# CS 594 – Advanced Crypto (Spring 2026)

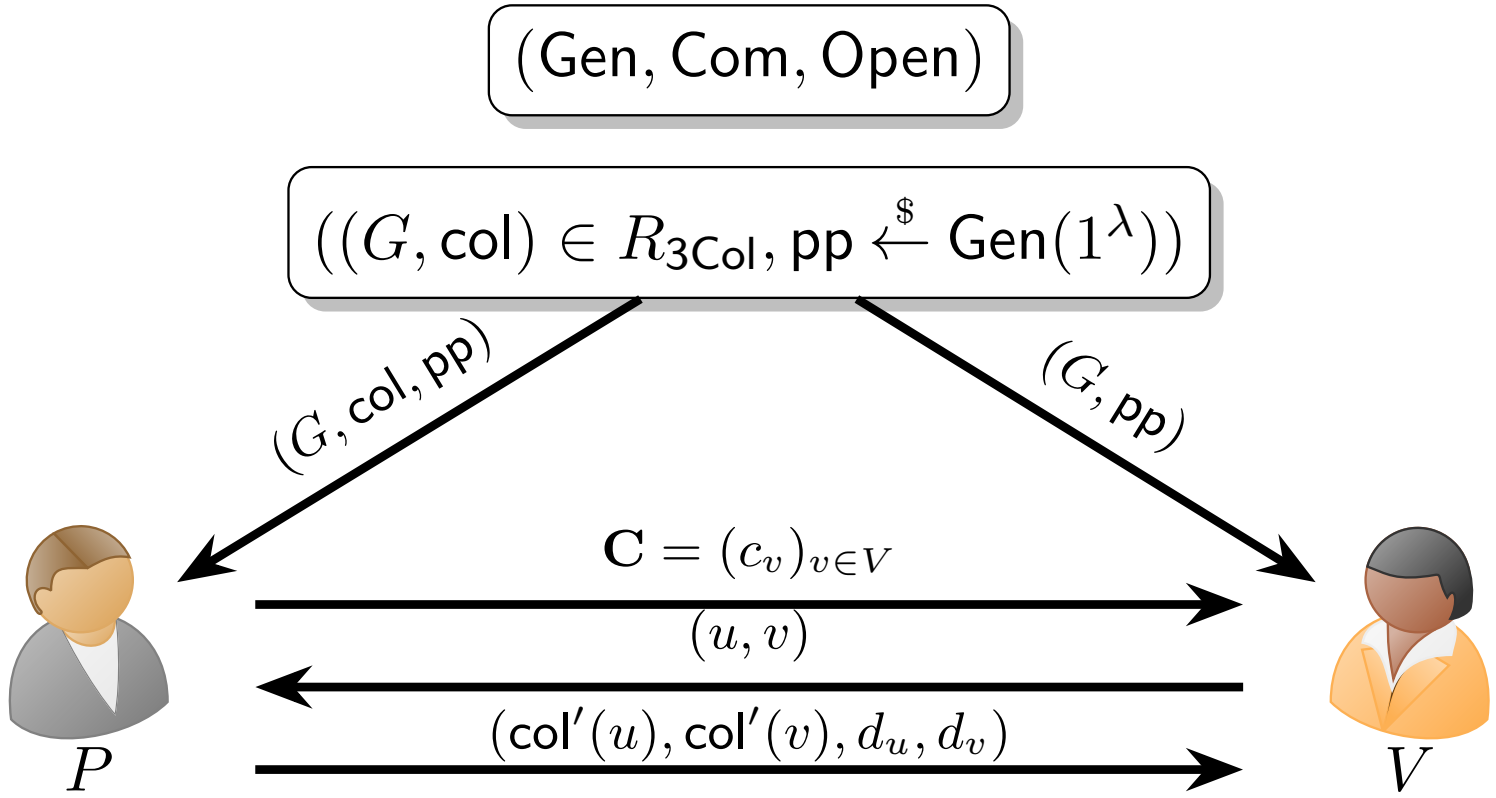
Alex Block

Lecture 14

March 04, 2026

# Sequential Repetition and Zero-knowledge

# Recall: 3Col Proof




(P1) Sample random permutation $\pi$ and compute $(c_v, d_v) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathsf{col}'(v))$, $\forall v \in V$

(V1) Sample $(u, v) \xleftarrow{\$} E$.

(V2) Check the following.

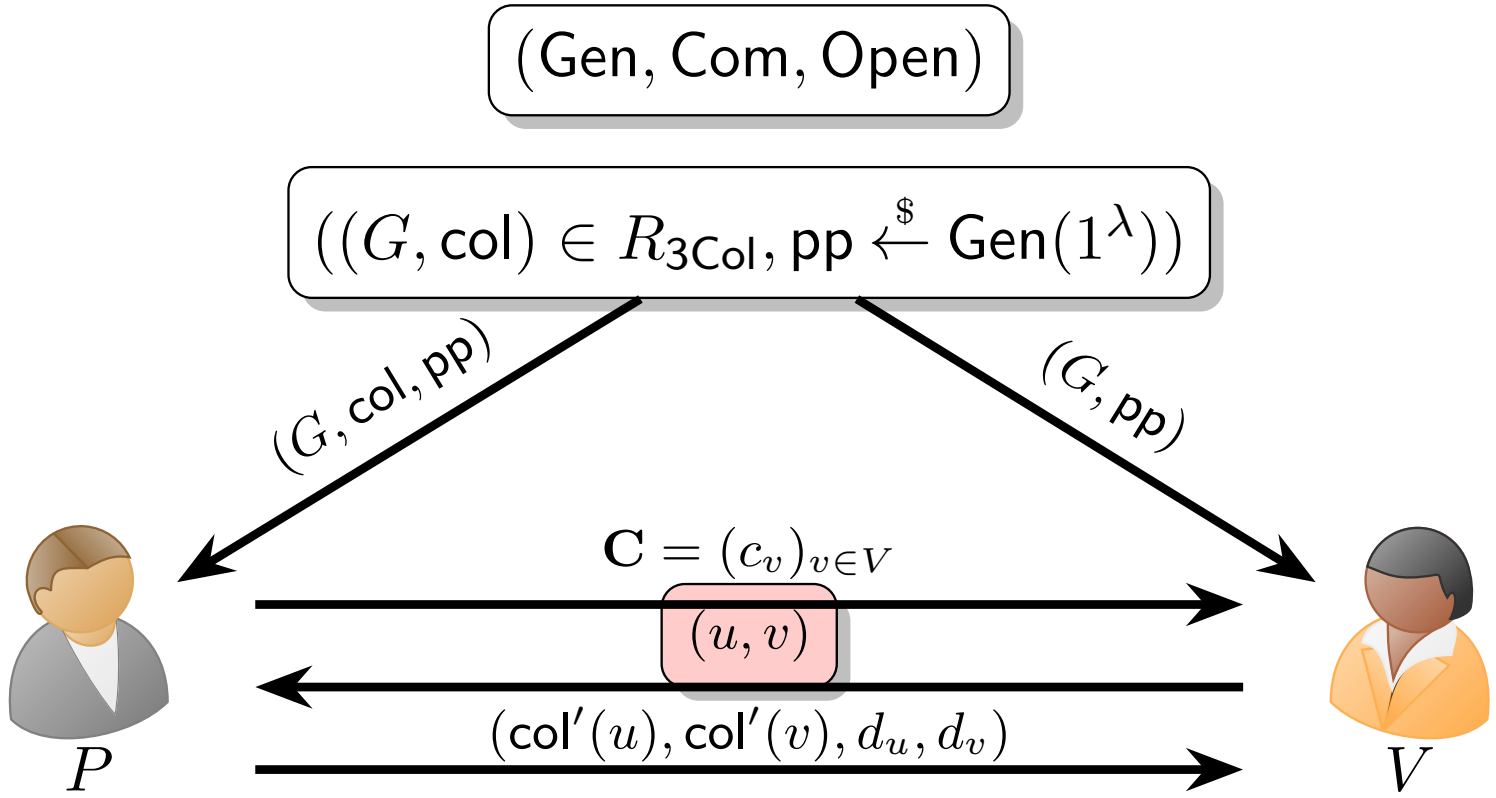
(1) $\mathsf{col}'(u), \mathsf{col}'(v) \in \{0, 1, 2\}$

(2) $\mathsf{col}'(u) \neq \mathsf{col}'(v)$

(3) $\mathsf{Open}(\mathsf{col}'(u), c_u, d_u) = 1$

(4) $\mathsf{Open}(\mathsf{col}'(v), c_v, d_v) = 1$

# Recall: 3Col Proof




(P1) Sample random permutation $\pi$ and compute $(c_v, d_v) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathsf{col}'(v))$, $\forall v \in V$

(V1) Sample $(u, v) \xleftarrow{\$} E$.

(V2) Check the following.

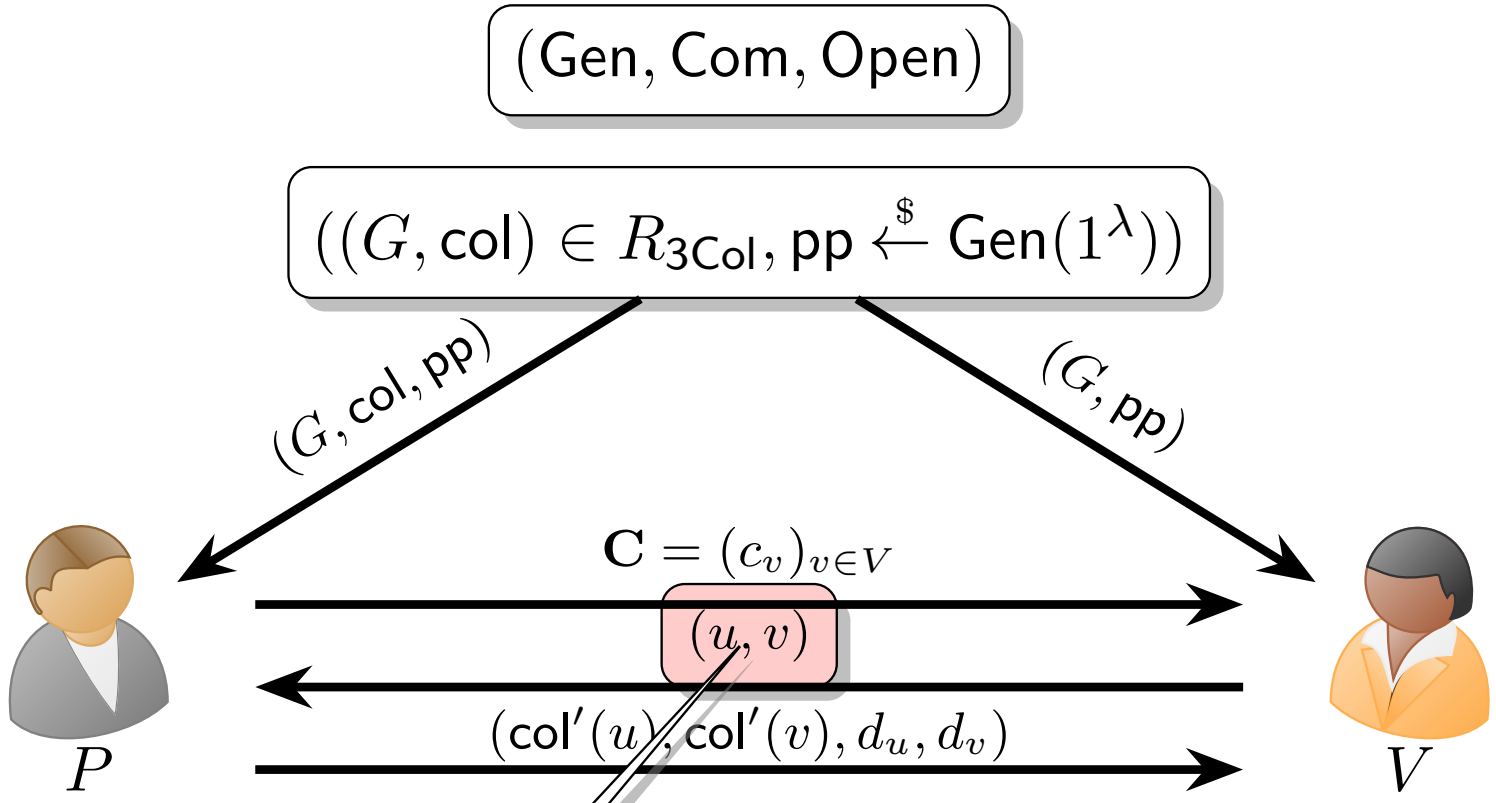
(1) $\mathsf{col}'(u), \mathsf{col}'(v) \in \{0, 1, 2\}$

(2) $\mathsf{col}'(u) \neq \mathsf{col}'(v)$

(3) $\mathsf{Open}(\mathsf{col}'(u), c_u, d_u) = 1$

(4) $\mathsf{Open}(\mathsf{col}'(v), c_v, d_v) = 1$

# Recall: 3Col Proof




(P1) Sample random permutation $\pi$ and compute $(c_v, d_v) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathsf{col}'(v))$, $\forall v \in V$

Soundness error $1 - 1/|E|$

(V1) Sample $(u, v) \xleftarrow{\$} E$.

(V2) Check the following.

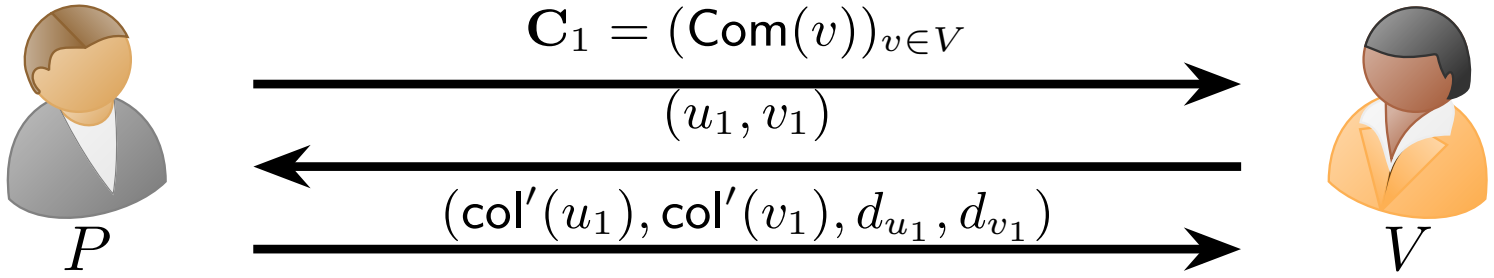
(1) $\mathsf{col}'(u), \mathsf{col}'(v) \in \{0, 1, 2\}$

(2) $\mathsf{col}'(u) \neq \mathsf{col}'(v)$

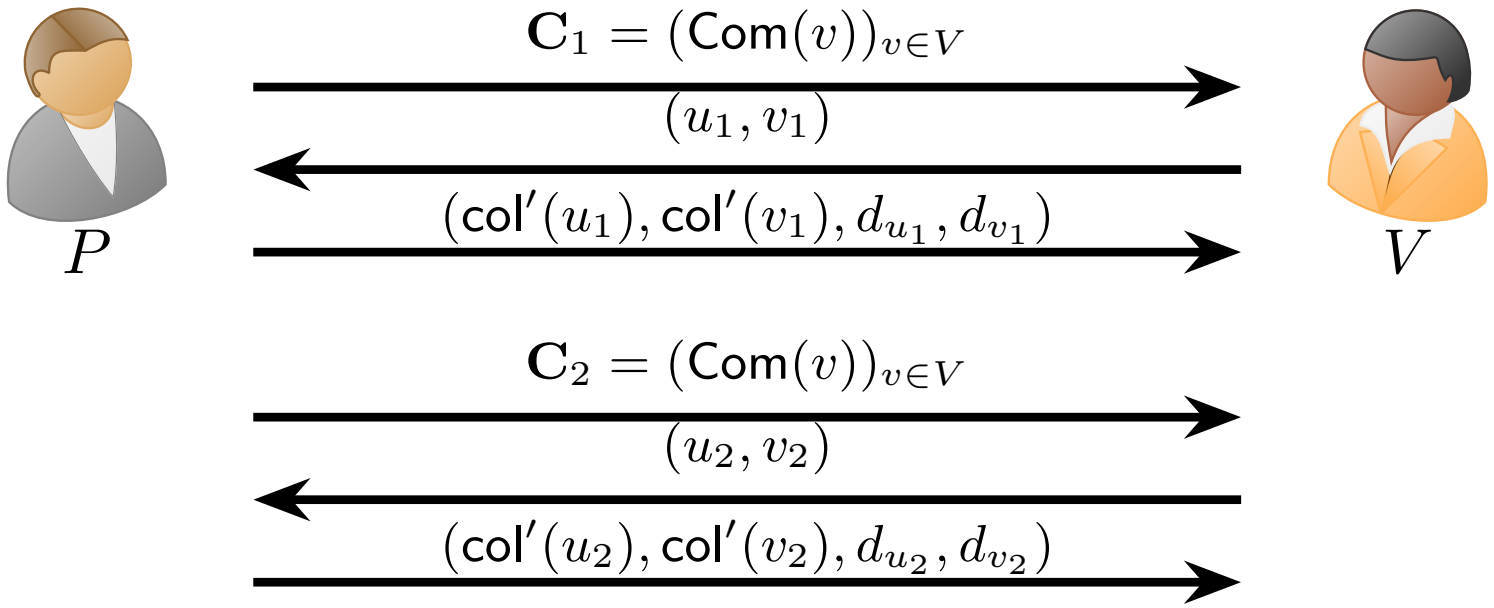
(3) $\mathsf{Open}(\mathsf{col}'(u), c_u, d_u) = 1$

