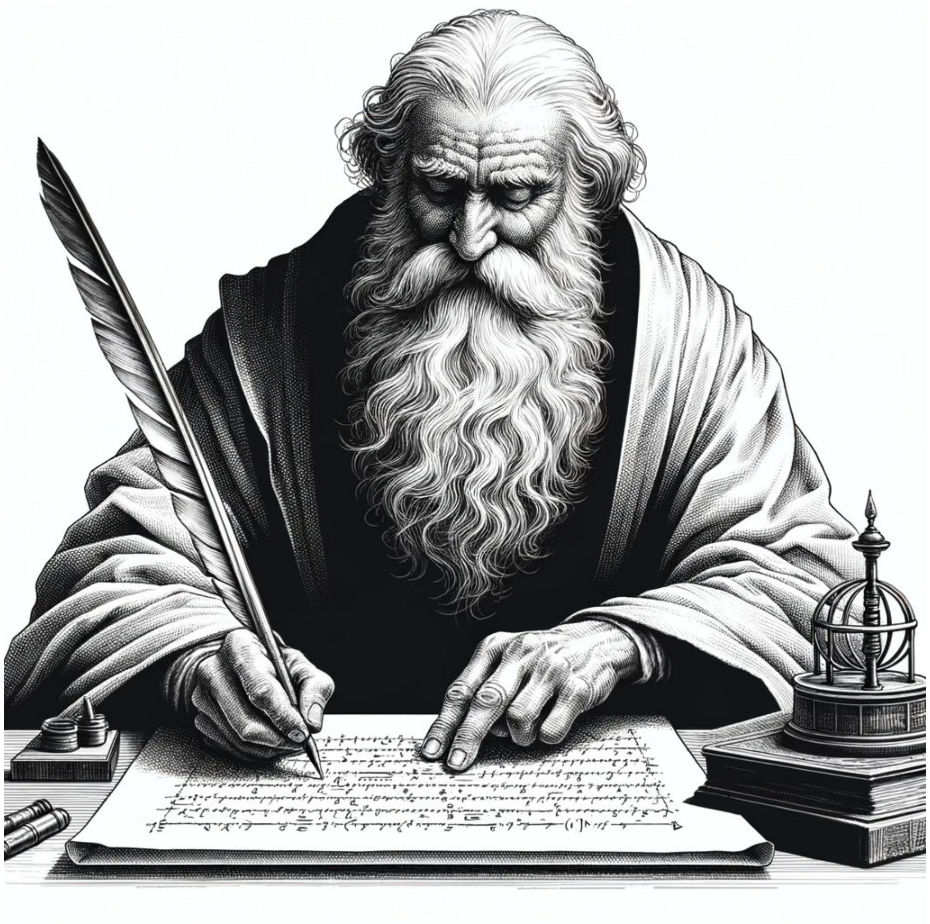
Journey To Zero-Knowledge

Davide Galassi – January 2024

Introduction

- **What?** To prove the correctness of a statement without revealing any details beside the validity of the statement itself
- **Why?** To unlock an entire new class of authentication protocols, secure multiparty computation, scalability solutions and last but not least pure philosophical amusement
- **How?** Well… This is what this presentation is mostly about



Classical Proofs

Deductive Reasoning

- Fundamental method of logical thinking and the bedrock of any form of ancient or modern proof
- Direct and intuitive derivation of a **conclusion** from a set of **premises**

