Proof Systems

Claim: $x \in L$



$P(x, w)$





$V(x)$

Output: $b \in \{0,1\}$

Definition: a proof system for a language $L \subset \{0,1\}^*$ consists of a pair of interactive algorithms (P, V) run on an input $x \in \{0,1\}^n$

- V will always be a $\mathbf{poly}(n)$-time algorithm
- P can be inefficient, or $\mathbf{poly}(n)$-time given a witness w

It must satisfy the following properties:

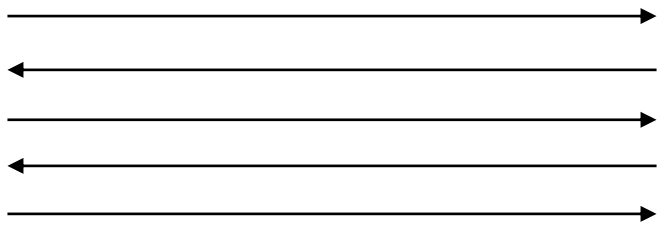
1. **Completeness**: if $x \in L$ and the two parties run (P, V), then V outputs 1 (accept)
2. **Soundness**: if $x \notin L$, then for any strategy P^*, V accepts with probability $\mathbf{negl}(n)$.

Honest-Verifier Zero Knowledge

Claim: $x \in L$



$P(x, w)$





$V(x)$

Output: $b \in \{0,1\}$

Definition: a proof system (P, V) for a language L is honest-verifier zero knowledge if there exists a polynomial-time simulator

$$\mathrm{Sim}(x) \to \tau$$

such that for all $x \in L$ and all valid w,

$$\mathrm{Sim}(x) \approx_c \mathrm{view}_V(x, w)$$

Simulation paradigm: everything V learned, V could have generated on their own!

Malicious-Verifier Zero Knowledge

Claim: $x \in L$



$P(x, w)$





$V(x)$

Output: $b \in \{0,1\}$

Definition: a proof system (P, V) for a language L is malicious verifier zero knowledge if for all polynomial-time V^*, there exists a polynomial-time simulator

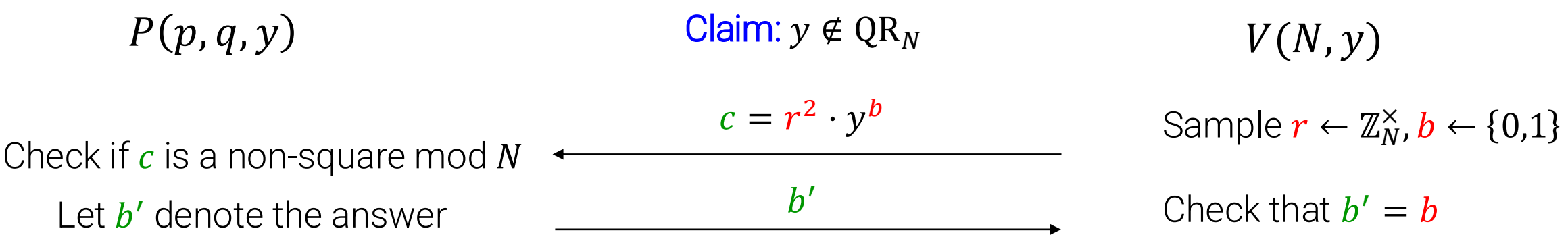
$$\mathrm{Sim}^*(x) \to \tau$$

such that for all $x \in L$ and all valid w,

$$\mathrm{Sim}^*(x) \approx_c \mathrm{view}_{V^*}(x, w)$$

Simulation paradigm: everything V^* learned, V^* could have generated on their own!

Proof Systems: Quadratic Non-Residuosity



Is this protocol malicious-verifier ZK?

- Intuition: malicious V^* could send c to be any number mod N, where V^* does not know if c is a square modulo N
- Honest P will tell V^* this information, so V^* has learned something

Proof Systems: Quadratic Non-Residuosity

$P(p, q, y)$

Claim: $y \notin \mathrm{QR}_N$

$V(N, y)$

$c = r^2 \cdot y^b$

Check if c is a non-square mod N

Let b' denote the answer

b'

Sample $r \leftarrow \mathbb{Z}_N^\times, b \leftarrow \{0,1\}$

Check that $b' = b$

Magic: there's a way to change this protocol, to make it fully ZK!

A little too complicated; we will switch to a different example.

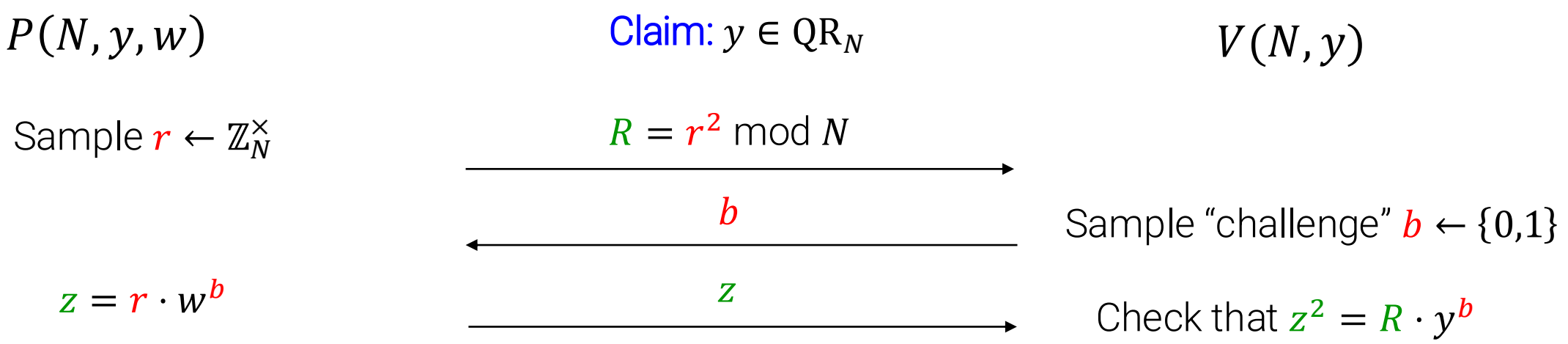
Zero-Knowledge Proof for QR

$P(N, y, w)$ Claim: $y \in \mathrm{QR}_N$ $V(N, y)$

Example: quadratic residuosity

- $L_{\mathbf{QR}} = \{(N, y): y \text{ is a square mod } N\}$
- Witness w: $w^2 \equiv y \bmod N$

