Zero-Knowledge Interactive Proofs

Michele Ciampi

THE UNIVERSITY of EDINBURGH

Two parties for a proof

- Merlin (prover) has unbounded resources
- Arthur (verifier) has limited resources

Theorem/statement $\mathbf{x}$

$\xrightarrow{\pi}$

The proof is efficient: $\mathbf{x}$ is an NP statement and $\boldsymbol{\pi}$ is its certificate/witness/proof

Graph Isomorphism

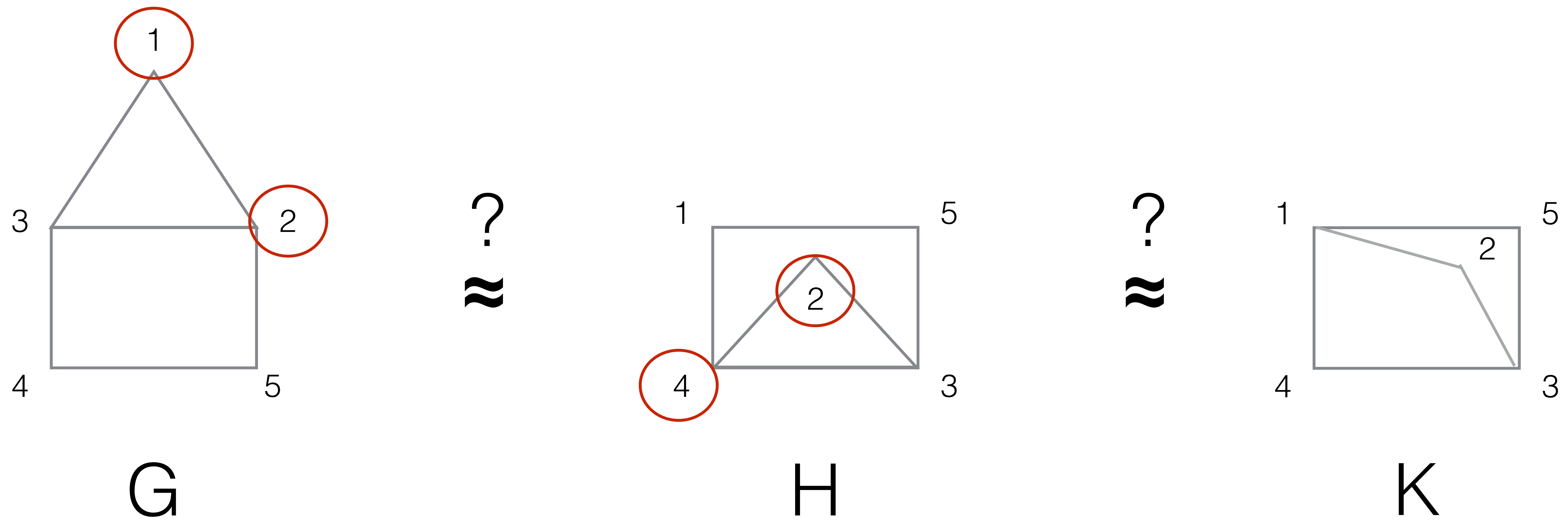
An isomorphism of graphs **G** and **H** is a bijection (permutation) π between the vertex sets of **G** and **H**

$$\pi: V(\mathbf{G}) \longrightarrow V(\mathbf{H})$$

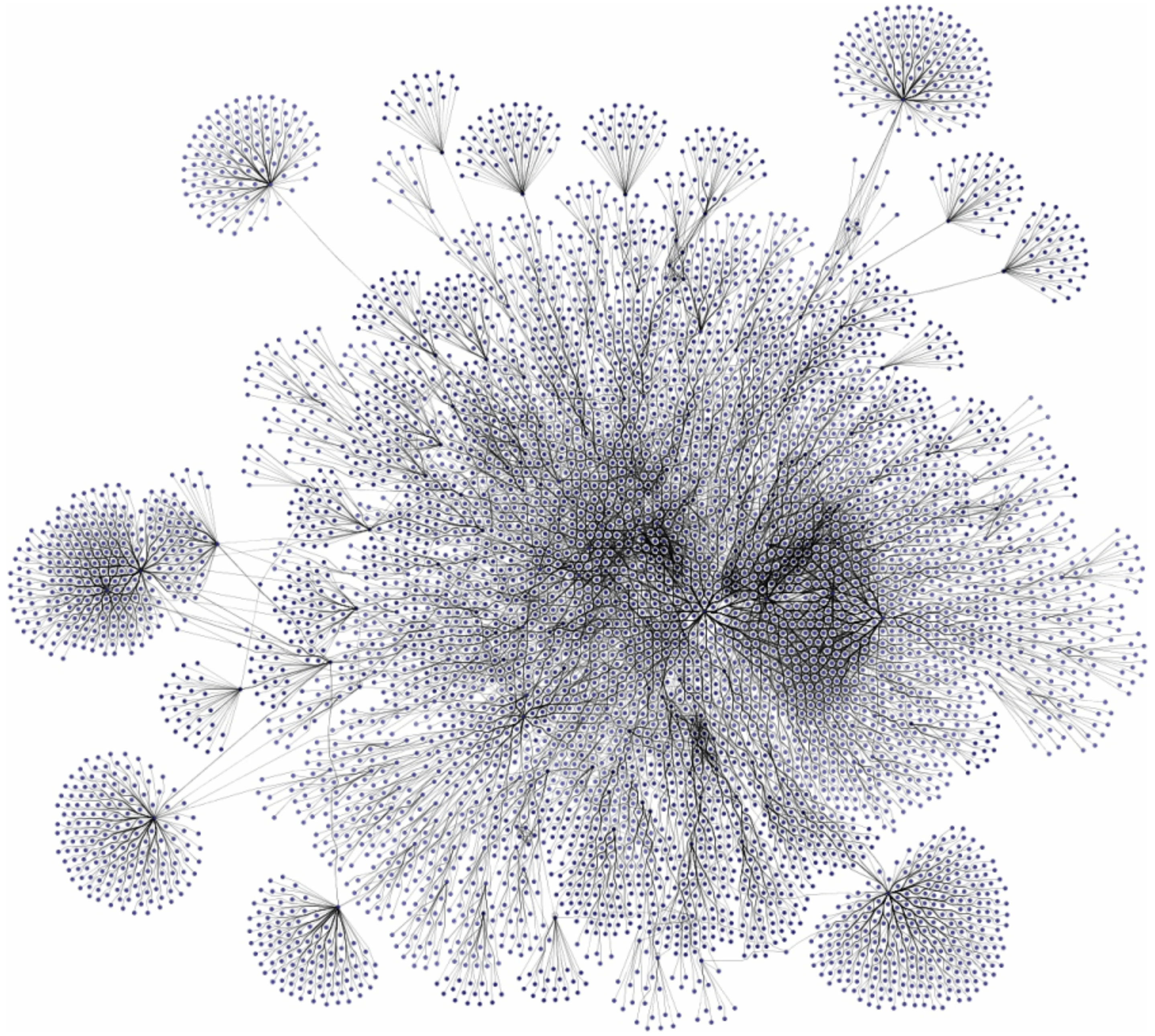
such that any two vertices u and v of **G** are adjacent in **G** if and only if π(u) and π(v) are adjacent in **H**.

