ON ZERO-KNOWLEDGE PROOFS

MARTA BELLÉS MUÑOZ

SEMINARI INFORMAL DE MATEMÀTIQUES DE BARCELONA

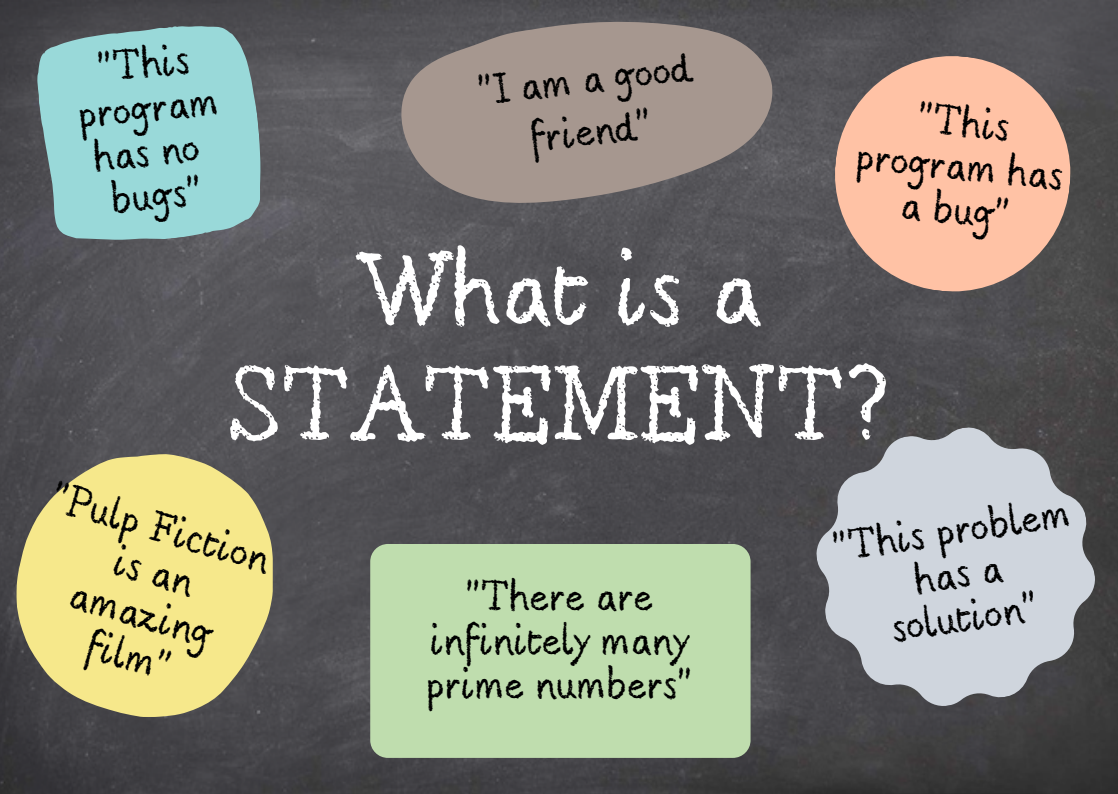
15TH DECEMBER 2021

SIMBa

ZERO-KNOWLEDGE PROOF

What is a PROOF?

Evidence or argument establishing a fact or the truth of a statement.


"This program has no bugs"
"I am a good friend"
"This program has a bug"
What is a STATEMENT?
"Pulp Fiction is an amazing film"
"There are infinitely many prime numbers"
"This problem has a solution"

" There are infinitely many prime numbers "

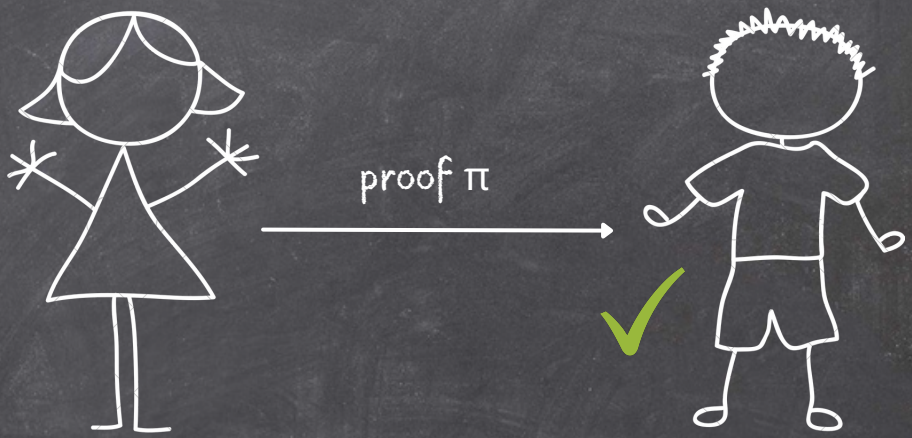
Proof

Assume there are only n prime numbers, denote them p1,...,pn. Consider the number

$$p = p1 \times p2 \times p3 \times \ldots \times pn + 1.$$

Clearly, p is greater than all primes, so it can't be one of them. Thus, it must be divisible by at at least one of them, say pi for some i. But when we divide p by pi we get reminder 1, which contradicts our assumption. Hence, the statement is true.

PROOFS



ZERO-KNOWLEDGE PROOFS

1. **Completeness**
   An honest prover can convince a verifier.

2. **Soundness**
   A malicious prover cannot convince a verifier.

3. **Zero-knowledge**
   The verifier learns nothing beyond the fact that the statement is true.

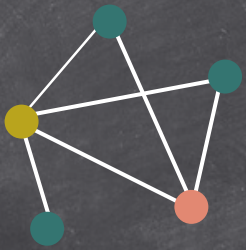
A PRATICAL EXAMPLE

~~ The 3-coloring problem ~~

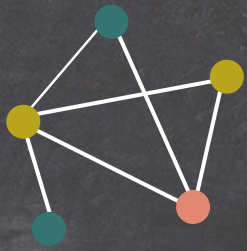
THE 3-COLORING PROBLEM

Definition (coloring)

A coloring of a graph is a labeling of their vertices with colors such that no two adjecent vertices sharing the same edge have the same color.



