Interactive Proofs

Prashant Nalini Vasudevan

Mathematical Proof

Sequence of claims leading to theorems from axioms

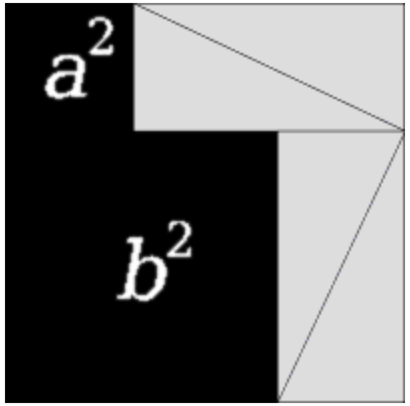
Theorem: $(a+b)^2 = a^2 + 2ab + b^2$

Proof:

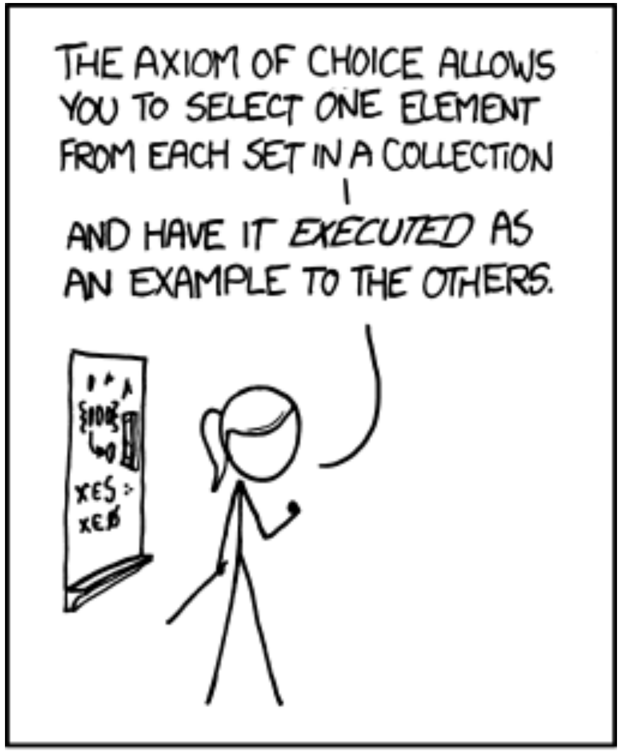
$$\begin{aligned}(a+b)^2 &= (a+b)\cdot(a+b)\\ &= a\cdot a + a\cdot b + b\cdot a + b\cdot b\\ &= a^2 + 2ab + b\text{^}2\end{aligned}$$

Verification: Verify each claim

Other kinds of proofs



Proof by picture



MY MATH TEACHER WAS A BIG BELIEVER IN PROOF BY INTIMIDATION.

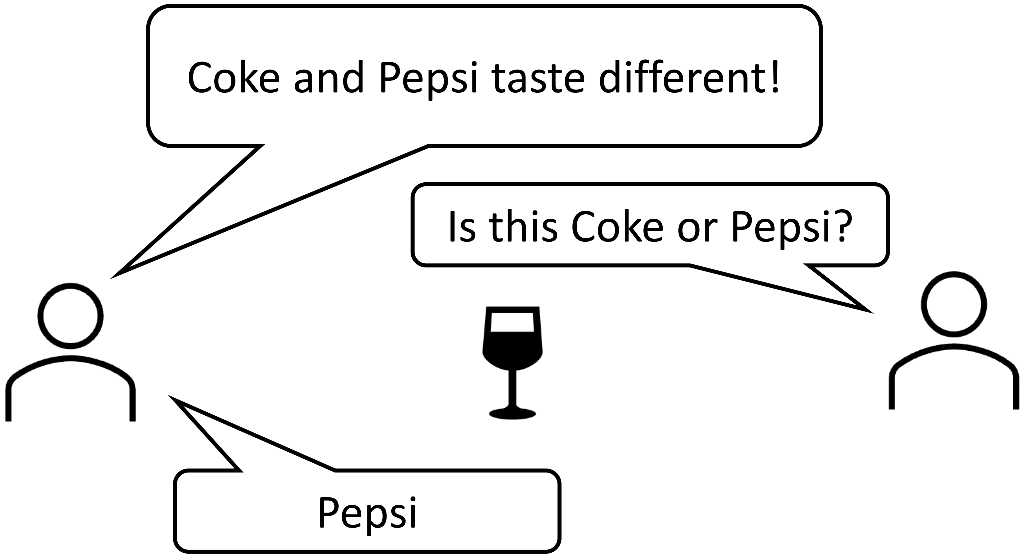
Sources: Wikipedia, xkcd

What is a Proof?

