Zero-Knowledge Proofs

263-4665-00L

Lecturer: Jonathan Bootle

Lecturer

- Jonathan Bootle
- jonathan.bootle@inf.ethz.ch

- PhD on zero-knowledge at UCL
- Now at IBM Research – Zurich

- Started studying cryptography to make a secure secret-santa protocol!

Teaching assistant

- Karen Klein
- karen.klein@inf.ethz.ch

- PhD at ISTA
- Postdoc in the Foundations of Cryptography group

- Started studying cryptography for cool applications of number theory and combinatorics.

Teaching assistant

- Varun Maram
- vmaram@inf.ethz.ch

- MSc at ETHZ
- PhD in the Applied Cryptography group

- Started studying cryptography after reading “Digital Fortress” by Dan Brown.

Teaching assistant

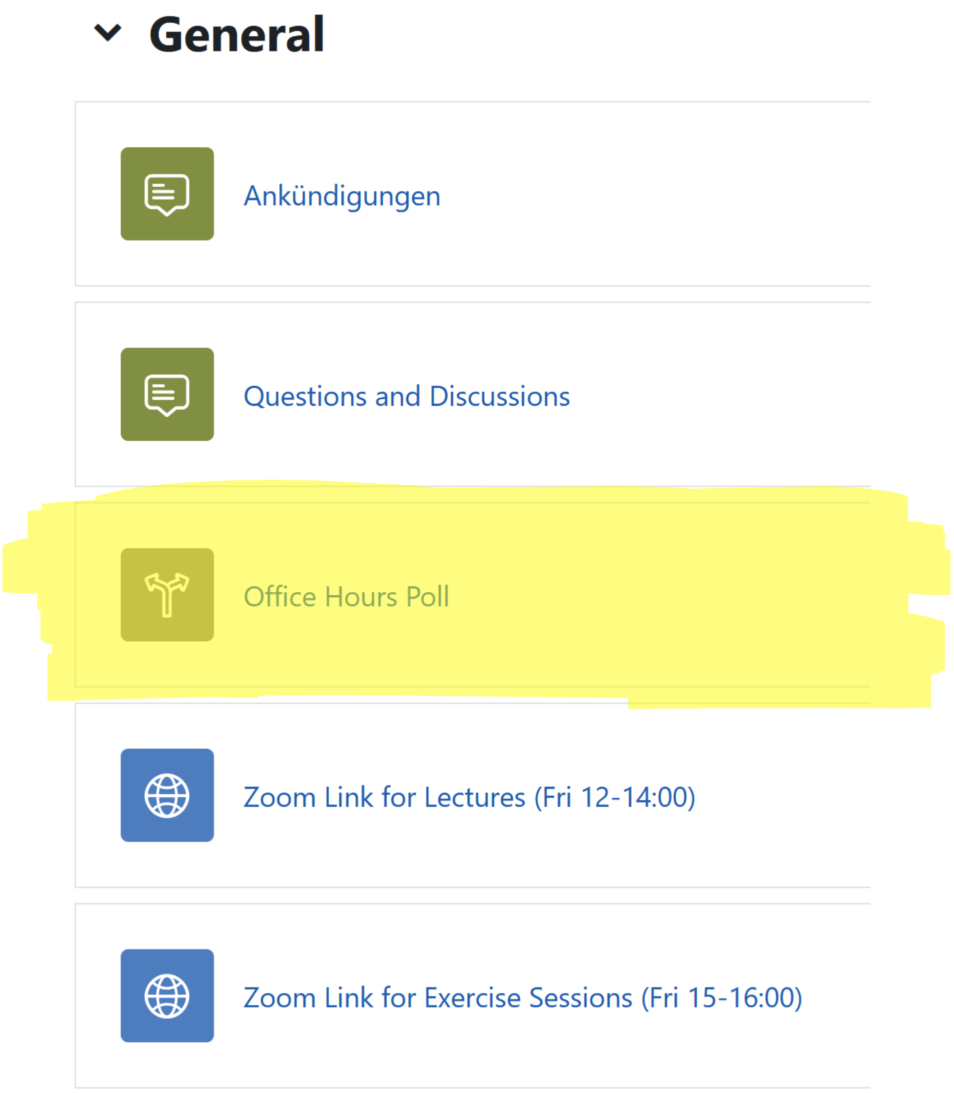
- Antonio Merino Gallardo
- antonio.merinogallardo@inf.ethz.ch

- MSc at ETHZ

- Started studying cryptography after reading about the threats posed by quantum computers.

