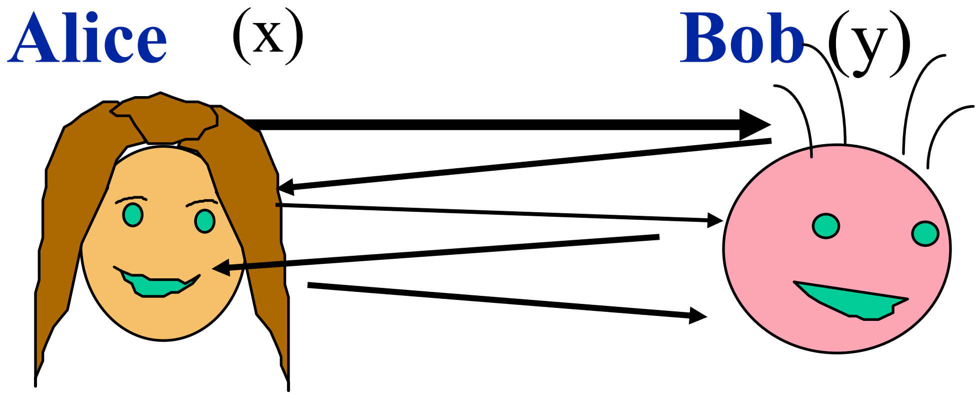
Lecture 14

Zero Knowledge I

From Secure Communication to Complex Interactions



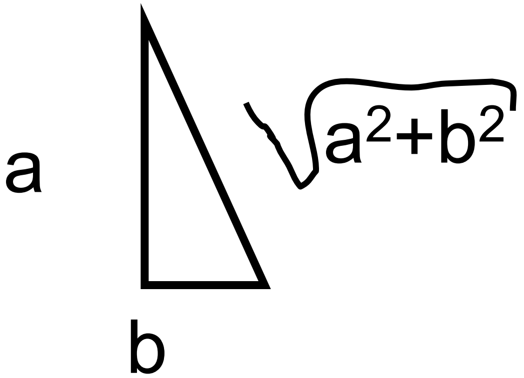
Now doing much more than communicating securely:

- Complex interactions: games, computations, proofs
- Complex Adversaries: Alice or Bob, adaptively chosen
- Complex Properties: correctness, simultaneity, fairness
- Joined by others: auctions, bidding, elections, e-commerce

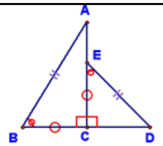
Classical Proofs

 . .
. .



. . .



Given: $\overline{AC} \perp \overline{BD}$
$\overline{BC} \cong \overline{EC}$
$\overline{AB}$ is not $\cong$ to $\overline{ED}$

Prove: $\angle B$ is not $\cong$ to $\angle CED$

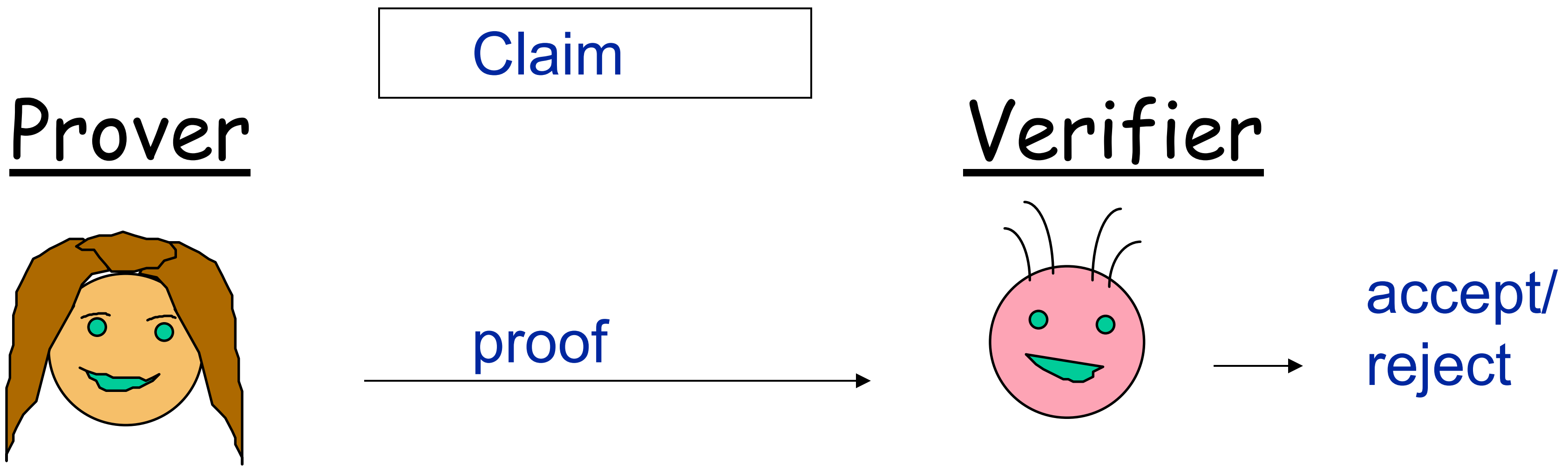
Statements	Reasons
A 1. Assume: $\angle B \cong \angle CED$	1. Assumption
2. $\overline{AC} \perp \overline{BD}$	2. Given
3. $\angle BCA$ and $\angle DCE$ are right $\angle$s	3. Defn. of $\perp$ segs
A 4. $\angle BCA \cong \angle ECD$	4. RAT
S 5. $\overline{BC} \cong \overline{EC}$	5. Given
6. $\triangle BCA \cong \triangle ECD$	6. ASA (1, 5, 4)
7. $\overline{AB} \cong \overline{ED}$	7. CPCTC
8. $\overline{AB}$ is not $\cong$ to $\overline{ED}$	8. Given

But statement 7 contradicts statement 8. Consequently, the assumption must be false.
$\therefore \angle B$ is not $\cong$ to $\angle CED$

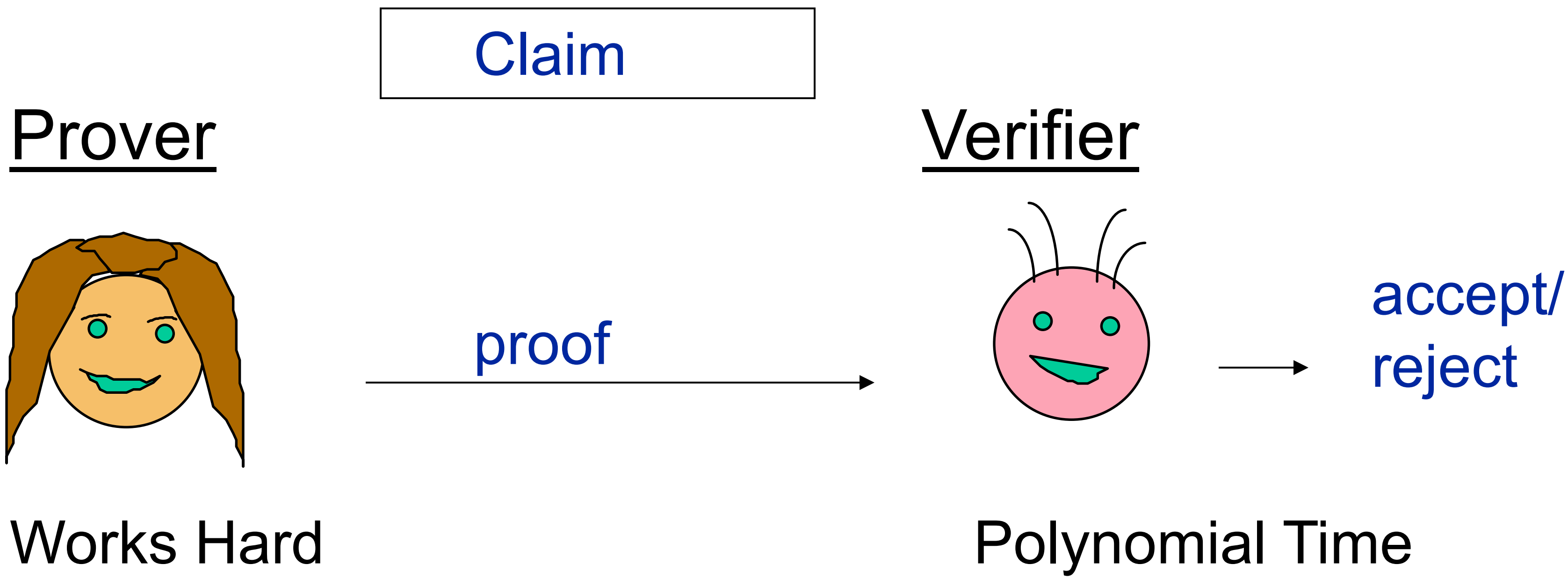
. . .