A proof is anything that convinces me that a statement is true

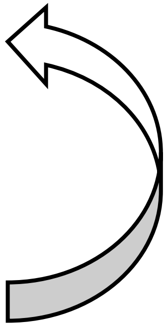
"verifier" of the proof

An “Interactive” Proof



Crucial usage of randomness

Fill glass with either Coke or Pepsi at random

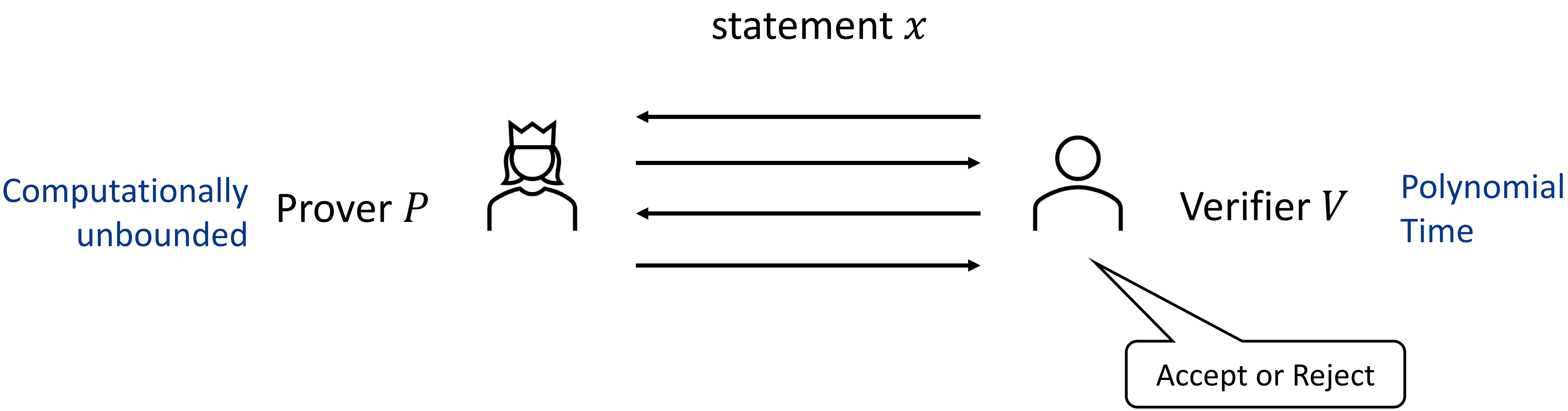


Repeat several times

- If they taste different, can always answer correctly
- If not, some answer will be wrong (with high probability)
- You know whether the glass has Coke or Pepsi, so you can check

Interactive Proofs

[Goldwasser-Micali-Rackoff 85, Babai 85]



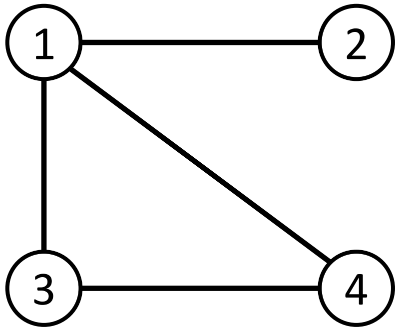
- **Completeness:** If statement is true, Verifier should Accept with high probability

- **Soundness:** If statement is false, Verifier should Reject with high probability, even if Prover cheats

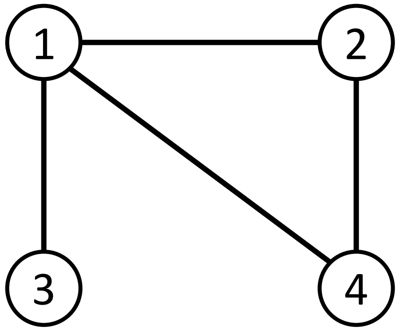
Example: Graph Non-Isomorphism

Definition: Two graphs G_0 and G_1 are isomorphic if there is a relabelling of the vertices of G_0 that makes it the same as G_1