π

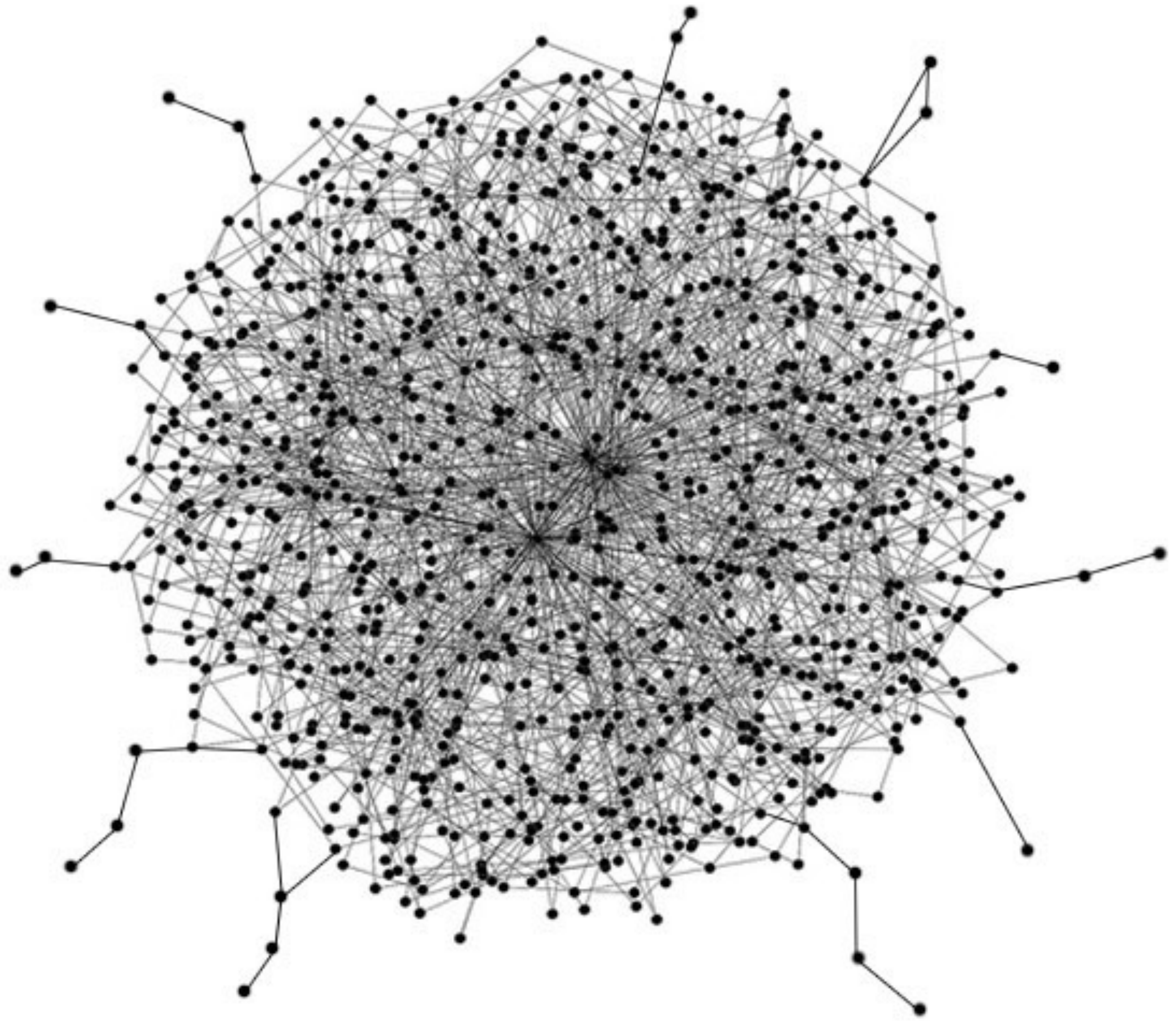
G	**H**
1	2
2	4
3	3
4	5
5	1



Graph Isomorphism



$\stackrel{?}{\approx}$



The problem belongs to NP

We do not know if it is in P: best known algorithm is quasi-polynomial time

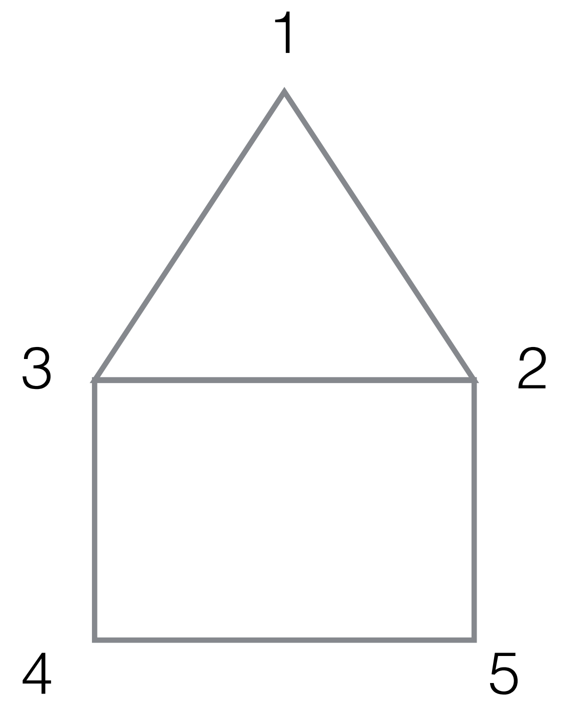
Graph Isomorphism

Witness

π

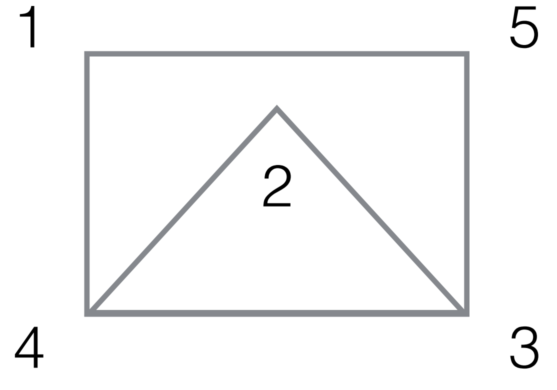
G	H
1	2
2	4
3	3
4	5
5	1

Thm



G

$\approx$



H

π: 1—>2, 4—>1,…

Interactive Proofs

- Suppose now that I want to prove that two graphs are **not isomorphic** or that an equation has no solutions.
- Introduced by Goldwasser, Micali and Rackoff
  - A proof is described as a game between a *prover* and a *verifier*
  - The theorem is true if and only if the prover wins the game always.
  - If the theorem is false then the prover loses the game with 50% probability

Prover (Merlin)

Verifier (Arthur)

Interactive Proofs

A simple example first

Prover

Verifier

Interactive Proofs

A simple example first

Prover

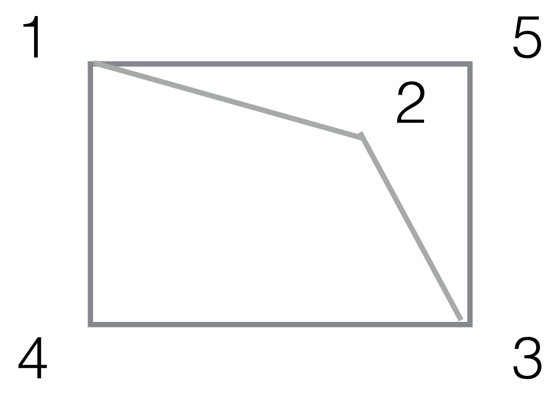
Verifier

If the pencils are both red, then the prover convinces the verifier with a 50% probability