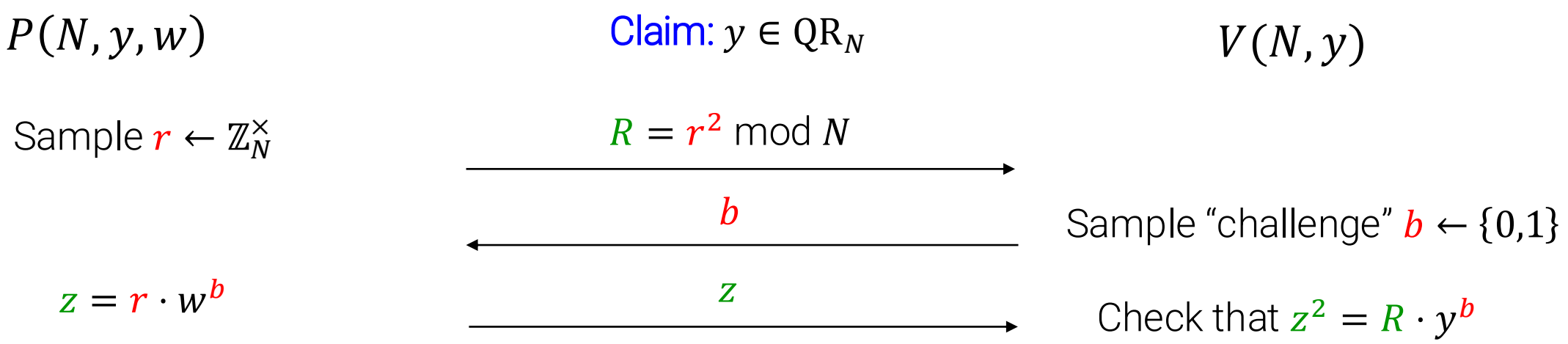
Zero-Knowledge Proof for QR



Intuition: "I'll show you that I could send you a square root if I wanted… but I won't!"

"I know a square root of R and a square root of $R \cdot y$, but you only get to see one"

Zero-Knowledge Proof for QR

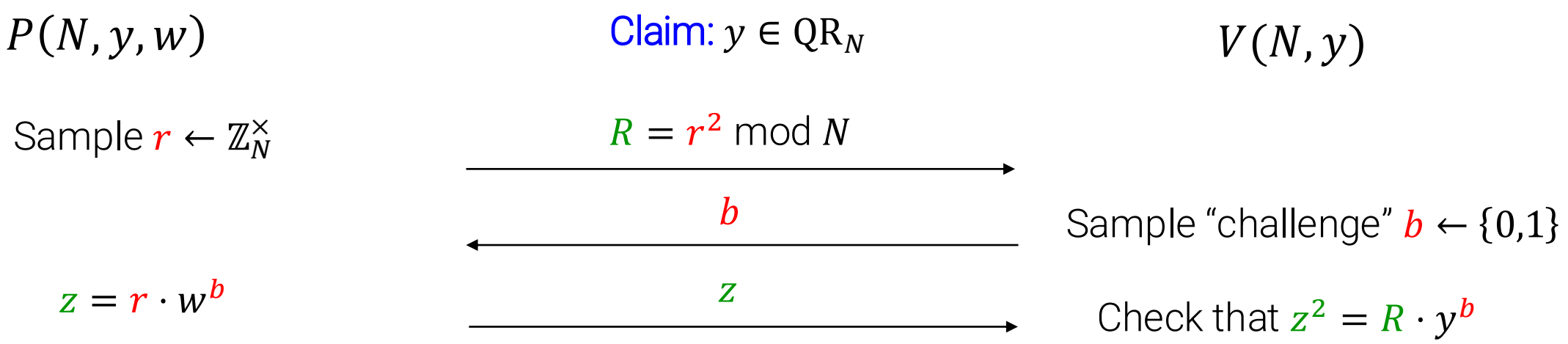


Completeness: $\left(r \cdot w^b\right)^2 \equiv R \cdot y^b$

Soundness: if y is not a square, then impossible to have answers to both challenges!

- For any $R \in \mathbb{Z}_N^\times$, either R is not a square or $R \cdot y$ is not a square
- For any P^*, V rejects with probability at least ½
- Repeat the protocol (sequentially!) to improve the soundness error.