(4) $\mathsf{Open}(\mathsf{col}'(v), c_v, d_v) = 1$

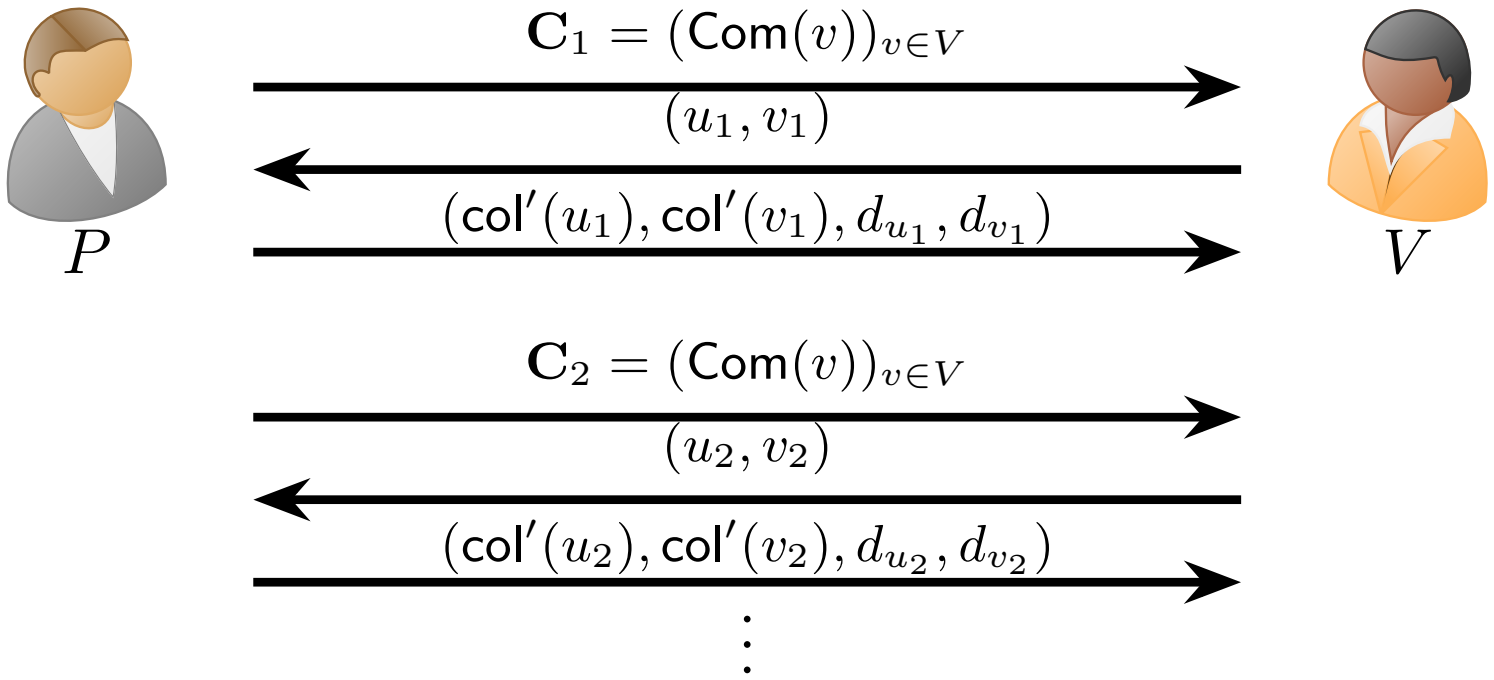
# Soundness Amp.: Sequential Repetition



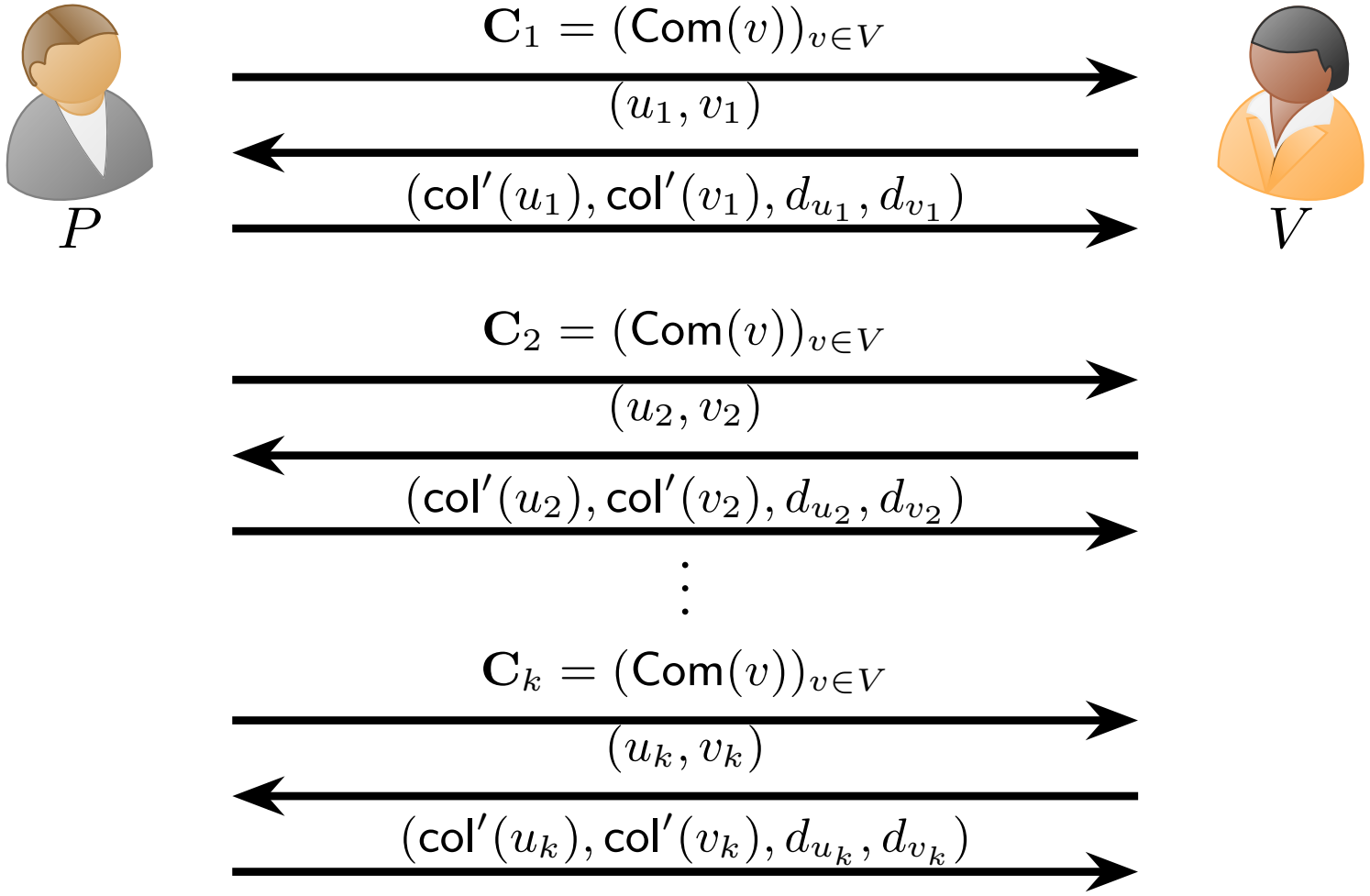
# Soundness Amp.: Sequential Repetition



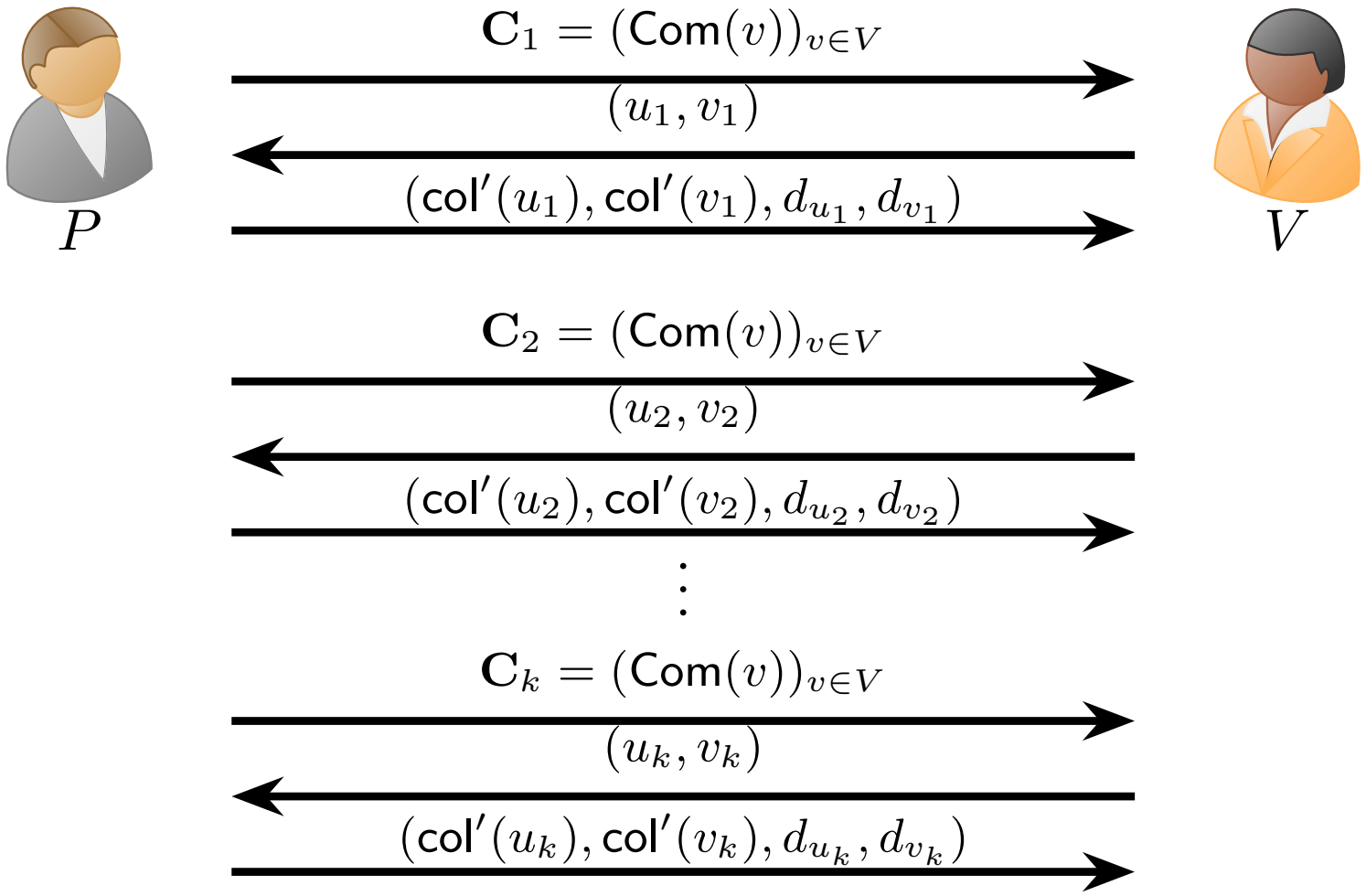
# Soundness Amp.: Sequential Repetition



# Soundness Amp.: Sequential Repetition



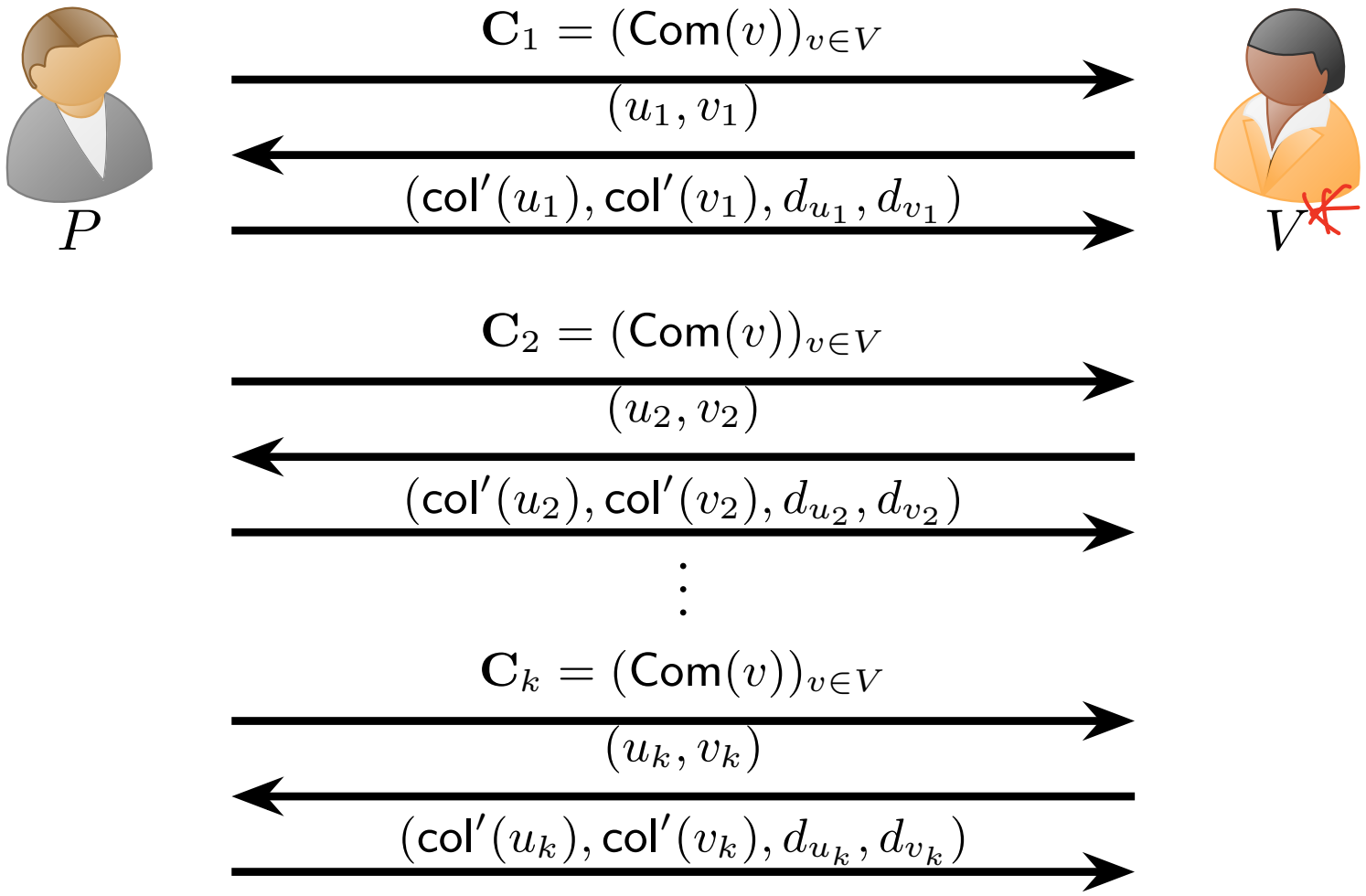
# Soundness Amp.: Sequential Repetition




**Soundness Error**

$$(1 - 1/|E|)^k$$

# Soundness Amp.: Sequential Repetition




**Soundness Error**

$(1 - 1/|E|)^k$

**Zero-knowledge?**

Some issues!

# Does Sequential Rep. Preserve ZK?

- Soundness error can be arbitrarily small, and for $k = \text{poly}(\lambda)$ it is negligible.

# Does Sequential Rep. Preserve ZK?

- Soundness error can be arbitrarily small, and for $k = \text{poly}(\lambda)$ it is negligible.

- But is Zero-knowledge preserved?

# Does Sequential Rep. Preserve ZK?

- Soundness error can be arbitrarily small, and for $k = \text{poly}(\lambda)$ it is negligible.

- But is Zero-knowledge preserved?
  - Issue: $V^*$ might be gaining more information from multiple rounds of interaction than from a single round!

# Does Sequential Rep. Preserve ZK?

- Soundness error can be arbitrarily small, and for $k = \text{poly}(\lambda)$ it is negligible.
- But is Zero-knowledge preserved?
  - Issue: $V^*$ might be gaining more information from multiple rounds of interaction than from a single round!
- Solution: *auxiliary input zero-knowledge*.

# Does Sequential Rep. Preserve ZK?

- Soundness error can be arbitrarily small, and for $k = \text{poly}(\lambda)$ it is negligible.

- But is Zero-knowledge preserved?
  - Issue: $V^*$ might be gaining more information from multiple rounds of interaction than from a single round!

- Solution: *auxiliary input zero-knowledge*.
  - Solves the above issue.

# Does Sequential Rep. Preserve ZK?

- Soundness error can be arbitrarily small, and for $k = \text{poly}(\lambda)$ it is negligible.

- But is Zero-knowledge preserved?
  - Issue: $V^*$ might be gaining more information from multiple rounds of interaction than from a single round!

- Solution: *auxiliary input zero-knowledge*.
  - Solves the above issue.
  - Also allows us to argue about zero-knowledge when composed with other protocols.

# Does Sequential Rep. Preserve ZK?

- Soundness error can be arbitrarily small, and for $k = \text{poly}(\lambda)$ it is negligible.

- But is Zero-knowledge preserved?
  - Issue: $V^*$ might be gaining more information from multiple rounds of interaction than from a single round!

- Solution: *auxiliary input zero-knowledge*.
  - Solves the above issue.
  - Also allows us to argue about zero-knowledge when composed with other protocols.

- In some sense, this is the "right" definition of zero-knowledge.