We can repeat the proof many times to make this probability small

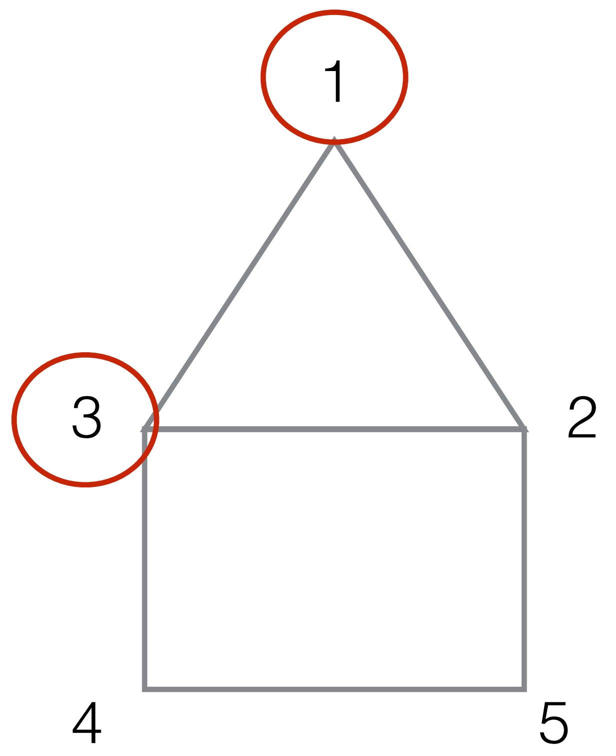
Graph Non-Isomorphism

Thm



H

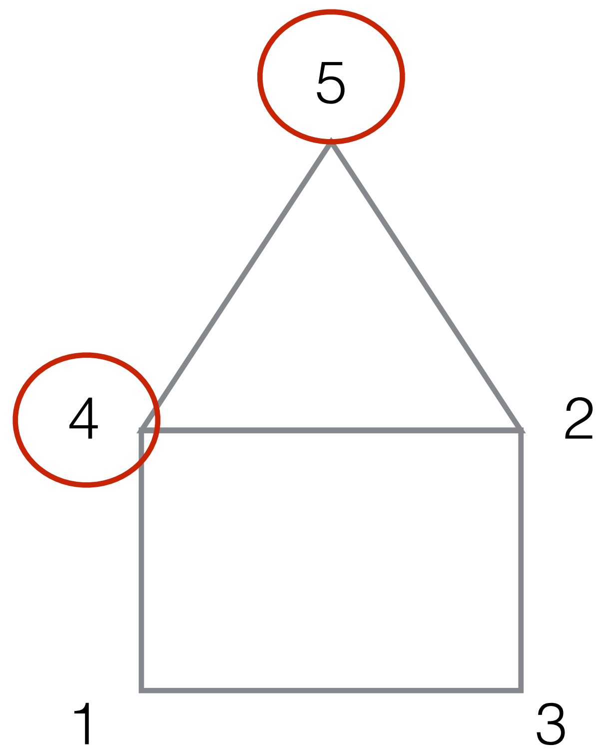
$\not\approx$



G

π'

$\approx$



C

π'

G	C
1	5
2	2
3	4
4	1
5	3

C cannot be isomorphic to H (due to transitivity)

Unbounded

C

G

G $\Leftarrow$

π' $\Leftarrow$ Random permutations

C $\Leftarrow$ π'(G)

Poly

Interactive Proofs (formal definition)

Definition 4.2.6 (Generalized Interactive Proof): *Let* $\mathtt{c}, \mathtt{s} : \mathbb{N} \to \mathbb{R}$ *be functions satisfying* $\mathtt{c}(n) > \mathtt{s}(n) + \frac{1}{p(n)}$ *for some polynomial* $p(\cdot)$. *An interactive pair* (P, V) *is called a (generalized) interactive proof system for the language* L*, with* **completeness bound** $\mathtt{c}(\cdot)$ *and* **soundness bound** $\mathtt{s}(\cdot)$*, if*

- (modified) completeness: *for every* $x \in L$,

$$\Pr\left[\langle P, V\rangle(x) = 1\right] \geq \mathtt{c}(|x|)$$

- (modified) soundness: *for every* $x \notin L$ *and every interactive machine* B,

$$\Pr\left[\langle B, V\rangle(x) = 1\right] \leq \mathtt{s}(|x|)$$

In the previous example c(|x|)=1 and s(|x|)=1/2

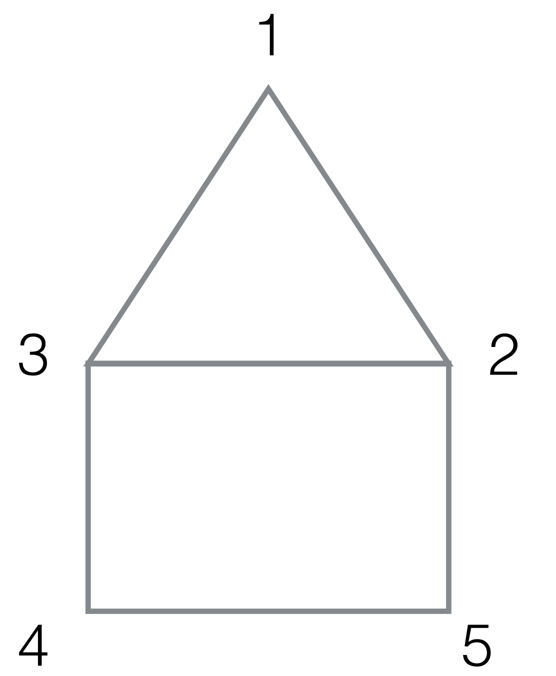
Zero-Knowledge (ZK)

Witness

π

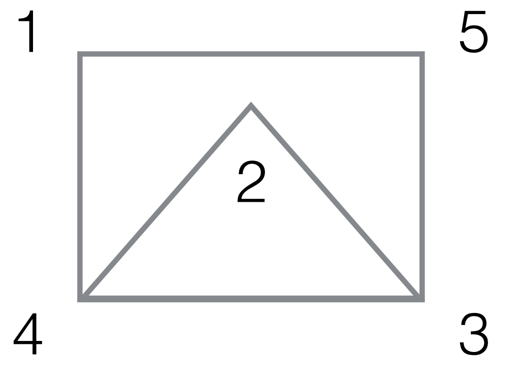
G	**H**
1	2
2	4
3	3
4	5
5	1

Thm



G

$\approx$



H

π: 1—>2, 2—>4,…

- How much knowledge is transmitted to the verifier?
- We would like to transmit only one bit: 1 if the theorem is true and 0 otherwise.
- E.g. For the case of graph isomorphism, the prover does not want to disclose the witness