Syllogism (*Aristotle*): two premises and one conclusion

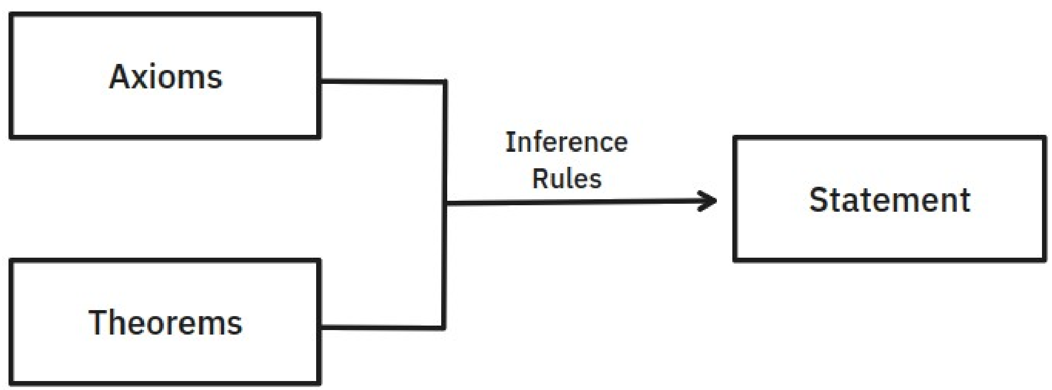
Premise: All men are mortal

Premise: Socrates is a man

Conclusion: Socrates is mortal

Proofs in Mathematics

- Application of deductive reasoning to mathematical abstract objects and concepts
- Establish the correctness of a **statement** from a set of **axioms** and previously proven theorems by using **inference rules**



(premises) (conclusion)

Validity and Soundness

- A proof is **valid** if the conclusion logically follows the premises
- A proof is **sound** if it is valid and all its premises are true

  Example of a valid but not sound proof (syllogism):

  - Premise: All prime numbers are odd (wrong!)
  - Premise: 2 is a prime number
  - Conclusion: 2 is odd

- We can reach invalid conclusions even though we constructed an apparently correct proof. Just because of a bad premise

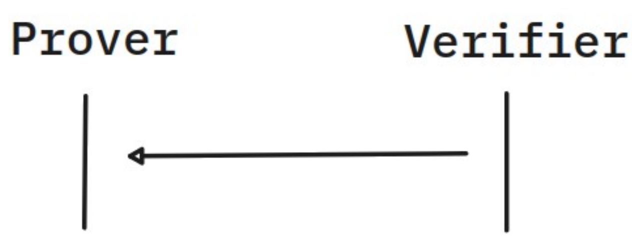
Proof Systems

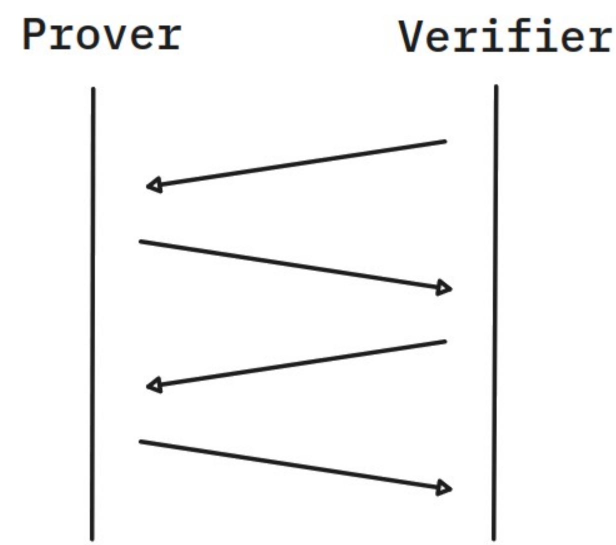
- **Formal** approach for construction and evaluation of proofs
- Components:
  - ***Statement*** (x): assertion to be proven
  - ***Proof*** (π): steps to establish validity of the statement
  - ***Prover*** (P): proof construction algorithm (P(x) = π)
  - ***Verifier*** (V): proof verification algorithm (V(x, π) = true/false)
- Formalization is paramounth to enter the machines world

Interactive Proofs

Interactive Proof Systems

- **Generalization** of classical proof system
- The prover incrementally convinces the verifier by actively **exchanging messages**