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

Let $(P, V)$ be a $k$-round interactive proof for a language $L$.

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

Let $(P, V)$ be a $k$-round interactive proof for a language $L$. Then, we say that the proof system has *auxiliary input zero-knowledge* if

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

Let $(P, V)$ be a $k$-round interactive proof for a language $L$. Then, we say that the proof system has *auxiliary input zero-knowledge* if for every PPT verifier algorithm $V^*$,

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

Let $(P, V)$ be a $k$-round interactive proof for a language $L$. Then, we say that the proof system has *auxiliary input zero-knowledge* if for every PPT verifier algorithm $V^*$, there exists a PPT algorithm $S$ (the *simulator*), which can depend on $V^*$, such that

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

Let $(P, V)$ be a $k$-round interactive proof for a language $L$. Then, we say that the proof system has *auxiliary input zero-knowledge* if for every PPT verifier algorithm $V^*$, there exists a PPT algorithm $S$ (the *simulator*), which can depend on $V^*$, such that for all $x \in L$

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

Let $(P, V)$ be a $k$-round interactive proof for a language $L$. Then, we say that the proof system has *auxiliary input zero-knowledge* if for every PPT verifier algorithm $V^*$, there exists a PPT algorithm $S$ (the *simulator*), which can depend on $V^*$, such that for all $x \in L$ and for any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(|x|)}$,

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

Let $(P, V)$ be a $k$-round interactive proof for a language $L$. Then, we say that the proof system has *auxiliary input zero-knowledge* if for every PPT verifier algorithm $V^*$, there exists a PPT algorithm $S$ (the *simulator*), which can depend on $V^*$, such that for all $x \in L$ and for any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(|x|)}$, the distribution $S(x, z)$ is "indistinguishable" from $\mathrm{View}_{V^*}(\langle P, V^*(z)\rangle(x))$.

# Auxiliary Input Zero-knowledge

**Definition 1 (Auxiliary Input Zero-knowledge, Informal)**

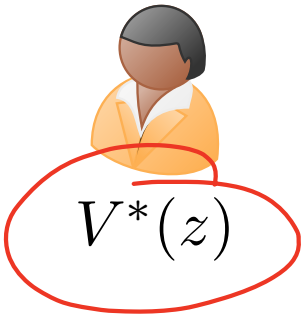
Let $(P, V)$ be a $k$-round interactive proof for a language $L$. Then, we say that the proof system has *auxiliary input zero-knowledge* if for every PPT verifier algorithm $V^*$, there exists a PPT algorithm $S$ (the *simulator*), which can depend on $V^*$, such that for all $x \in L$ and for any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(|x|)}$, the distribution $S(x, z)$ is "indistinguishable" from $\mathrm{View}_{V^*}(\langle P, V^*(z)\rangle(x))$. Here, $\mathrm{View}_{V^*}(\langle P, V^*(z)\rangle(x))$ denotes the distribution over proofs/transcripts generated by the interaction between $P$ and $V^*$.

# Simulation Idea for Sequential Repetition

# Simulation Idea for Sequential Repetition

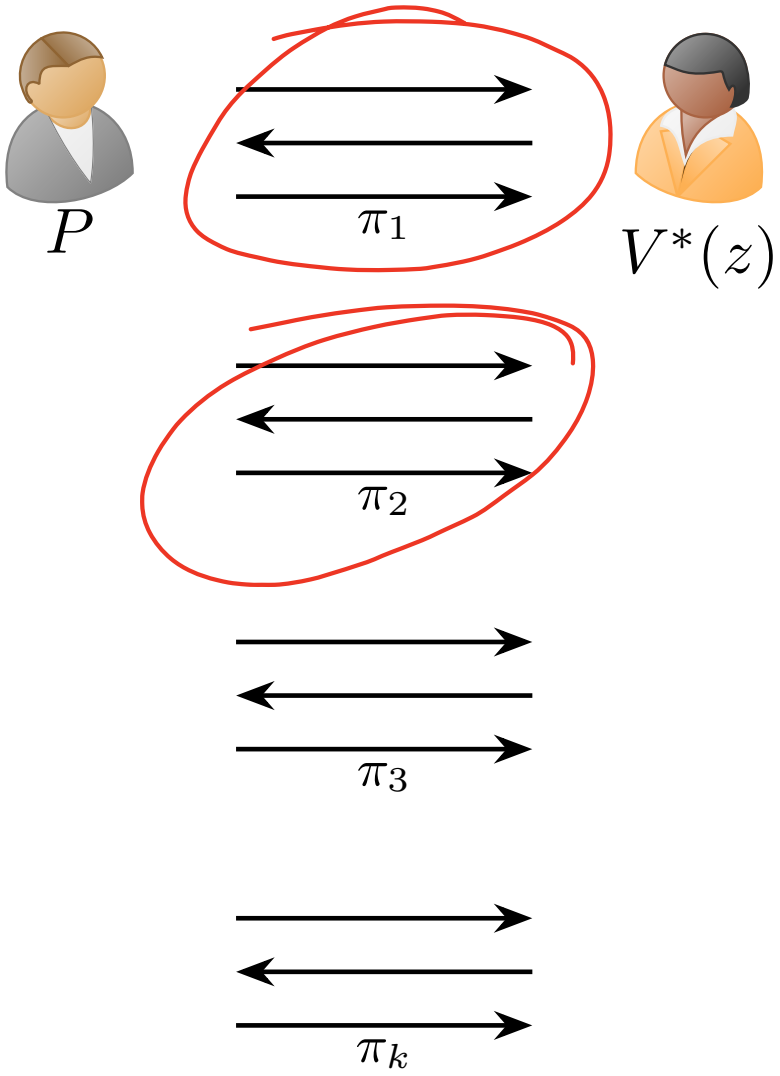
# Simulation Idea for Sequential Repetition

$P$

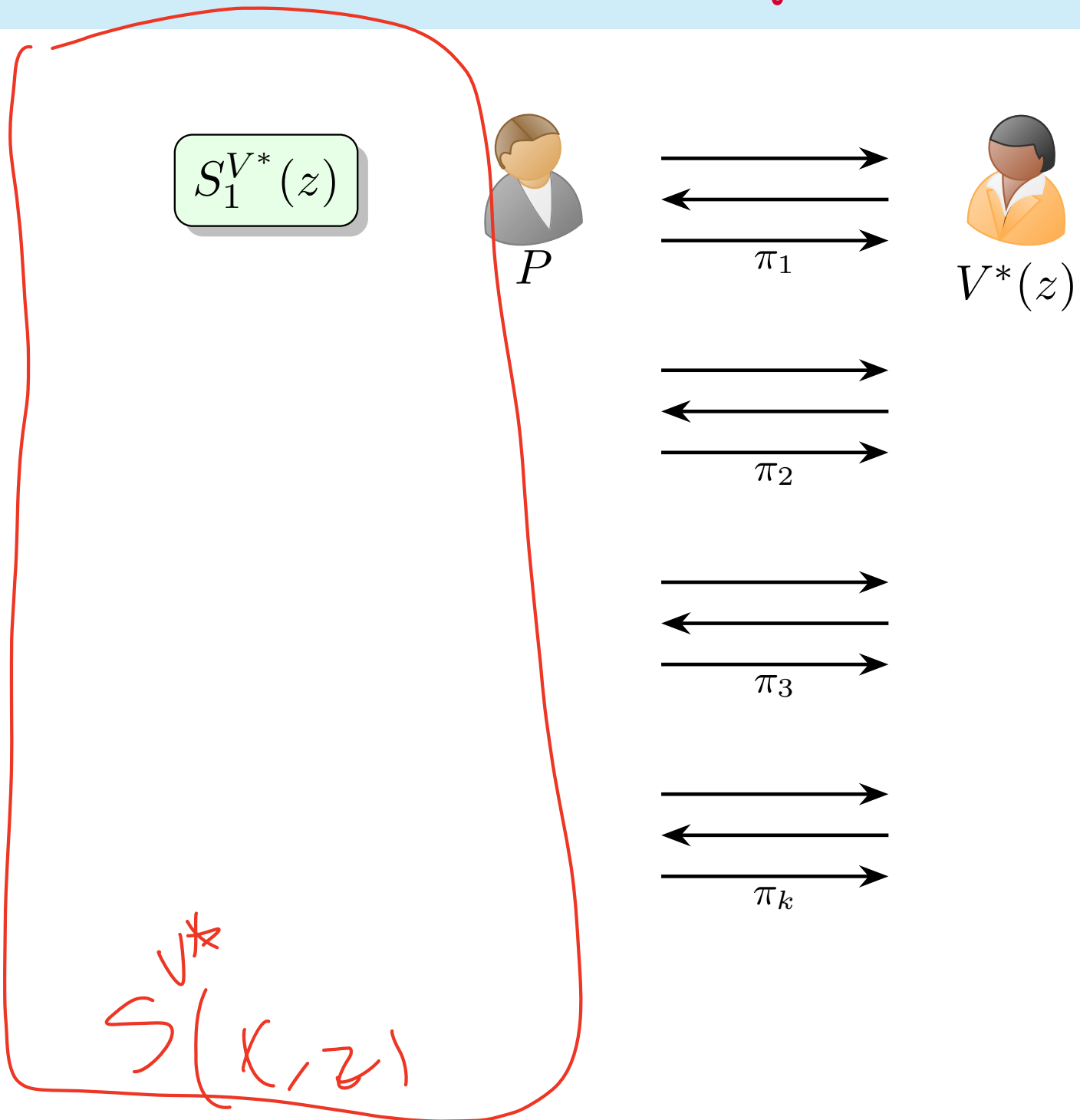


$V^*(z)$

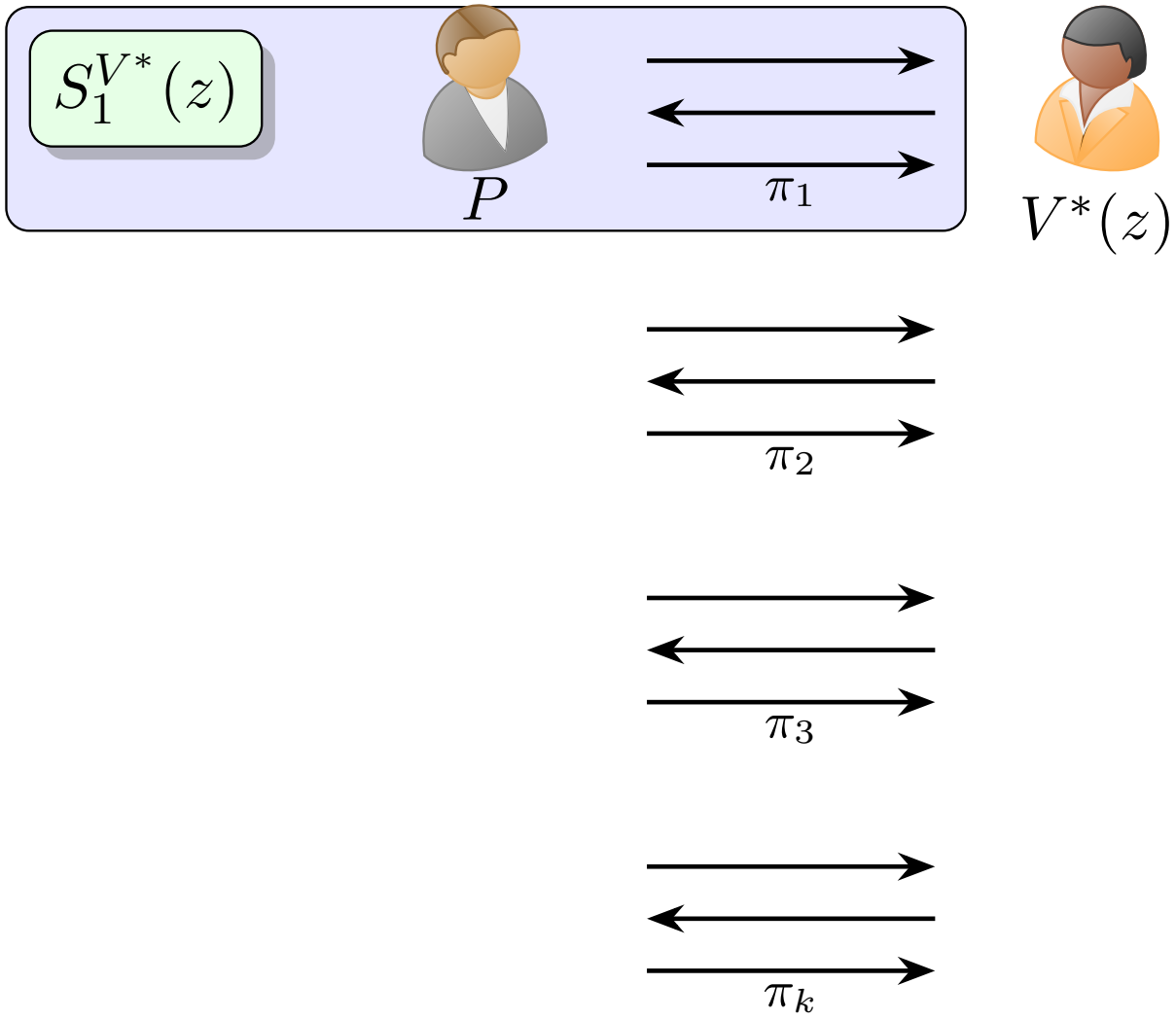
# Simulation Idea for Sequential Repetition



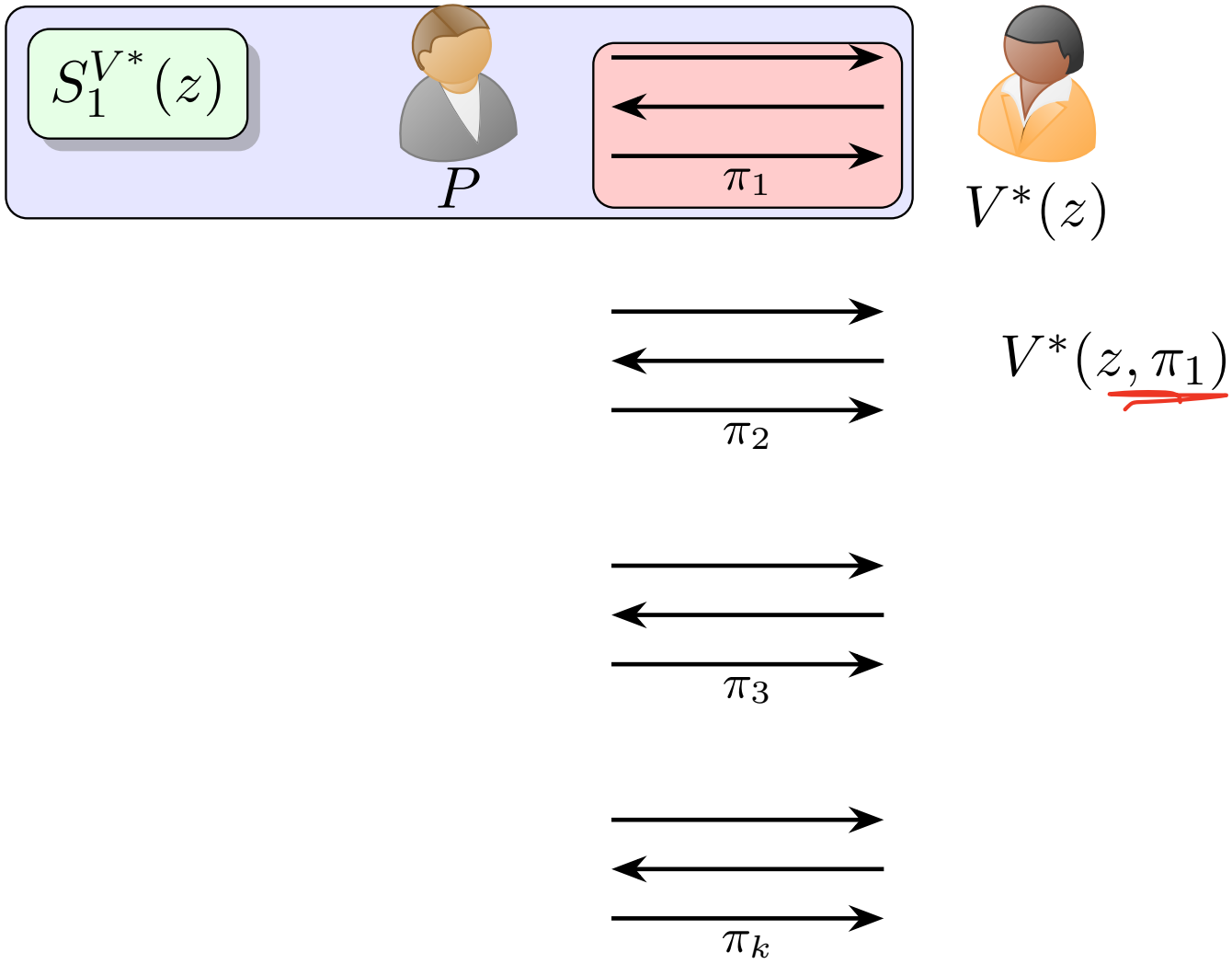
# Simulation Idea for Sequential Repetition