ZK for Graph Isomorphism

Witness

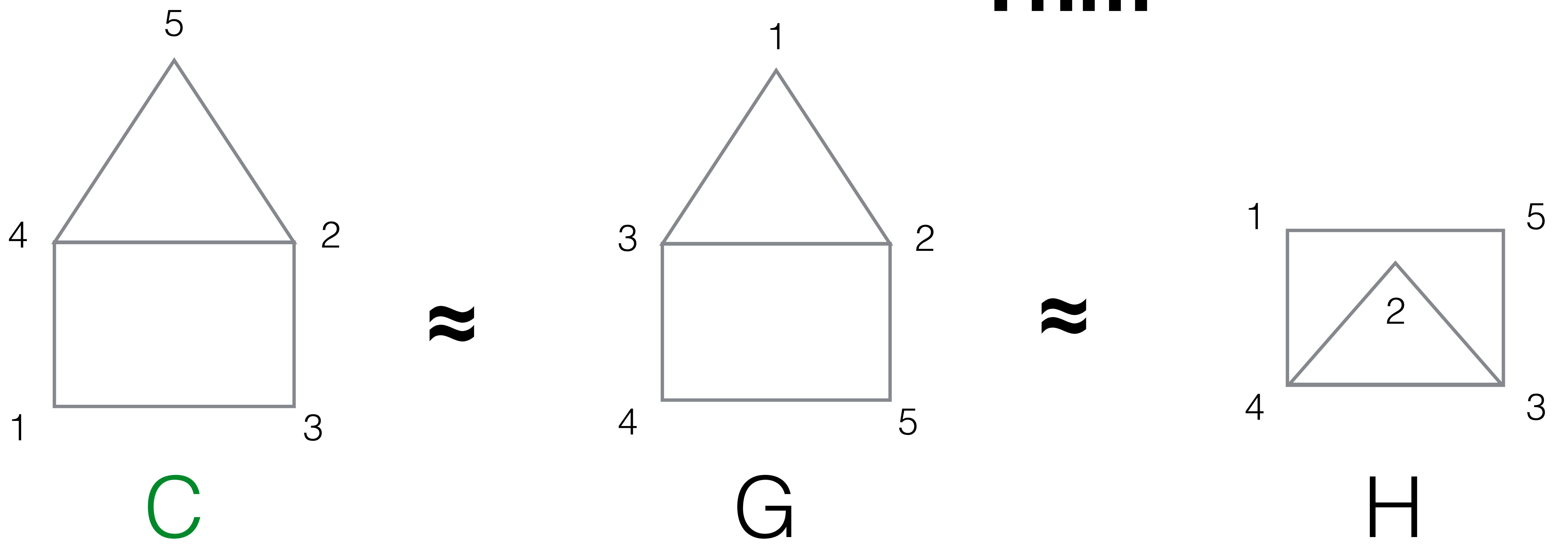
π

G	H
1	2
2	4
3	3
4	5
5	1

π'