A classical proof system is an IP system with a single message

Probabilistic Proof Systems

- Both parties can introduce some **randomness** into the protocol messages
- Originally proposed by *Goldwasser*, *Micali* and *Reckoff* in 1985
  - Prover is assumed to have **unbounded** resources
  - Verifier operates has **polynomially bounded** resources (with respect to the statement size)
  - Both parties has access to a **private** random generator
- Allows proving an entire new class of problems which can’t be proven using deterministic interactive proof systems

IP Systems Characteristics

- **Completeness**: if the statement is true then *V(x, π) = true* with high probability.
- **Soundness**: if the statement is false then *V(x, π) = true* with negligible probability.
- **Efficiency**: *V(x, π)* must run in polynomial time with respect to the length of *x*.

We must be able to prove these characteristics

Tetrachromacy

- Condition enabling some individuals to perceive a broader spectrum of colors than the typical trichromat
- There are two apparently identical marbles and Peggy states she can distinguish the two

  Protocol:

  - Peggy places the two marbles in front of the Victor and turns her back
  - Victor flips a coin and based on the outcome he may swap the marbles
  - Peggy turns and tells Victor if the positions were swapped

Quadratic Non Residue

- y is a quadratic residue *(QR)* modulo n iff $y = x^2 \bmod n$
- Peggy wants to prove to Victor that y is not a quadratic residue (QNR)

Protocol:

- Victor toss a coin and, depending on the toss result, sends to Peggy $t = z^2$ or $t = z^2 \cdot y$ for some secret integer z
- Peggy, leveraging its unlimited computational power, finds out if t is a QNR and tells it to Victor

Note: if y is a QR then t is always a *QR*.

Zero Knowledge Proofs

The Issue

- How much knowledge is leaked to a verifier or any other observer during protocol execution?
- What is the minimal quantity of knowledge which must be shared in order to validate a proof?

- The prover may want to minimize this knowledge, ideally to one single bit of information

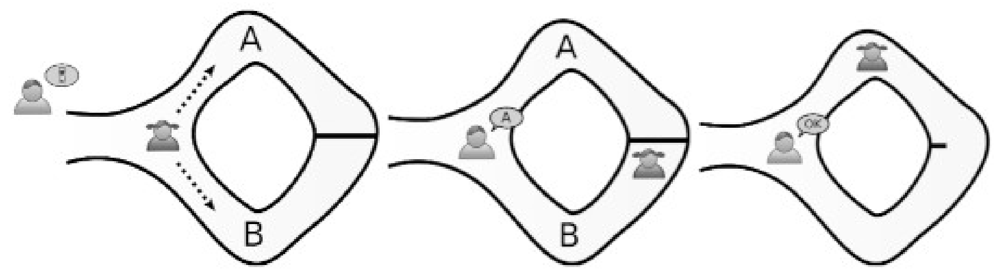
ZKP Systems Characteristics

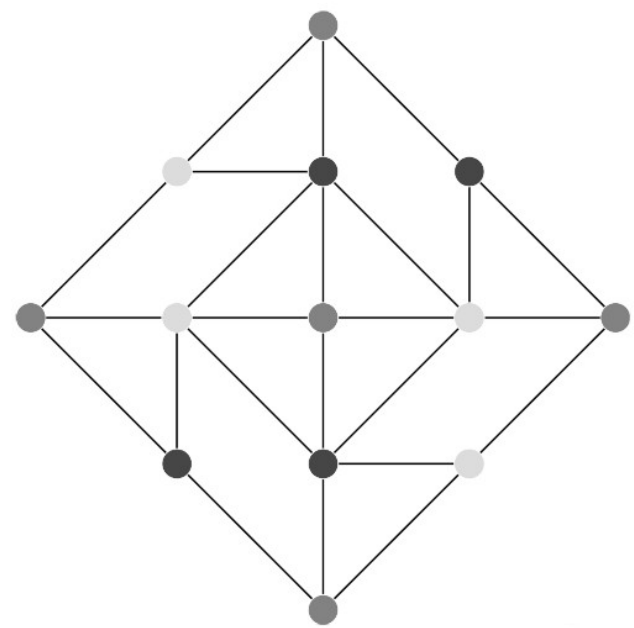
- Completeness
- Soundness
- Efficiency

- **Zero-Knowledgeness** (yes… that’s how is called)
  - **Simulator** existance: an algorithm which is able to convince the prover about the statement without the prover possessing any knowledge
  - Requires the protocol to operate under special condition which must not be realistically accomplished in normal operation circumstances: a **time machine**