Prime-
Number Thm

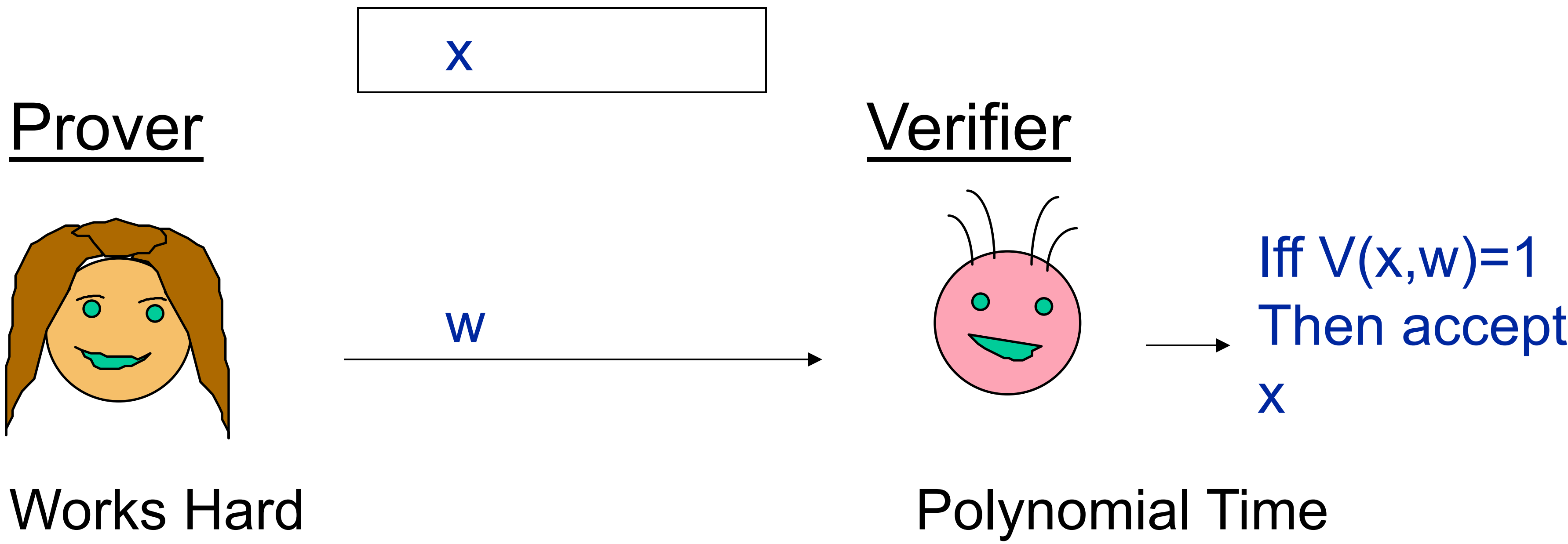
Proofs



Efficiently Verifiable Proofs (NP)

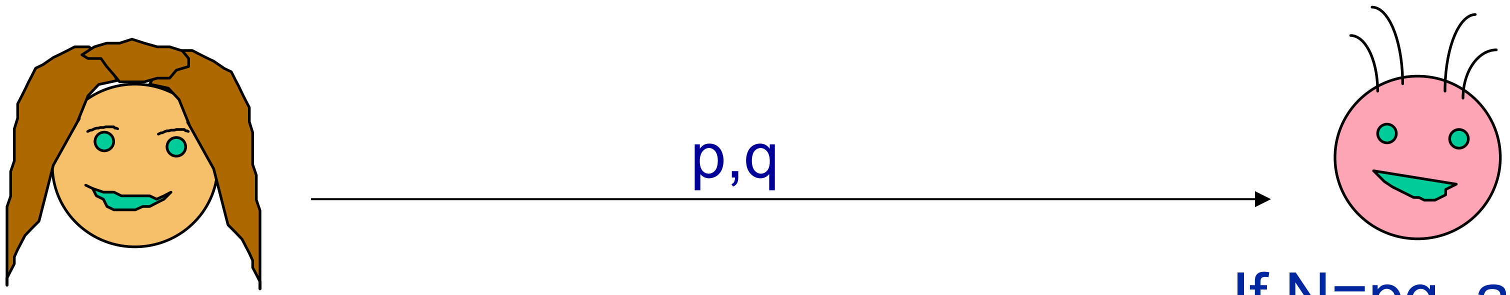


Efficiently Verifiable Proofs (NP)



NP = decision problems D for which there is a short and polynomial time verifiable proofs (witness)of x∈ D

Example: N is a product of 2 large primes



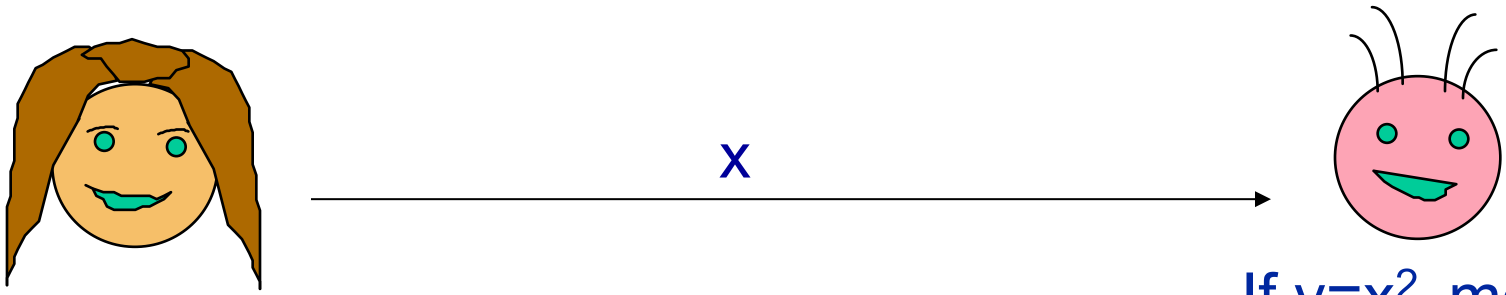
If N=pq, accept
Else reject

After interaction, Bob knows:

1) N is product of 2 primes

2) Also the factors of N

Example: y is a quadratic residue mod N (i.e $y=x^2$ mod N)



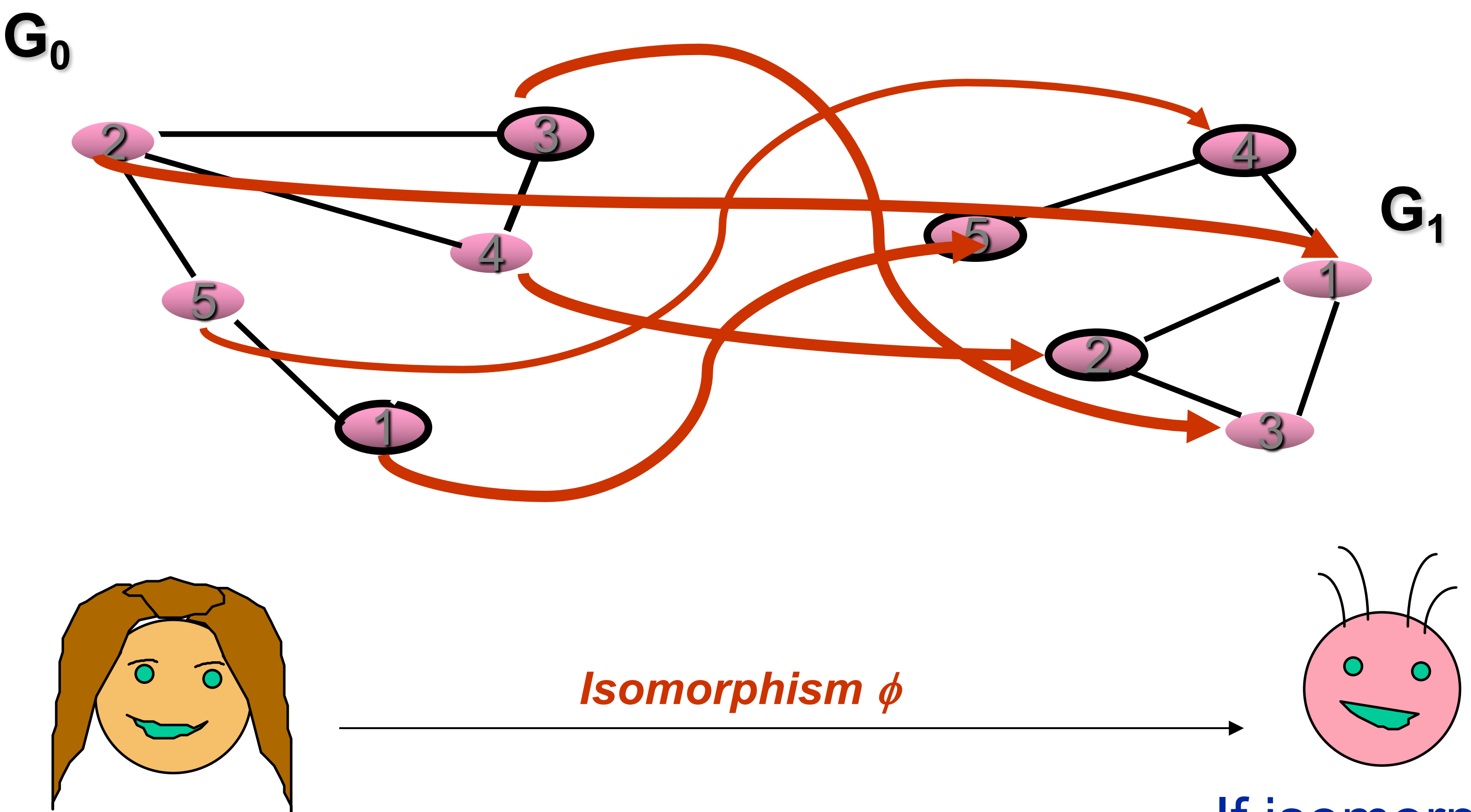
If $y=x^2$ mod N,
Accept
Else reject

After interaction, Bob knows:

1) y is a quadratic residue mod

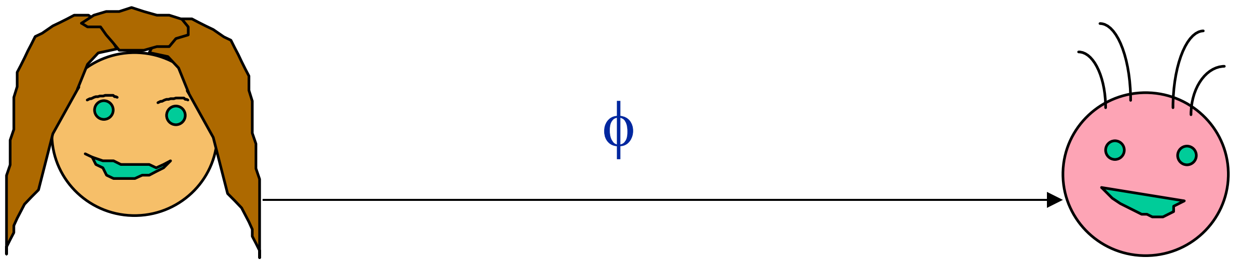
2) Square root of y

Example: G_0 is isomorphic to G_1



If isomorphism is good, accept
Else reject

G_0 isomorphic to G_1



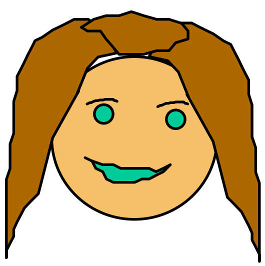
After interaction, Bob knows:

1) G_0 is isomorphic to G_1

2) Also the isomorphism

Is there any other way?

Zero Knowledge Proofs



Main Idea:

Prove that

I could prove it

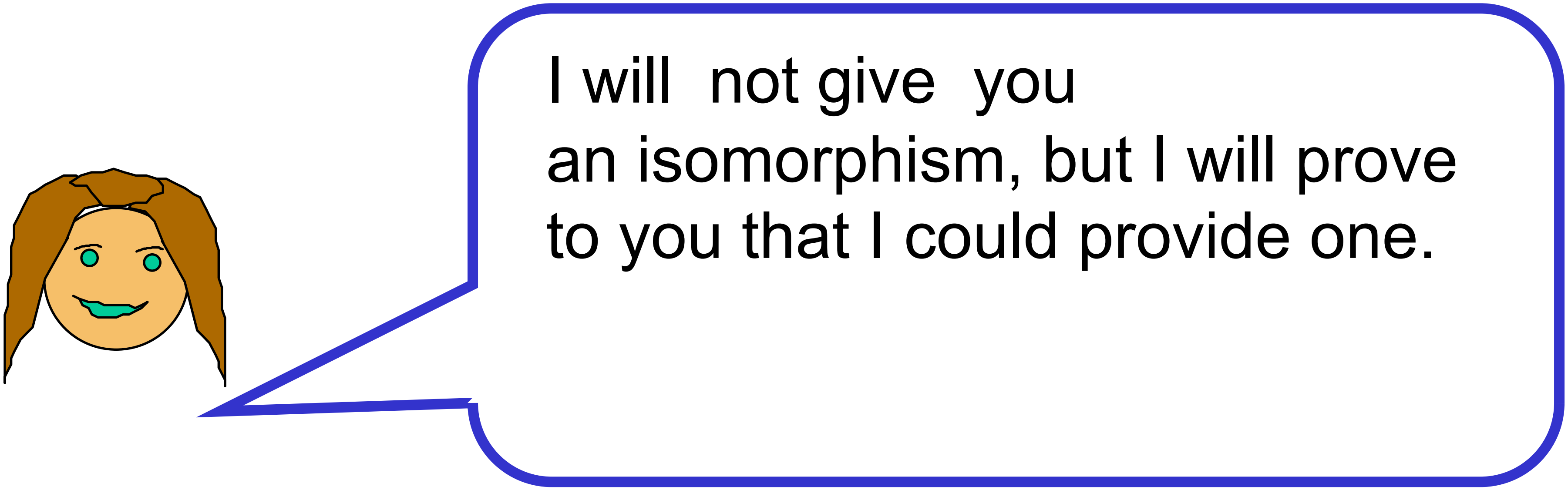
If I felt like it

Two New Ingredients

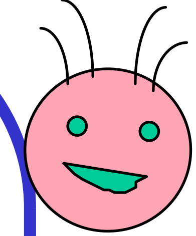
Interactive and Probabilistic Proofs

Non-trivial interaction: rather than "reading" proof, verifier engages in an non-trivial **interaction** with the **prover.**

Randomness: verifier is randomized (tosses coins as a primitive operation), and can err with some small probability


I will not give you
an isomorphism, but I will prove
to you that I could provide one.

HOW?



I will produce a random graph H for which

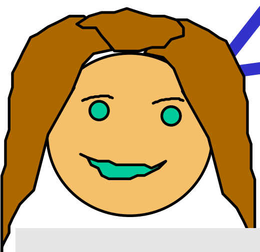
1: I can give you an isomorphism γ_0 from G_0 to H

OR

2: I can give you an isomorphism γ_1 from G_1 to H

Hence, there is an isomorphism σ from G_0 to G_1 directly

YOU randomly choose if I should demonstrate my ability to do #1 or #2.



Proof:

$H = \gamma_0(G_0)$,

$H = \gamma_1(G_1)$,

Thus

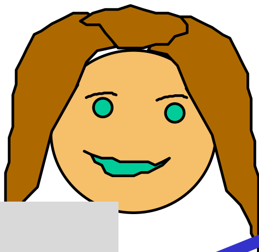
$G_1 = \gamma_1^{-1}(\gamma_0(G_0))$

$Set\ \sigma = \gamma_1^{-1} \cdot \gamma_0$

POINT IS: If I can do both, there exists an isomorphism from G_0 to G_1

REPEAT K INDEPDENT TIMES.

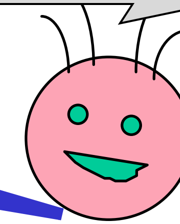
An Interactive Proof



Choose random γ_0 permutation of vertices of G_0. Set $H=\gamma_0(G_0)$

Graph H

Toss coin b



b

If b=0: send γ_0

If b=1: send $\gamma_0 \sigma^{-1}$ (where $\sigma(G_0)=G_1$)

Claims:

(1) Statement true $\Rightarrow$ can answer correctly for b= 0 and 1

(2) Statement false $\Rightarrow$ $prob_b$(catch a mistake) = $1-1/2^k$

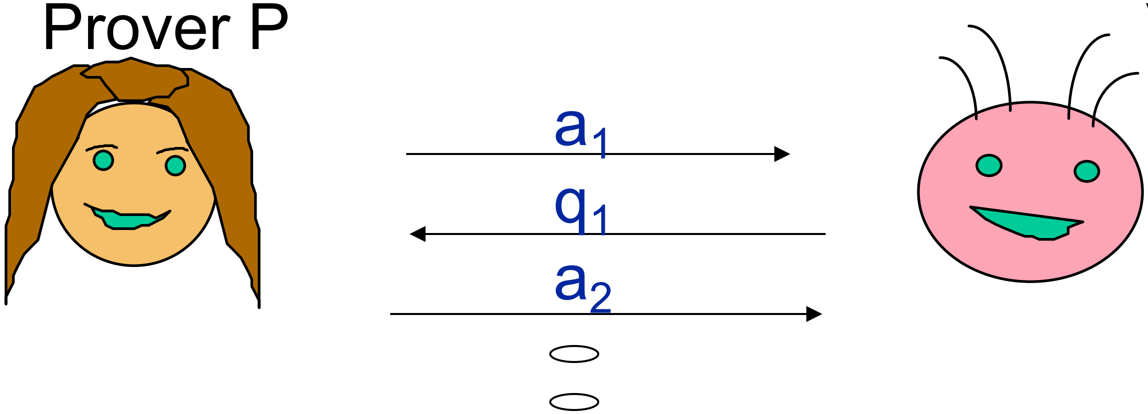
(3) Zero Knowledge (to be defined)

