CS 65500
Advanced Cryptography

Lecture 15: Zero-Knowledge Proofs

Instructor: Aarushi Goel

Spring 2025

Agenda

→ Zero-Knowledge Proofs:

* Motivation
* Definition
* Construction

Example 1: Where's Waldo?

Example 2: Sudoku

Hey Bob, look at this very hard Sudoku

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

Last week you gave me a puzzle with no solution. I wasted 3 hrs on that!

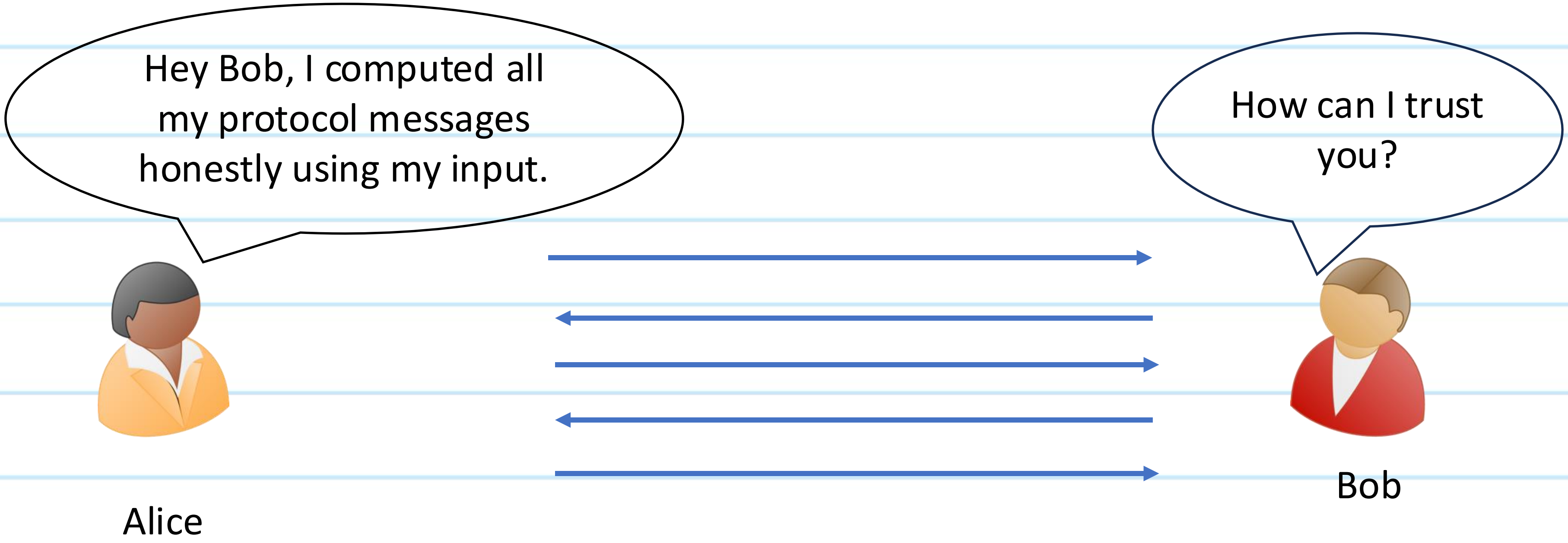
Alice

Bob

Trust me! This one has a solution.

Example 3: Secure Multiparty Computation

Alice and Bob are participating in an MPC protocol that is only proven to be semi-honest secure.
But Bob suspects that Alice might be malicious.