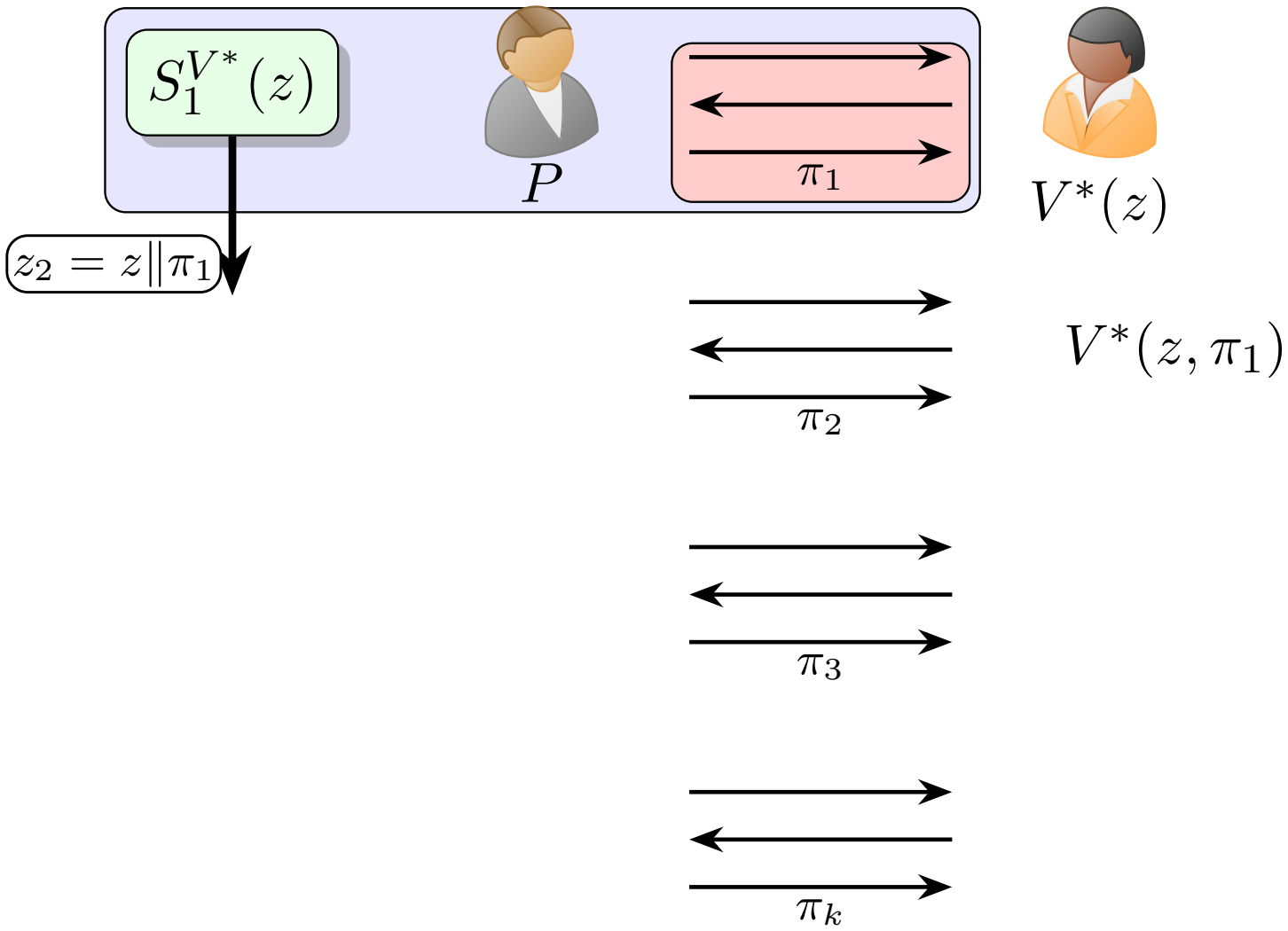
# Simulation Idea for Sequential Repetition



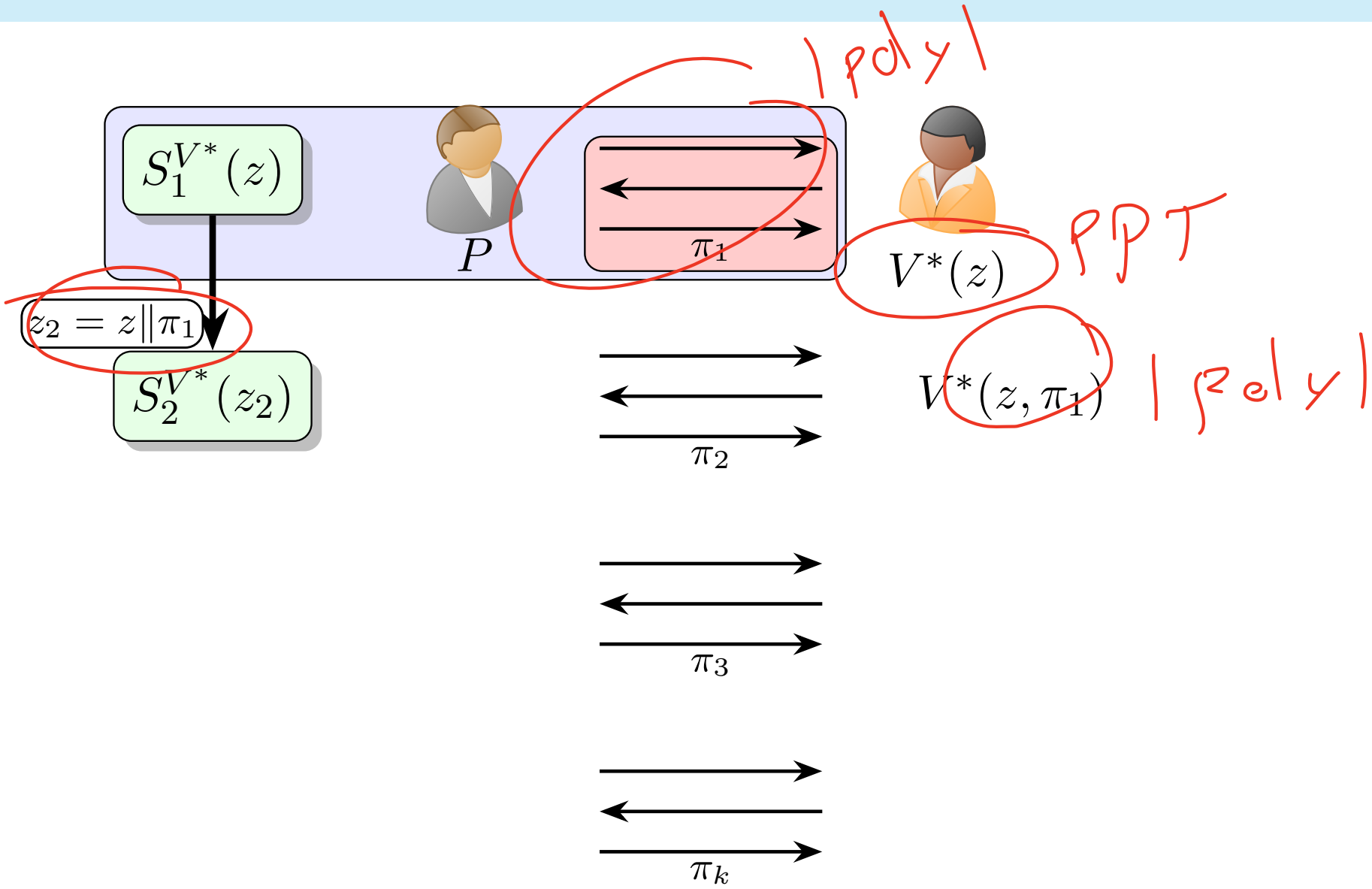
# Simulation Idea for Sequential Repetition



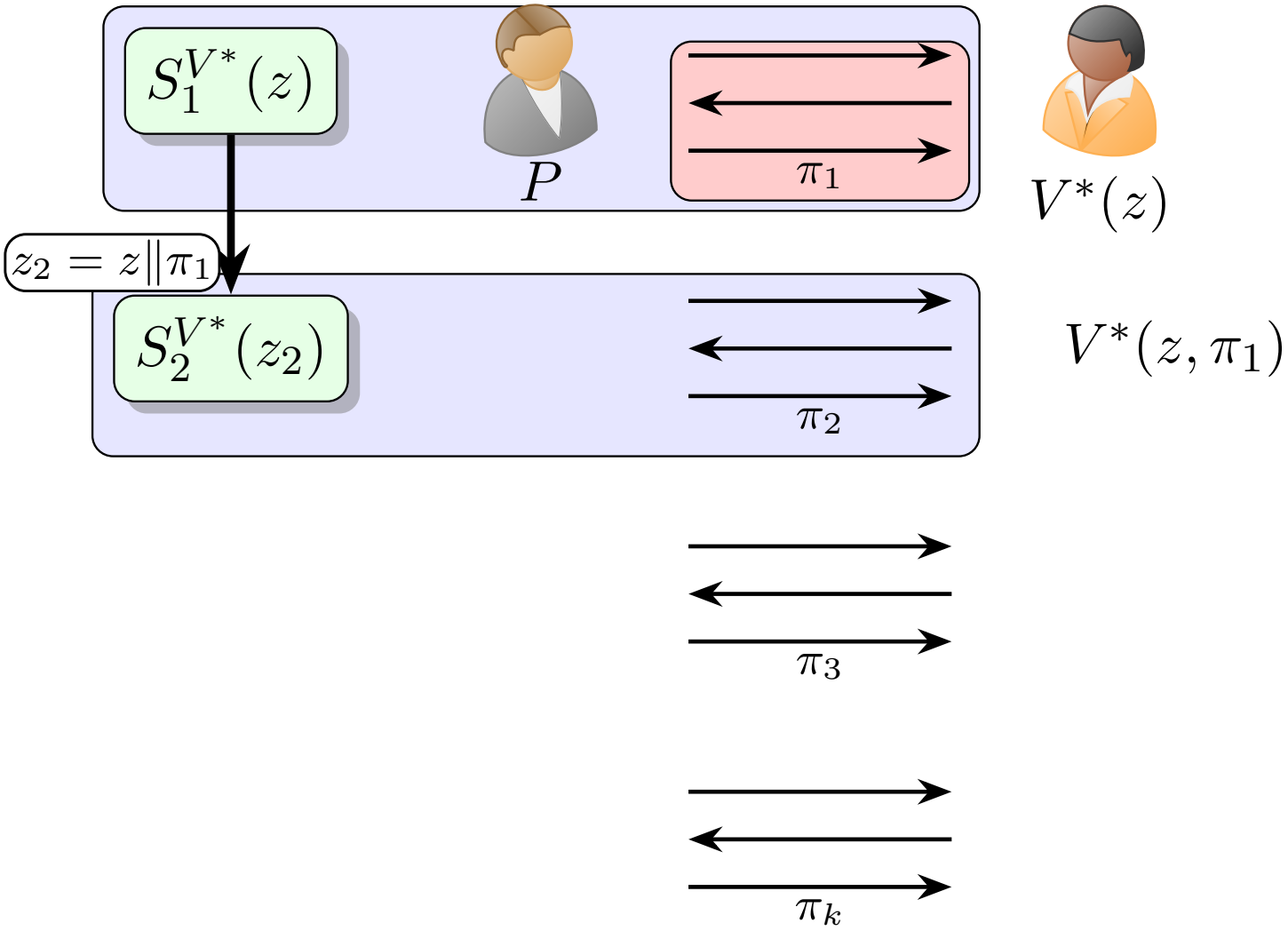
# Simulation Idea for Sequential Repetition



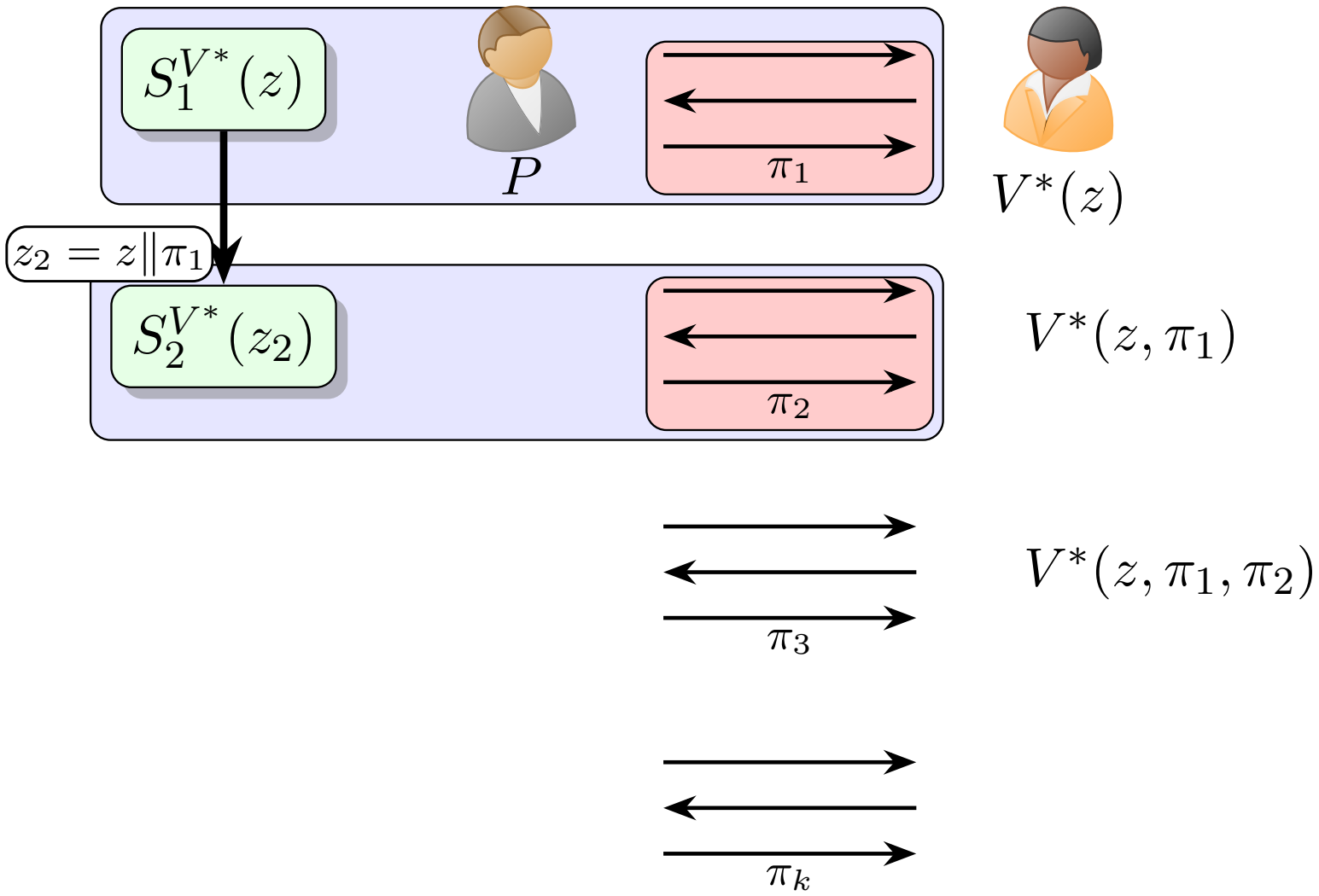
# Simulation Idea for Sequential Repetition



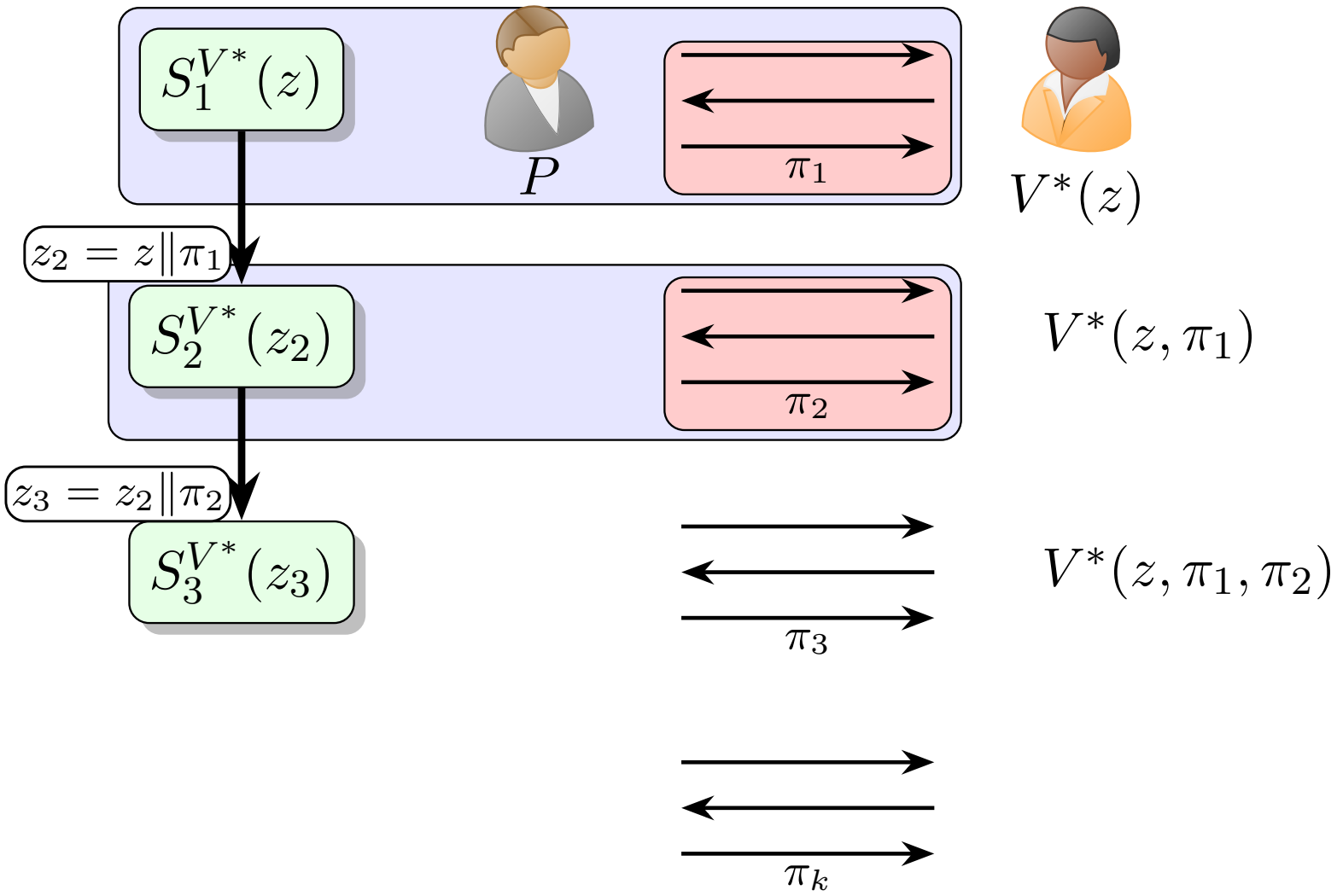
# Simulation Idea for Sequential Repetition