G	C
1	5
2	2
3	4
4	1
5	3

Thm

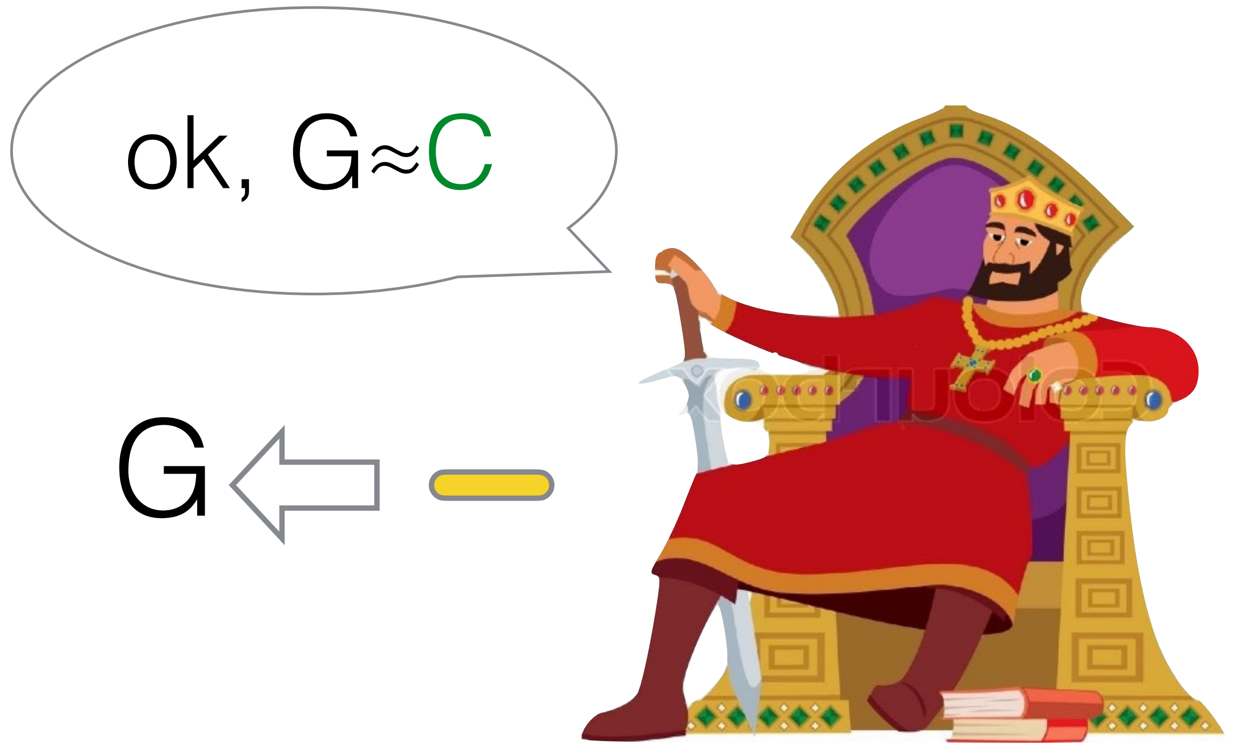


$C \Leftarrow \pi'(G)$

$\xrightarrow{C}$

$\xleftarrow{G}$

$\xrightarrow{\pi'}$



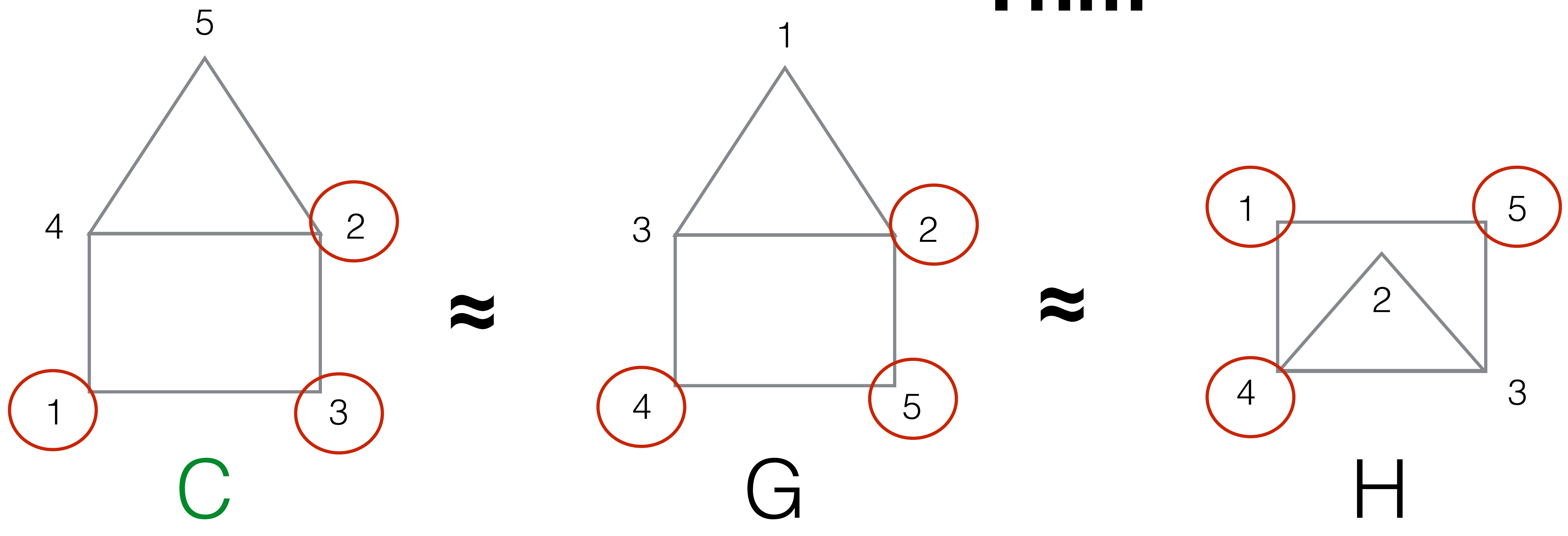
$G \Leftarrow$

ZK for Graph Isomorphism

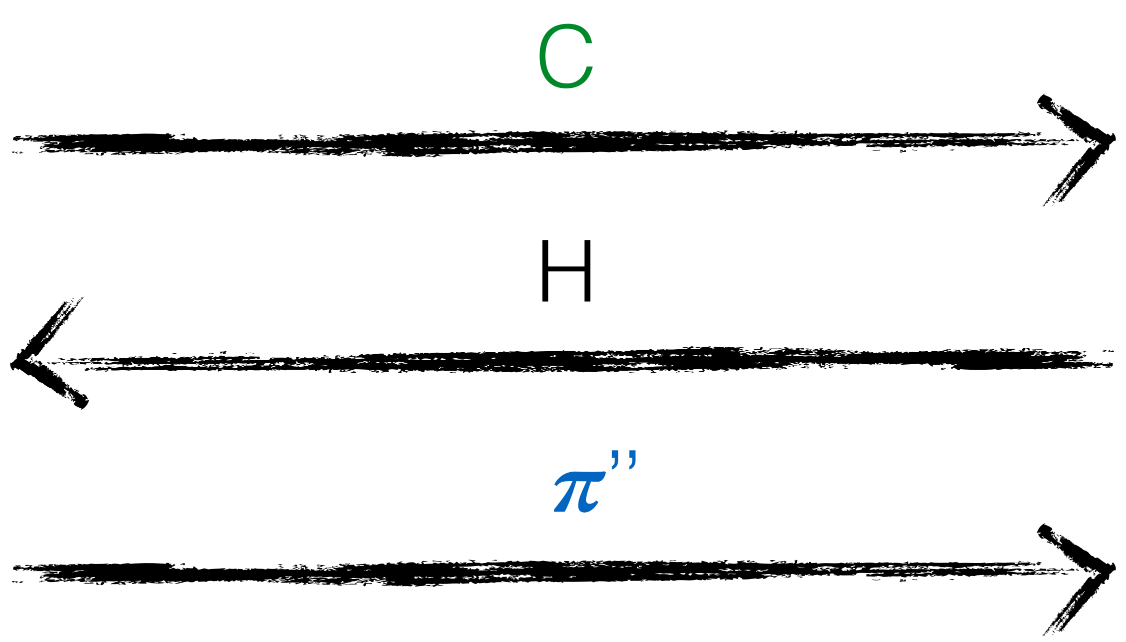
Witness

π		π'		π''	
H	**G**	**G**	**C**	**C**	**H**
2	1	1	5	1	
4	2	2	2	2	
3	3	3	4	3	
5	4	4	1	4	
1	5	5	3	5	

Thm



$C \Leftarrow \pi'(G)$