Many Intuitive Examples

- Where is Waldo?
- Ali Baba Cave
- Sudoku
- Graph 3-Coloring
- Graphs isomorphism
- …

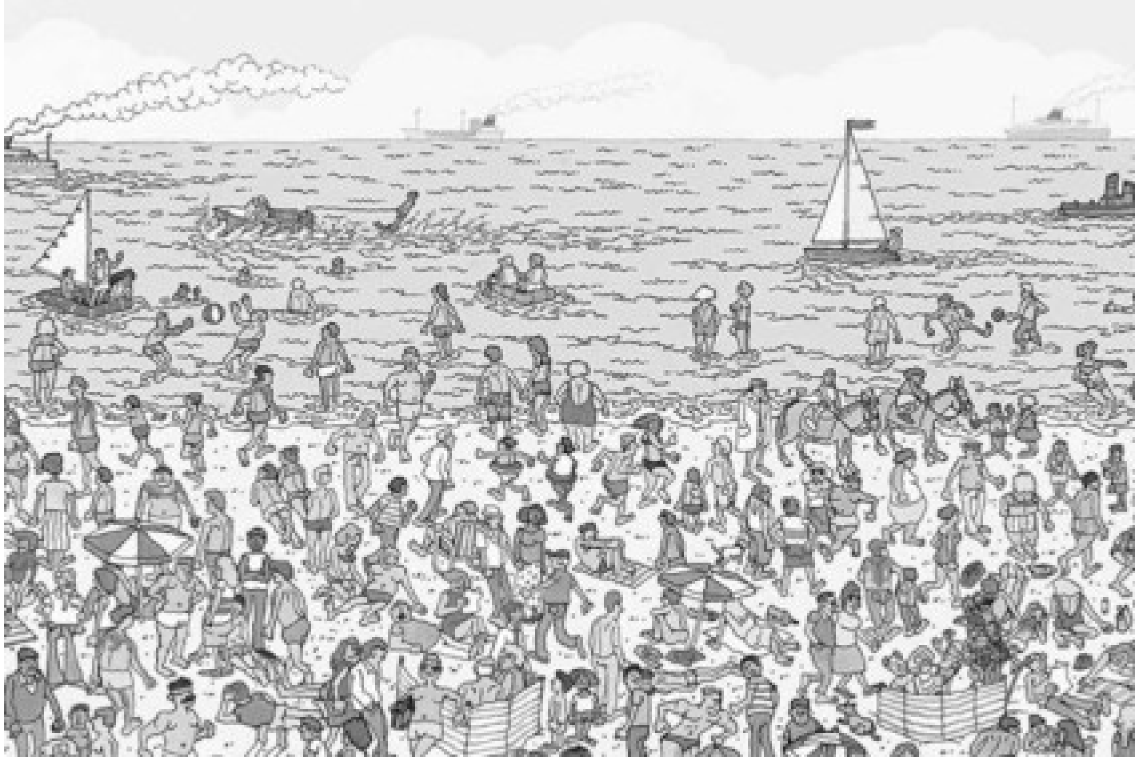




4		1	2	9			7	5
2			3			8		
	7			8				6
			1		3		6	2
1		5				4		3
7	3		6		8			
6				2			3	
		7			1			4
8	9			6	5	1		7