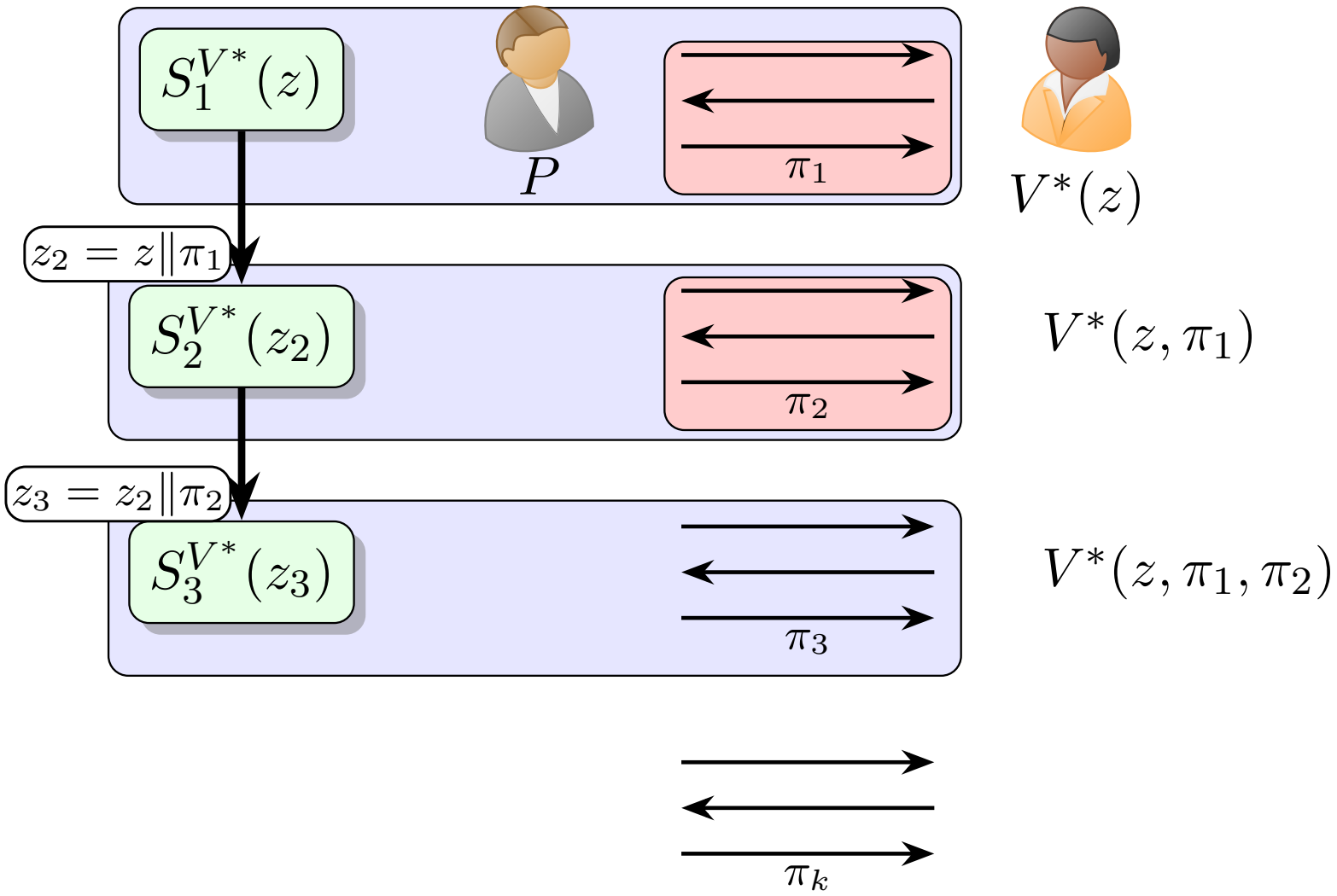
# Simulation Idea for Sequential Repetition



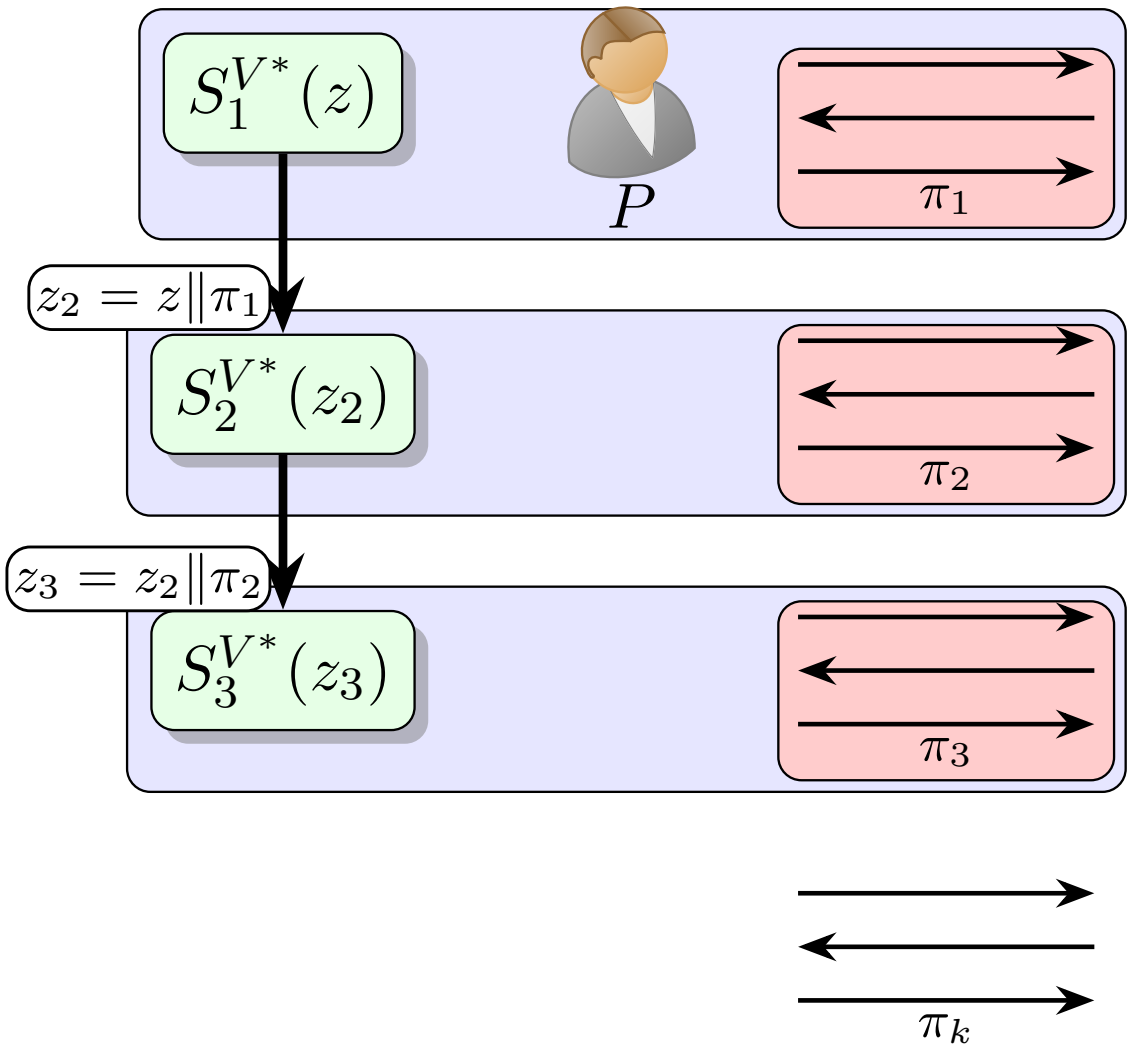
# Simulation Idea for Sequential Repetition