Zero-Knowledge Proof for QR



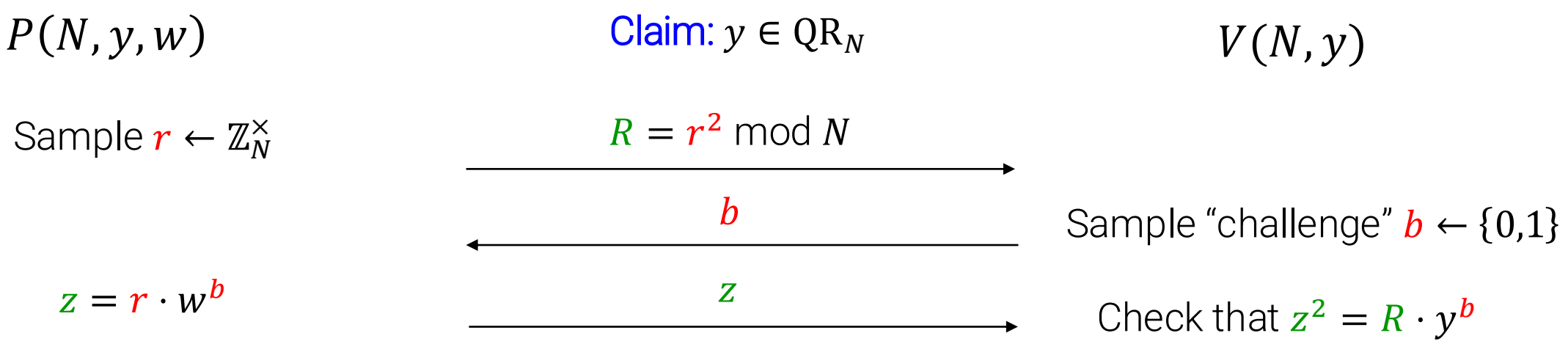
Claim: this proof system is HVZK

Proof: define the simulator

$$\mathrm{Sim}(N, y) = \Big((N, y), b, \overbrace{s^2 \cdot y^{-b}}^{R}, \overbrace{s}^{z}\Big)$$

If $y \equiv w^2$, the output of $\mathrm{Sim}(N, y)$ is identical to $\mathrm{view}_V(N, y, w)$

Zero-Knowledge Proof for QR

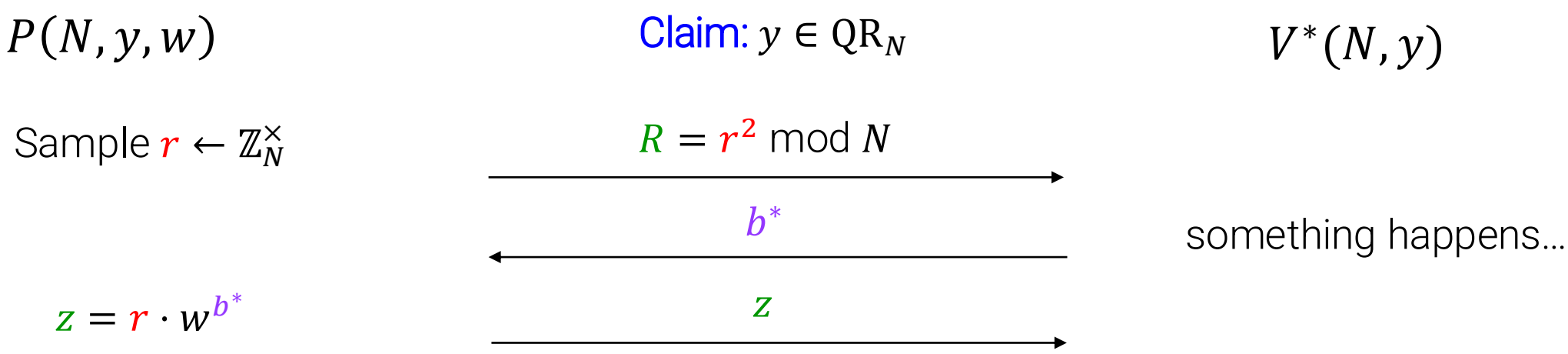


Claim: this proof system is ZK

Proof: ???

Any ideas?

Zero-Knowledge Proof for QR



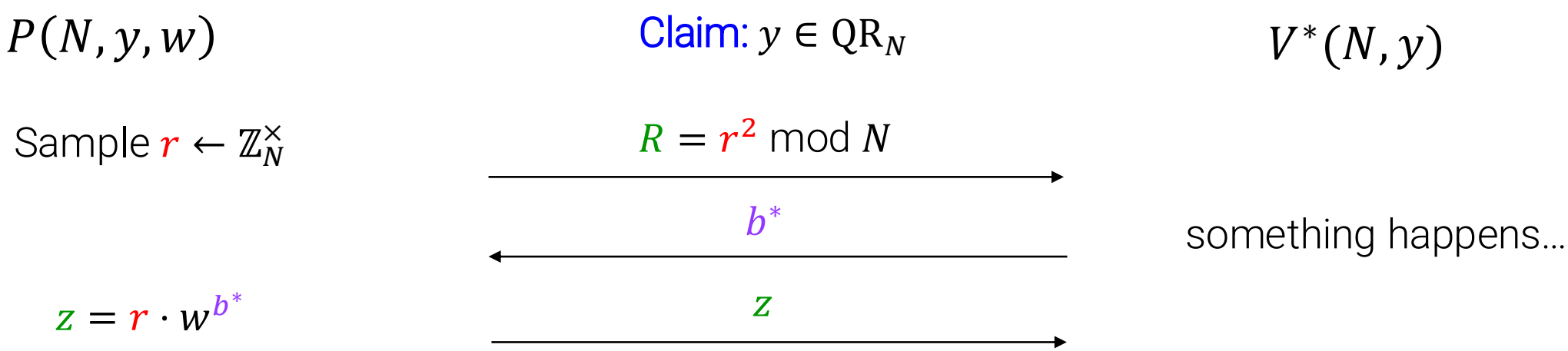
Claim: this proof system is ZK

Proof: Given V^*, need to define $\mathbf{Sim}^*$

Can we use $\mathbf{Sim}^* = \mathbf{Sim}$?