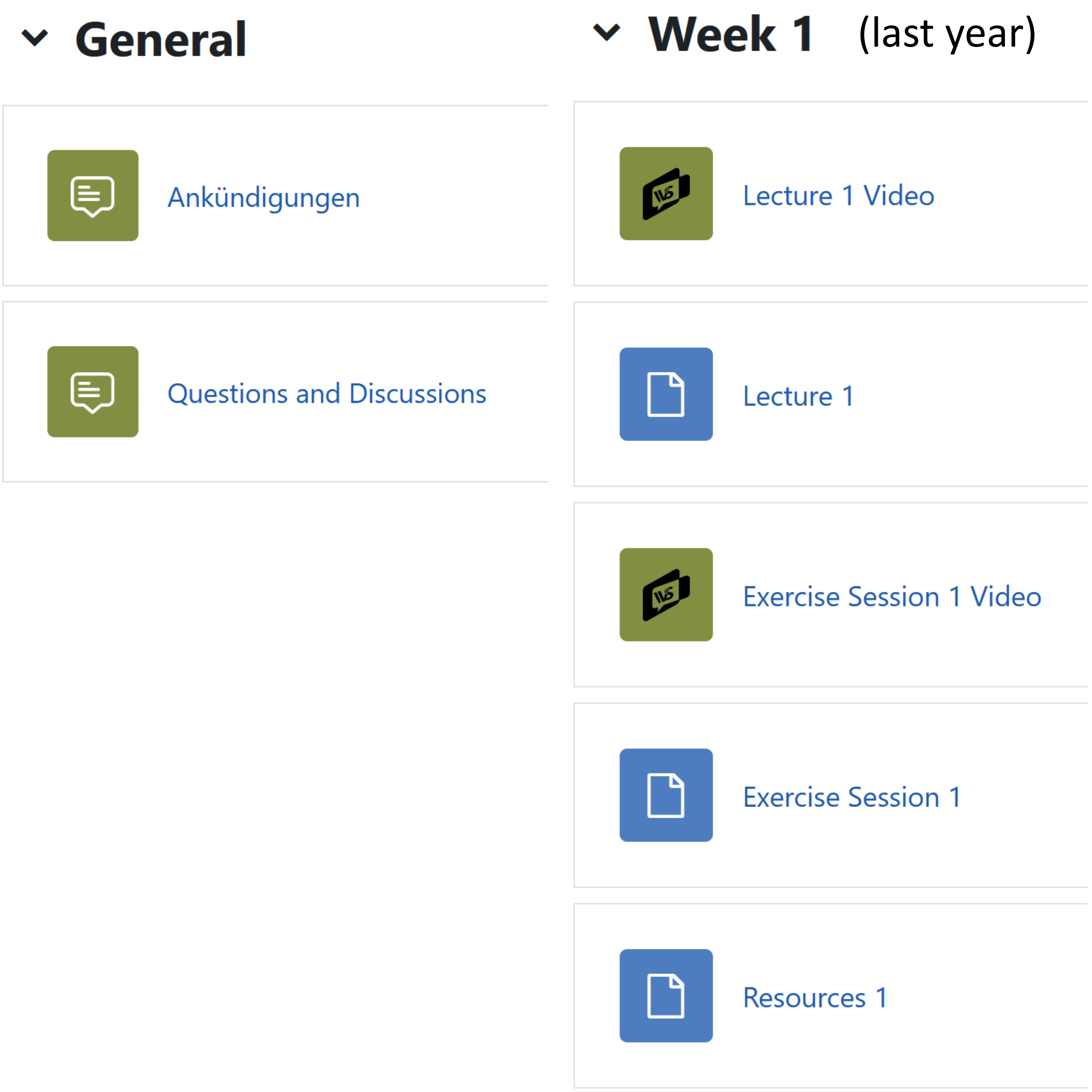
Time and Location

- Lectures on Fridays, 12:00-14:00, CHN G 42.
- Remote only on 24/11/2023.
- Exercise sessions on Fridays, 15:00-16:00, in CHN F 42.
- Online on Zoom, and recorded.
- Office hours poll closes 23:59, 25/09