ok, H≈C

H ⇐ –

If the graphs are non-isomorphic then the prover convinces the verifier with a 50% probability

We can repeat the proof many times to make this probability small

Zero Knowledge

- The notion of zero knowledge requires the existence of a simulator **S** that:
  - knows **only** that the theorem is true
  - is efficient
  - generates a transcript that is distributed similarly* to the real one (when the verifier is honest)
  - has black-box access to the adversary

Honest-Verifier ZK for Graph Isomorphism

π

H	G
2	1
4	2
3	3
5	4
1	5

Thm G ≈ H

$\xrightarrow{C}$

$\xleftarrow{G}$

$\xrightarrow{\pi'}$

ok, G≈C

G ⇐ —

Sim **Thm** G ≈ H

G ⇐ —

$\xleftarrow{C}$

π^* ⇐ Random permutation of G

$\xrightarrow{G}$

C ⇐ π^*(G)

$\xleftarrow{\pi^*}$

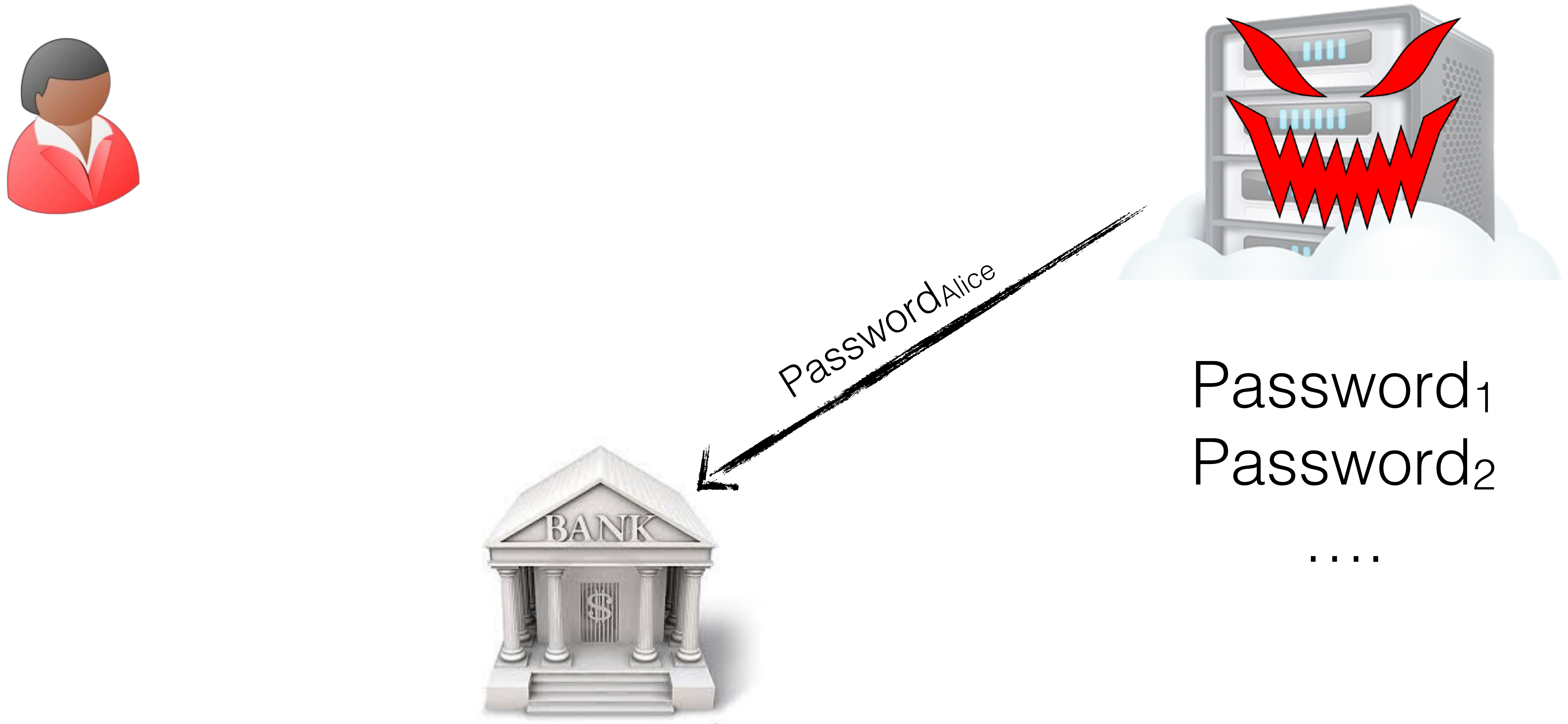
Why do we care?