Coloring ✓



Coloring ✗

THE 3-COLORING PROBLEM

Definition (coloring)

A coloring of a graph is a labeling of their vertices with colors such that no two adjecent vertices sharing the same edge have the same color.

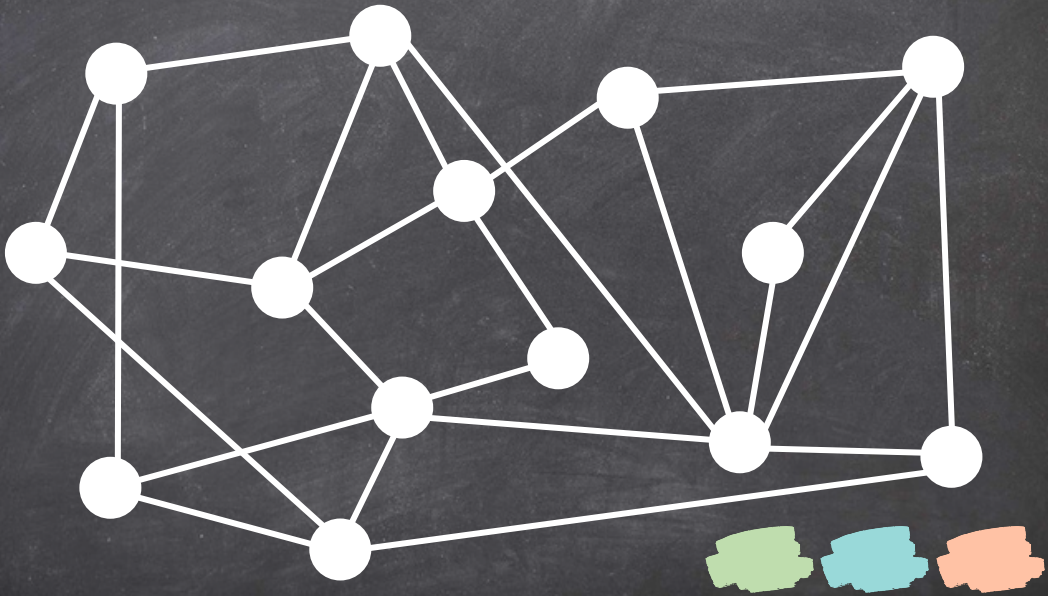
Theorem (4-coloring)

Every planar map can be colored with four colors.

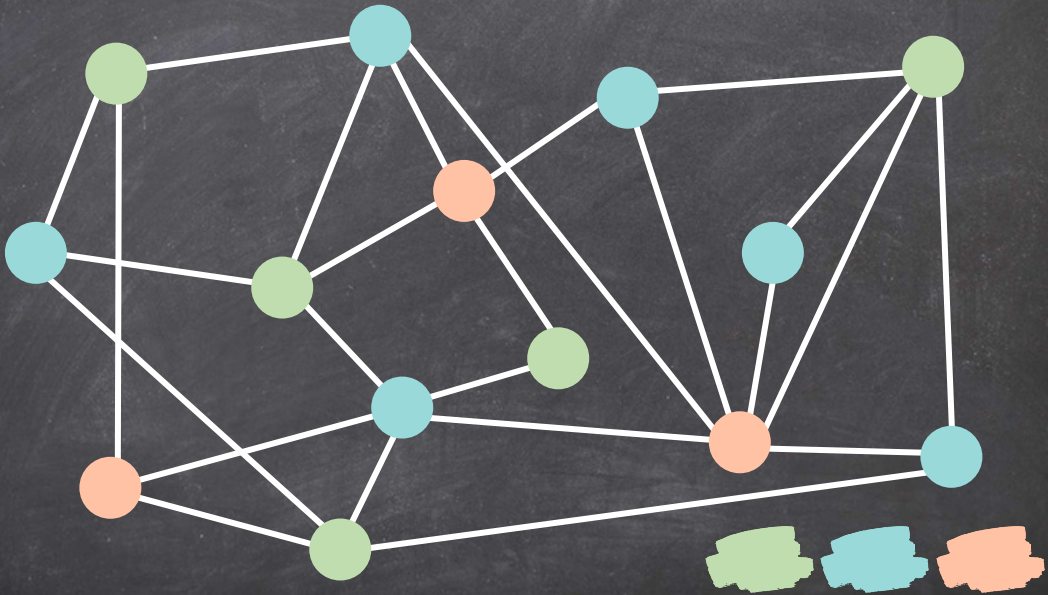
Problem (3-coloring)

It is NP-complete in complexity to decide whether an arbitrary planar map can be colored with just three colors.