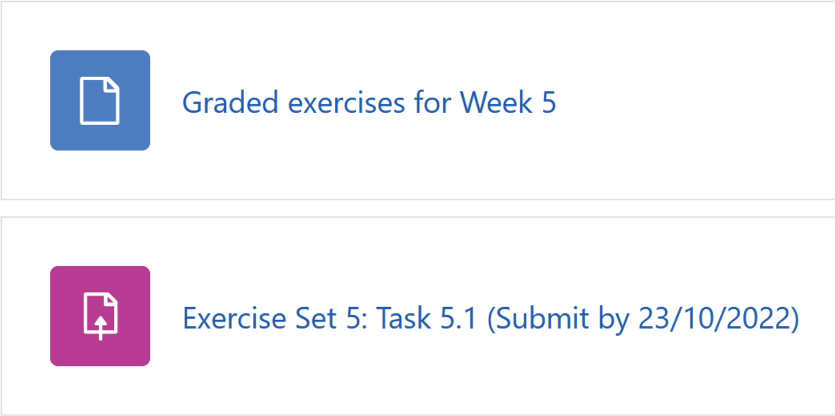
Materials

- Posted on Moodle
- Lecture slides, shortly before each lecture
- Exercise sheets, about a week before each exercise session
- Lecture and exercise session recordings, once processed
- Links to external resources such as papers and notes



Examination

(last year)



- This course = 5 ECTS credits

- 70% session examination, 2 hour written exam
  - Practice exam questions available later in the course

- 30% graded exercise sheets
  - 10% week 5 – submit through Moodle on or before 20/10/2023
  - 10% week 9 – submit through Moodle or before 17/11/2023
  - 10% week 13 – submit through Moodle on or before 15/12/2023
- Other weeks ungraded, but **essential** for practice and final exam

Course Outline (13 lectures)

1. **Introduction and definitions** ~2 lectures

2. **Sigma protocols** ~3 lectures

3. **ZK arguments with short proofs** ~4 lectures

4. **Non-interactive zero-knowledge** ~3 lectures

5. **Bonus material?** ~1 lecture

Goals of the course

- To understand what it means for a zero-knowledge proof to be secure

- To construct and analyse various types of zero-knowledge proofs

- To understand some applications of zero-knowledge proofs

Prerequisites

- Familiarity with cryptography (e.g. security games, p.p.t. adversaries, indistinguishability as in IND-CPA security)
- Groups, finite fields, modular arithmetic
- Vectors and matrices
- **Polynomial arithmetic over e.g.** $\mathbb{F}[\boldsymbol{X}], \mathbb{F}[\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3}]$
- Probability
- May use all of the above **at the same time**
- Proofs: ~~deep~~ ~~difficult~~ dirty
- Review maths at https://shoup.net/ntb/ntb-v2.pdf

Why you should study zero-knowledge proofs

- Philosophy: new perspectives on proof, knowledge and learning

- Theory: simulation techniques, links to complexity theory

- Practice: used in digital signatures, e-voting, mix nets, verifiable computation, blockchains and more…

- Because they are interesting!
- If you are ready for a challenge…

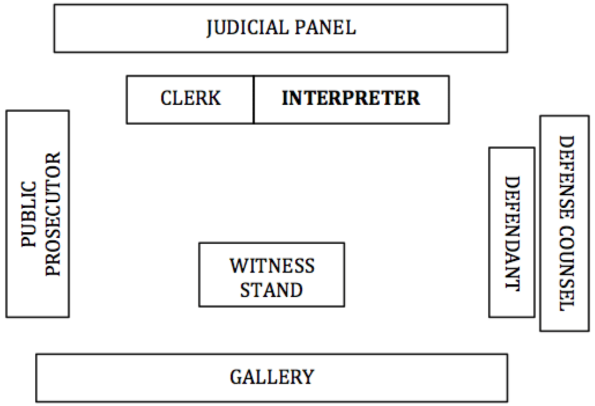
Lecture 1: Interactive proofs and zero-knowledge

What happens when you add interaction, randomness, and cryptographic assumptions to mathematical proofs?

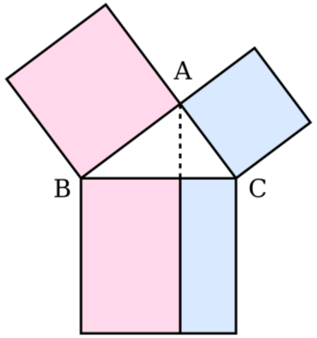
Types of proof

Proof:
"the process or an instance of establishing the validity of a statement especially by derivation from other statements
in accordance with principles of reasoning"
https://www.merriam-webster.com/dictionary/proof

Real life



Formal



This course

Static
Deterministic

Interactive
Randomised

Mathematical proofs

Known for over 2000 years!