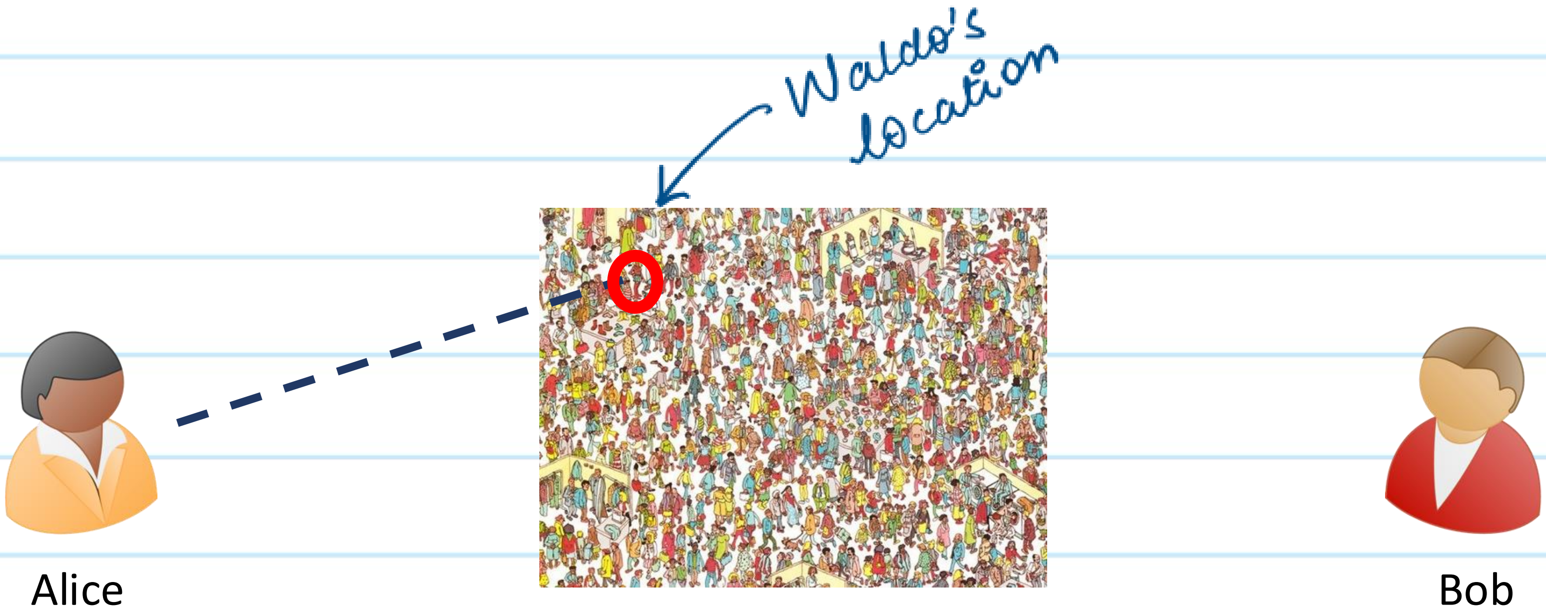


In all these examples, Alice can use classical mathematical proofs to convince Bob. But classical proofs away too much information. Can Alice still convince Bob without giving away too much information?