Interactive Proofs[GMR85]

Statement: T

Prover P

Verifier V



Probabilistic Polynomial time algorithm

Accepts /Rejects

(P,V) is an interactive proof system for T if

Completeness: if T is true, then V will always accept

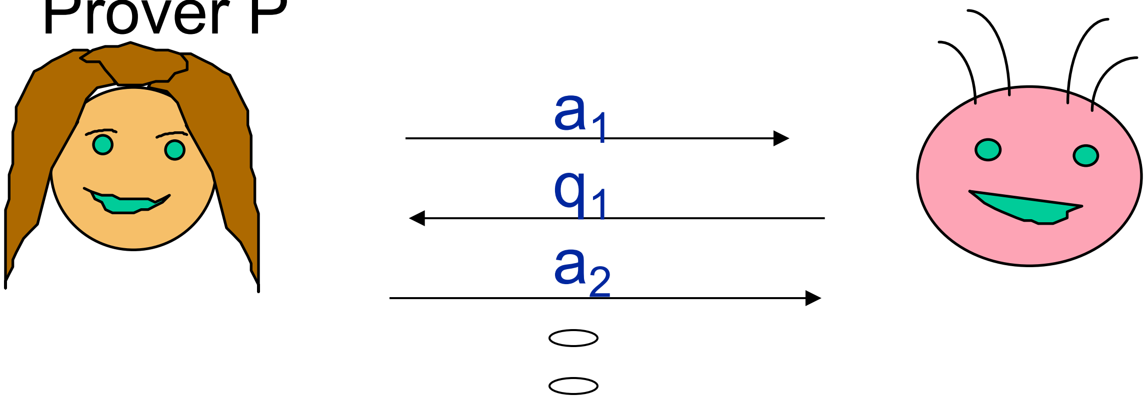
Soundness: if T is false, then regardless of prover P*strategy, V will reject with overwhelming probability

Interactive Proofs for Language Membership [GMR85]

Statement: $x \in L$

Prover P

Verifier V



Probabilistic Polynomial time algorithm

Accepts /Rejects

(P,V) is an interactive proof system for L if

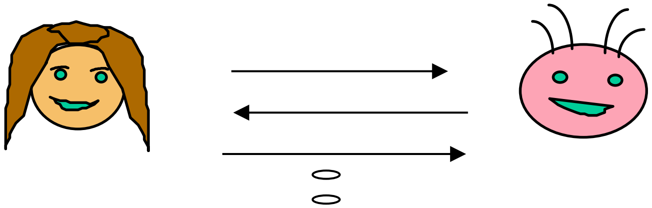
Completeness: if $x \in L$, then Prob[(P,V)[x]= accept]=1

Soundness: if $x \notin L$, then $\forall P^*$
Prob[(P*,V)[x]=accept]=neg (|x|)

Remarks: Interactive Proofs

Prover P

Verifier V



Probabilistic
Polynomial time

Accepts
/Rejects

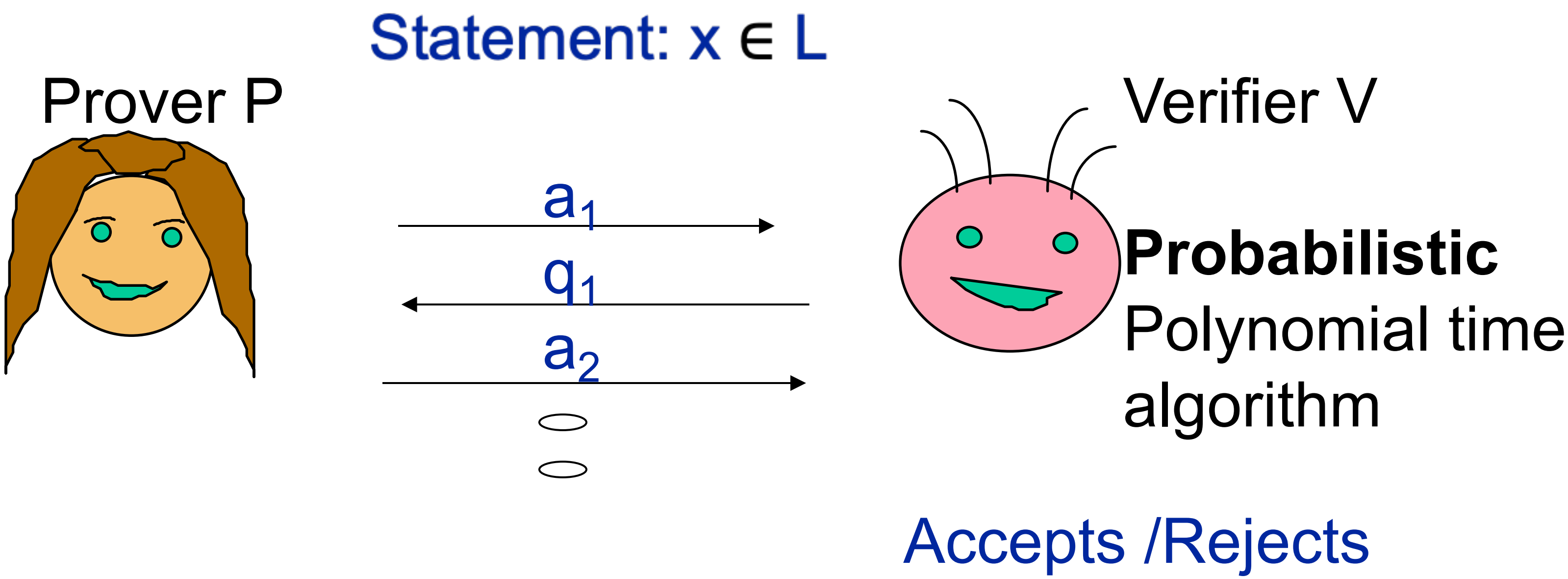
- P and V are a pair of interactive Algorithms, each having private inputs and private coins as well as a common public input.

- V additionally must run in polynomial time

- (P,V) satisfy completeness c(x) & soundness s(x) if
  $x \in L$, Prob((P,V)[x]= accepts)> c(x)
  $x \notin L$, $\forall P^*$, Prob[(P^*,V)[x]=accepts]<s(x)

- Suffice to require: c(x)=2/3 and s(x)=1/3

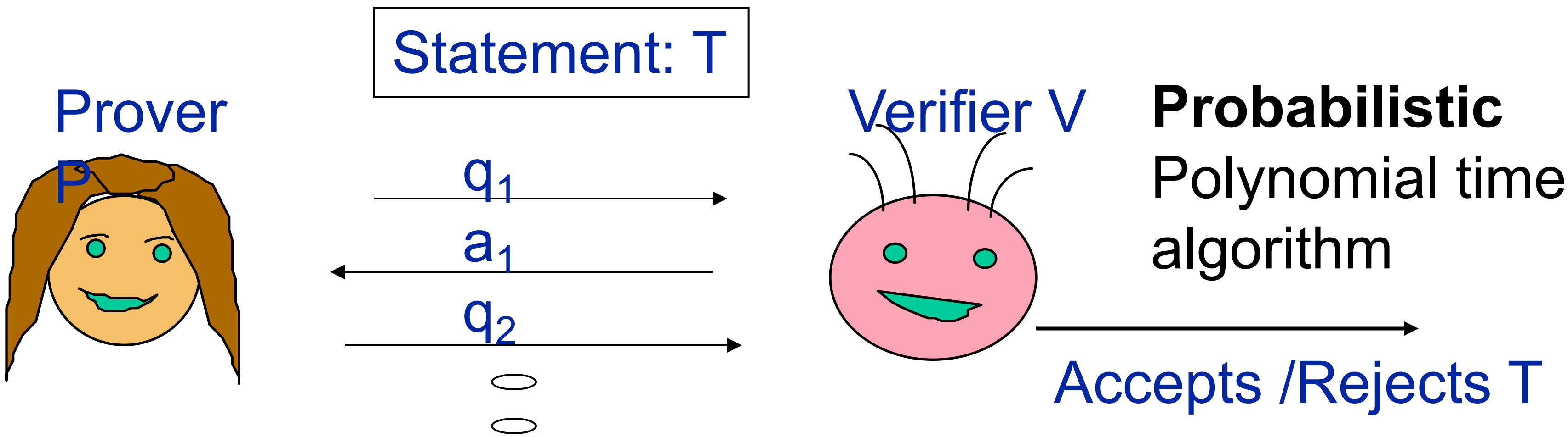
Class IP



IP = {L s.t. there exists (P,V) interactive proof system for L with completeness c(x)=2/3
and soundness s(x)=1/3}

Is IP greater than NP?