# Simulation Idea for Sequential Repetition



# Simulation Idea for Sequential Repetition



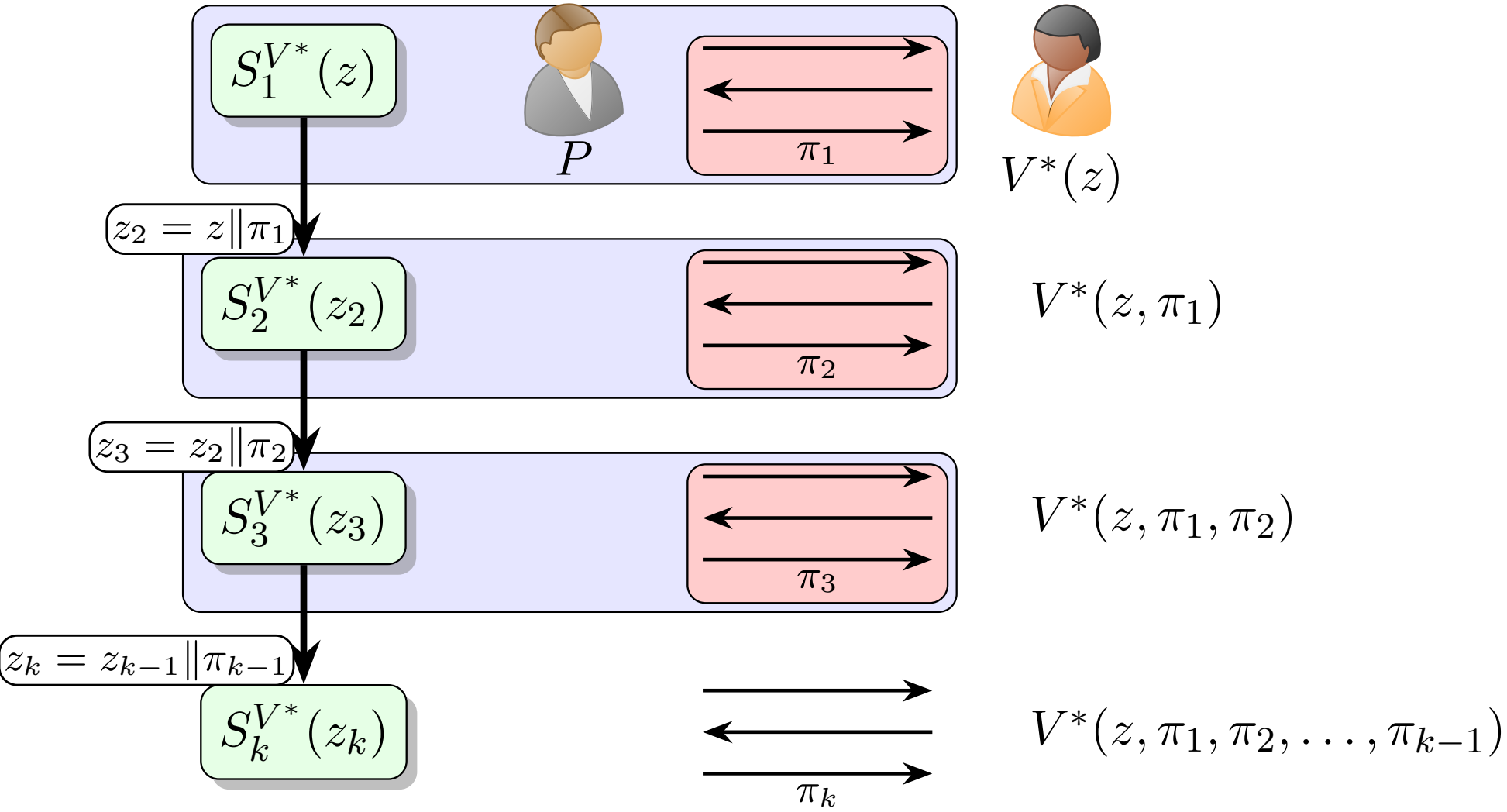

$V^*(z)$

$V^*(z, \pi_1)$

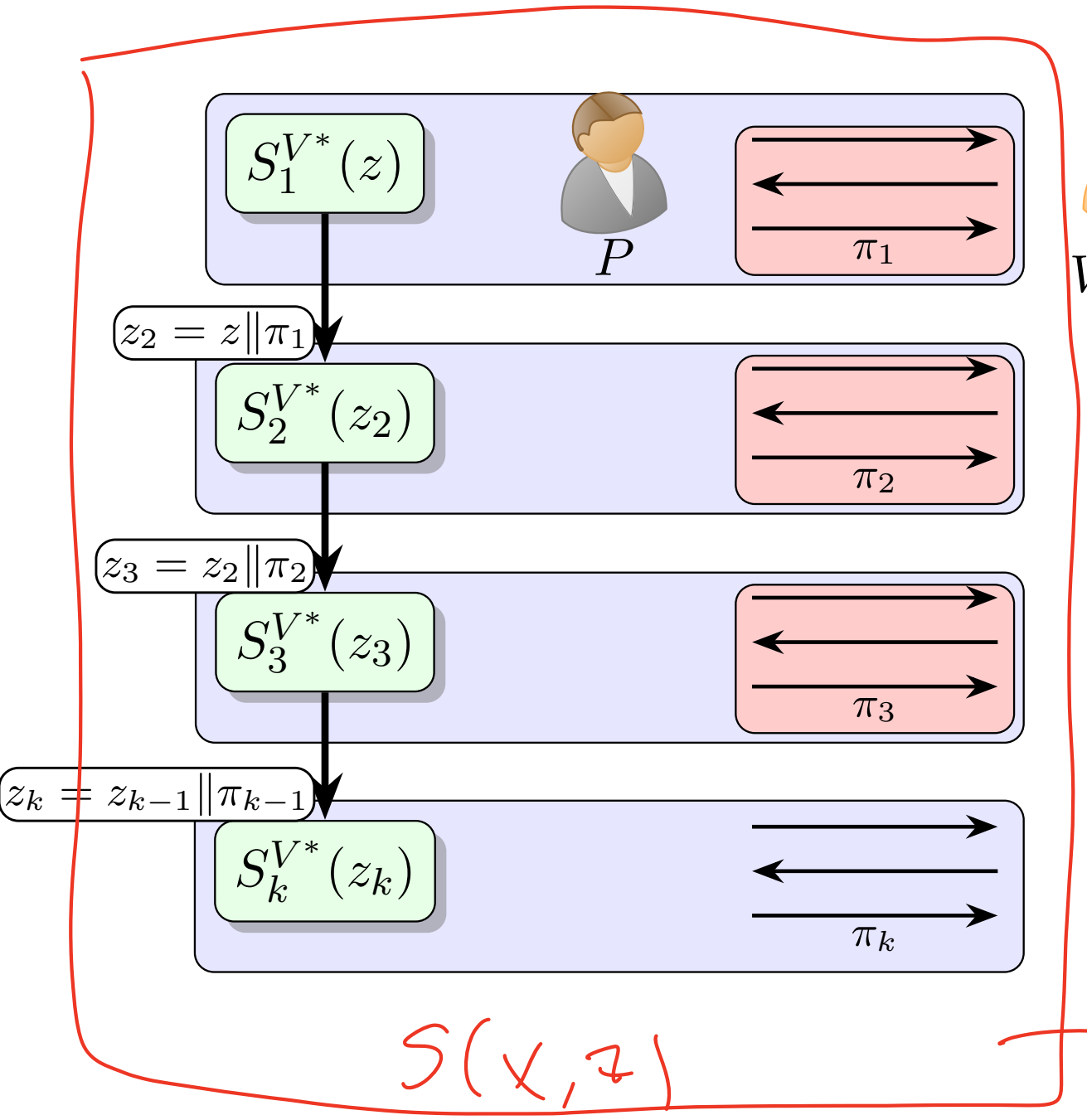
$V^*(z, \pi_1, \pi_2)$

$V^*(z, \pi_1, \pi_2, \ldots, \pi_{k-1})$

# Simulation Idea for Sequential Repetition



# Simulation Idea for Sequential Repetition




$V^*(z)$

$V^*(z, \pi_1)$

$V^*(z, \pi_1, \pi_2)$

$(\pi_1, \pi_2, \ldots, \pi_k)$

$V^*(z, \pi_1, \pi_2, \ldots, \pi_{k-1})$

$\|$

$(\pi_1^*, \pi_2^*, \ldots, \pi_k^*)$

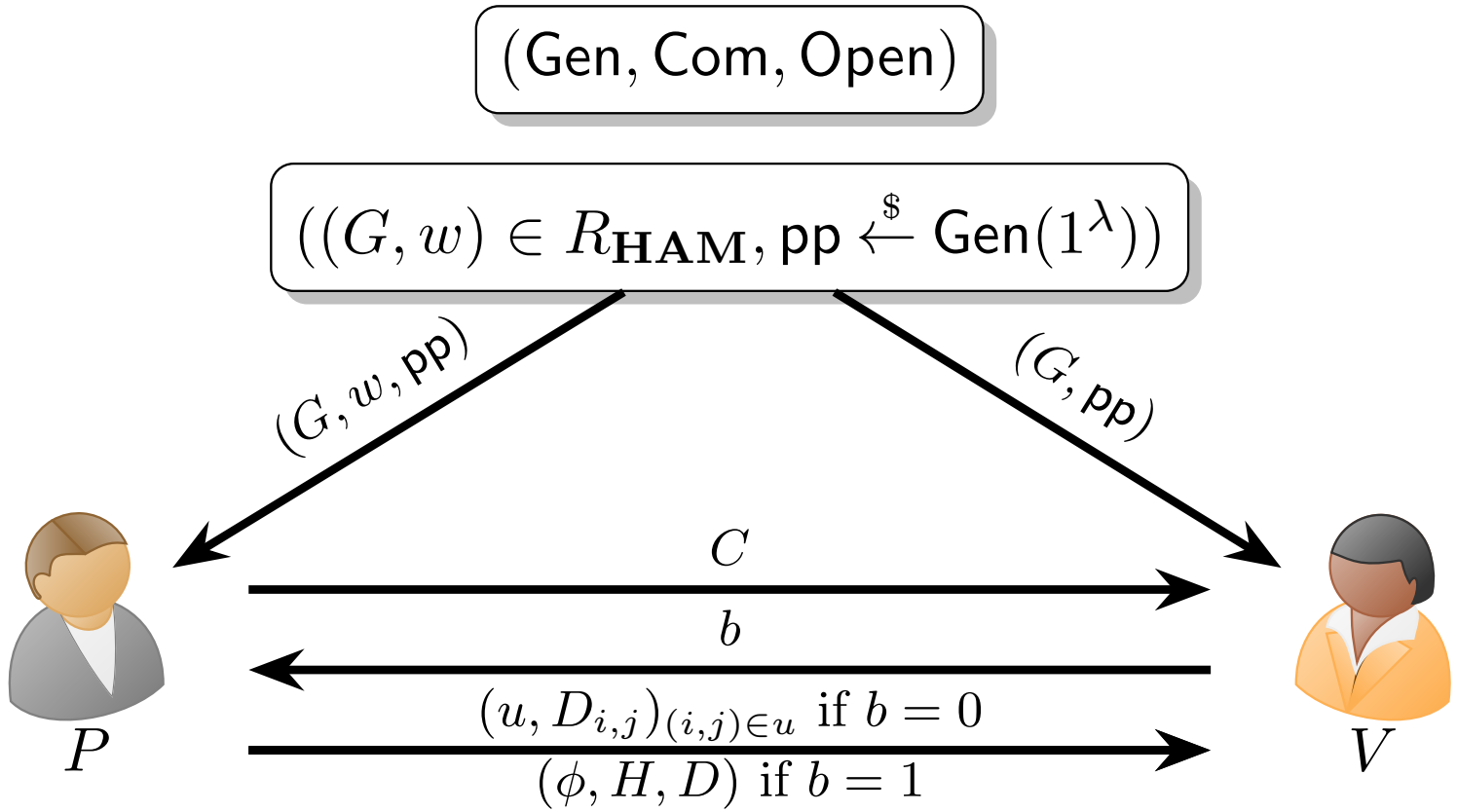
# Constant-round Zero-knowledge for NP

# More ZK for NP?

Goal: proofs for every $L \in \mathbf{NP}$

- in computational ZK;
- with *negligible* soundness error; and
- a constant number of rounds

# Recall: CZK Proof for HAM




(P1) Sample random permutation $\phi$ and compute $(C, D) \leftarrow \mathsf{Com}(\phi(G))$

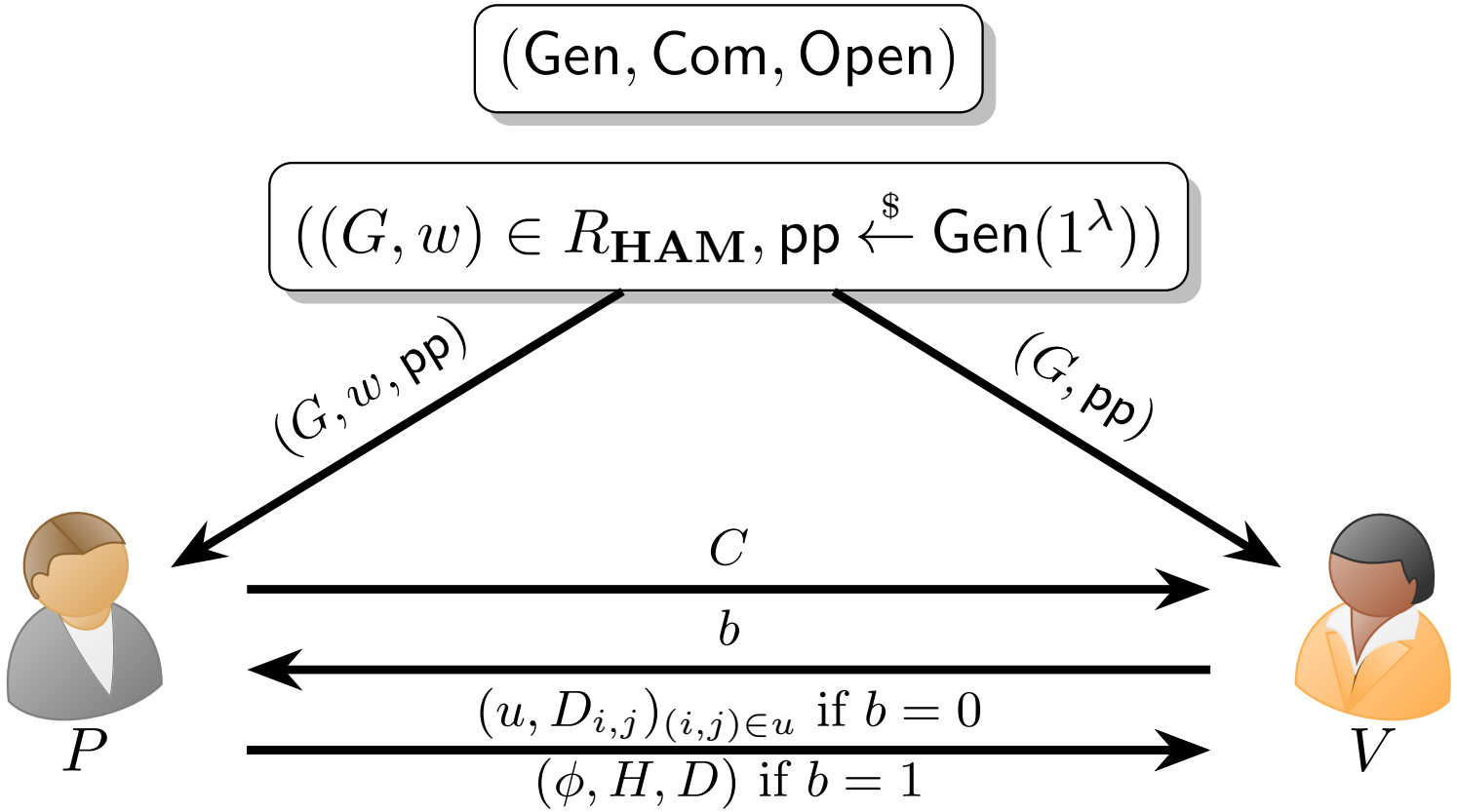
(P2) Set $u = \phi(w)$.

(V1) Sample $b \xleftarrow{\$} \{0, 1\}$.

(V2-0) $b = 0$: check that $u$ is a cycle and $\forall (i, j) \in u$, check $1 = \mathsf{Open}(1, C_{i,j}, D_{i,j})$

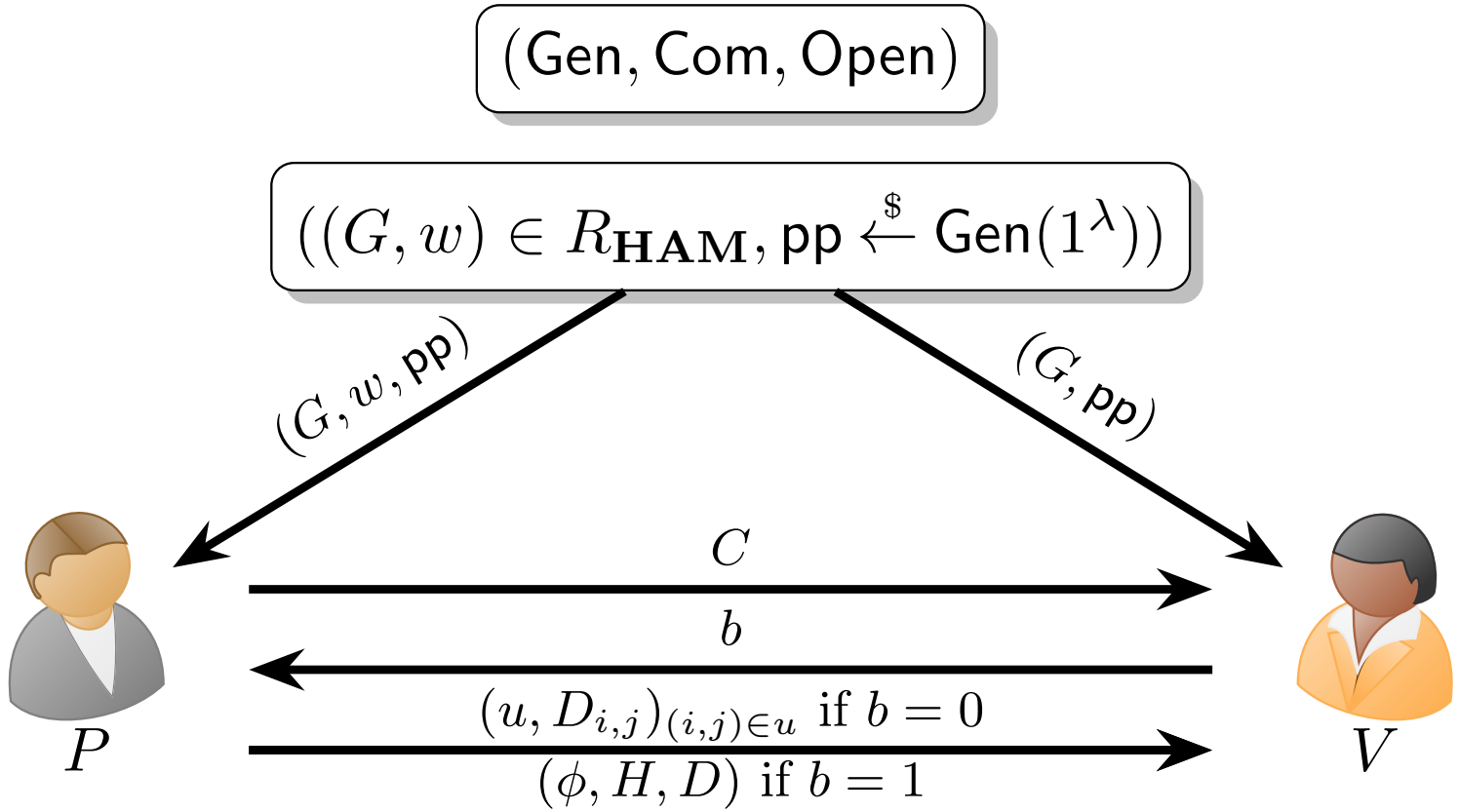
(V2-1) $b = 1$: check that $H = \phi(G)$ and $1 = \mathsf{Open}(H, C, D)$

# Recall: CZK Proof for HAM



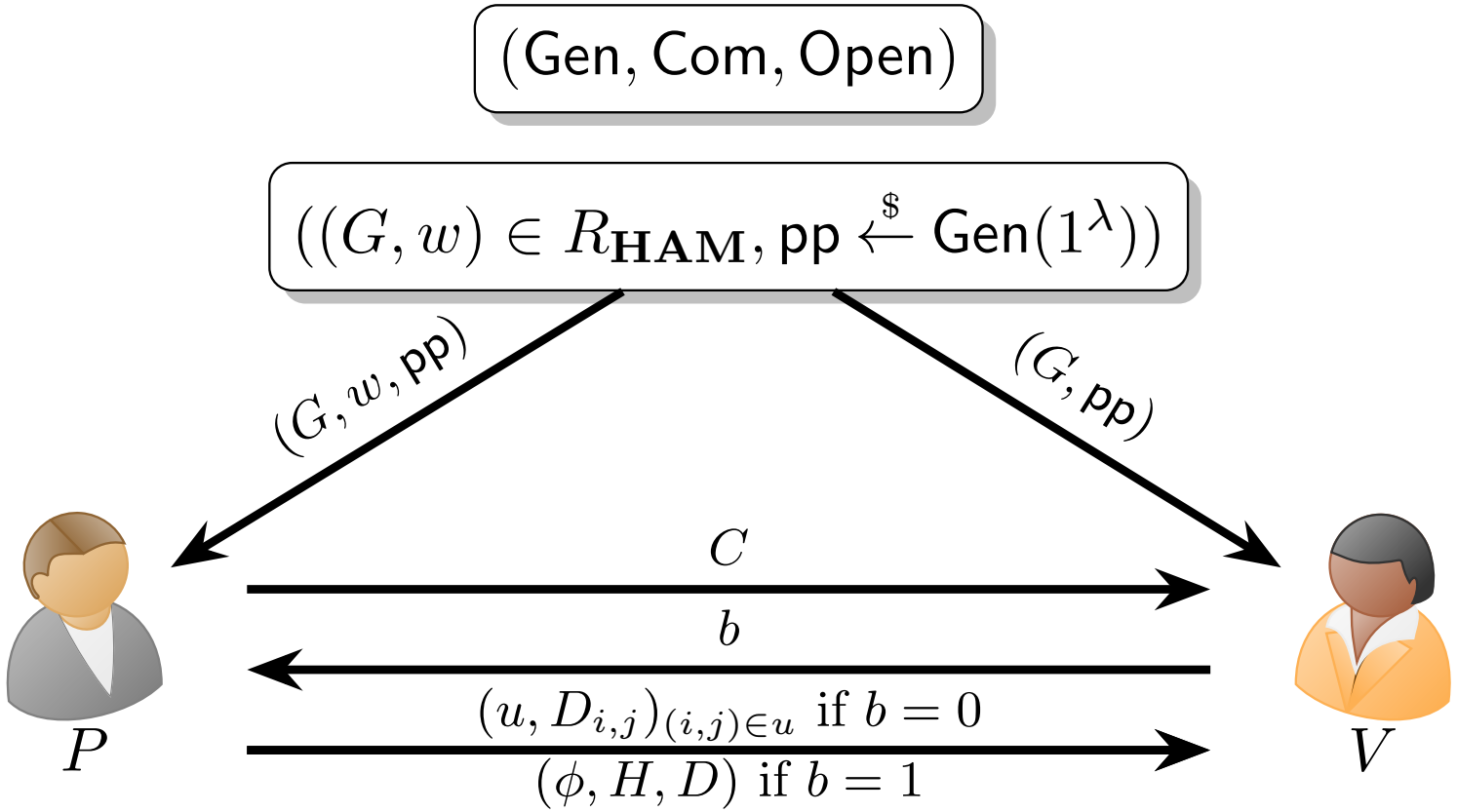

- To get negligible soundness error, sequential repetition for $k = \text{poly}(\lambda)$.

# Recall: CZK Proof for HAM



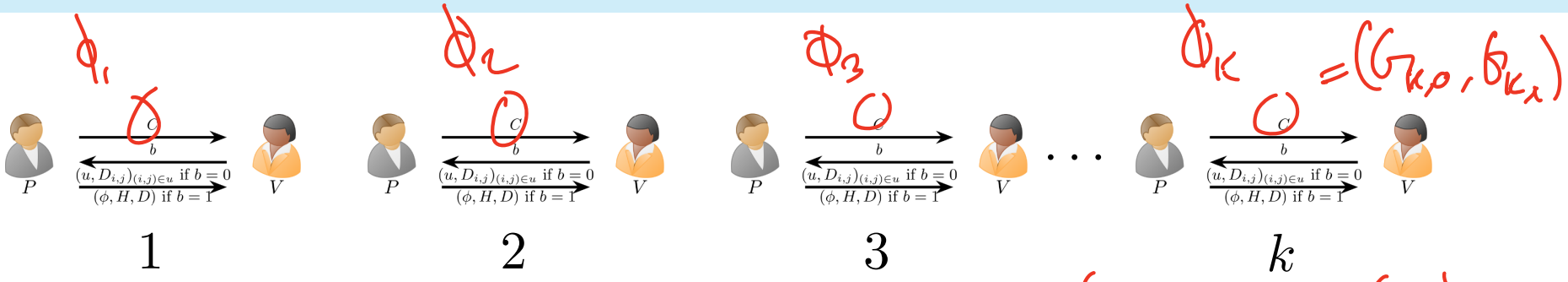

- To get negligible soundness error, sequential repetition for $k = \mathrm{poly}(\lambda)$
- Not constant round!

# Recall: CZK Proof for HAM




- To get negligible soundness error, sequential repetition for $k = \text{poly}(\lambda)$.
- Not constant round!
  - What about parallel repetition?