G_0



G_1

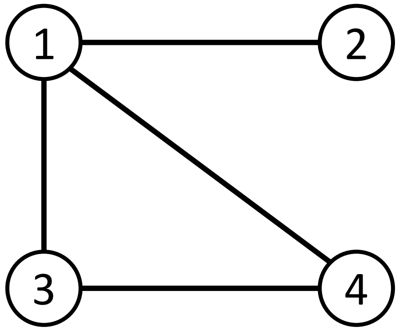


G_0 and G_1 isomorphic by relabelling: $1 \to 1, 2 \to 3, 3 \to 2, 4 \to 4$

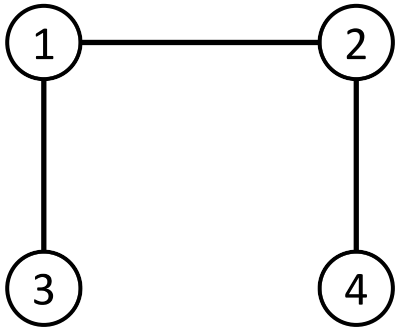
Example: Graph Non-Isomorphism

Definition: Two graphs G_0 and G_1 are isomorphic if there is a relabelling of the vertices of G_0 that makes it the same as G_1

G_0



G_1



G_0 and G_1 not isomorphic

Example: Graph Non-Isomorphism

Definition: Two graphs G_0 and G_1 are isomorphic if there is a relabelling of the vertices of G_0 that makes it the same as G_1

Not known how to decide in polynomial time

Easy to prove (G_0, G_1) are isomorphic – just show relabelling

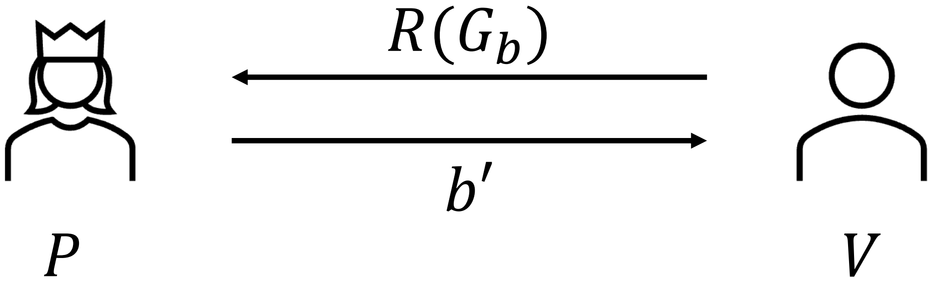
How to prove G_0 and G_1 are *not* isomorphic?