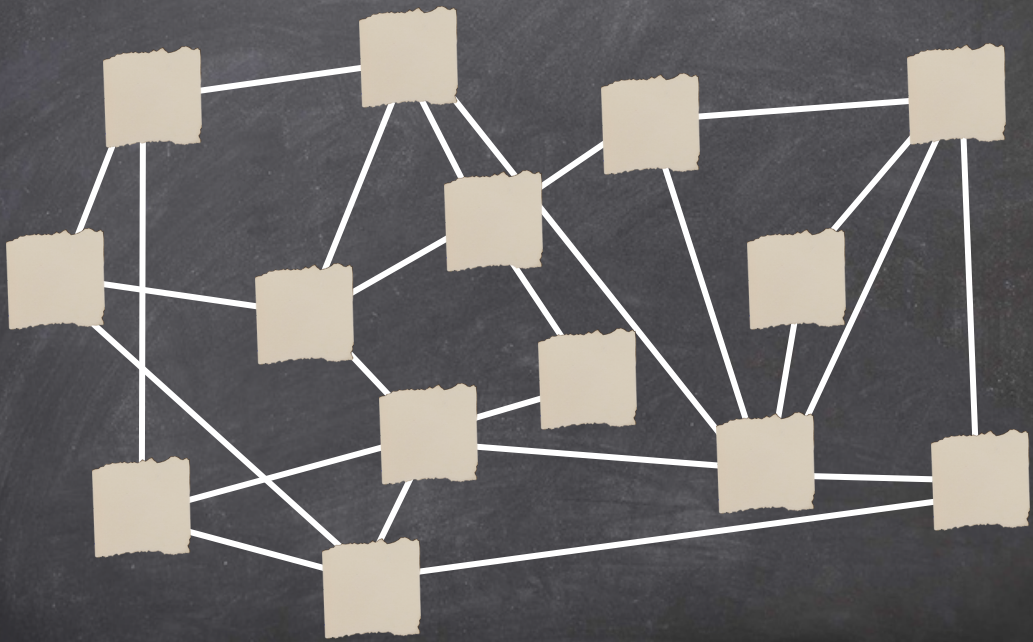
THE 3-COLORING PROBLEM



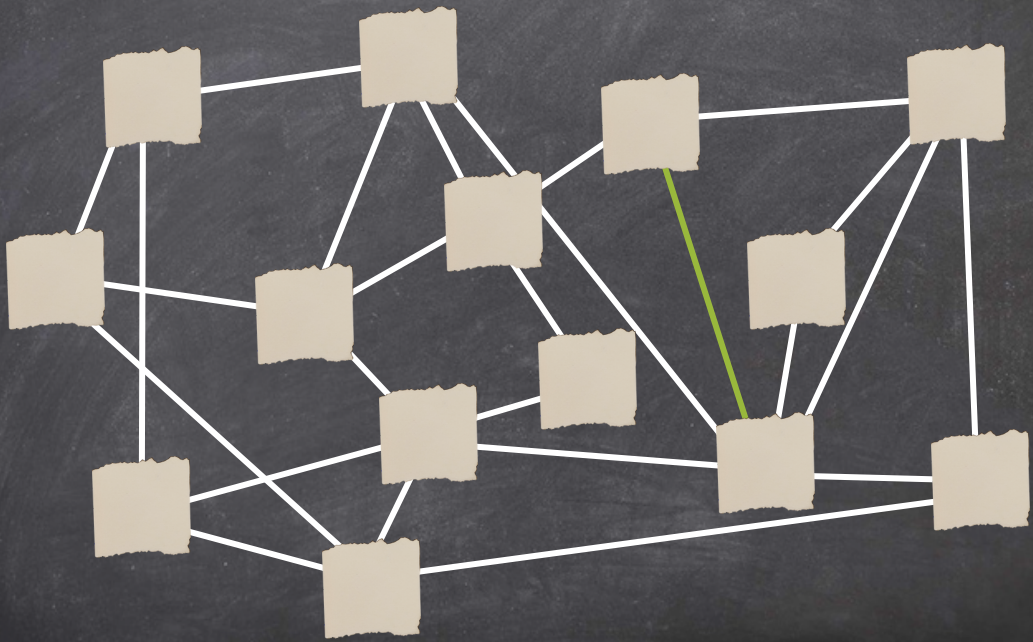
THE 3-COLORING PROBLEM



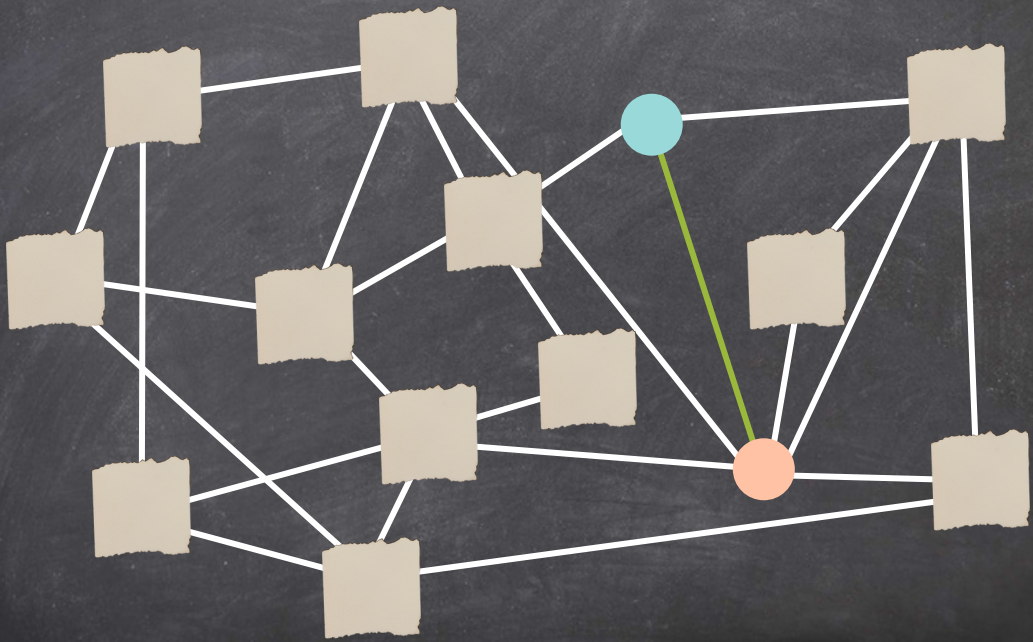
THE 3-COLORING PROBLEM



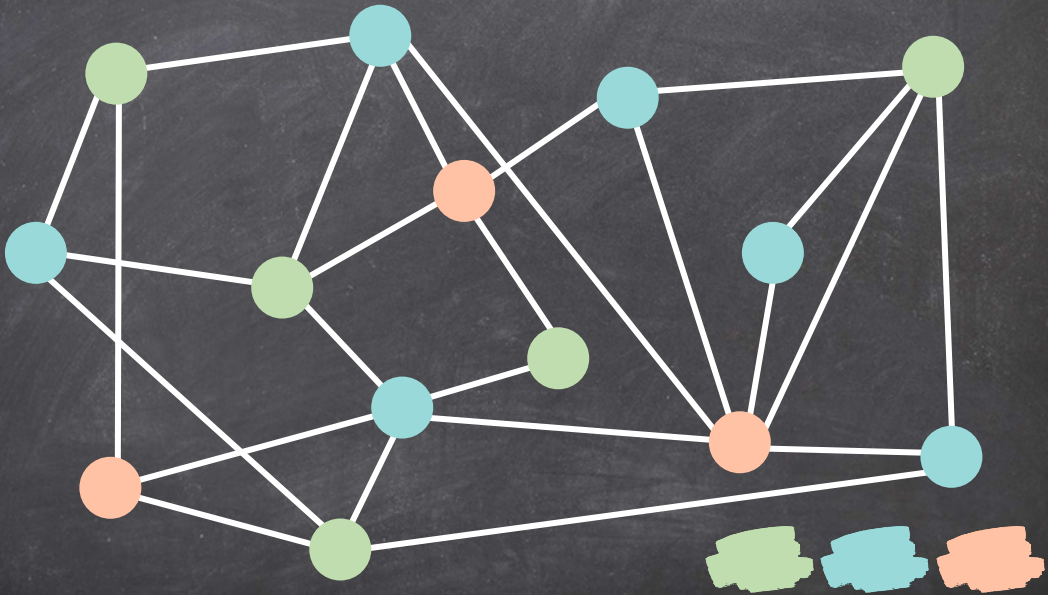
THE 3-COLORING PROBLEM



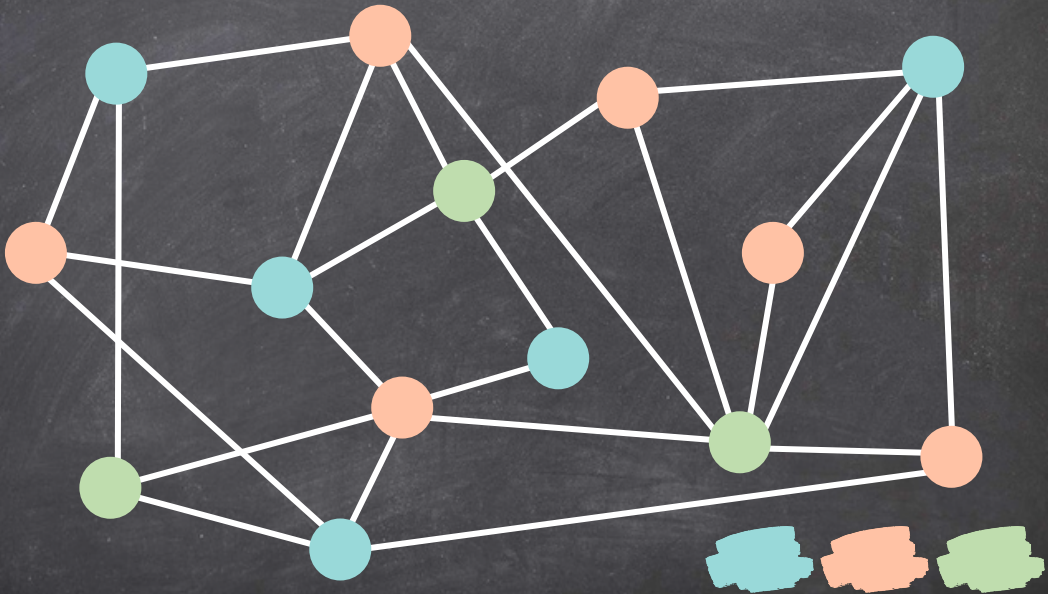
THE 3-COLORING PROBLEM



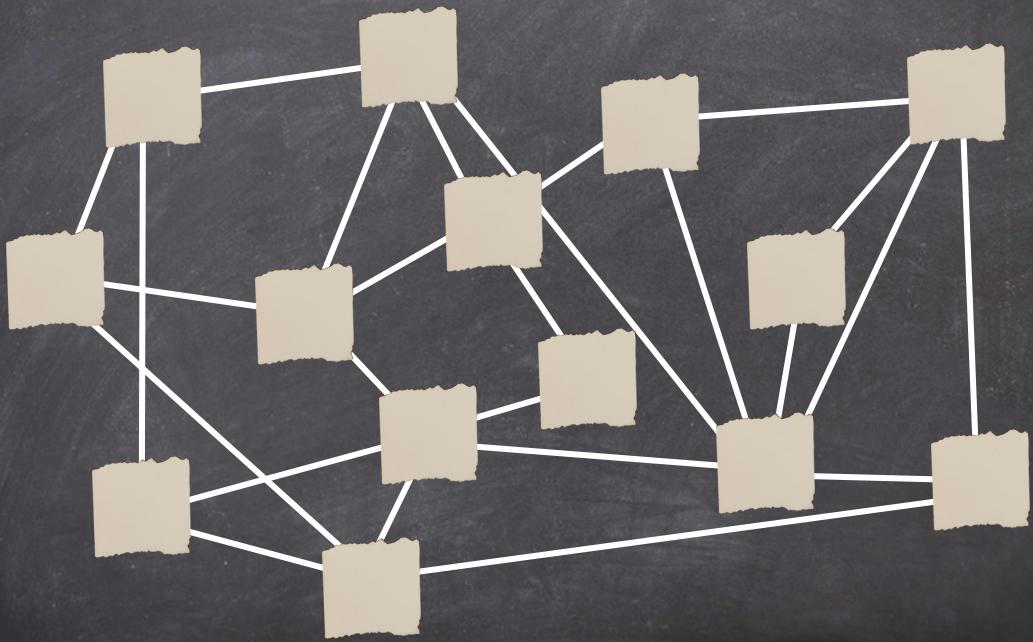
THE 3-COLORING PROBLEM