- We know how to construct ZK proofs for any NP-language (with both efficient prover and verifier)
- CCA-encryption scheme
- Multi-party computation
- Identification schemes
- Privacy-preserving blockchains

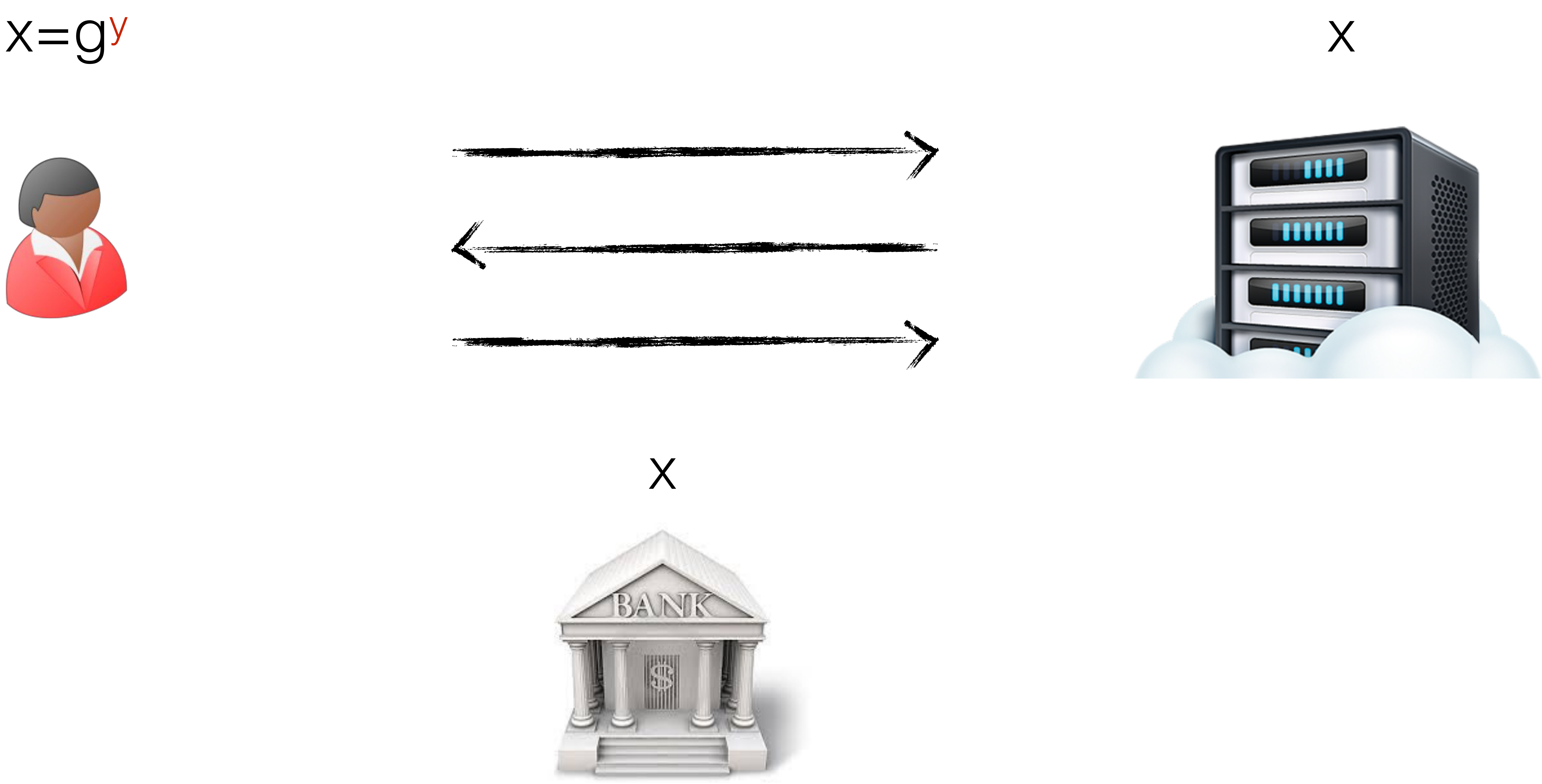
Identification scheme

Password$_1$
Password$_2$
....

Identification scheme



Identification scheme



Identification scheme

y, $x_1=g^y$

x_2

x_3

x_4

I know y s.t. either
$x_1=g^y$ or
$x_2=g^y$ or
$x_3=g^y$ or
$x_4=g^y$

x_1,x_2,x_3,x_4

End

The references are for the book of Goldreich Oded: Foundations of Cryptography: Volume 1, Basic Tools (see the link on learn)

- Sec. 4.2 until (included) Sec. 4.2.2 with no proofs
- Sec. 4.3 until (included) Sec. 4.3.2 with no proofs
- Sec. 4.7 until (included) Definition 4.7.2 with no proofs