IP for Graph Non-Isomorphism

G_0, G_1

Pick uniformly random relabelling R, random bit b

If $R(G_b)$ isomorphic to G_0, set $b' = 0$, else $b' = 1$



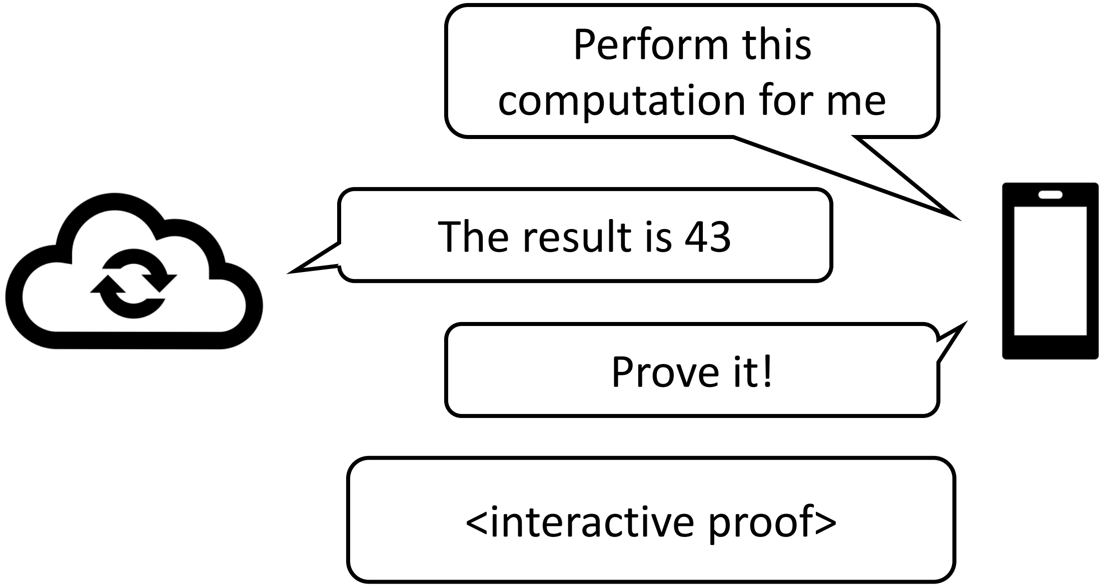
Accept iff $b' = b$

Completeness: If (G_0, G_1) are non-isomorphic, then $R(G_b)$ is isomorphic to one of G_0 or G_1, but not both. Prover can thus learn b given $R(G_b)$. So V always accepts.

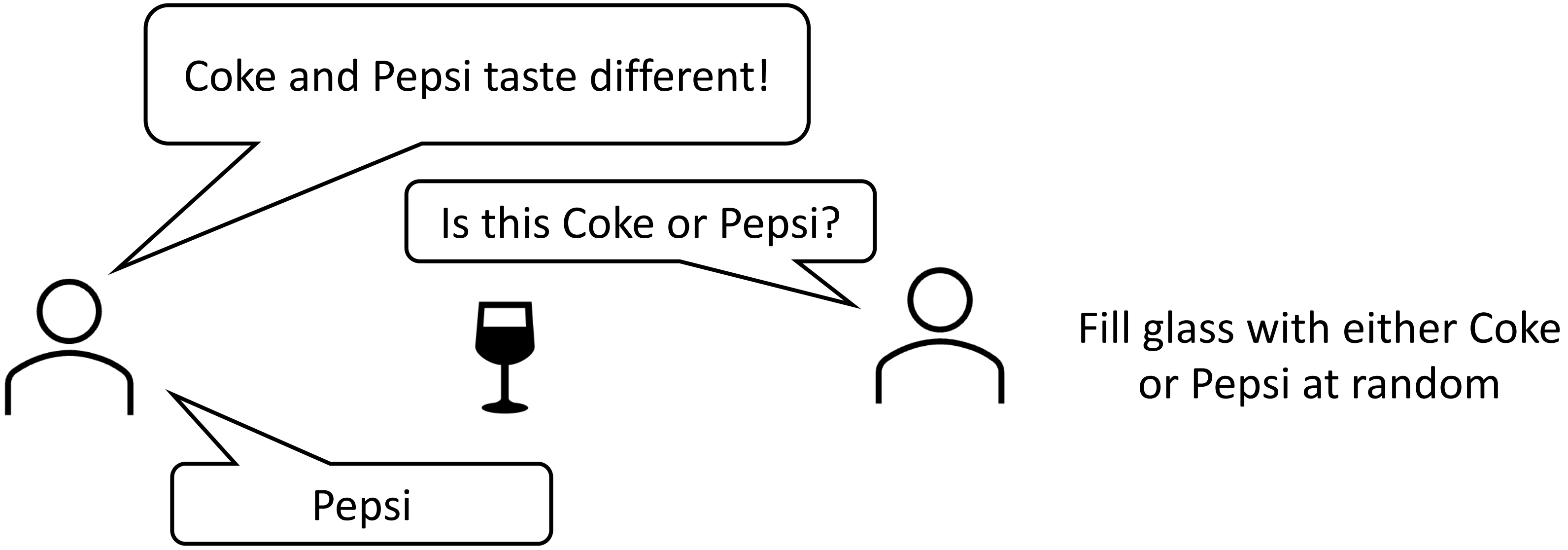
Soundness: If (G_0, G_1) are isomorphic, then $R(G_b)$ could have been produced either from G_0 or G_1, so prover learns nothing about b. So V accepts with probability at most ½.

Interactive Proofs

- Fundamentally new notion of what it means to prove something
- Connected to various other concepts in complexity theory
- Potential real-world applications, e.g. delegation of computation



Zero-Knowledge Proofs



If you cannot distinguish between Coke and Pepsi before the proof, you still cannot after the proof!