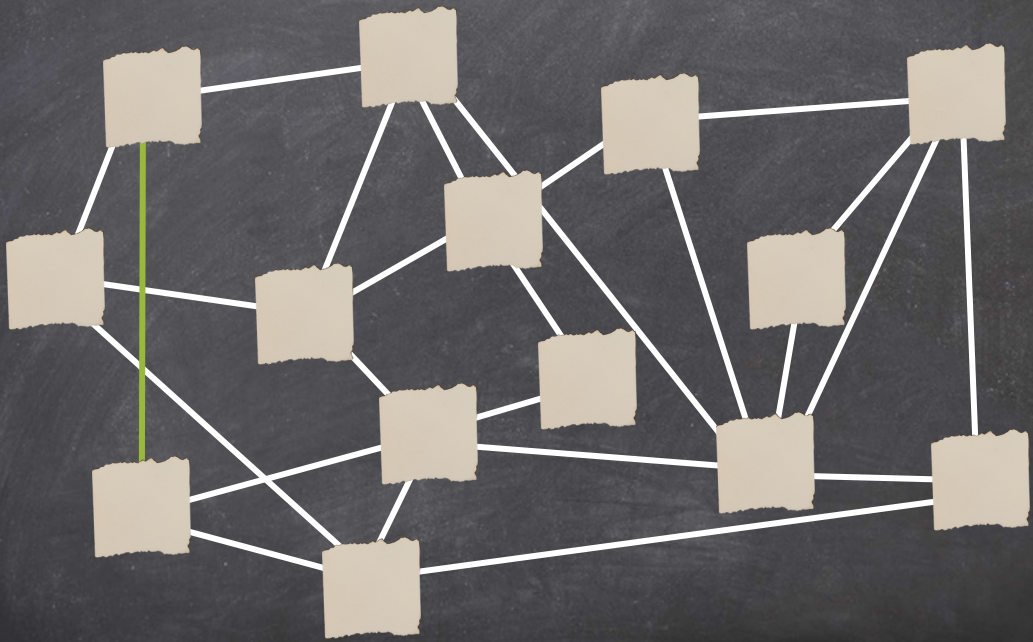
THE 3-COLORING PROBLEM



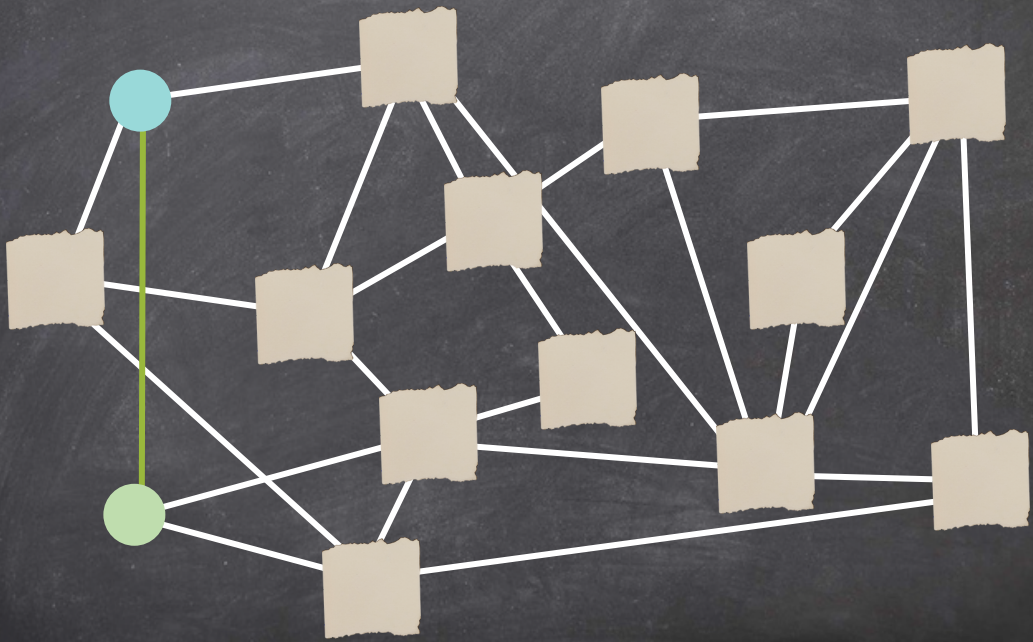
THE 3-COLORING PROBLEM



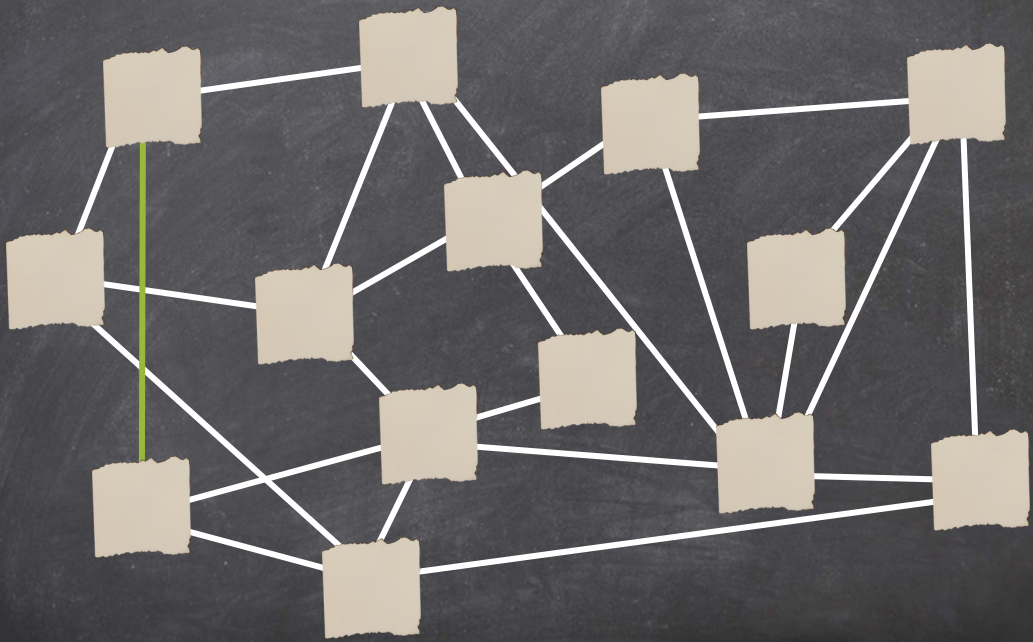
THE 3-COLORING PROBLEM



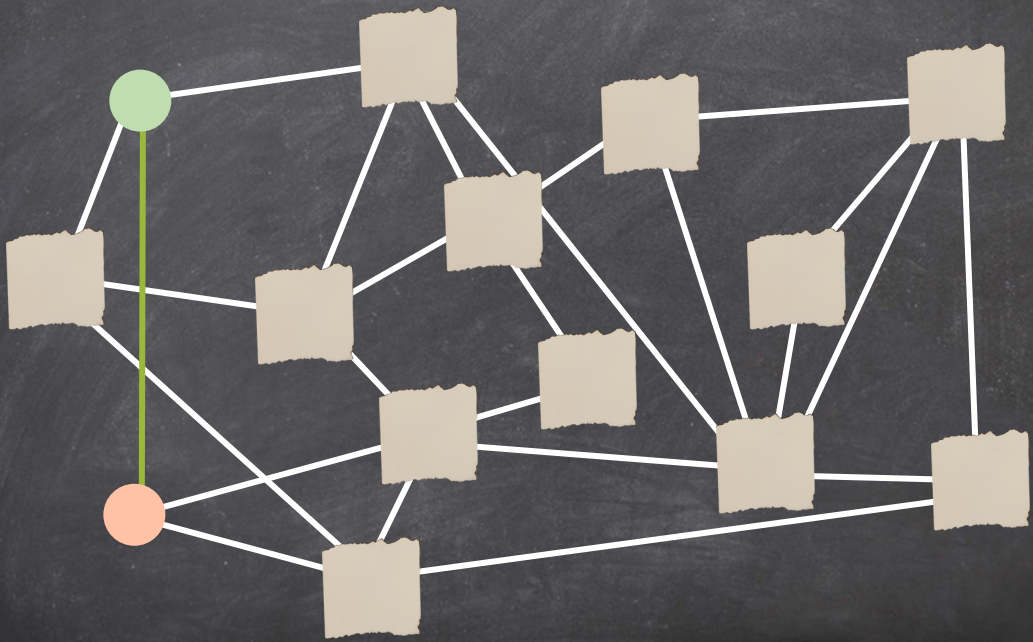
THE 3-COLORING PROBLEM



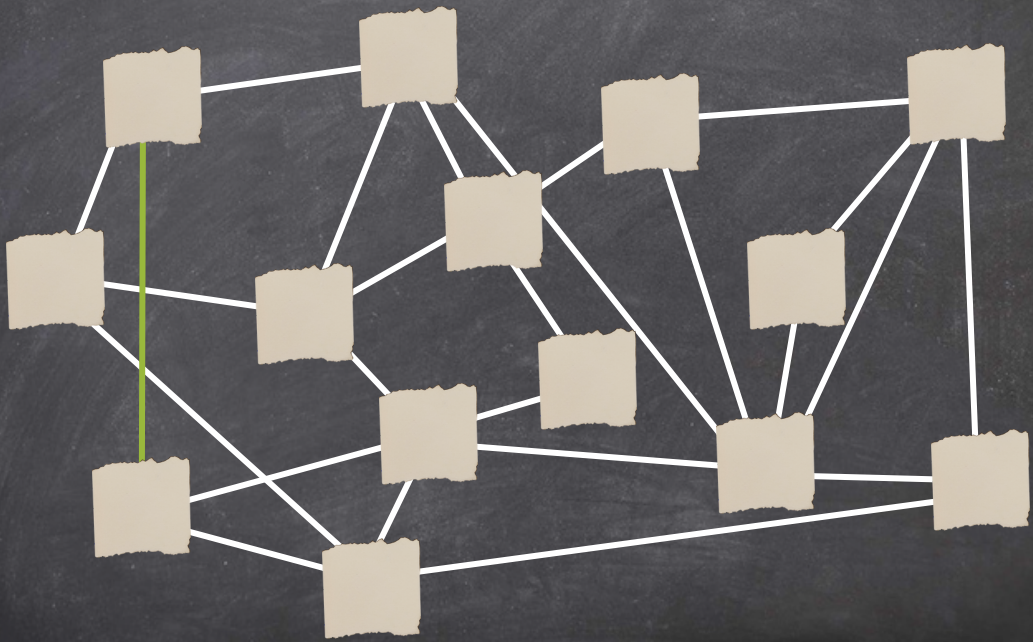
THE 3-COLORING PROBLEM



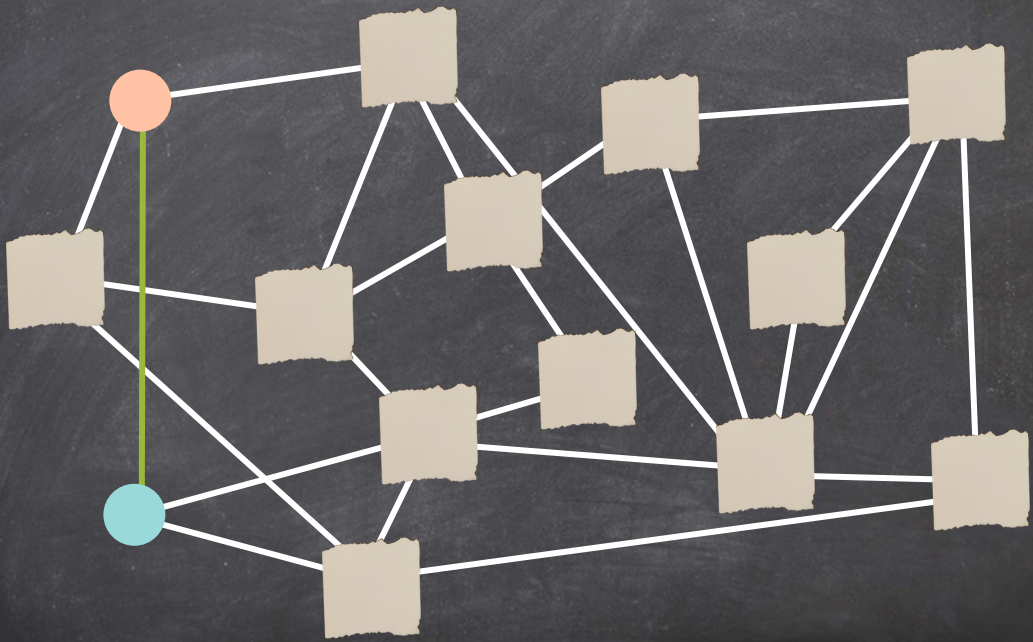
THE 3-COLORING PROBLEM



THE 3-COLORING PROBLEM



THE 3-COLORING PROBLEM



ZERO-KNOWLEDGE PROOFS

1. **Completeness**
   An honest prover can convince a verifier.

2. **Soundness**
   A malicious prover cannot convince a verifier.

3. **Zero-knowledge**
   The verifier learns nothing beyond the fact that the statement is true.

ZERO-KNOWLEDGE PROOFS

1. Completeness

   An honest prover can convince a verifier. ✓

2. Soundness

   A malicious prover cannot convince a verifier ...except with some small probability.

3. Zero-knowledge

   The verifier learns nothing beyond the fact that the statement is true. ✓

PUBLIC KEY CRYPTOGRAPHY

Public parameters

prime p, generator g of Z_p^*

Pair of keys

secret s, public key $y = g^s \bmod p$

No efficient classical algorithm is known for computing discrete logarithms in general.