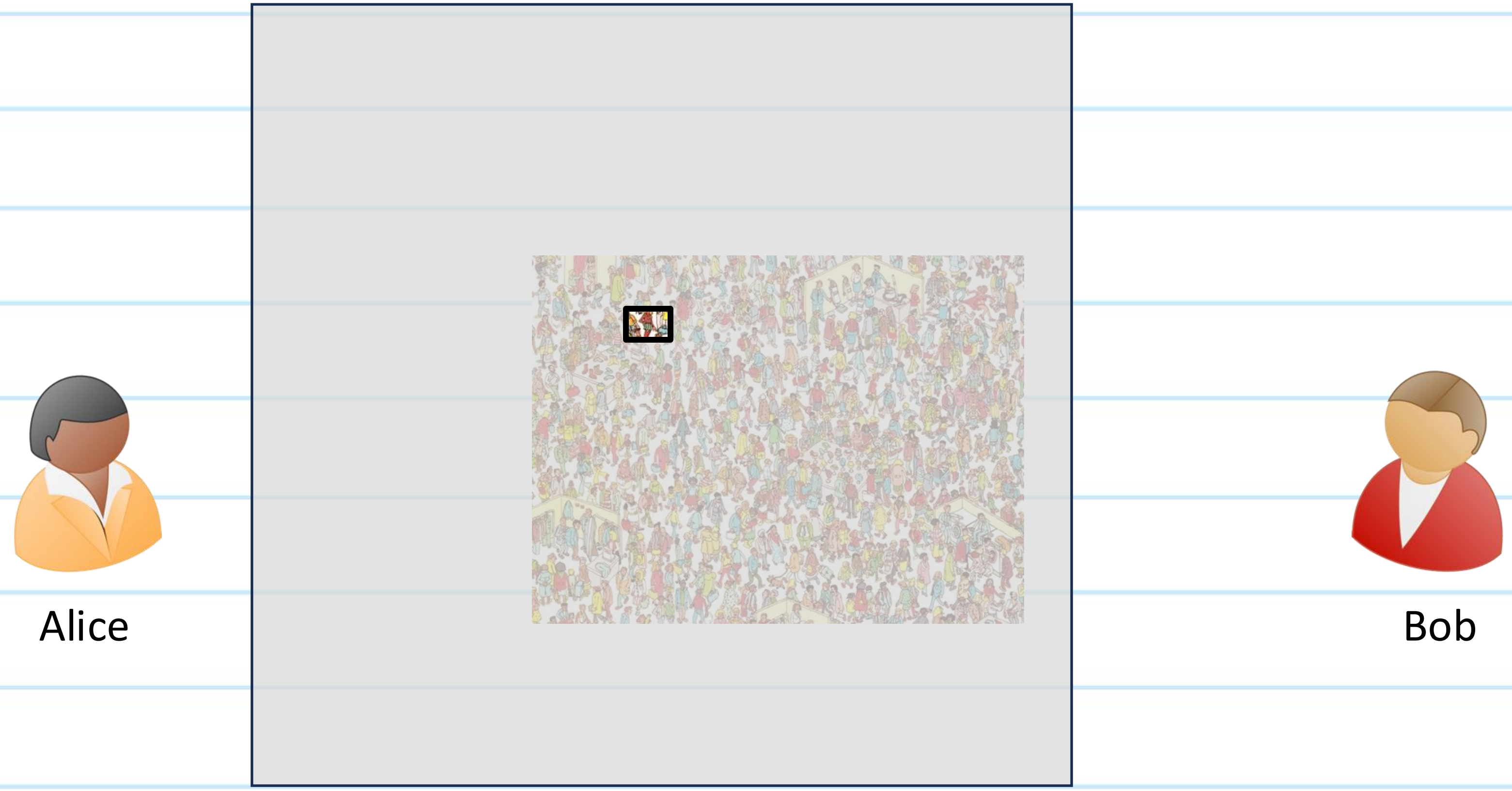
Are these conflicting requirements?

Alice wants to convince Bob:	Bob should not learn:
1 Waldo is in the picture	1. Waldo's location
2. Sudoku puzzle has a solution	2. Sudoku solution
3. MPC protocol messages were computed honestly using some input.	3. Input.

A Potential Knowledge-Hiding Solution for Where's Waldo?