$$\mathbf{Sim}(N, y) = \left((N, y), b, s^2 \cdot y^{-b}, s \right)$$

Zero-Knowledge Proof for QR



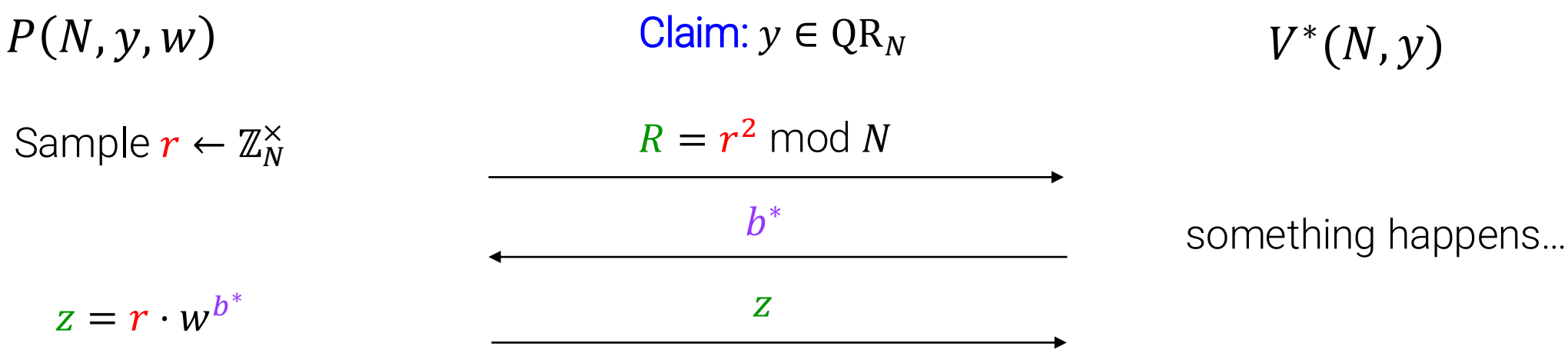
Claim: this proof system is ZK

Proof: Given V^*, need to define $\mathbf{Sim}^*$

Can we use $\mathbf{Sim}^* = \mathbf{Sim}$? No! What if $V^*(R) = 0$ for all R?

$$\mathbf{Sim}^*(N, y) = \left((N, y), 0, s^2, s\right)$$

Zero-Knowledge Proof for QR



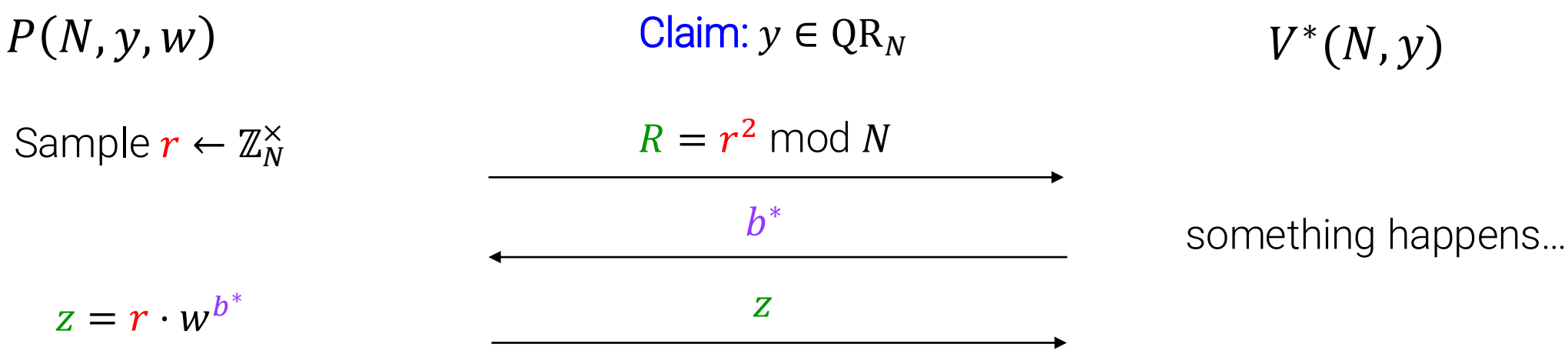
Claim: this proof system is ZK

Proof: Given V^*, need to define $\mathbf{Sim}^*$

Can we use $\mathbf{Sim}^* = \mathbf{Sim}$? No! What if $V^*(R) = 1$ for all R?

$$\mathbf{Sim}^*(N, y) = \left((N, y), 1, s^2 \cdot y^{-1}, s\right)$$

Zero-Knowledge Proof for QR

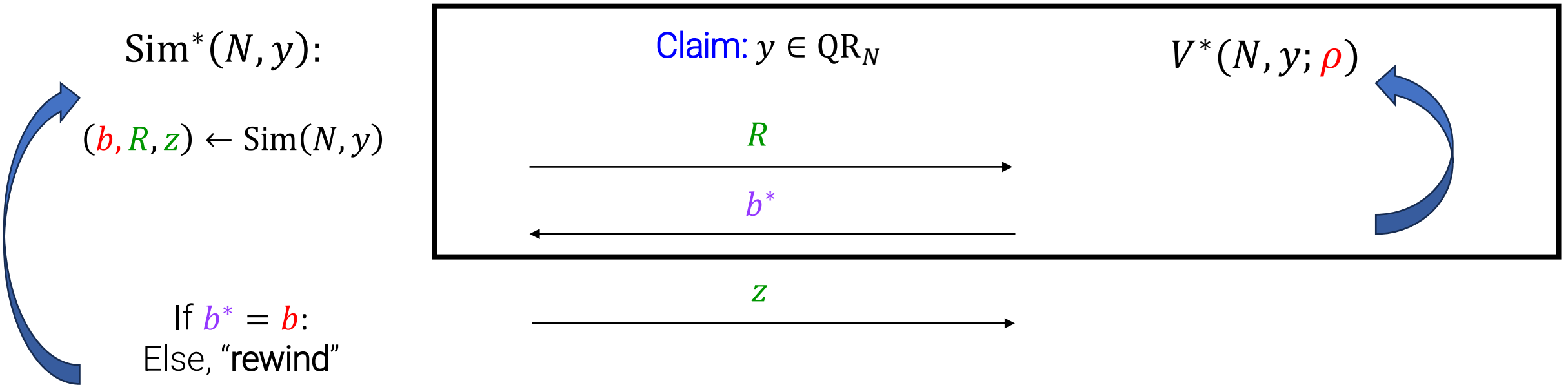


Claim: this proof system is ZK

Proof: Given V^*, need to define $\mathbf{Sim}^*$

Hard case: what if $V^*(R)$ **depends on** R?