Statement to prove $\sqrt{2} \notin \mathbb{Q}$

Prover
'proof writer'

P

Proof

1. Suppose $\sqrt{2} = \frac{a}{b} \in \mathbb{Q}, \gcd(a, b) = 1$
2. $\sqrt{2}b = a$ so $2b^2 = a^2$
3. $2 \mid a^2$ so $2 \mid a$
4. $\exists c$ with $a = 2c$
5. $2b^2 = 4c^2$ so $b^2 = 2c^2$
6. $2 \mid b^2$ so $2 \mid b$
7. Contradiction to $\gcd(a, b) = 1$

V

Verifier
'proof checker'

- All errors detectable by V
- The same 1-directional message every time

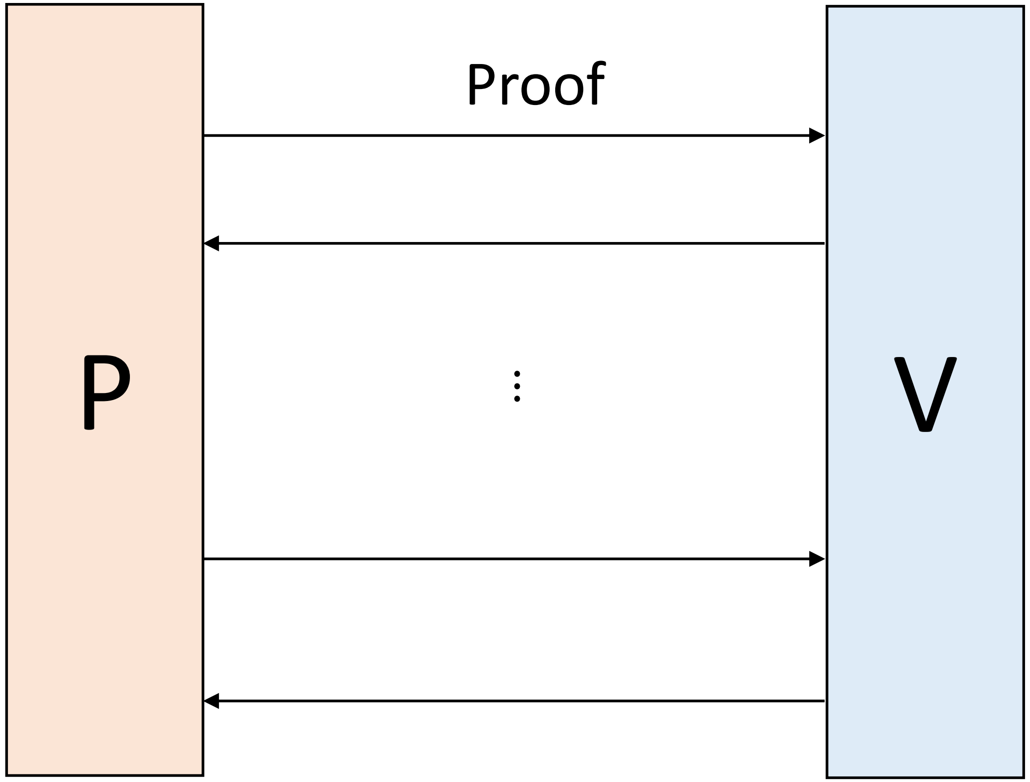
- Sequence of logical assertions
- Each must be implied by earlier ones

Interactive proofs (IP)

“The Knowledge Complexity of Interactive Proof Systems”, 1985

Statement to prove

Prover
‘proof writer’



Verifier
‘proof checker’

1. Increased proving power
2. Counterintuitive properties
3. Drastic efficiency gains

- Most errors detectable by V
- Randomised, adaptive, 2-directional messages

Agenda

- Definitions of complexity classes and IPs
- IP for graph non-isomorphism – increased proving power
- IP for graph isomorphism – counterintuitive properties

Course Outline (13 lectures)