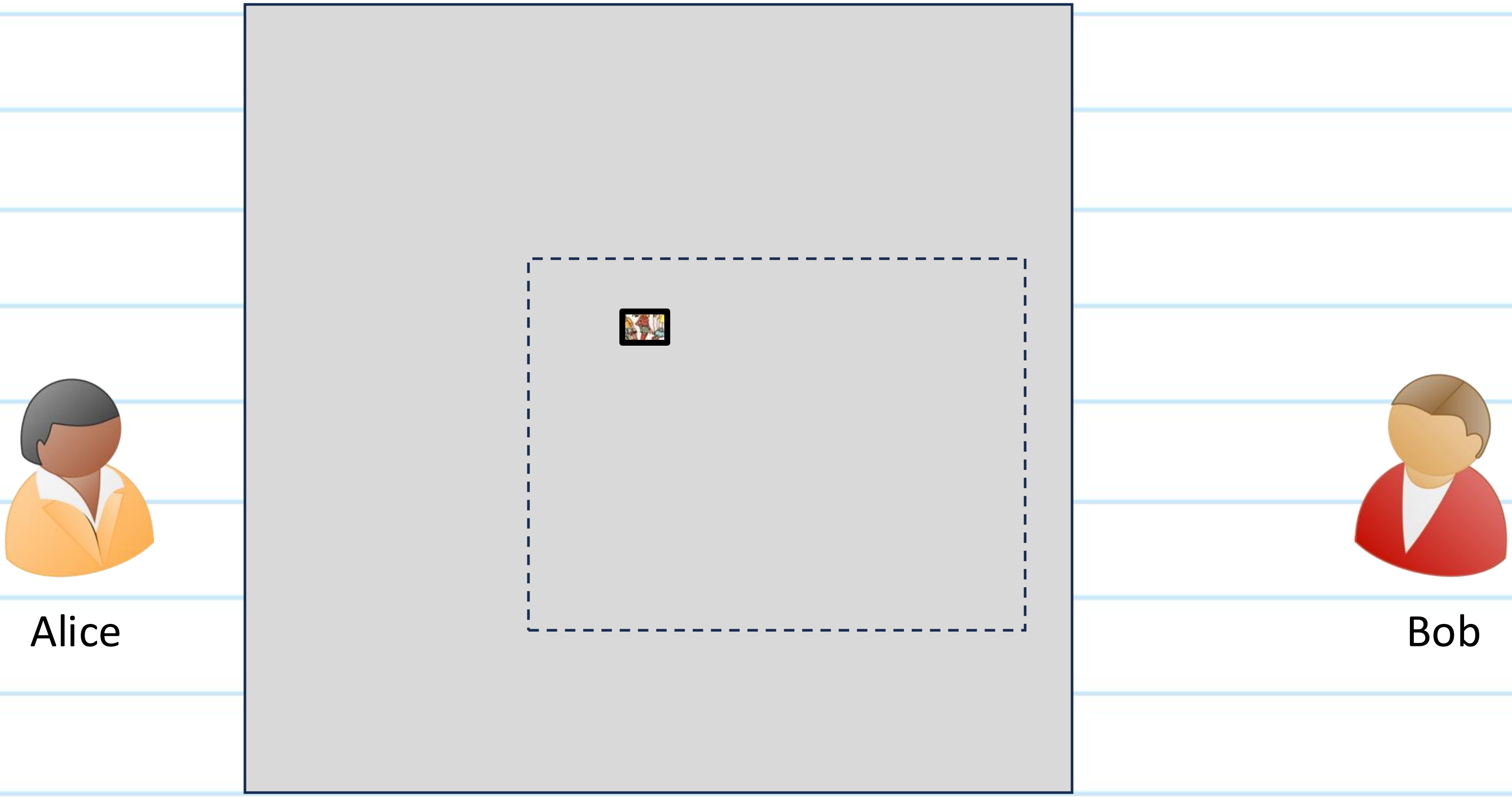


A Potential Knowledge-Hiding Solution for Where's Waldo?