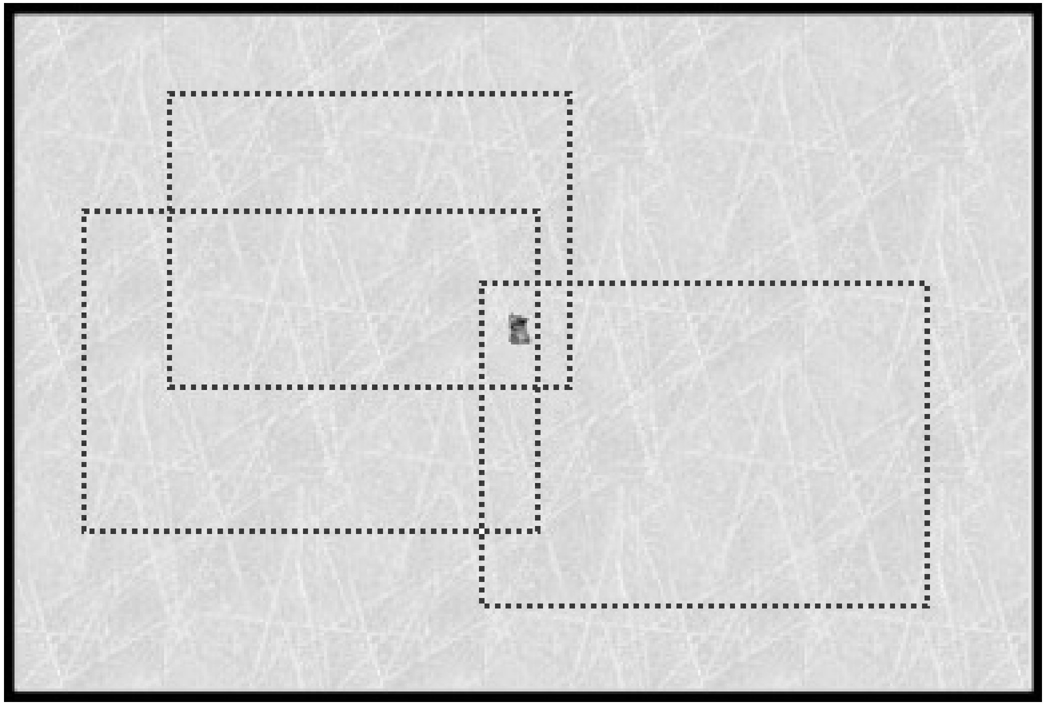
Where is Waldo?

Peggy wants to prove her knowledge about Waldo’s position in the illustration without revealing its position to Victor



Where is Waldo?

The illustration relative position is unknown to Victor



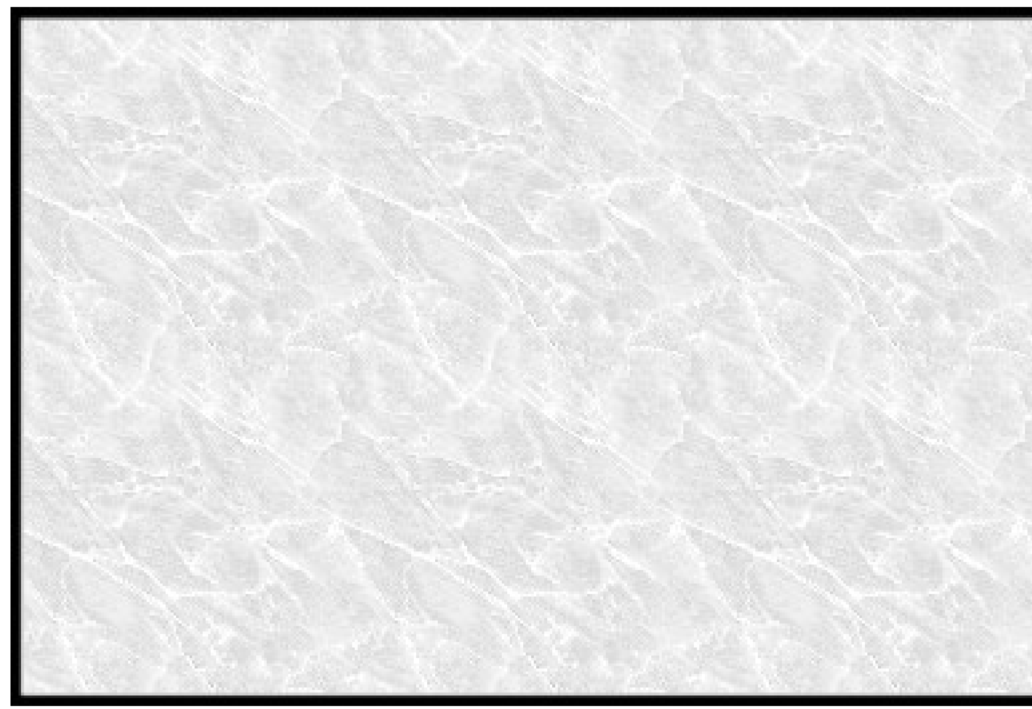
Where is Waldo?

