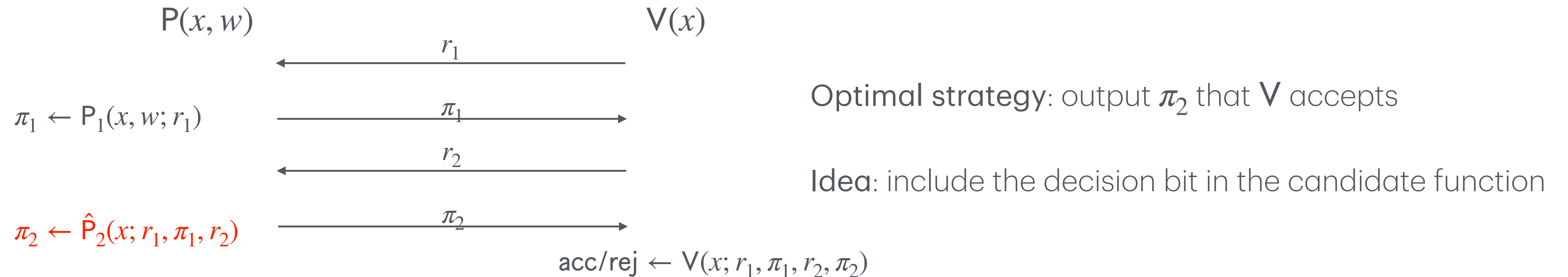
Example: 2-round public-coin ZK



Example: 2-round public-coin ZK



Example: 2-round public-coin ZK



Candidate function

$$f_{2,x}(\rho): (r_1, \pi_1, r_2, \pi_2) \leftarrow \text{Sim}(x; \rho), a \leftarrow V(x; r_1, \pi_1, r_2, \pi_2), \text{ output } (r_1, \pi_1, r_2, a)$$

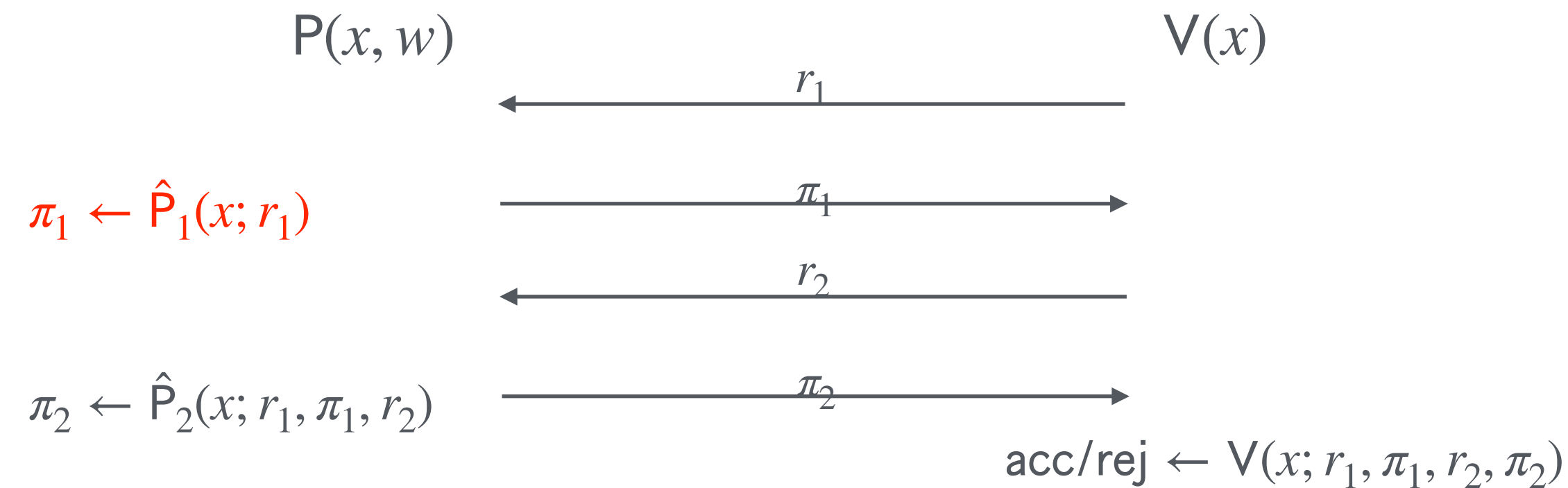
If we can invert $f_{2,x}$

$$\hat{P}_2(x; r_1, \pi_1, r_2): \hat{\rho} \leftarrow f_{2,x}^{-1}(r_1, \pi_1, r_2, 1), \text{ output } \hat{\pi}_2 \leftarrow \text{Sim}_4(x; \hat{\rho})$$

When $x \in \mathcal{L}$

$$\Pr_{P_1, \hat{P}_2} [V \text{ accepts}] \geq 1 - \epsilon_z$$

Example: 2-round public-coin ZK



Optimal strategy: output π_1 that maximizes **Val**

Idea: obtain the information from the inverter

Define $\mathbf{Val}(x; r_1, \pi_1) = \Pr_{\hat{P}_2}[\mathbf{V} \text{ accepts}]$