Zero Knowledge Interactive Proofs



After interactive proof, V "knows":

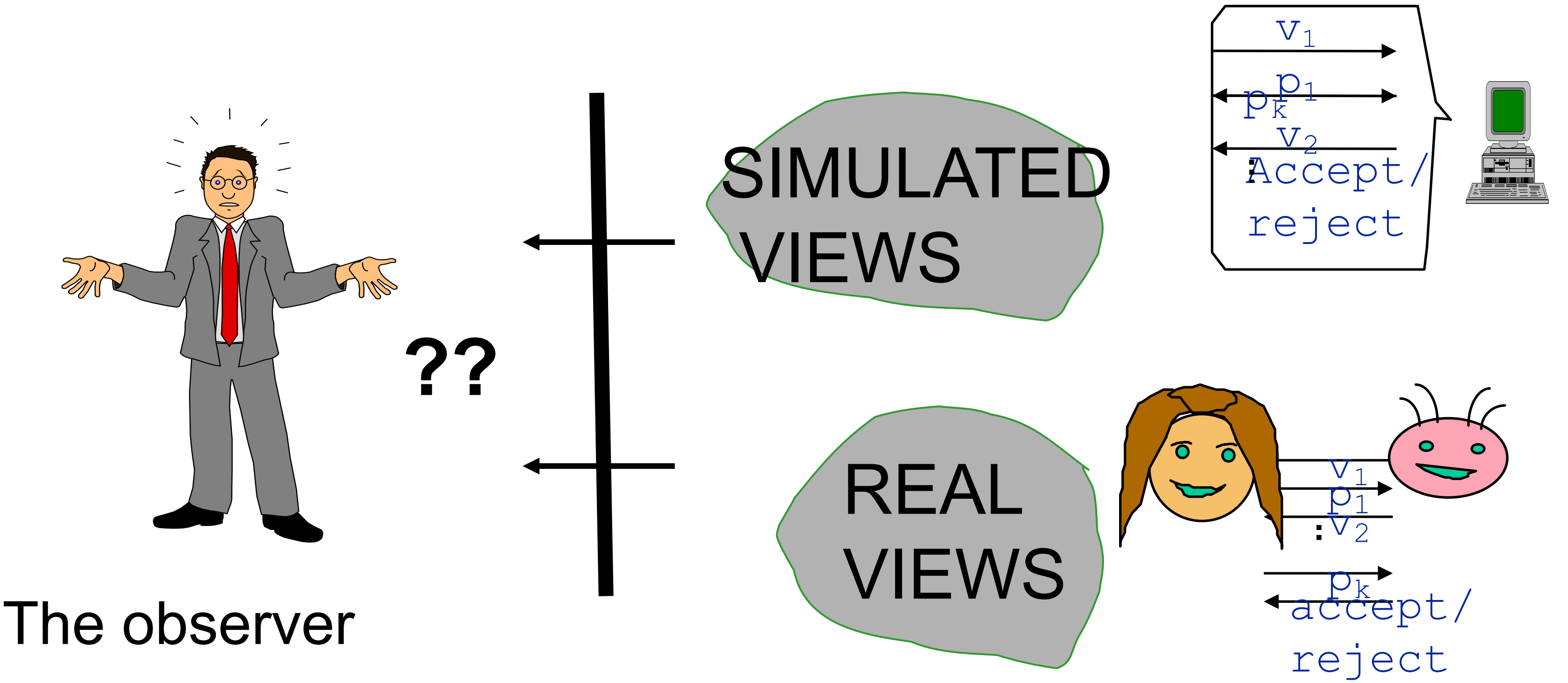
- T is true (or $x \in L$)
- A **view** of interaction (=transcript + coins V tossed)

P gives Zero- Knowledge to V: when T is true,

the **view** gives V nothing he couldn't have obtained on his own without interacting

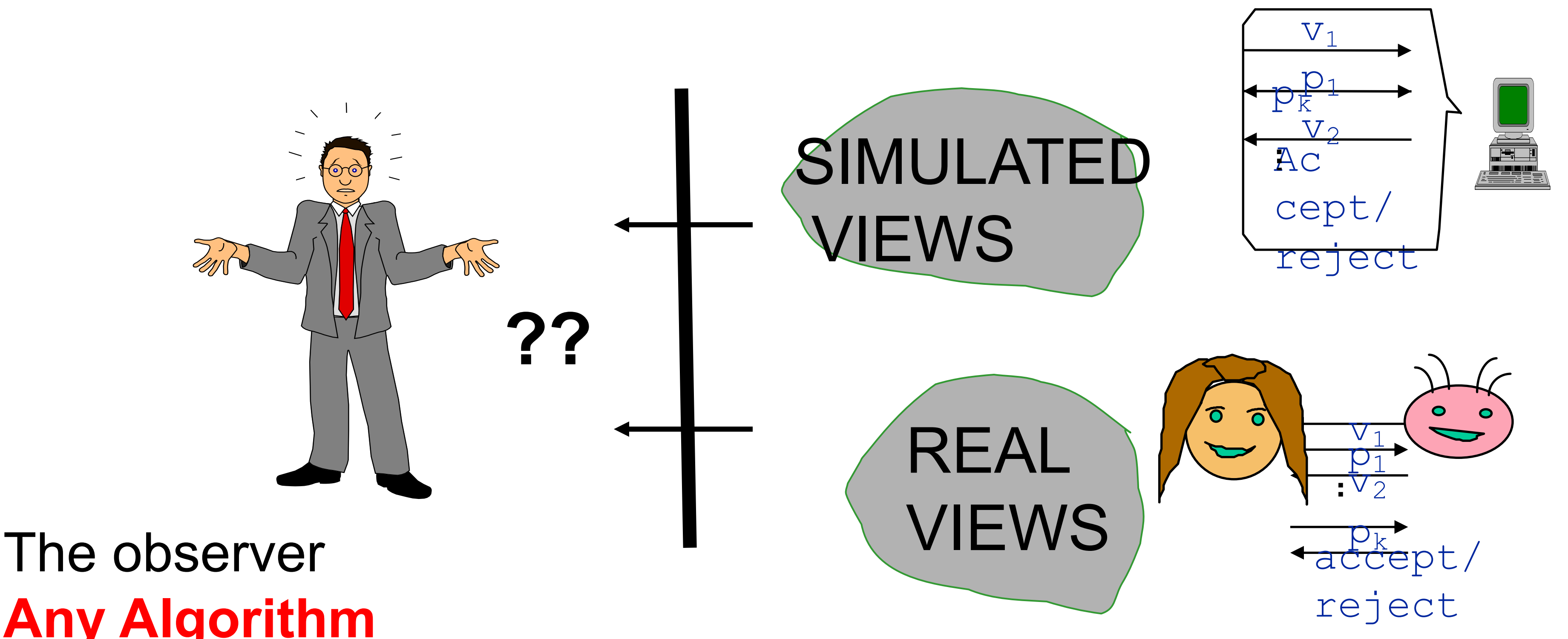
How Do we Capture Getting “Nothing Extra”(when T is true)

If: the verifier’s view can be efficiently simulated so that `simulated views’ and `real views’ are indistinguishable by an observer



Perfect Zero Knowledge (when T is true)

If: the verifier's view can be efficiently simulated so that `Simulated views' = `real views'



Formal Definition: Perfect Zero-Knowledge

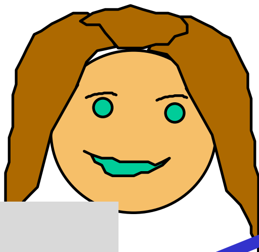
For a given P and V on input x, define probability space $View_{(P,V)}(x) = \{(q_1,a_1,q_2,a_2,\ldots,\text{coins of } V)\}$ (over coins of V and P)

(P,V) is **honest** verifier perfect zero-knowledge for L if:

$\exists$SIM a polynomial time randomized algorithm s.t. $\forall$x in L, $View_{(P,V)}(x) = SIM(x)$

Will allow SIM Expected polynomial time

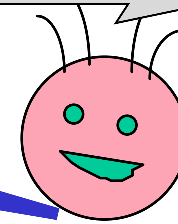
Recall: Isomorphism Example



Choose random γ_0 permutation of vertices of G_0. Set $H=\gamma_0(G_0)$

Graph H

Toss coin b



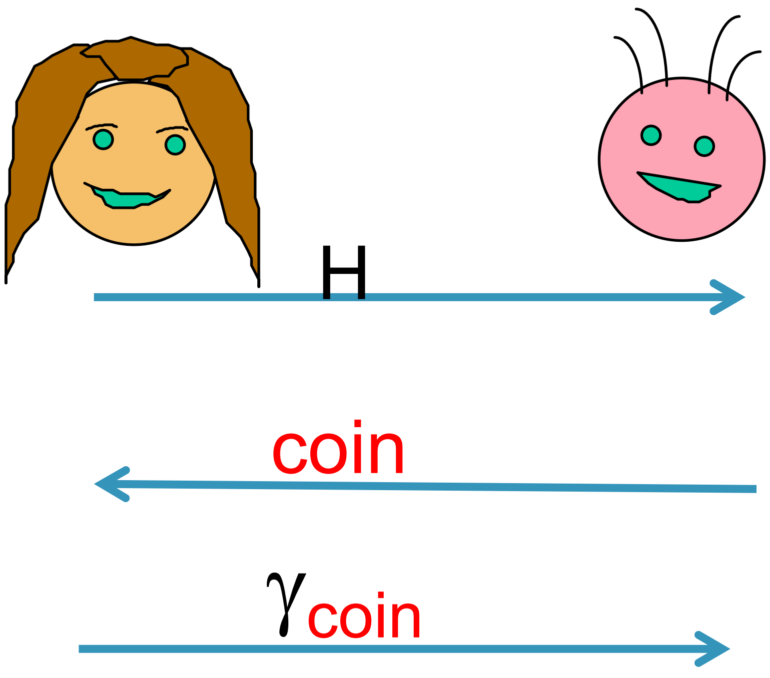
b

If b=0: send γ_0

If b=1: send $\gamma_0 \sigma^{-1}$ (where $\sigma(G_0)=G_1$)