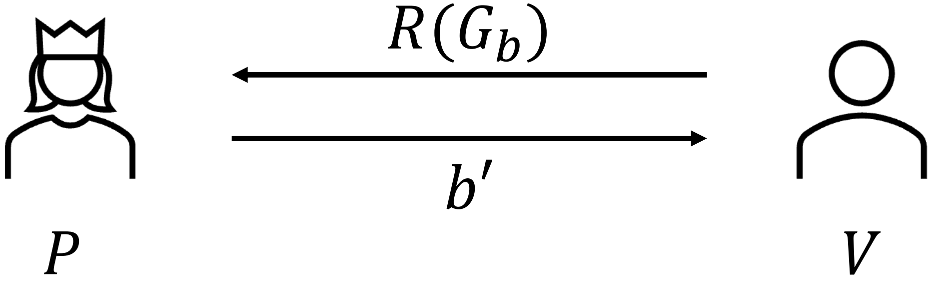
Zero-Knowledge: Verifier learns nothing during the proof except the truth of the statement being proven

Zero-Knowledge Proofs

G_0, G_1

Pick uniformly random relabelling R, random bit b

If $R(G_b)$ isomorphic to G_0, set $b' = 0$, else $b' = 1$



Accept iff $b' = b$

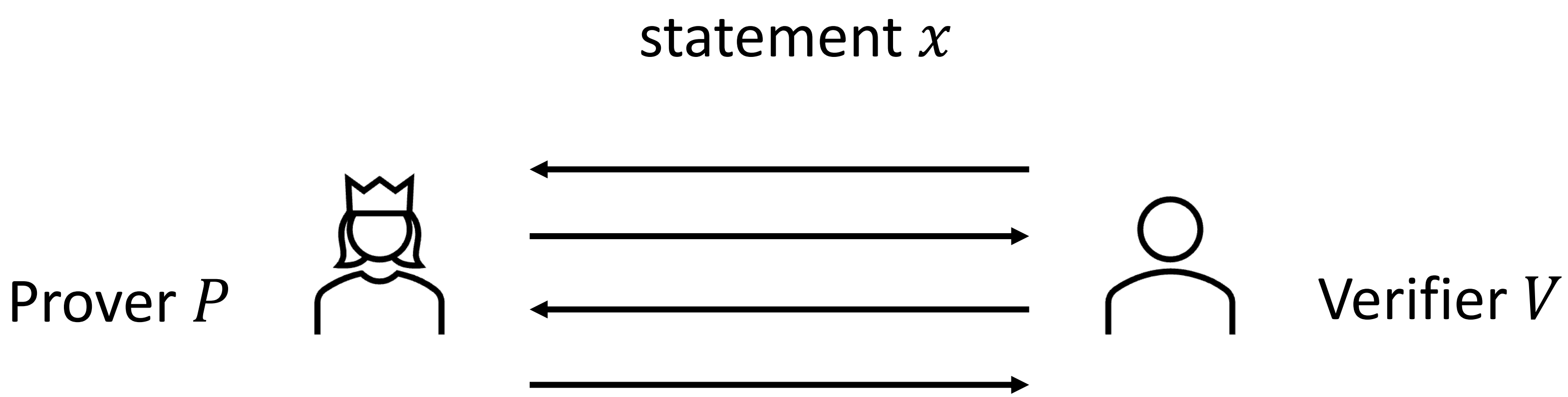
What did V learn in this proof?

If (G_0, G_1) isomorphic: V always receives $b' = 0$

If (G_0, G_1) non-isomorphic: V receives $b' = b$, but it already knew b!

Zero-Knowledge Proofs

[Goldwasser-Micali-Rackoff 85]

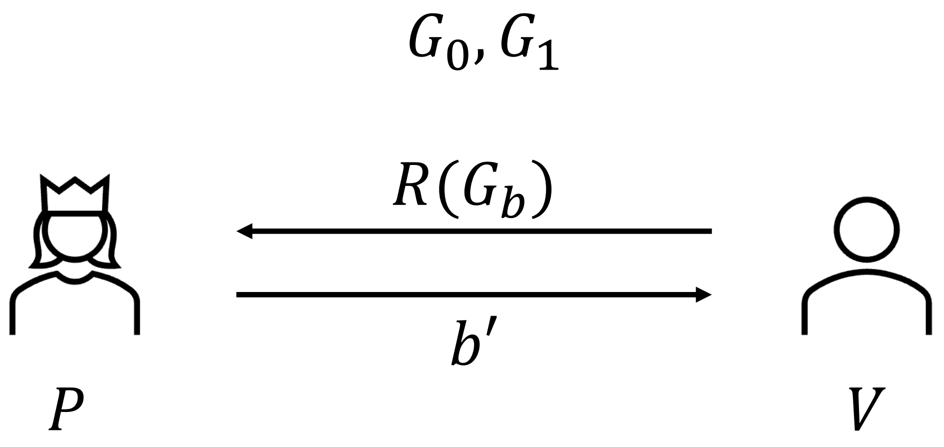


Informally: The Verifier learns nothing during the proof except the truth of the statement being proven