Victor has two choices:

1) See Waldo’s face

2) See the illustration

Peggy has ½ chance to cheat

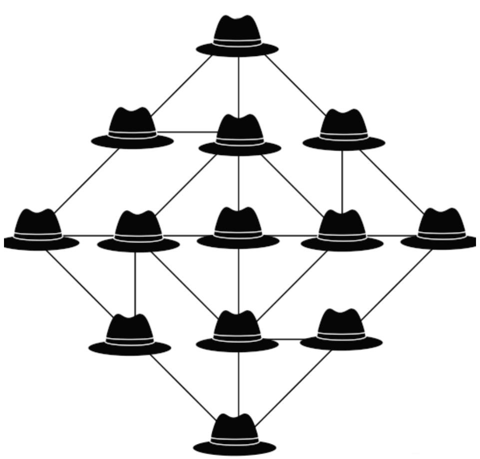


Graph 3-Coloring

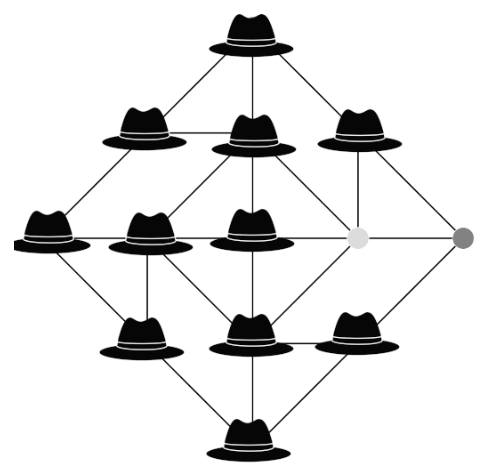
- Given a Graph, Peggy wants to prove that she knows a solution for the 3-Coloring problem

  Protocol

  - Applies the solution using random colors and covers the vertices
  - Victor asks to uncover two vertices connected by an edge



(E-1)/E chances to cheat



Proofs for all NP

- Graph 3-Coloring problem is **NP-complete**
  - Any NP problem can be mapped to an instance of 3-coloring problem
  - There is a ZK proof for all NP problems
- Very **inefficient** but revolutionary
- When possible ZK proofs leverage ad-hoc properties of the problem domain

Schnorr's Protocol

- Prove knowledge of discrete logarithm of $y = g^x$
- Foundation for Schnorr's signatures and DSA

Protocol:

- Peggy selects a random k and sends $r = g^k$
- Victor sends a random challenge c
- Peggy responds with $s = x \cdot c + k$
- *Victor checks if* $g^s = y^c \cdot r$

Future Directions

- **Proof of Computation** to assess the results of any arbitrary computation
- Bleeding Edge application of ZK proofs for **scalability** by offloading work
- Succint and constant proofs size with **zk-SNARK**s and **zk-STARK**s

https://datawok.net/posts/journey-to-zero-knowledge