View of Bob=
{(H, b, random isomorphism from G_b to H}

Zero Knowledge



SIMULATOR M:

- toss coin to
- If coin=head:
  choose random γ_0
  set $H = \gamma_0 (G_0)$
- If coin=tail
  choose random γ_1
  set $H = \gamma_1 (G_1)$

View of Bob=
{(H, coin, random isomorphism of G_b to H}

What if V is not honest: Perfect Zero-Knowledge (Final def)

For a given P and V on input x, define probability space $View_{(P,V)}(x) = \{(q_1,a_1,q_2,a_2,\ldots,coins)\}$ (over coins of V and P)

(P,V) is honest verifier perfect zero-knowledge for L if:

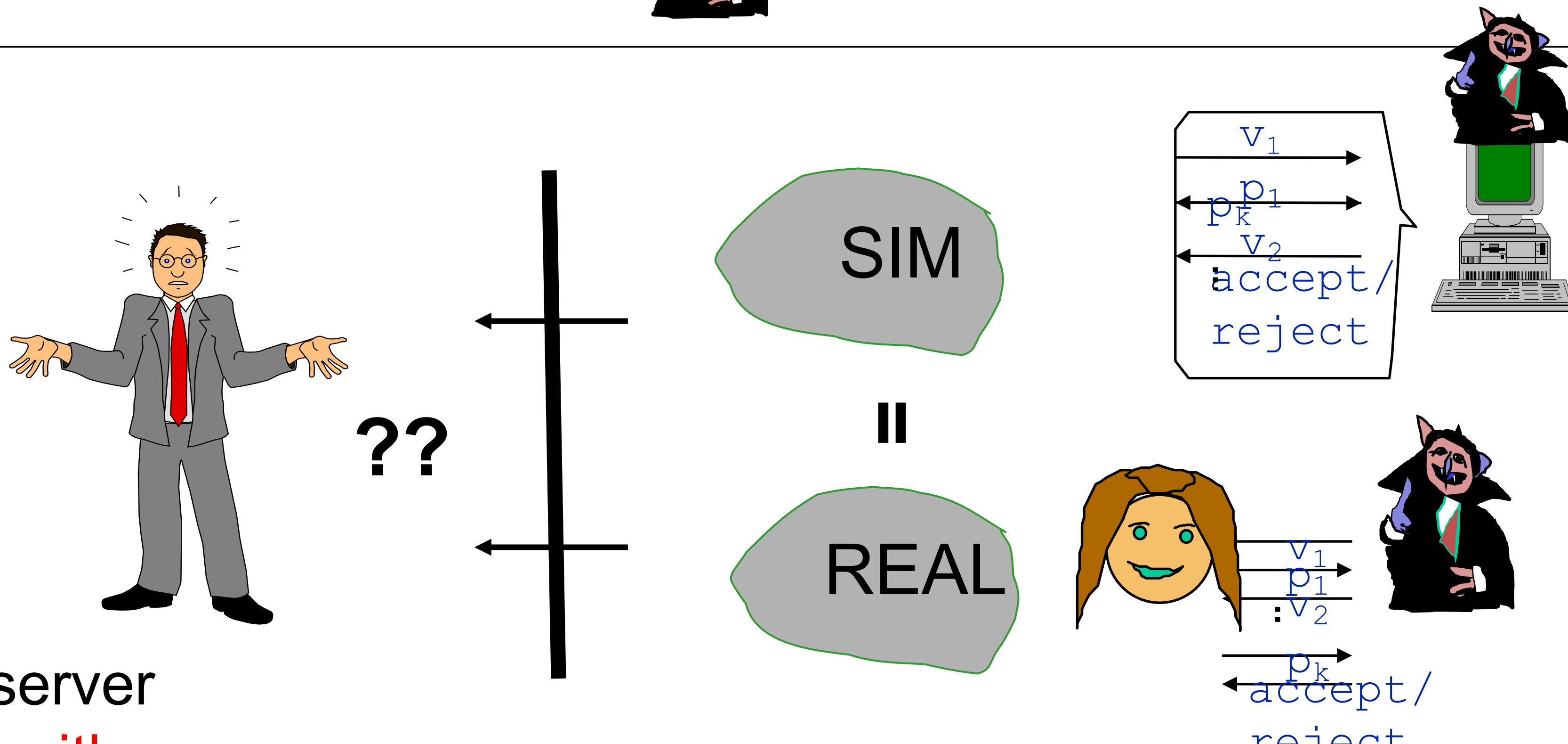
∃SIM an expected polynomial time randomized algorithm s.t. ∀x in L, $View_{(P,V)}(x) = SIM(x)$

(P,V) is perfect zero-knowledge for L if : **∀PPT V***

∃SIM an expected polynomial time randomized algorithm s.t. ∀x in L, $View_{(P,V^*)}(x) = SIM(x)$

Prover Gives **Perfect** Zero Knowledge

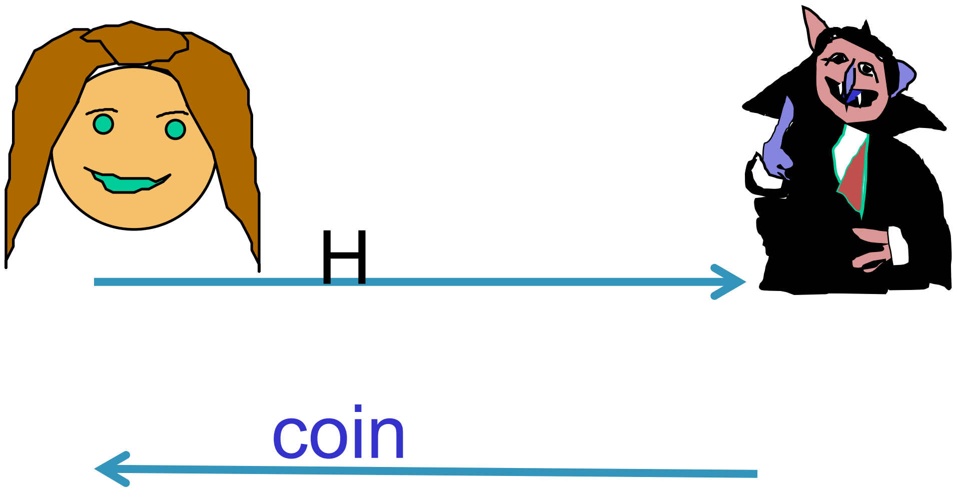
- If: we can efficiently simulate the view of any verifier s.t. `Simulated views' = `real verifier" for any poly time verifier 



The observer

Any Algorithm

Zero Knowledge Proof that G_1 isomorphic to G_2



if coin=coin. answer

Else abort and try again

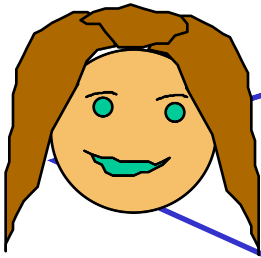
SIMULATOR SIM:

1. toss coin

2. If coin=head:
   choose random γ_0
   set $H = \gamma_0(G_0)$
   If coin=tail
   choose random γ_1
   set $H = \gamma_1(G_{21}$
3. Feed H to V*= 
4. If V* outputs
   coin==coin
   output (H, coin, γ_{coin})
   Else abort and
   goto 1 again.

Claim:
prob[coin=coin] = ½,
Expected [number of repetitions of SIM] = 2.
For k repetitions, SIM expected trials = 2k

Claim: $y = x^2 \mod N$ is solvable



Repeat 100 times ns

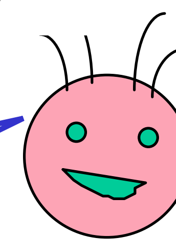
$z = [r^2 \bmod n]$

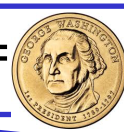
$zy = [(rx)^2 \bmod n]$

- If I gave you solutions to both, that is r and rx, you would be convinced that the claim is true but also know x
- Instead, I will give you a solution to only one equation, either r or rx but you ca choose which!