Zero-Knowledge Proof for QR

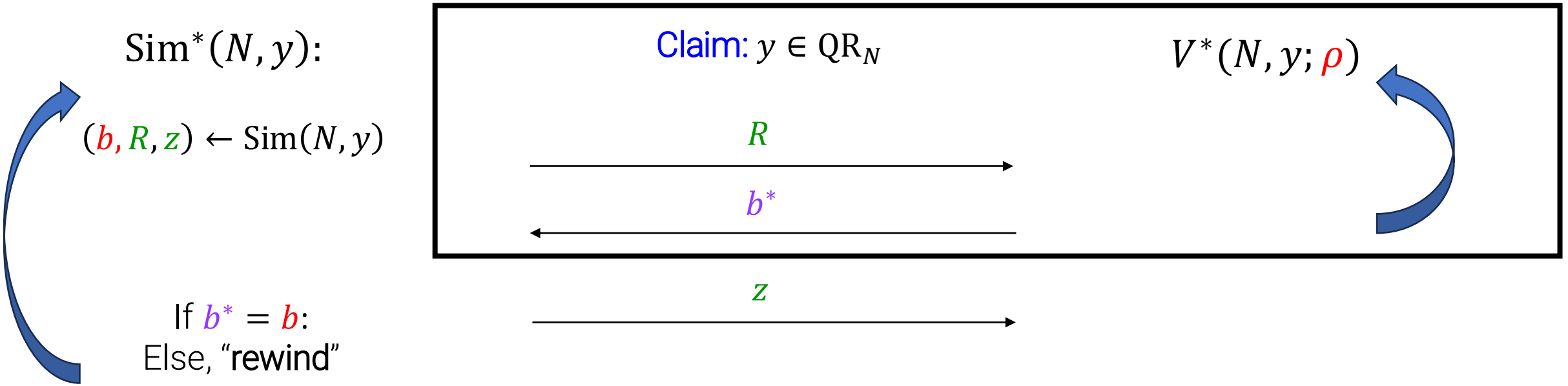


Randomness of V^*
(sampled once)

$$\mathrm{Sim}^*(N, y) = \big((N, y), \rho, R, z\big)$$

From final execution
of the loop

Zero-Knowledge Proof for QR



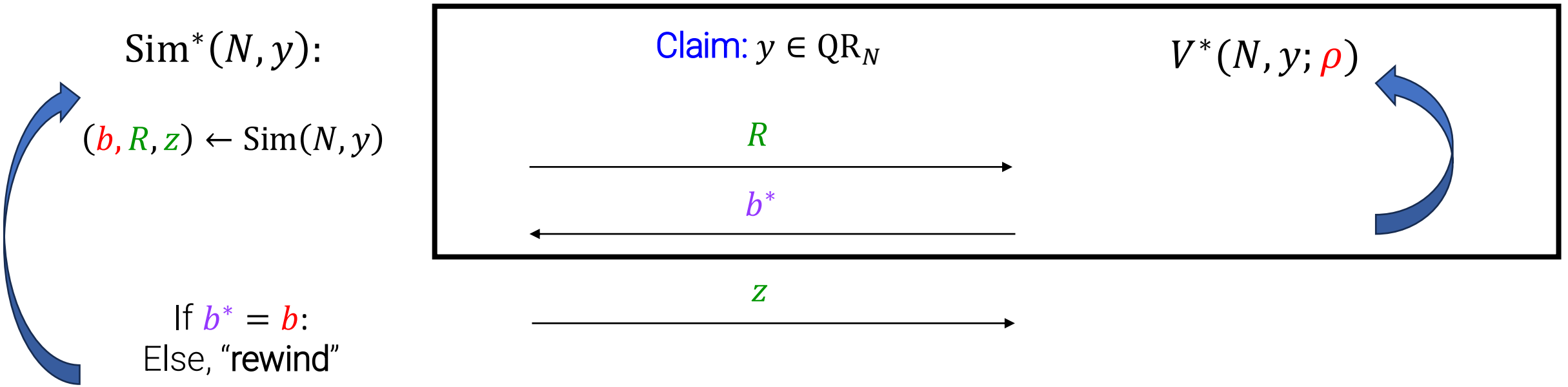
Claim: Sim^* terminates within n attempts with probability $1 - 2^{-n}$

Proof: in one attempt, we have:

$$\Pr[b^* = b] = 1/2$$

Why? Because $b^* = V^*(N, y, R; \rho)$ and R is independent of b!