# Parallel Repetition: Exponential Simulation!

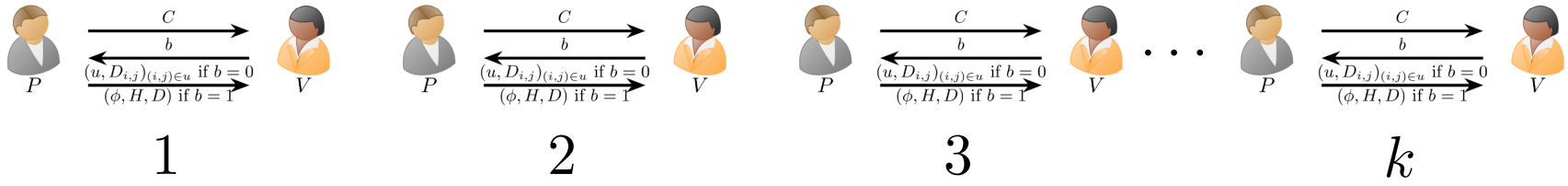



$\vec{b} \in \{0,1\}^k$

$\sigma_{i,0} = \phi_i(\omega)$

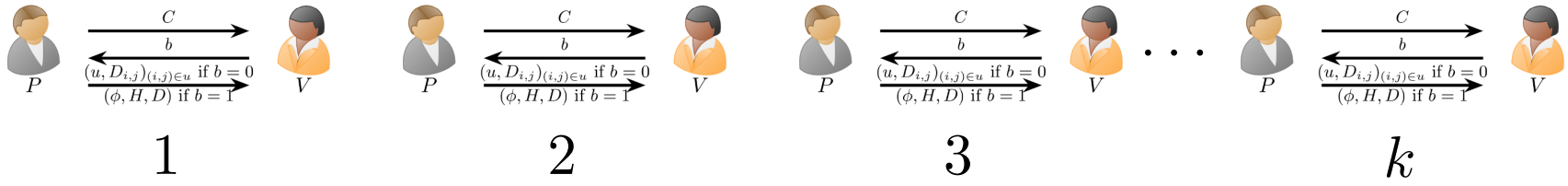
$\sigma_{i,1} = \phi_i(\sigma)$

# Parallel Repetition: Exponential Simulation!



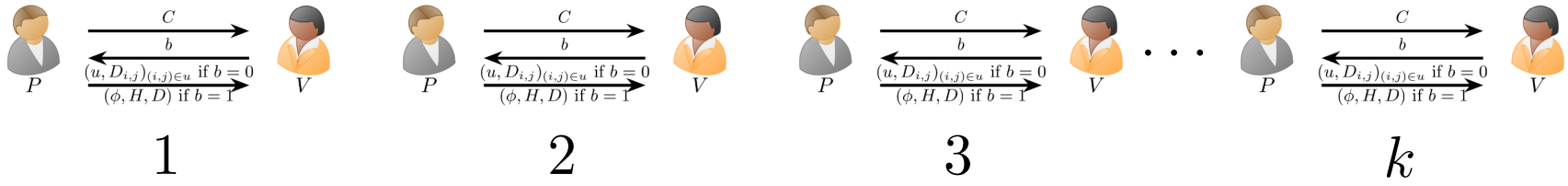

✓ Soundness error $2^{-k}$!

# Parallel Repetition: Exponential Simulation!



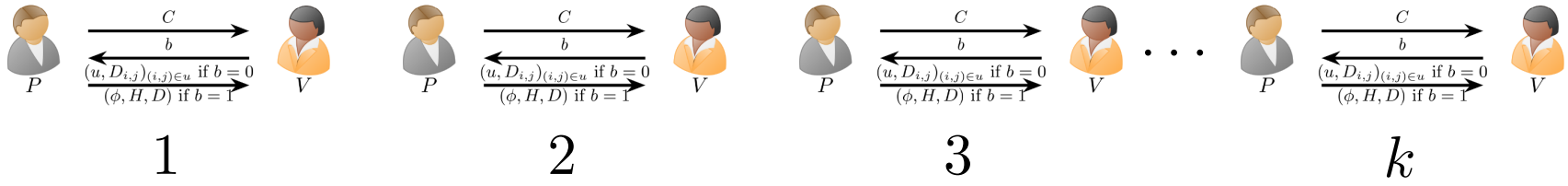

- ✓ Soundness error $2^{-k}$!
- ✓ Constant number of rounds! (3 rounds).

# Parallel Repetition: Exponential Simulation!



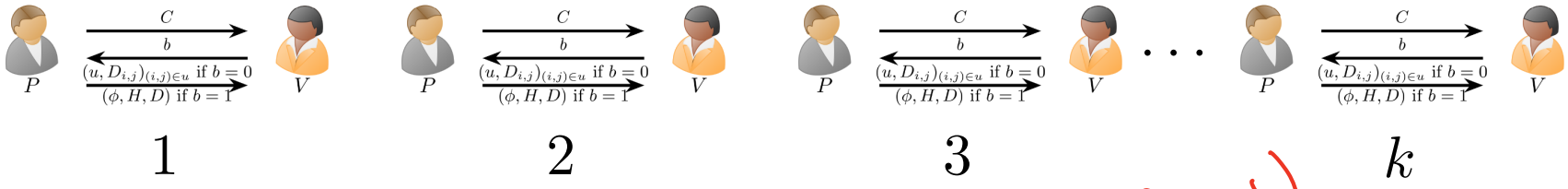

- ✓ Soundness error $2^{-k}$!
- ✓ Constant number of rounds! (3 rounds).
- × Can't simulate in expected polynomial time!

# Parallel Repetition: Exponential Simulation!




- ✓ Soundness error $2^{-k}$!
- ✓ Constant number of rounds! (3 rounds).
- × Can't simulate in expected polynomial time!
  - In original simulation, $S$ guessed the bit $b$ from $V^*$.

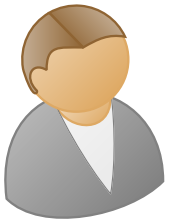
# Parallel Repetition: Exponential Simulation!




poly(|x|)
↑
$\in L/\bar{L}$

- ✓ Soundness error $2^{-k}$!
- ✓ Constant number of rounds! (3 rounds).
- × Can't simulate in expected polynomial time!
  - In original simulation, $S$ guessed the bit $b$ from $V^*$.
  - In parallel repetition, $S$ must guess $\mathbf{b} \in \{0,1\}^k$, which gives $2^k$ expected number of simulation attempts.

# Fix (with no proof)

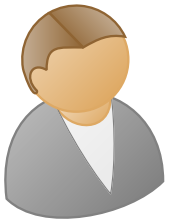


$P$

$V$

# Fix (with no proof)

- Simple fix: add 2 more rounds!

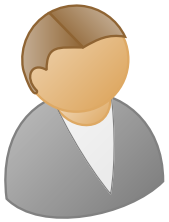


$P$

$V$

# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.



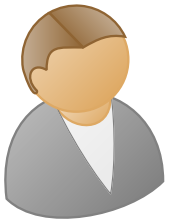
$P$

$V$

# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.

$(\mathsf{Com}_1, \mathsf{Com}_2)$



$P$

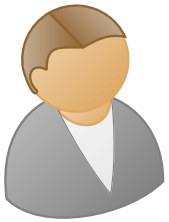
$V$

# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.

$$(\mathsf{Com}_1, \mathsf{Com}_2)$$

$$((G, w) \in R_{\mathbf{HAM}}, \mathsf{pp}_{1,2} \xleftarrow{\$} \mathsf{Gen}_{1,2}(1^\lambda))$$



$P$

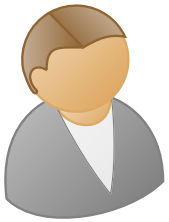
$V$

# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.

$$(\mathsf{Com}_1, \mathsf{Com}_2)$$

$$((G, w) \in R_{\mathbf{HAM}}, \mathsf{pp}_{1,2} \xleftarrow{\$} \mathsf{Gen}_{1,2}(1^\lambda))$$

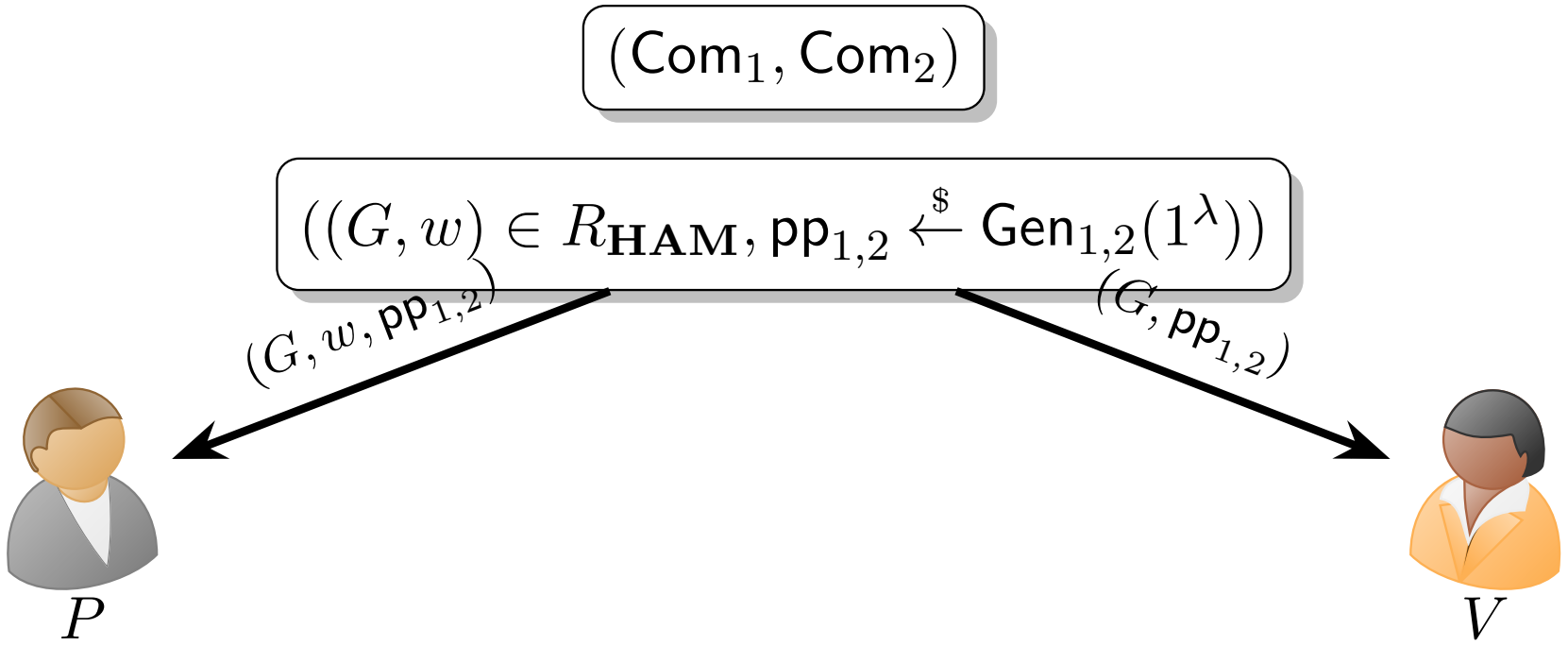


$P$

$V$

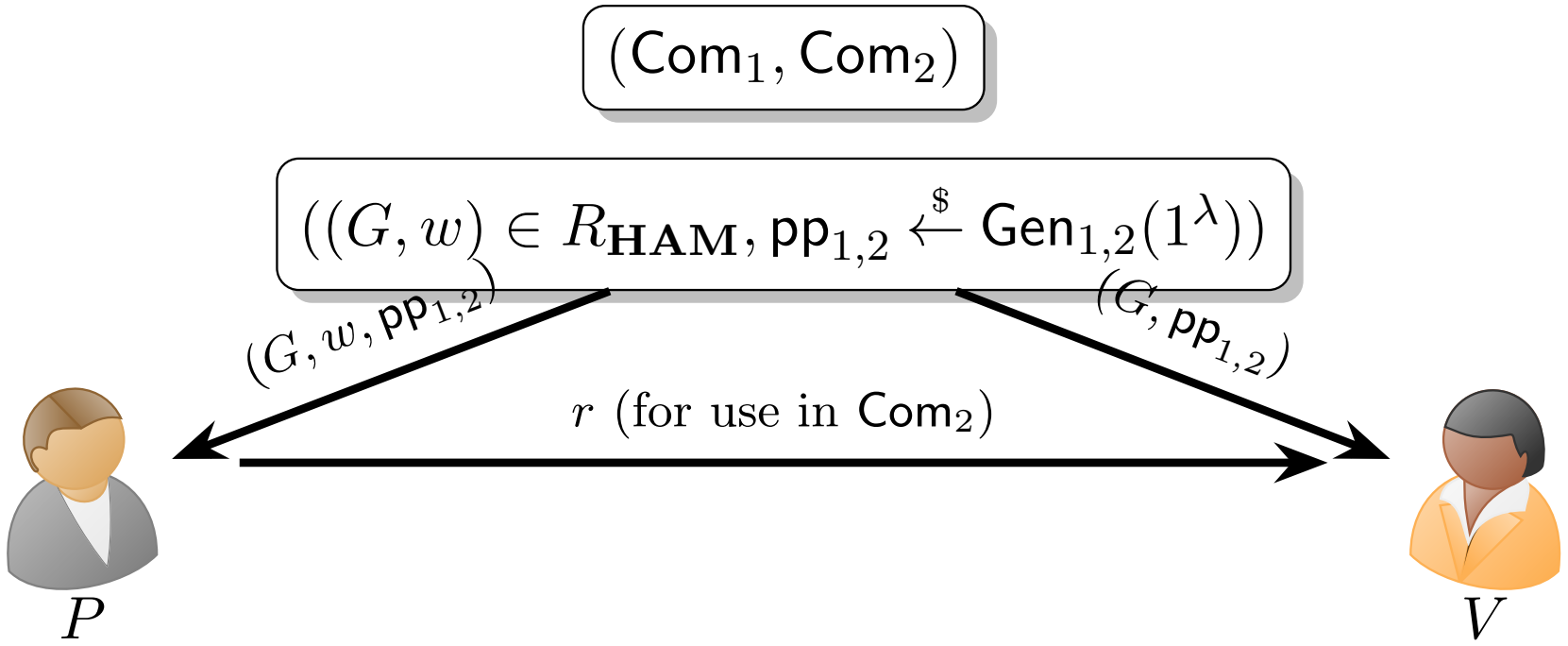
# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.



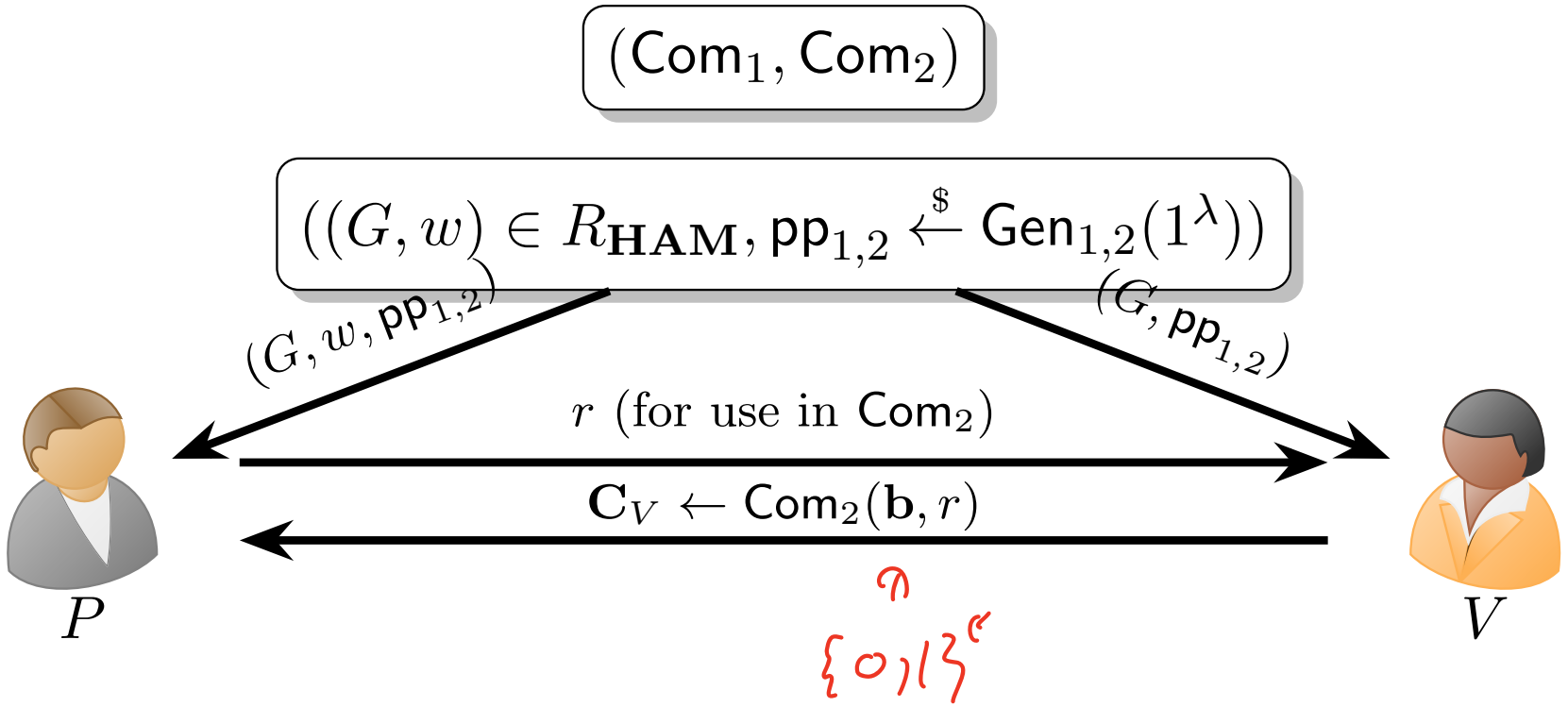
# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.