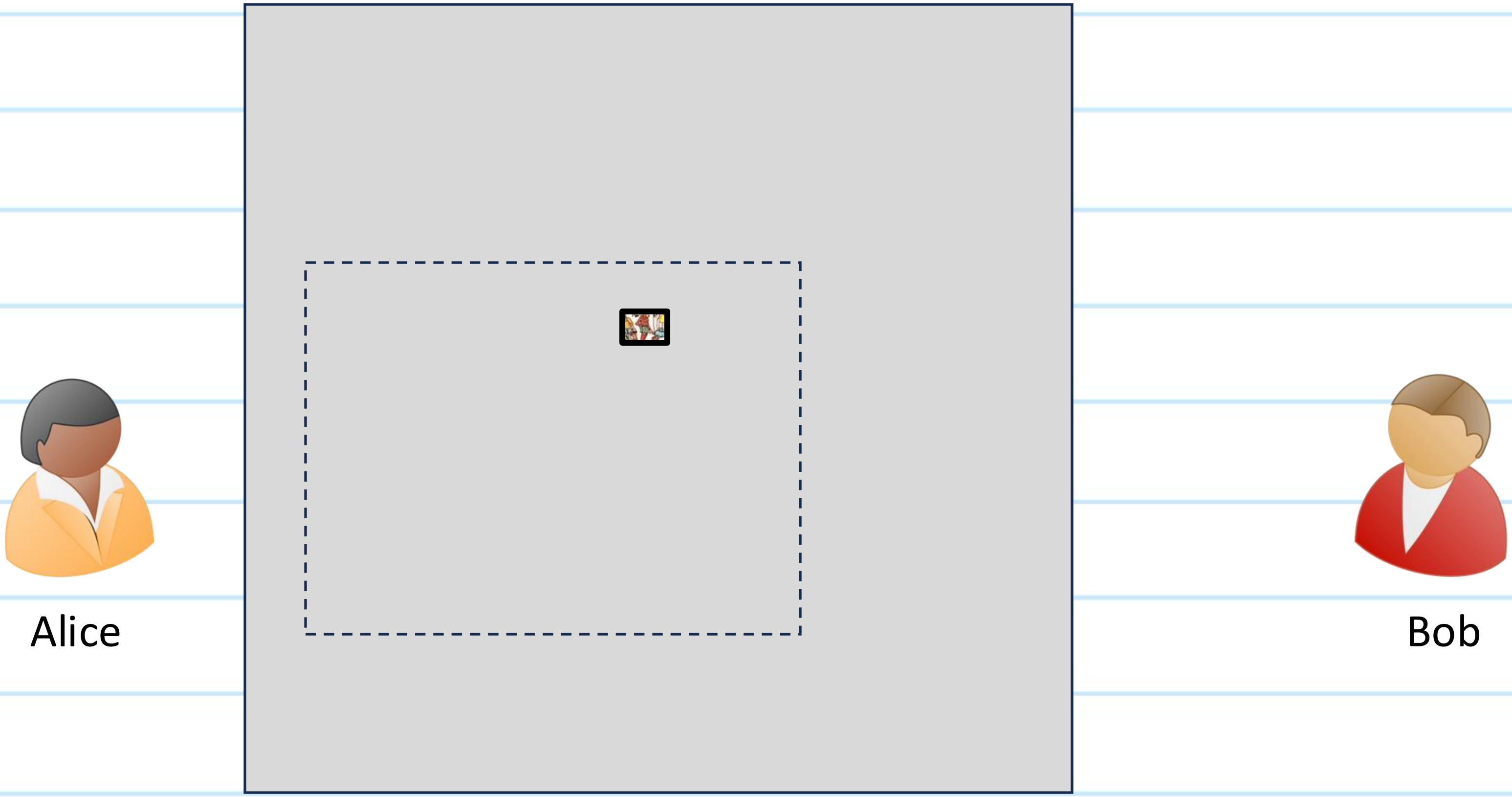
Alice places an opaque cardboard with a hole over the picture revealing Waldo.

A Potential Knowledge-Hiding Solution for Where's Waldo?



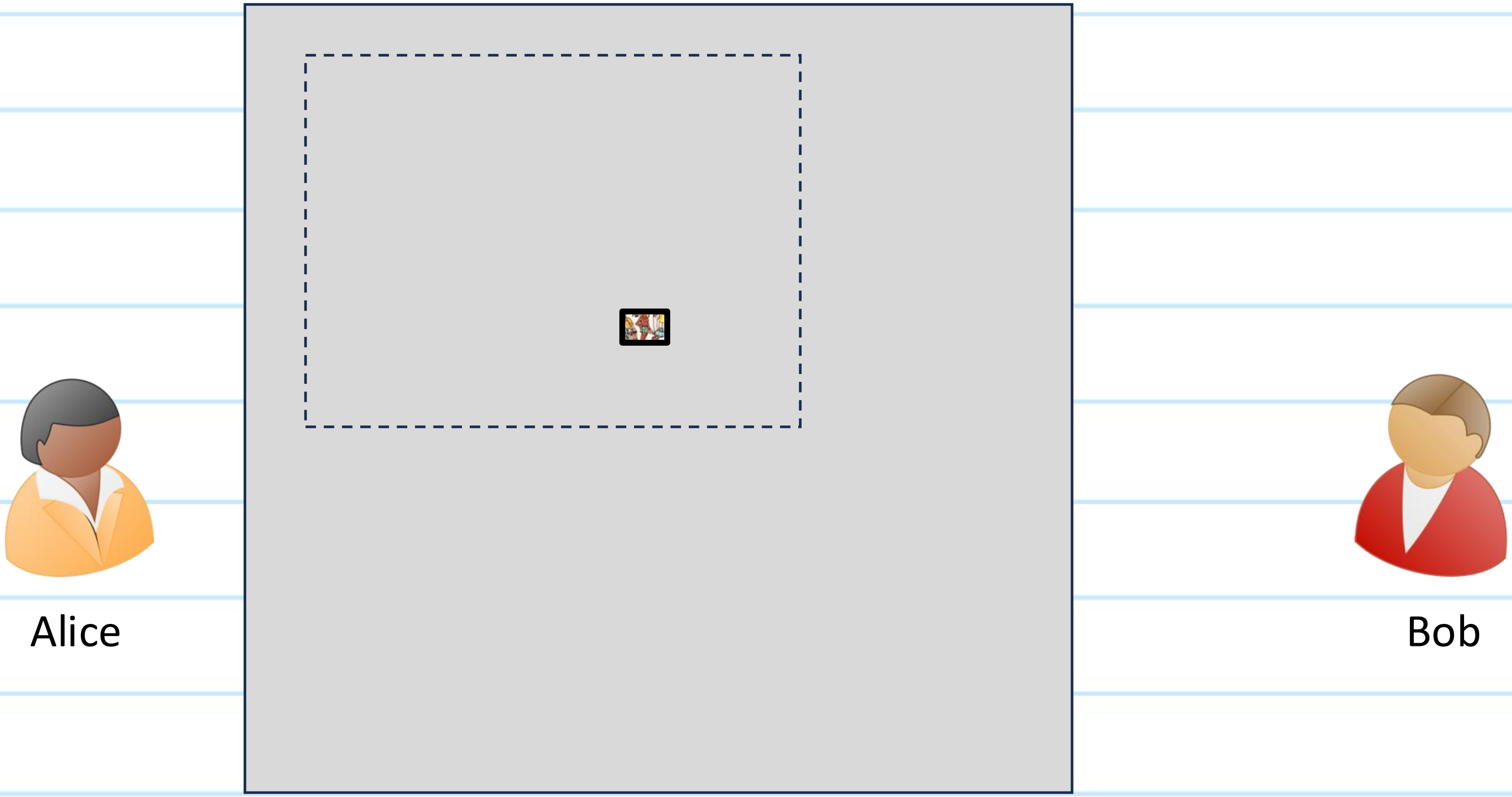
Bob gets no information about Waldo's actual location in the picture.