Measure the probability that $\hat{P}_2$ convinces the verifier given the transcript (r_1, π_1)

Suppose **Val** can be computed deterministically

Candidate function

$$f_{1,x}(\rho): (r_1, \pi_1, r_2, \pi_2) \leftarrow \mathbf{Sim}(x; \rho), \text{ output } (r_1, \mathbf{Val}(x; r_1, \pi_1))$$

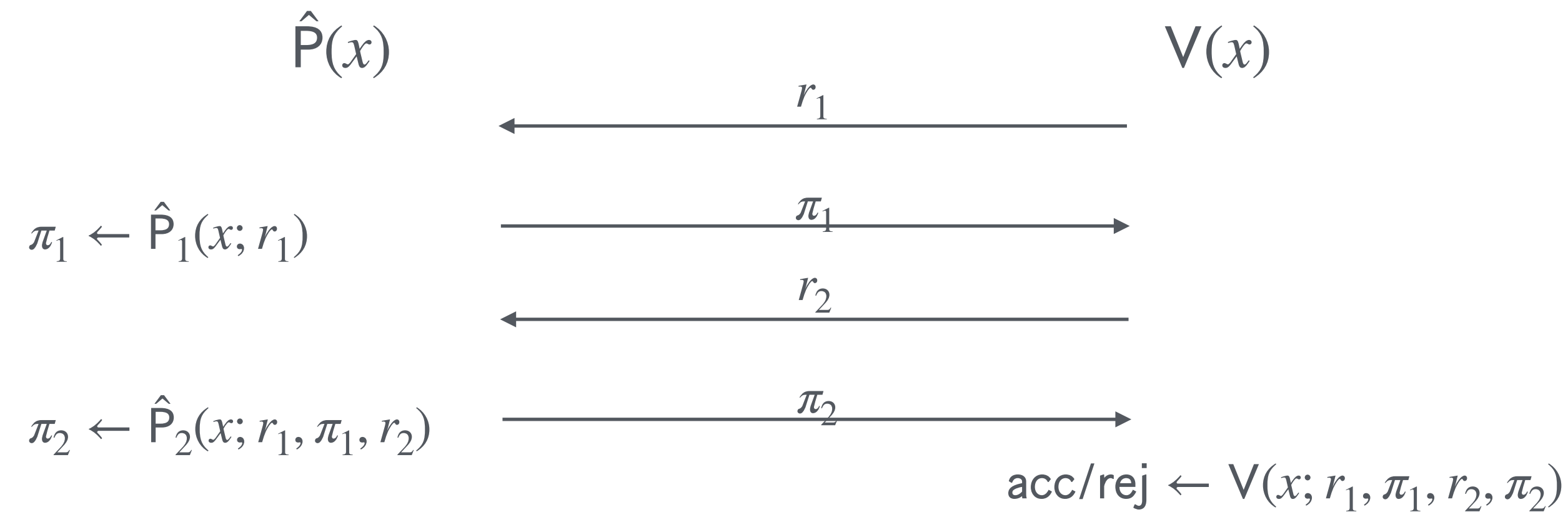
If we can invert $f_{1,x}$

$$\hat{P}_1(x; r_1): \hat{a} \leftarrow \text{the maximum } a \text{ s.t. } f_{1,x}^{-1} \text{ can invert } (r_1, a), \text{ and } \hat{\rho} \leftarrow f_{1,x}^{-1}(r_1, \hat{a}), \text{ output } \hat{\pi}_1 \leftarrow \mathbf{Sim}_2(x; \hat{\rho})$$

When $x \in \mathcal{L}$

$$\Pr_{\hat{P}_1, \hat{P}_2} [\mathbf{V} \text{ accepts}] \geq 1 - 2\epsilon_z$$

Example: 2-round public-coin ZK



ZK simulator $(r_1, \pi_1, r_2, \pi_2) \leftarrow \text{Sim}(x; \rho)$

Assume $\epsilon_c = 0$ and $\epsilon_s + 2\epsilon_z < 1$

The distinguisher runs the protocol

When $x \in \mathcal{L}$, $\Pr_{\hat{P}_1, \hat{P}_2} [V \text{ accepts}] \geq 1 - 2\epsilon_z$

Contradict the worst-case hardness of $\mathcal{L}$

When $x \notin \mathcal{L}$, $\Pr_{\hat{P}_1, \hat{P}_2} [V \text{ accepts}] \leq \epsilon_s$

At least one of the functions is one-way

Weak NIZK & $\epsilon_c + \epsilon_s + \epsilon_z < 1 \implies$ OWFs

Weak public-coin ZK under parameter constraints $\implies$ OWFs

- Include the verifier's decision bit (or the acceptance likelihood) in the function's output
- Use the inverters to construct efficient prover's strategies

Q & A