Zero-Knowledge Proof for QR

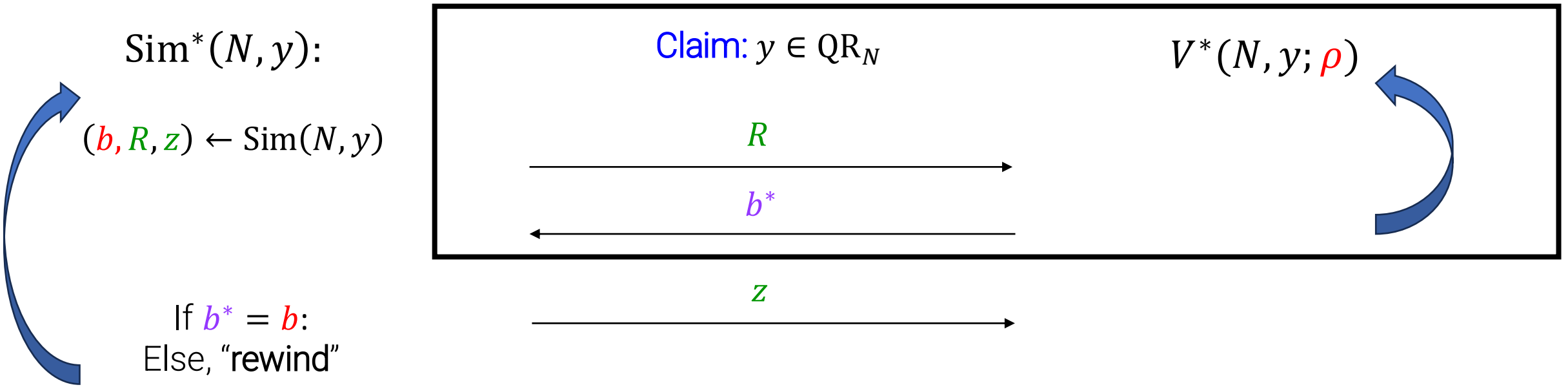


Claim: Output of Sim^* identical to $\mathbf{view}_{V^*}(N, y, w)$

Proof: Output of Sim^* is $\big((N, y), \rho, R, z\big)$, where

$$(b, R, z) \leftarrow \mathrm{Sim}(N, y) \mid b = b^*$$

Zero-Knowledge Proof for QR

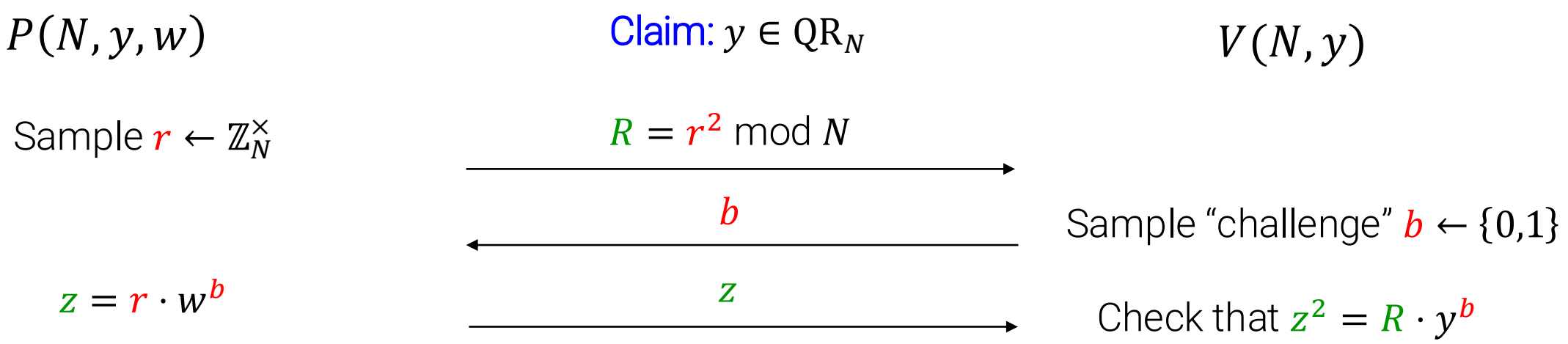


Claim: Output of Sim^* identical to $\mathbf{view}_{V^*}(N, y, w)$

Proof: Output of Sim^* is $\big((N, y), \rho, R, z\big)$, where

$$z = s \leftarrow \mathbb{Z}_N^\times \qquad R = s^2 \cdot y^{-b^*} \qquad \equiv \qquad z = r \cdot y^{b^*} \qquad R = r^2$$

Zero-Knowledge Proof for QR



Summary:

- Prover message defined two claims (R is a square, $R \cdot y$ is a square)
- If both claims are true, y must be a square
- Proof of each individual claim can be simulated
- Simulate V^* by guessing which b^* they will send, and rewinding if wrong

Zero-Knowledge Proofs

Theorem (we just proved):

The language $L_{\mathbf{QR}}$ has a perfect zero knowledge (PZK) proof system.

- PZK: Simulation indistinguishability is perfect

Question: for which languages are PZK proofs possible?

- Factoring: unknown (may also depend on the formulation of the language)
- 3-coloring: unknown but strong evidence against

Next: computational zero knowledge



Zero-Knowledge Proofs for NP





Theorem [Goldreich-Micali-Wigderson]:

Assuming OWFs, every language $L \in \mathbf{NP}$ has a ZK proof system!

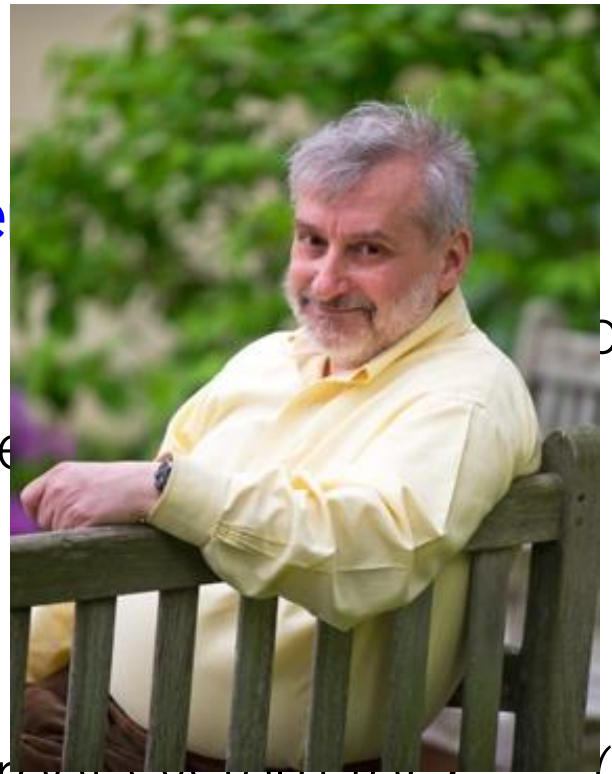
- Honest prover efficient given $\mathbf{NP}$-witness.
- ZK is computational