A Potential Knowledge-Hiding Solution for Where's Waldo?



Because Bob doesn't know where the picture is placed underneath the cardboard.

A Potential Knowledge-Hiding Solution for Where's Waldo?



Yet Bob is convinced that Waldo is in the picture and that Alice knows where Waldo is.

Zero-Knowledge Proofs

→ This was an example of a zero-Knowledge proof.

→ A zero-Knowledge proof is a way of convincing someone that x satisfies a property, without giving away any additional knowledge / information.

→ Introduced in the 80s by:

Shafi Goldwasser

Silvio Micali

Charlie Rackoff.

→ Shafi & Silvio won the Turing award in 2012.

What are x, Property and Additional Information?

1 Where's Waldo

What is x? Picture

What is the property? Waldo is in the picture

What is the additional information? Waldo's location

2. Sudoku Puzzle

What is x? Sudoku Puzzle

What is the property? Sudoku puzzle has a solution

What is the additional information? Sudoku Solution

3. MPC

What is x? MPC protocol messages

What is the property? The messages were computed using some input

What is the additional information? input

Zero-Knowledge Proofs

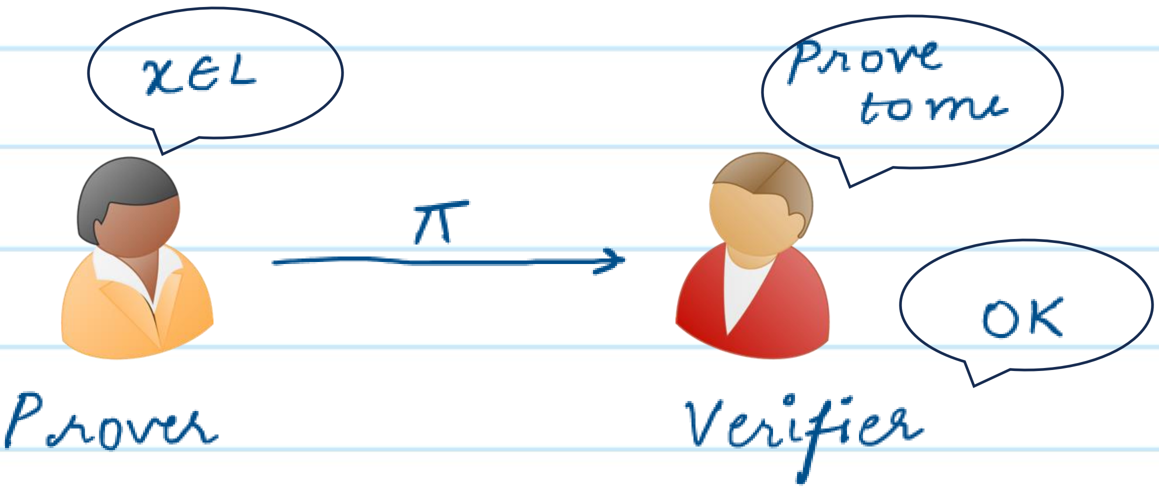
→ This can be generalized to statements in any NP language.