Schnorr Identification Protocol

Peggy

$(s, y = g^s \bmod p)$

generates random k

$x = g^k \bmod p$

random r

$t = k - sr$

$x = g^t y^r$ verifies

$g^k = g^{k - sr} (g^s)^r$

Victor

Schnorr Identification Protocol

Peggy

$(s, y = g^s \bmod p)$

generate random t

$x = g^t y^{-r} \bmod p$

random r

t

$x = g^t y^r$ verifies

Victor

Fake proof without s!

Schnorr Identification Protocol

Peggy

$(s, y = g^s \bmod p)$

Peggy convinced Victor that she knows s without revealing it!

Victor

Schnorr Identification Protocol

Peggy convinced Victor
that she knows a
discrete logarithm
without revealing it!

Peggy

$(s, y = g^s \bmod p)$

Victor

The Schnorr Identification Protocol

is a Zero-Knowledge protocol for proving discrete logarithms!!

Peggy

$(s, y = g^s \bmod p)$

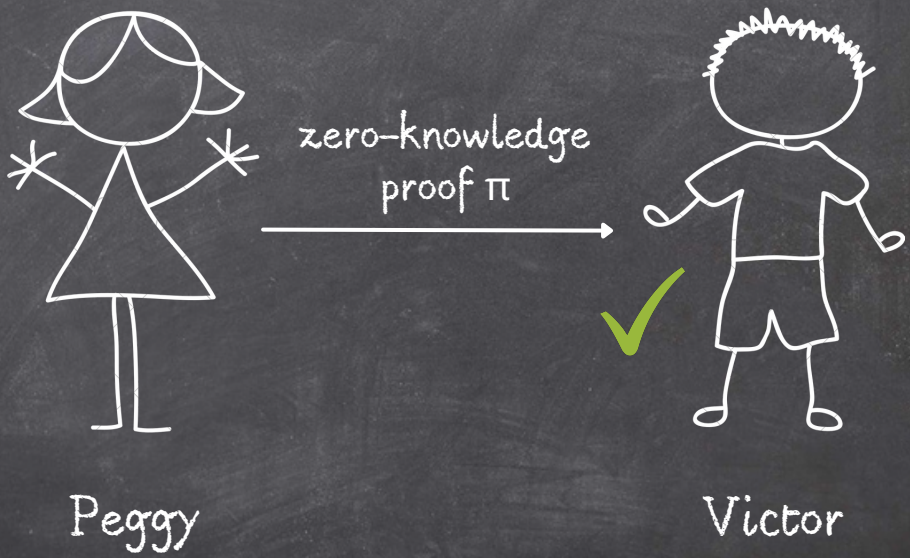
Victor

ZERO-KNOWLEDGE PROOFS

NON-INTERACTIVE
ZERO-KNOWLEDGE PROOFS

Fiat-Shamir transformation
(with hash functions)

NON-INTERACTIVE
ZERO-KNOWLEDGE PROOFS



How can we prove generic statements with zero-knowledge?

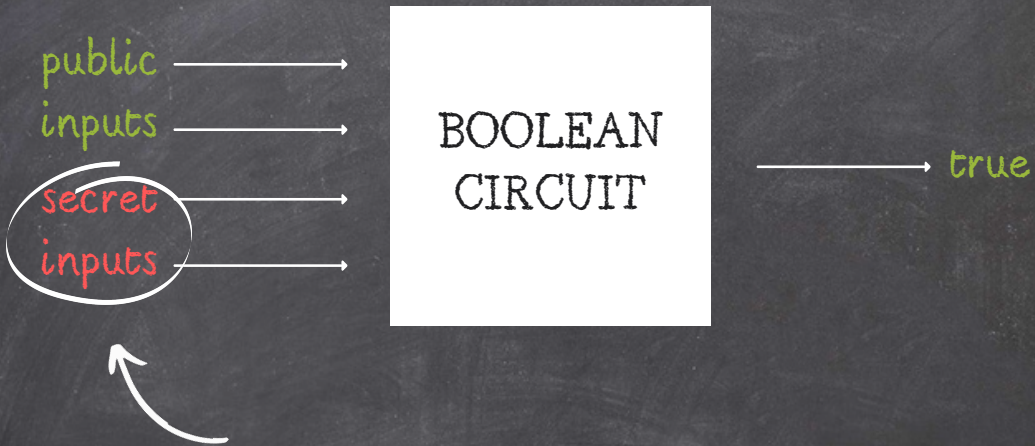
CIRCUIT SATISFIABILITY

The circuit satisfiability problem (CircuitSAT) is the decision problem of determining whether a given Boolean circuit has an assignment of its inputs that makes the output true.

CIRCUIT SATISFIABILITY

CircuitSAT can be proved with zero-knowledge.

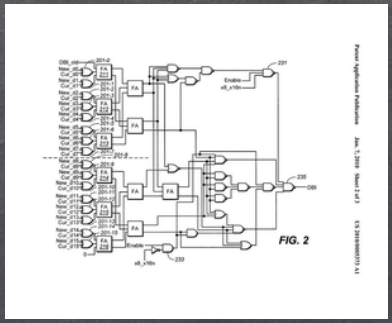
CIRCUIT SATISFIABILITY



CIRCUIT SATISFIABILITY

public inputs

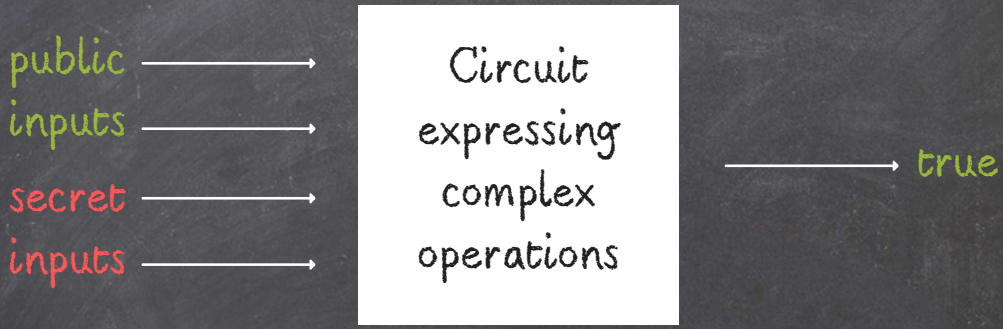
secret inputs



true

Any operation a computer can do can be expressed as a boolean circuit!