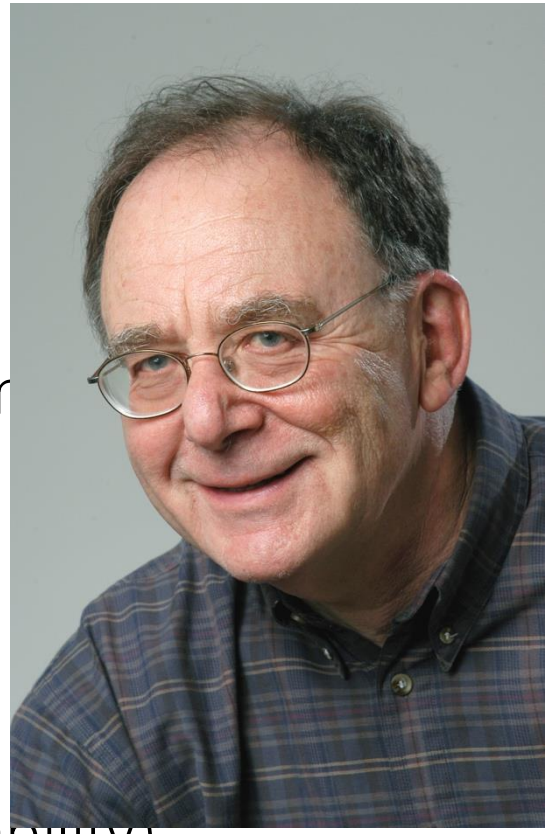
All languages with (non-interactive) proofs have ZK proofs!

Can be extended to all languages with interactive proofs ($\mathbf{IP}$)

Zero-Knowledge Proofs for NP







The -Wigde

Ass guage of system

- ent give
-

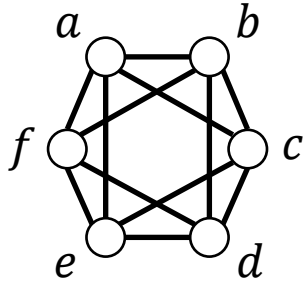
Proc

Step 1: It suffices to build a ZK proof system for $L_{\mathbf{3Col}}$ (3-colorability)

Makes use of “NP-completeness:” every $\mathbf{NP}$ problem can be efficiently converted into an instance of 3-coloring! ([Cook 1971], [Levin 1972], [Karp 1972])

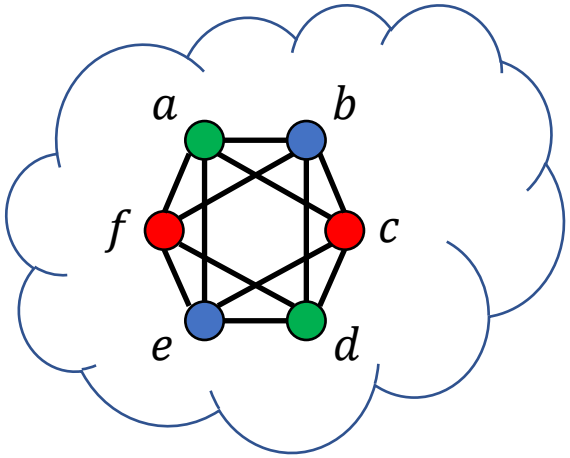
Zero Knowledge Proofs for NP

Claim: This graph has a 3-coloring.



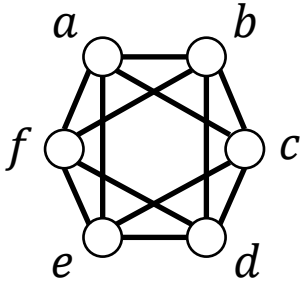
P V

Zero Knowledge Proofs for NP



Claim: This graph has a 3-coloring.

P

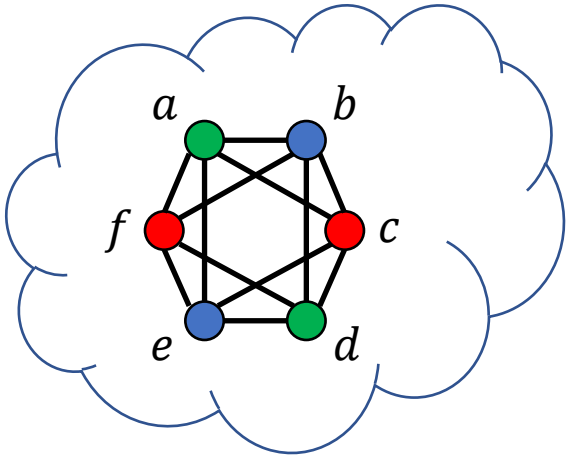


V

Intuition:

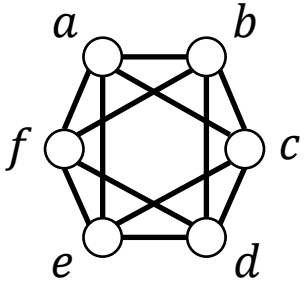
- P will send a message defining $|E(G)|$ different claims
- V will test one of these claims at random

Zero Knowledge Proofs for NP



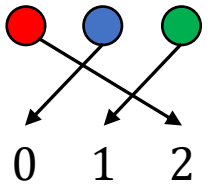
Claim: This graph has a 3-coloring.

P



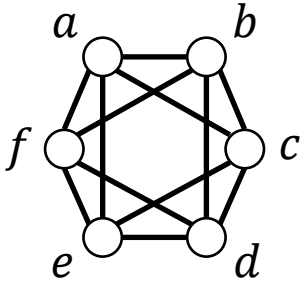
V

1) Randomize colors:



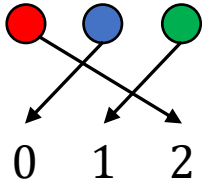
Zero Knowledge Proofs for NP

Claim: This graph has a 3-coloring.



P V

1) Randomize colors:

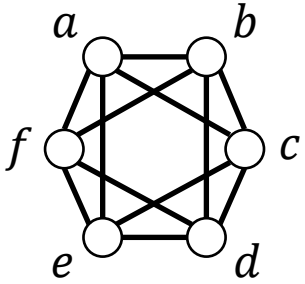


2) Commit:

a	b	c	d	e	f
1	0	2	1	0	2

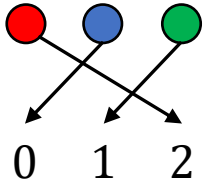
Zero Knowledge Proofs for NP

Claim: This graph has a 3-coloring.