→ Let L be a language in NP and let R be be associated relation.

→ Alice (Prover) wants to convince Bob (verifier) that a statement $x \in L$, without revealing the witness w, such that $R(x, w) = 1$

additional information

Property



Zero-Knowledge Proofs (Properties)

What Properties do we typically want from a proof?

→ **Completeness**: If $x \in L$, then an honest prover must be able to convince an honest verifier

→ **Soundness**: If $x \notin L$, then a cheating prover cannot find any purported proof that will convince an honest verifier

→ **Efficient Verification**: The proof must be finite and efficiently verifiable. e.g. Proof that there are infinitely many primes should not simply be a list of all primes. Not only would it take forever to generate that proof, it will also take forever to verify it

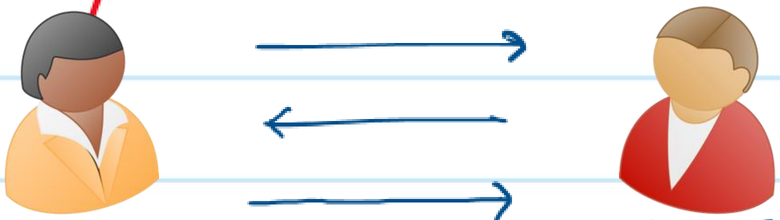
An additional property that we want from a zero-knowledge or information hiding proof:

→ **Zero-Knowledge:** If $x \in L$, then the convincing proof sent by an honest prover should not leak any information about w, where w is such that $R(x, w) = 1$.

If we didn't care about zero-knowledge, there is a trivial proof for all NP languages that satisfies all the other properties

→ The witness w is a trivial proof.

Interactive Proofs



Prover Verifier

→ We typically think of mathematical proofs as *one-shot* or non-interactive proofs.

→ But it doesn't have to be this way.

→ A proof can also be a conversation (or interactive protocol) that convinces the verifier that $x \in L$.

why interaction?

1 Interaction helps us prove statements not known to be in NP: i.e., IP = PSPACE

2 Necessary for zero-knowledge

Defining Interactive Proofs (without Zero-Knowledge)

Definition: A protocol π between a prover P and a verifier V is an interactive proof system for a language L if V is a PPT machine and the following properties hold:

- Completeness: $\forall x \in L$

$$\Pr\left[\mathrm{Out}_V\left[P(x) \longleftrightarrow V(x)\right] = 1\right] = 1$$

- Soundness: There exists a negligible function $\nu(.)$, s.t., $\forall x \notin L$, $\forall \lambda \in \mathbb{N}$ and all adversarial provers P^*,

$$\Pr\left[\mathrm{Out}_V\left[P^*(x) \longleftrightarrow V(x)\right] = 1\right] \leq \nu(\lambda)$$

We can also modify the above definition to consider PPT provers

Formalizing the Notion of Zero-Knowledge.

→ If an interactive proof convinced the verifier that $x \in L$, then this interactive proof should not leak any information about the witness w that the prover used to participate in in the interactive proof.

→ In other words, whatever the verifier saw during the interactive proof, it could have *simulated* on its own using x, L and the fact that $x \in L$.

Defining Zero-Knowledge

Definition: An interactive proof Π between P & V for a language L with witness relation R is said to be zero-knowledge if for every n.u. PPT adversary V^*, there exists a (expected) PPT simulator S, such that $\forall x \in L, \forall w \in R(x), \forall z \in \{0,1\}^*$ and $\forall \lambda \in \mathbb{N}$, the following two distributions are computationally indistinguishable:

1. $\{ \text{View}_{V^*} [P(x,w) \longleftrightarrow V^*(x,z)] \}$
2. $\{ S^{V^*}(1^\lambda, x, z, L) \}$

We can also consider the notions of statistical/perfect zero-knowledge against unbounded adversaries, if the above distributions are statistically close (or identical respectively)

Understanding the Notion of Zero-Knowledge.