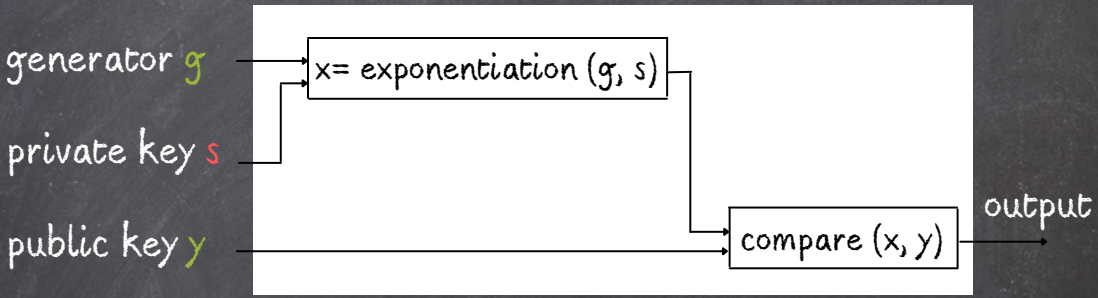
CIRCUIT SATISFIABILITY



Any operation a computer can do can be expressed as a boolean circuit!

CIRCUIT SATISFIABILITY

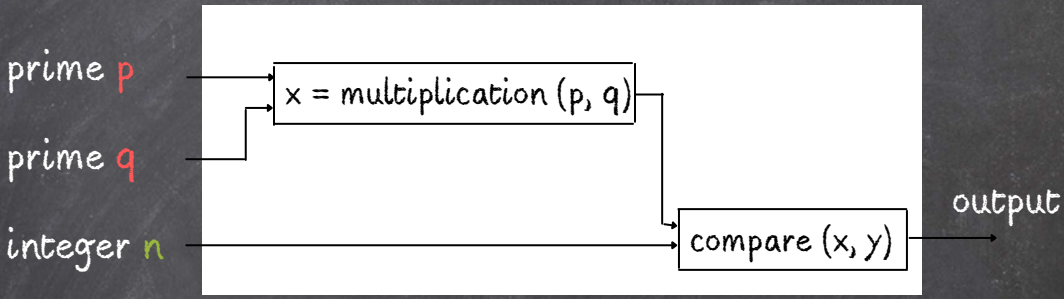
Circuit for the discrete logarithm problem



A zero-knowledge proof for this circuit would prove that, given g and y, you know an s that is the discrete logarithm of y.

CIRCUIT SATISFIABILITY

Circuit for the factorization problem



A zero-knowledge proof for this circuit would prove that you know the factorization (p,q) of a given composite number n.

APPLICATIONS

VERIFIABLE COMPUTATION

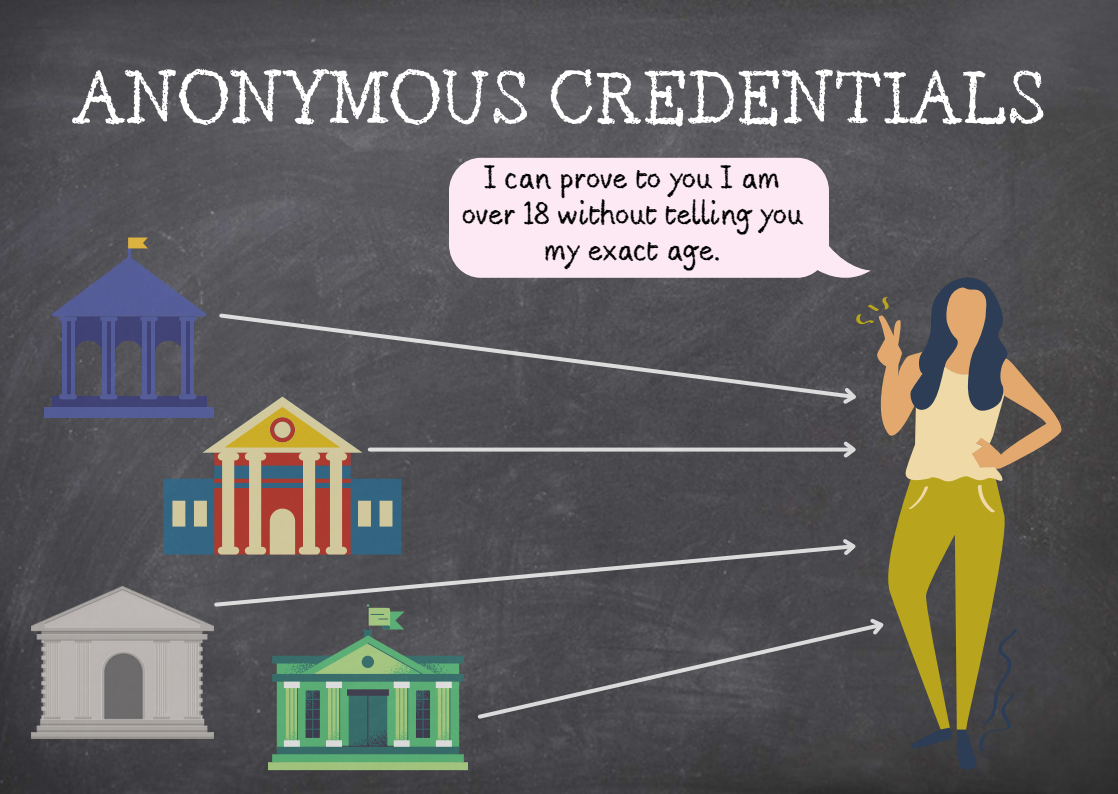
delegate a
computation

(result, proof of computation)

BLOCKCHAIN

- Proof that all transactions have been computed correctly.
- Privacy of payment details can be kept private while ensuring the correctness of the payment.
- Verifying a block ⟶ verifying a zero-knowledge proof.

ANONYMOUS CREDENTIALS



TAKE-AWAY EXERCISE

Given two identical balls of different color, how could you convince a color blind person that they are of different color?