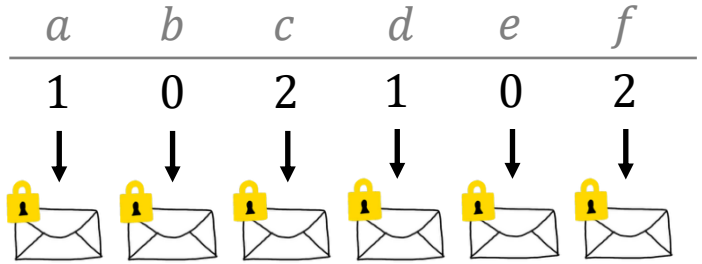


P V

1) Randomize colors:

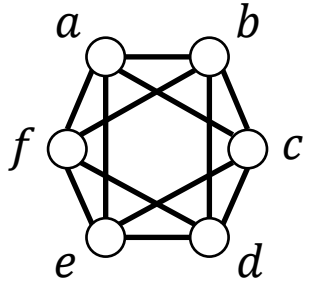


2) Commit:



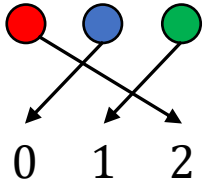
Zero Knowledge Proofs for NP

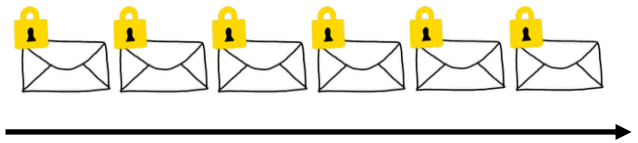
Claim: This graph has a 3-coloring.



P V

1) Randomize colors:





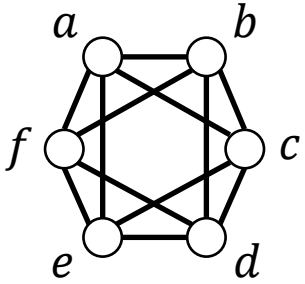
2) Commit:

a	b	c	d	e	f
1	0	2	1	0	2



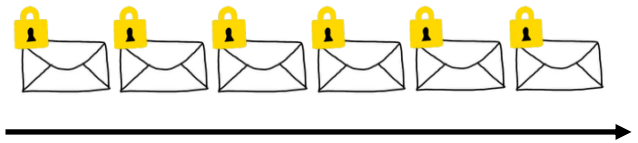
Zero Knowledge Proofs for NP

P



V

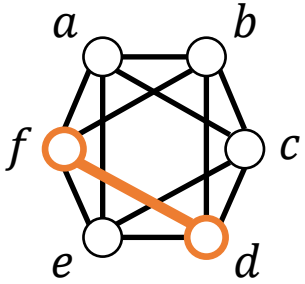
1) Randomize colors:



2) Commit:

Zero Knowledge Proofs for NP

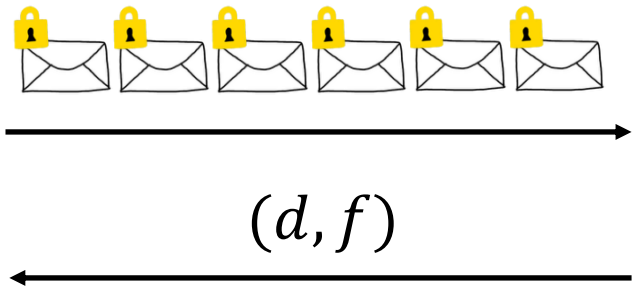
P



V

1) Randomize colors:

2) Commit:

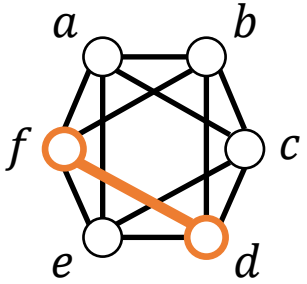


(d, f)

1) Sample a **challenge** edge.

Zero Knowledge Proofs for NP

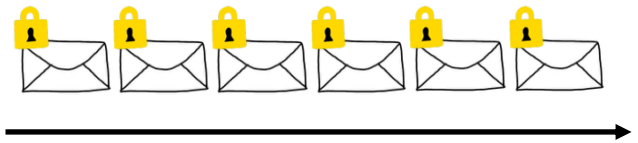
P



V

1) Randomize colors:

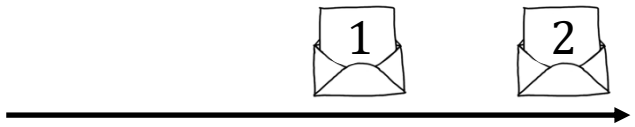
2) Commit:



(d, f)

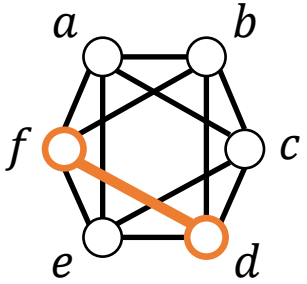
1) Sample a **challenge** edge.

3) Reveal edge colors



Zero Knowledge Proofs for NP

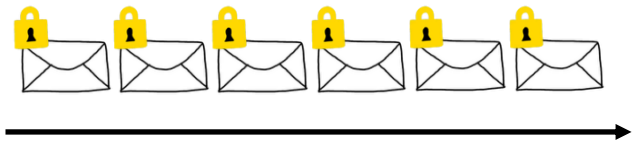
P



V

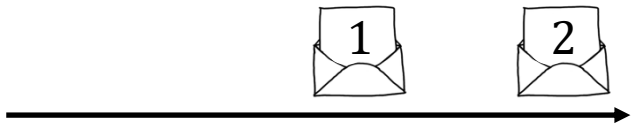
1) Randomize colors:

2) Commit:



(d, f)

1) Sample a **challenge** edge.



3) Reveal edge colors

2) Accept if colors are different.