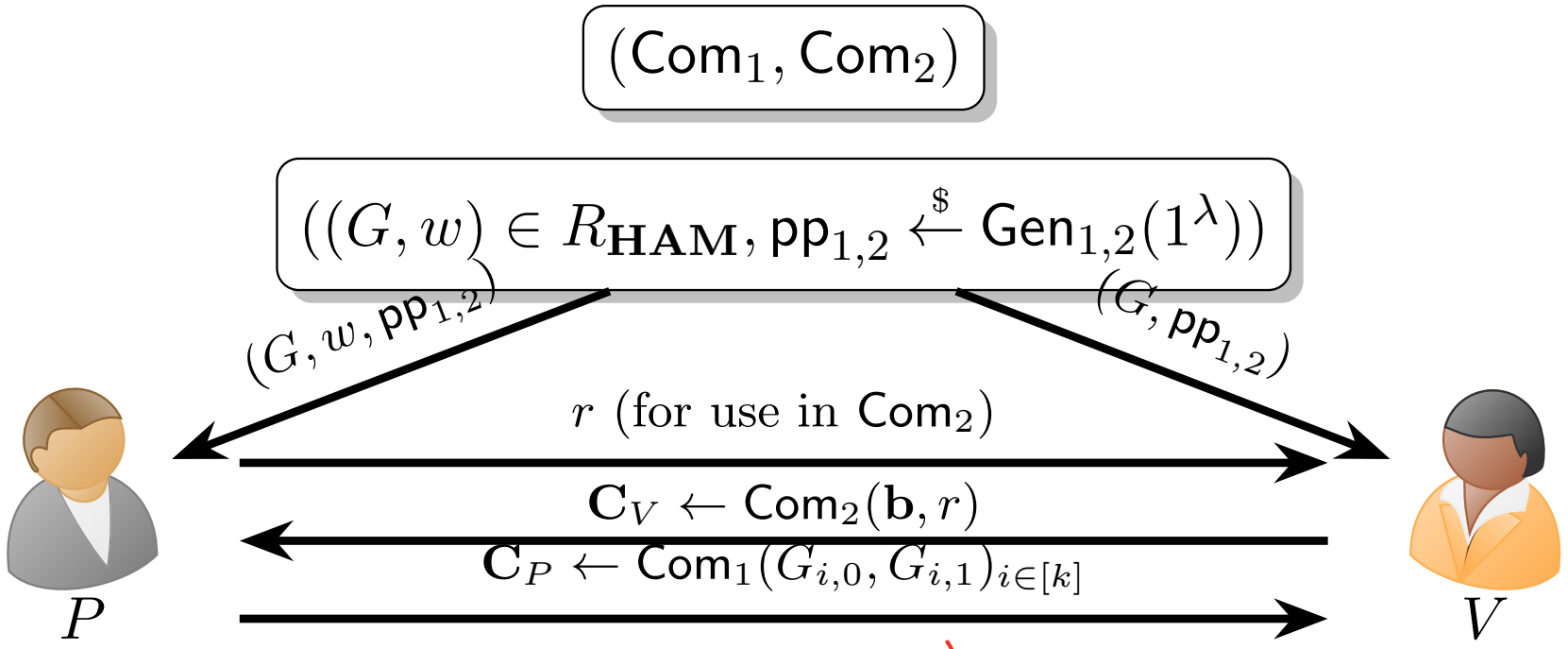
# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.



# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.

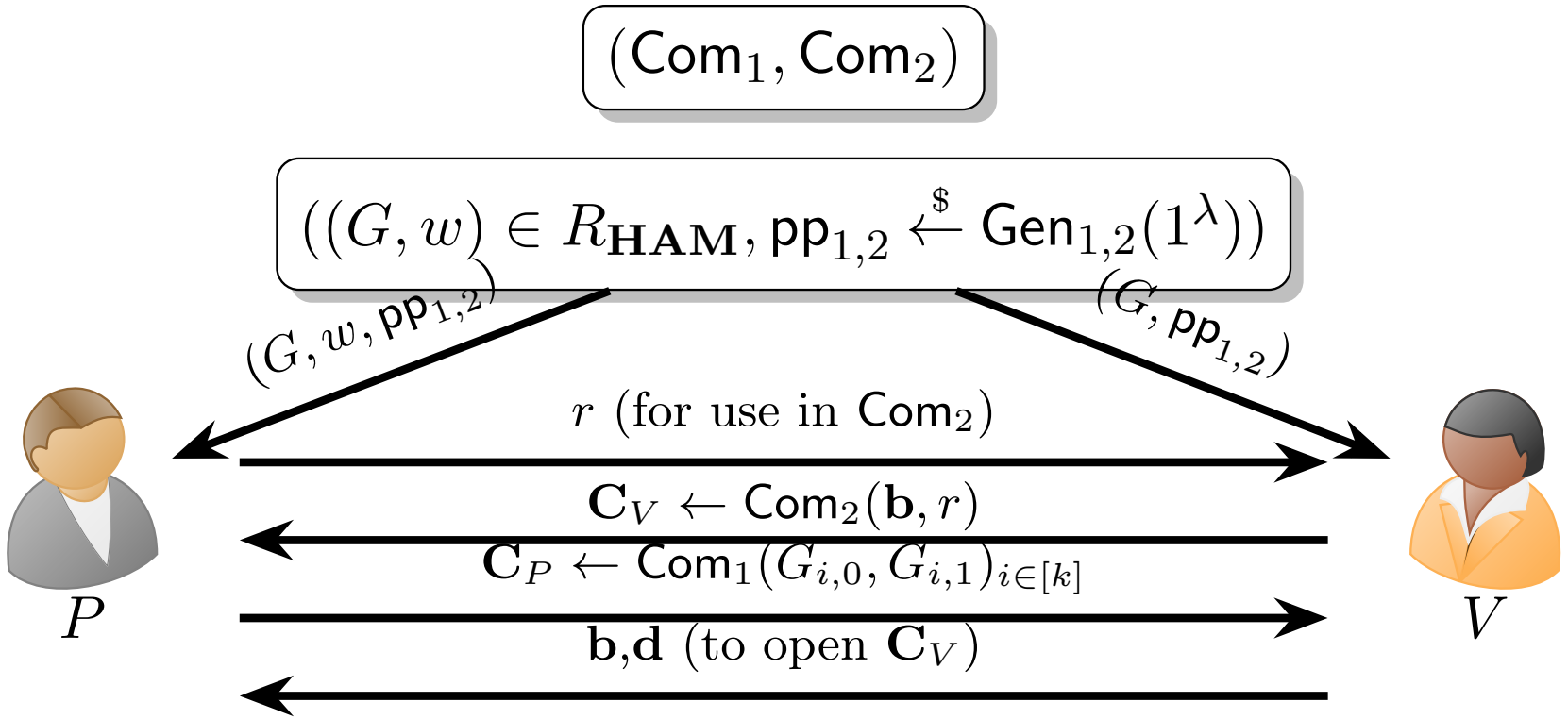



$G_{i,0} = \phi_i(w)$

$G_{i,1} = \pi(G_{i,0})$

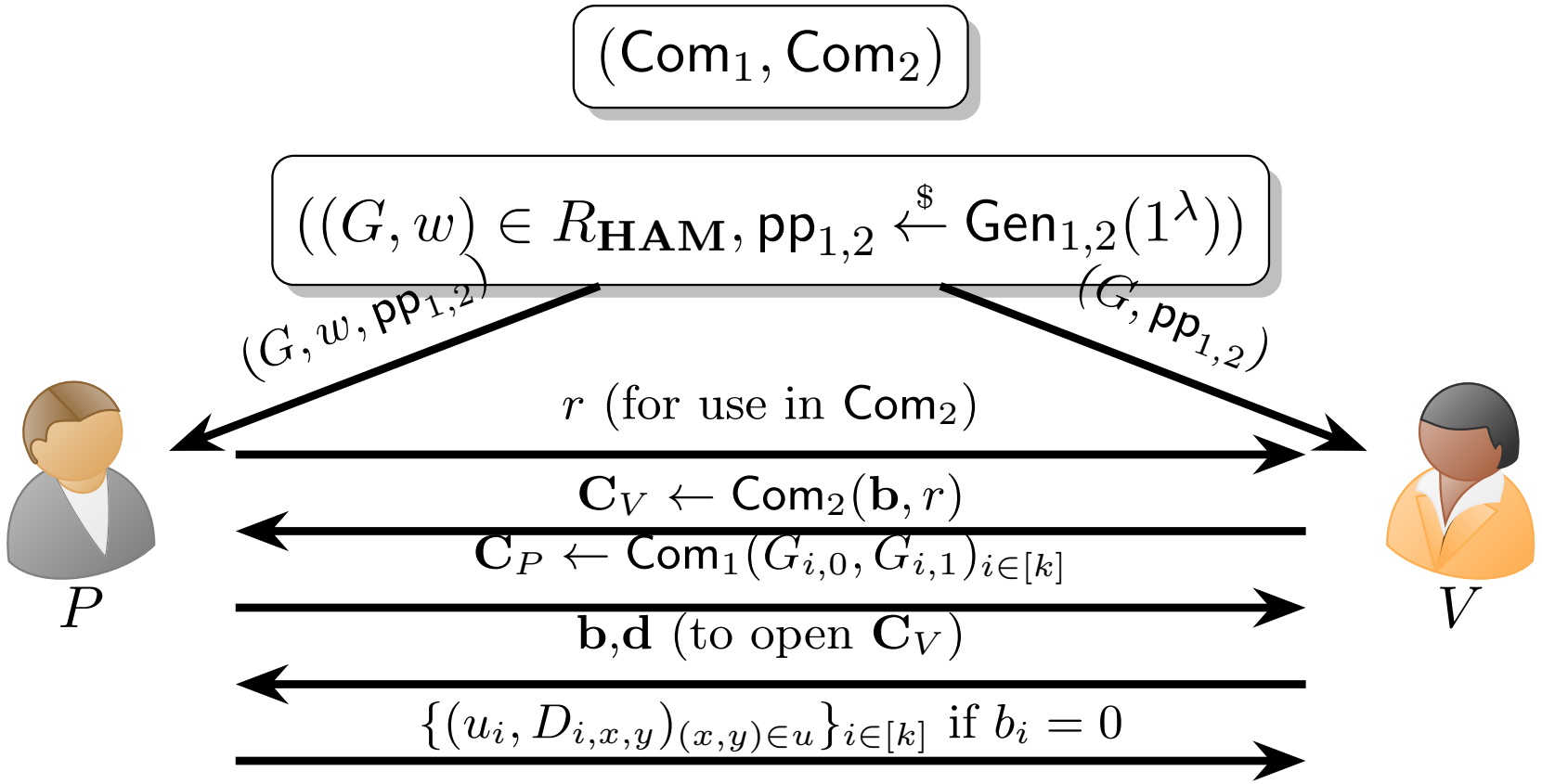
# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.



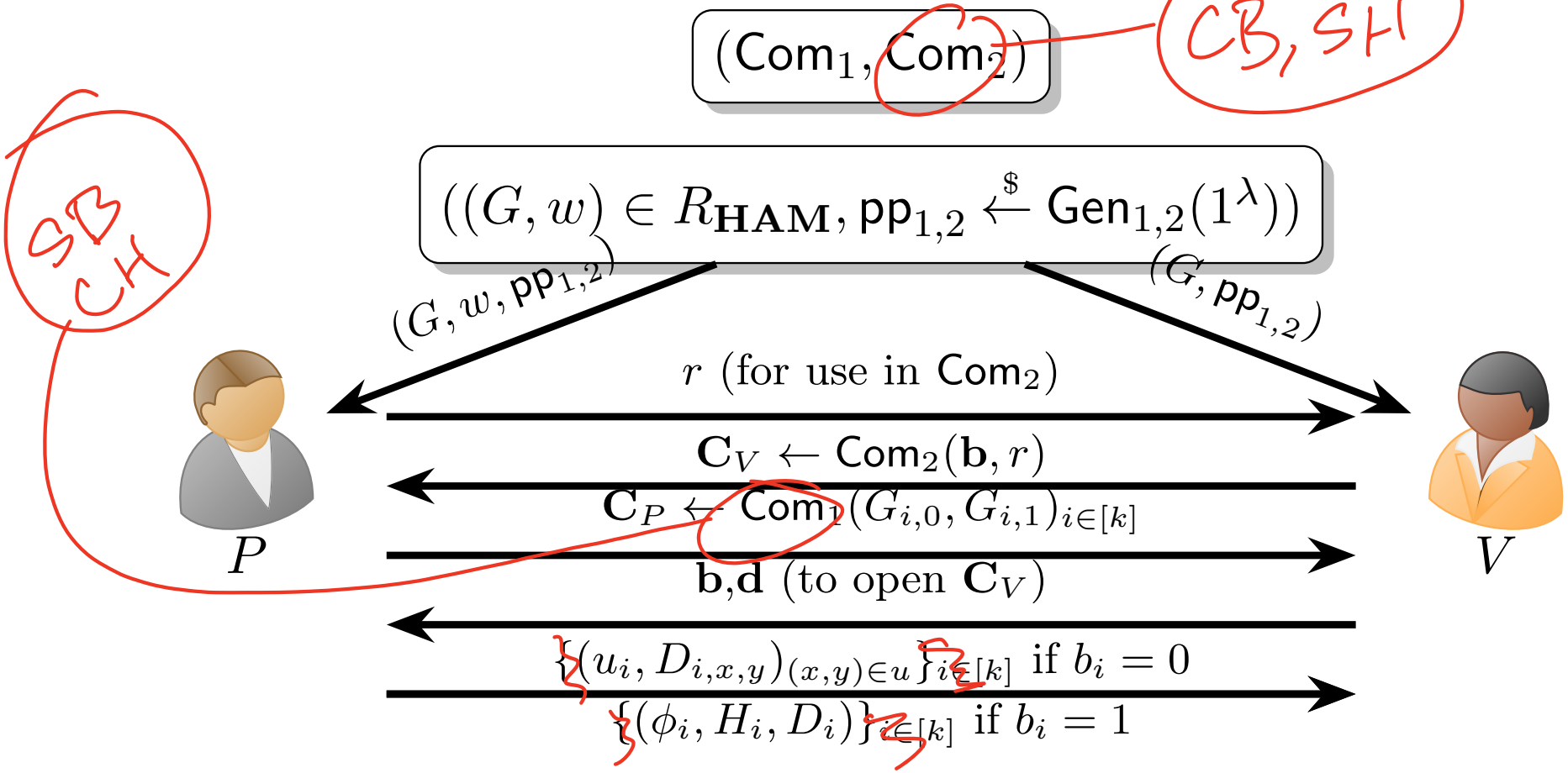
# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.



# Fix (with no proof)

- Simple fix: add 2 more rounds!
- Have $V$ *commit* to $\mathbf{b}$ ahead of time, then later open this commitment.
- $\mathsf{Com}_2$ should be *statistically hiding* and *computationally binding*.



# Final Notes on Zero-knowledge

# Final Notes on Zero-knowledge

- Above protocol yields the following theorem.

# Final Notes on Zero-knowledge

- Above protocol yields the following theorem.

**Theorem 1**

*If statistically-hiding commitments and statistically-binding commitments exist, then every $L \in \mathbf{NP}$ has a (computational) 5-round ZK proof with soundness error $2^{-k}$.*

# Final Notes on Zero-knowledge

- Above protocol yields the following theorem.

**Theorem 1**

*If statistically-hiding commitments and statistically-binding commitments exist, then every $L \in \mathbf{NP}$ has a (computational) 5-round ZK proof with soundness error $2^{-k}$.*

- This is optimal, at it is known that if any language $L$ has a 4-round ZK proof, then $L \in \mathbf{coMA}$.

# Final Notes on Zero-knowledge

- Above protocol yields the following theorem.

**Theorem 1**

*If statistically-hiding commitments and statistically-binding commitments exist, then every $L \in \mathbf{NP}$ has a (computational) 5-round ZK proof with soundness error $2^{-k}$.*

- This is optimal, at it is known that if any language $L$ has a 4-round ZK proof, then $L \in \mathbf{coMA}$.

- The proof that the previous protocol has CZK is very subtle!

# Final Notes on Zero-knowledge

- Above protocol yields the following theorem.

**Theorem 1**

*If statistically-hiding commitments and statistically-binding commitments exist, then every $L \in \mathbf{NP}$ has a (computational) 5-round ZK proof with soundness error $2^{-k}$.*

- This is optimal, at it is known that if any language $L$ has a 4-round ZK proof, then $L \in \mathbf{coMA}$.

- The proof that the previous protocol has CZK is very subtle!
  - See https://www.youtube.com/watch?v=cAI7Iw_bkZs&list=PL8Vt-7cSFnw29cLUVqAIuMlg1QJ-szVOK&index=4 for more details.

# Final Notes on Zero-knowledge

- Above protocol yields the following theorem.

**Theorem 1**

*If statistically-hiding commitments and statistically-binding commitments exist, then every $L \in \mathbf{NP}$ has a (computational) 5-round ZK proof with soundness error $2^{-k}$.*

- This is optimal, at it is known that if any language $L$ has a 4-round ZK proof, then $L \in \mathbf{coMA}$.

- The proof that the previous protocol has CZK is very subtle!
  - See https://www.youtube.com/watch?v=cAI7Iw_bkZs&list=PL8Vt-7cSFnw29cLUVqAIuMlg1QJ-szVOK&index=4 for more details.

- As of 2019, the GM**W** result showing **NP** $\subseteq$ **CZK** is **W**idgerson's favorite result by far.

# Next Time: Secure Multi-party Computation