Somewhat Formally: Knowing whether the statement is true or not, the Verifier can "simulate" its interaction with the Prover on its own

Zero-Knowledge Proofs

If $R(G_b)$ isomorphic to G_0,
set $b' = 0$, else $b' = 1$



Pick uniformly random relabelling R,
random bit b

Accept iff $b' = b$

What V sees in the actual proof:
$(R, b, R(G_b), b')$

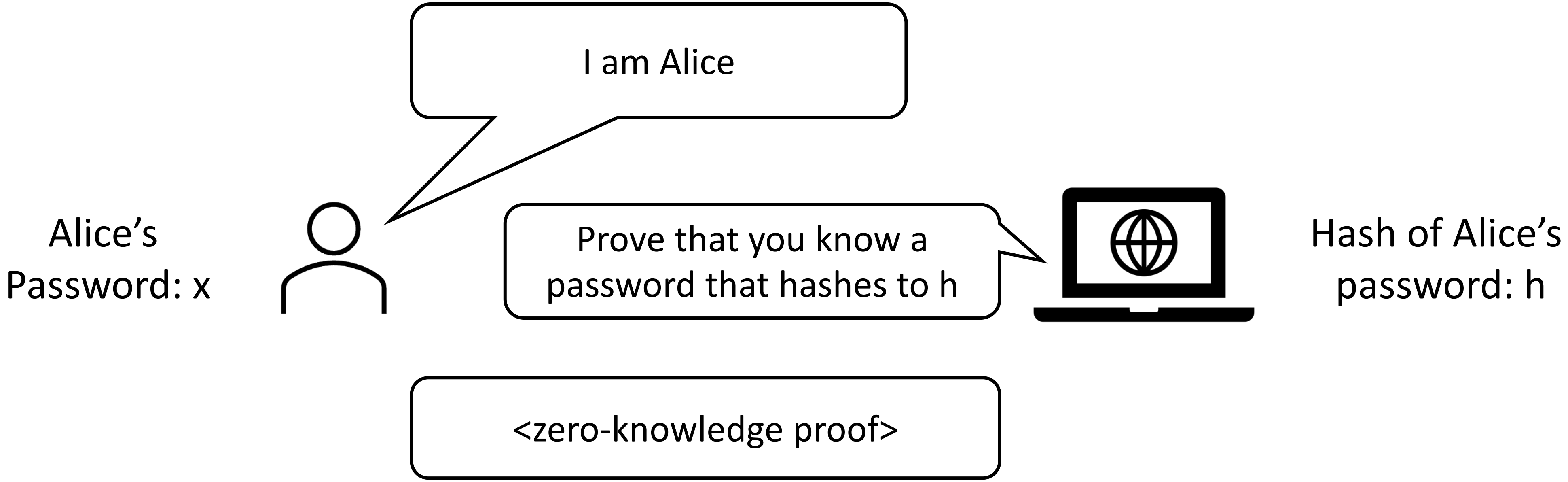
Simulation:
1. Sample R and b
2. If (G_0, G_1) are non-isomorphic, output $(R, b, R(G_b), b)$
3. Else, output $(R, b, R(G_b), 0)$

Computationally efficient, doesn't need prover!

Only needs to know whether (G_0, G_1) are isomorphic

Zero-Knowledge Proofs

- Connected to various concepts in cryptography
- Useful when data needs to be protected while allowing certain functionalities, e.g. authentication



Non-Classical Proof Systems

- New notions of what it means to “prove” something
- Vastly more “powerful” than classical mathematical proofs
- Usually involve randomness and/or interaction
- Many other examples:
  - Probabilistically Checkable Proofs,
  - Computationally Sound Interactive Proofs, etc.
- Lots of implications in theoretical computer science and cryptography
- Many practical applications – delegation, blockchains, etc.