$1-(1/2)^{100}$

Choose 1<r<n at random

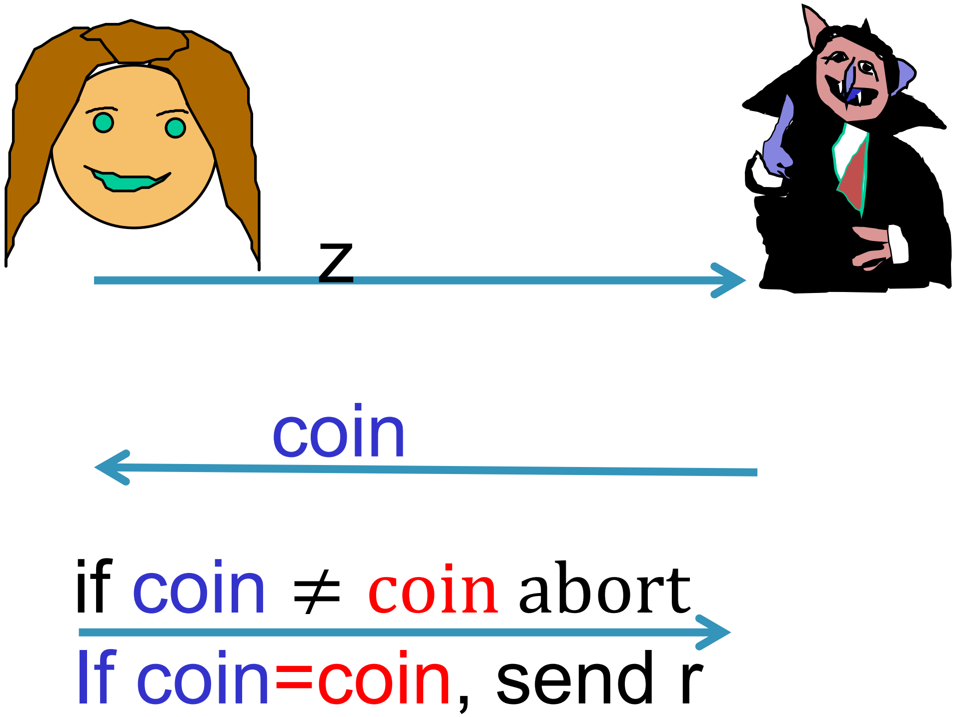


Flip a b=  to choose an equation

Gives a solution to the equation requested

Accepts claim only if gets correct solution

Zero Knowledge Proof that $Y=x^2 \bmod N$



SIMULATOR SIM:

1. toss coin

2. If coin=head:
   choose random r
   set $z=r^2 \bmod n$

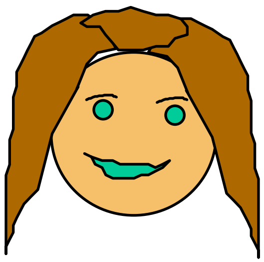
If coin=tail

choose random r

set $z=(ry^{-1})^2 \bmod n$

3. Feed z to V*= 
4. If V*(z) outputs coin≠coi
   abort and goto 1
   else for coin=head
   output(H, coin, r) &
   for coin=tail,
   output(H, coin, r)

Zero Knowledge Proof that $Y=x^2 \bmod N$



z

coin

if coin $\neq$ coin abort

If coin=coin, send r

SIMULATOR SIM:

1. toss coin

2. If coin=head:
   choose random r
   set $z=r^2 \bmod n$

If coin=tail

choose random r

set $z=(ry^{-1})^2 \bmod n$

3. Feed z to V*= 

4. If V*(z) outputs coin≠coin
   abort and goto 1
   ...or coin=head
   ...(H, coin, r) &
   ...n=tail,
   ...t(H, coin, r)

Claim:
prob[coin=coin] = ½,
Expected [number of repetitions of M] = 2.
For k repetitions, M expected trials = 2k

SIM: Expected Polynomial Time

- Analysis can be confusing
- Instead can change def to allow
  - SIM(x) to output ⊥ with probability at most 1/2 and require
  - View (x)= SIM(x) to be conditioned on the event that M(x) does not output ⊥
  - 1/2 can be relaxed to neg(x)

What Made it possible?

Randomness

- The statement to be proven has **many possible proofs** of which the prover chooses one ***at random***.
- Each such proof is made up of exactly 2 parts: seeing either part on its own gives the verifier no knowledge; seeing both parts imply 100% correctness.
- Verifier chooses **at random** which of the two parts of the proof he wants the prover to give him. The ability of the prover to provide either part, convinces the verifier

Recall, being able to quickly find a root of random number is equivalent to being able to factor n.

- Let A be an algorithm which can compute one root of a random input x.
- Pick r at random. Let $x=r^2$. $r_1 = A(x)$.
- With 50% chance r and r_1 are different and you can factor n. Repeat until n is factored.

Q: How to convert the proof that y is a quadratic residue to proving that you know the factorization of n

Actually, Alice seems to have proved more: that she actually "knows" the isomorphism (square root)

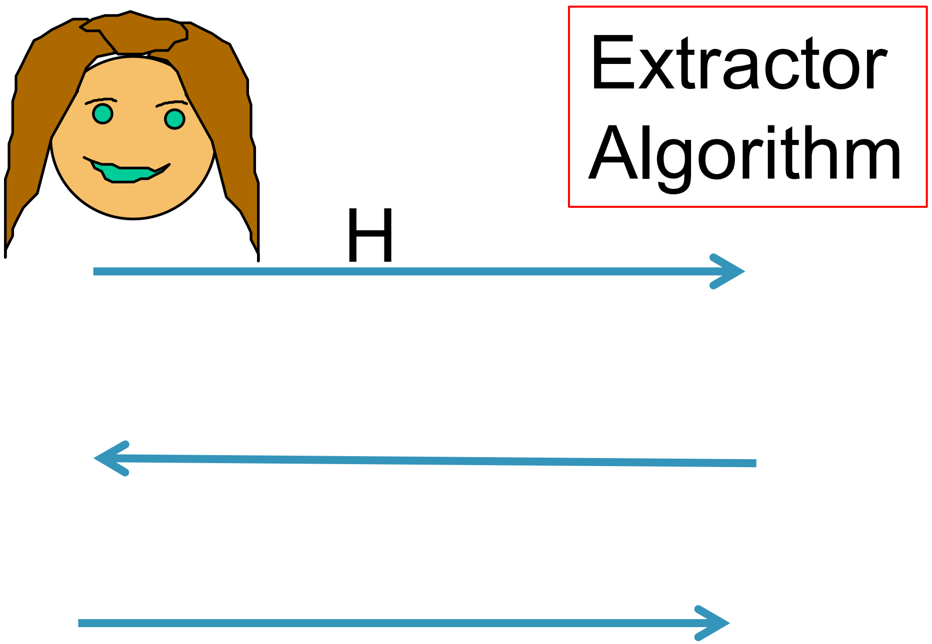
Let V be polynomial time relation. Let (x,w) ∈ V
V defines Language L_V= $\{x | \exists w \; s.t. \; V(x,w) = 1\}$.

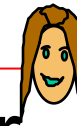
We say that (P,V) is a **proof of knowledge** for L_V
[or that P on x knows w] *if:*
∃an extractor algorithm E s.t. for all x
$E^P(x)$ outputs *w in expected polynomial time*

ZKPOK: zero knowledge proof of knowledge

This is called the rewinding technique

ZKPOK that Prover knows an isomorphism from G_1 to G_2



Extractor  :

1) On input H
   set coin=head
   Store γ_0
2) Rewind and 2nd time
   set coin=tail
   Store γ_1
3) Output $\gamma_1^{-1}(\gamma_0)$

ZKPOK

Let V be polynomial time relation. Let (x,w) ∈ V
V defines Language L_V= $\{x | \exists x \ s.t. R(x,w) = 1\}$.

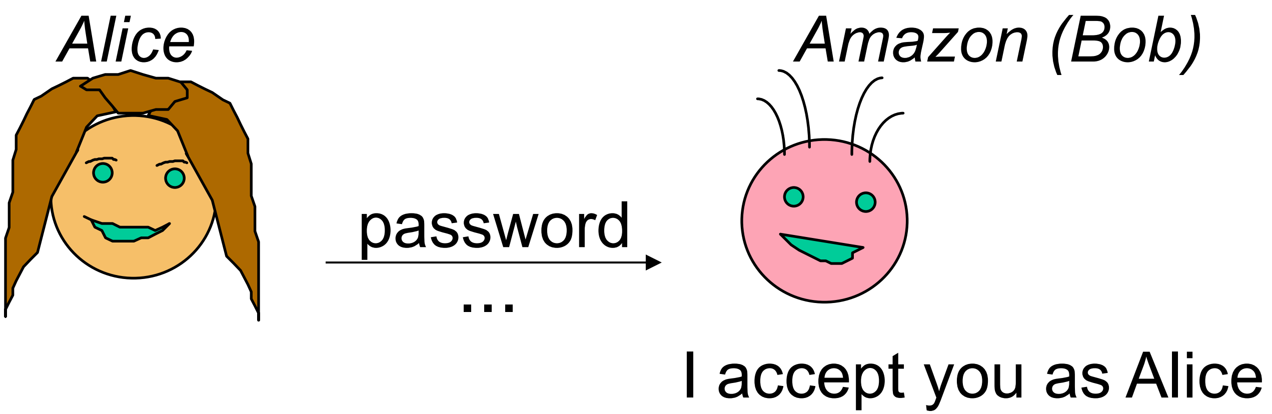
We say that (P,V) is a **proof of knowledge** for L_R
[or that P on x knows w] *if:*
∃an extractor algorithm E s.t. for all x and **for all P',**

If Prob[(P',V)[x] = accepts] = α, Then
E^P (x) outputs *w in expected polynomial time (|x|, 1/α)*

Why did we disturb the classical notion of proof ?