Paradox?

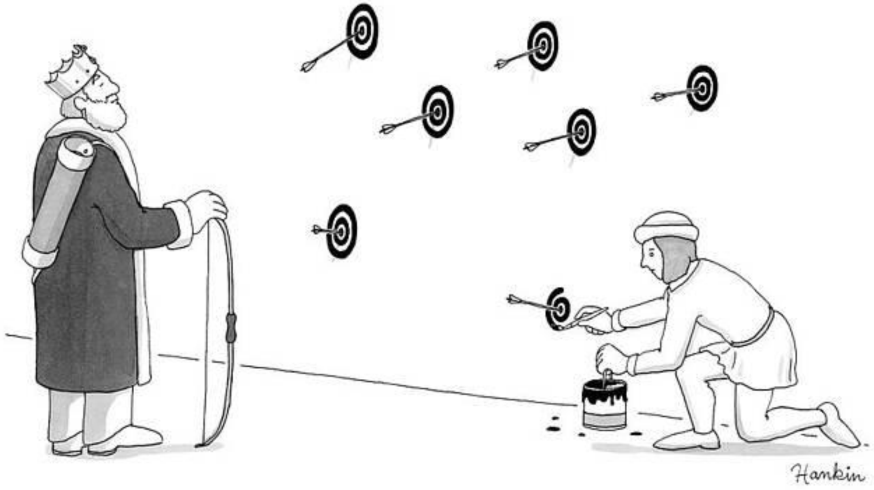
→ Protocol convinces V of the validity of x

→ Yet V could have generated the protocol transcript on its own

Not Really

Example

Shooting arrows at targets drawn randomly on a wall



VS

Drawing targets around arrows shot randomly on a wall

Both generate identical views, but only one is convincing of excellent shooting skills.

Graph Isomorphism

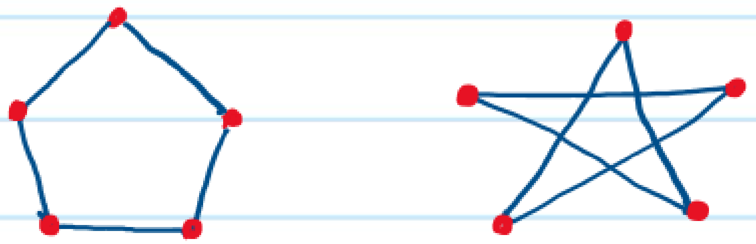
→ Let $G = (V, E)$ be a graph, wher V is the set of vertices & E is the set of edges

→ $G_0 = (V_0, E_0)$ and $G_1 = (V_1, E_1)$ are said to be isomorphic if there exists a permutation π, s.t.

* $V_1 = \{ \pi(v) \mid v \in V_0 \}$

* $E_1 = \{ (\pi(v_1), \pi(v_2)) \mid (v_1, v_2) \in E_0 \}$

in other words, $G_1 = \pi(G_0)$



Graph isomorphism is in NP.

Zero-Knowledge Proof for Graph Isomorphism

Prover wants to convince the verifier that graphs G_0 and G_1 are isomorphic without revealing π, where $\pi(G_0) = G_1$

Prover

(G_0, G_1, π)

Verifier

(G_0, G_1)

Prover: Sample a random permutation σ

Prover: Sample a random bit $b \in \{0,1\}$

$H = \sigma(G_b)$

Prover $\xrightarrow{H}$ Verifier

Verifier: Samples a random bit $b' \in \{0,1\}$

Prover $\xleftarrow{b'}$ Verifier

$$\tau = \begin{cases} \sigma & \text{if } b' = b \\ \sigma \cdot \pi^{-1} & \text{if } b = 0,\ b' = 1 \\ \sigma \cdot \pi & \text{if } b = 1,\ b' = 0 \end{cases}$$

Prover $\xrightarrow{\tau}$ Verifier

Verifier: Output 1 iff $H = \tau(G_{b'})$