- Preventing Identity Theft
- Proving Properties of secrets
- Can verify statements **not verifiable** efficiently with classical NP proofs
- Secure Protocols

Classicial Passwords: Identity Theft

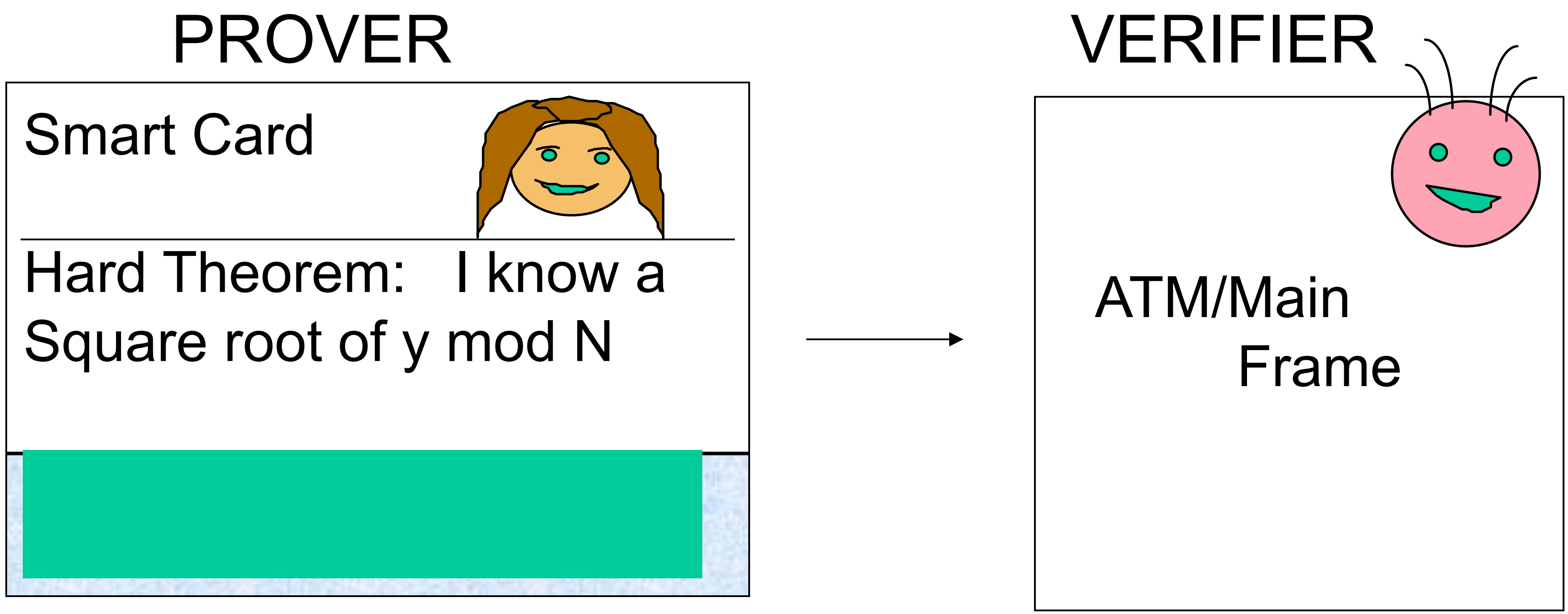


For Settings:

- Alice = Smart Card.
- Over the Net

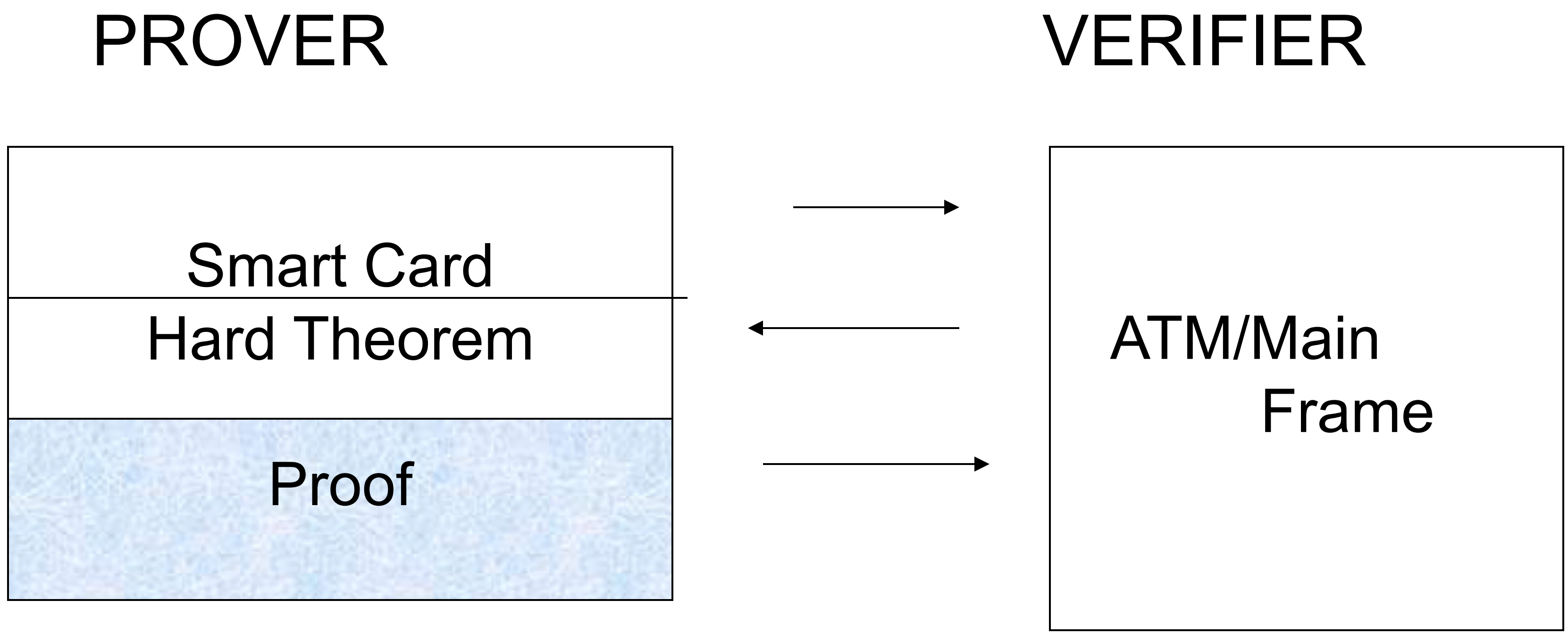
Passwords are no good

Zero Knowledge: Preventing Identity Theft



To identify itself prover proves that he knows a proof of the theorem.

More generally,

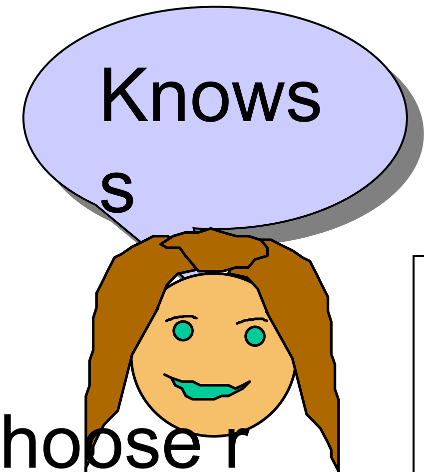


To identify itself Prover proves in zero-knowledge it knows a proof of the hard theorem.

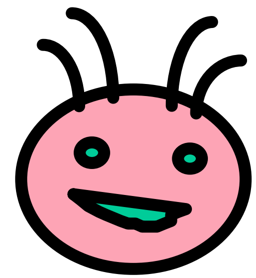
Schnorr Identification

Let G be a a cyclic group of prime order q,
Let both prover and verifier know y in G and

Claim: (P,V) is ZKPOX for the discrete log of y



Input: g, y



1. Choose r At random In Z_q

$R = g^r \bmod p$ →

2. Choose c At random in {0,1}

← c

3. Let z=r+cs

z=r+cs mod q →

4. Accept iff $g^z = R y^c \bmod p$,