1. **Introduction and definitions** ~2 lectures

2. **Sigma protocols** ~3 lectures

3. **ZK arguments with short proofs** ~4 lectures

4. **Non-interactive zero-knowledge** ~3 lectures

5. **Bonus material?** ~1 lecture

Drastic efficiency gains

Complexity classes

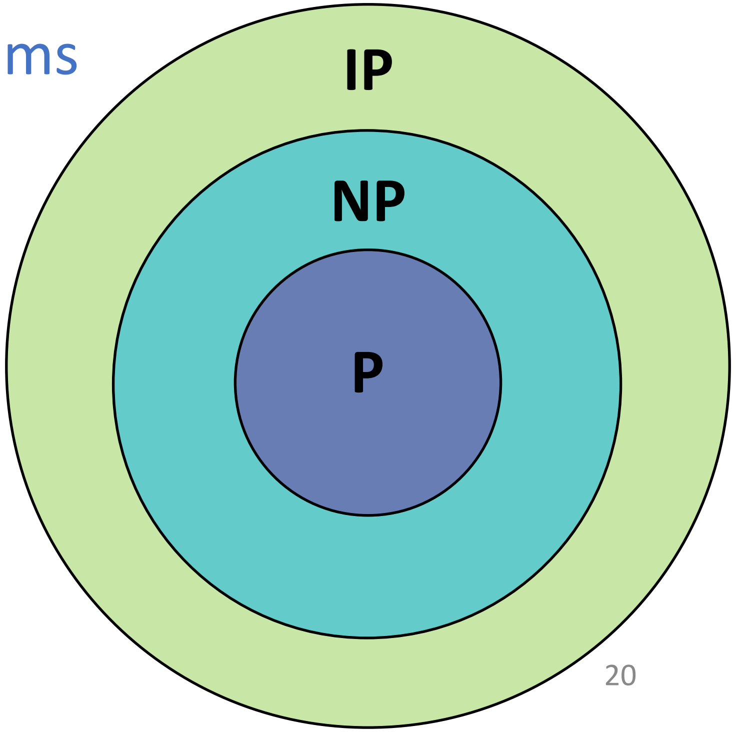
increased proving power = proofs for larger $\underbrace{\text{classes of statements}}_{\text{complexity classes}}$

Definition:

binary encoding of a YES/NO problem statement

- A *statement/instance* is an element $x \in \{0,1\}^*$.
- A *language* is a set $\mathcal{L} \subseteq \{0,1\}^*$. collection of problems
- A *complexity class* is a set of languages.

problem types with similar properties



Example:

- $\mathcal{L}_{G3C} = \{\text{graphs } G \text{ which have a } 3\text{−colouring}\}$.

Problem: decide whether G has a 3-colouring or not

The complexity class **P**

Definition:

P is the set of languages $\mathcal{L}$ for which

$\exists$ decision algorithm M and polynomial q — i.e. Turing Machine

such that $\forall x \in \{0,1\}^*$,

- $M(x) = 1 \iff x \in \mathcal{L}$; and
- M finishes in $\leq q(|x|)$ steps on input x. — i.e. polynomial time, efficient

Problems which are easy to solve $\longleftrightarrow$ Efficient checks that $x \in \mathcal{L}$

The complexity class **NP**

Definition:

NP is the set of languages $\mathcal{L}$ for which

$\exists$ language $\mathcal{R}_{\mathcal{L}}$ and polynomial q such that

- $\forall x \in \mathcal{L},\ \exists w$ with $|w| \leq q(|x|)$ and $(x, w) \in \mathcal{R}_{\mathcal{L}}$;
- $\forall x \notin \mathcal{L},\ \nexists w$ with $(x, w) \in \mathcal{R}_{\mathcal{L}}$; and
- $\mathcal{R}_{\mathcal{L}} \in$ **P**.

We call w the *witness* to x.

a *solution* to x

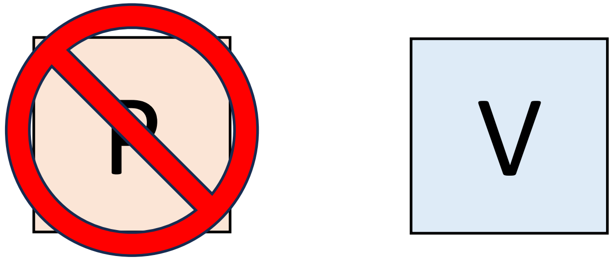
Problems whose YES solutions are easy to check $\longleftrightarrow$ Efficient checks that $(x, w) \in \mathcal{R}_{\mathcal{L}}$

It could be hard to find w (if $\mathbf{P} \neq \mathbf{NP}$).

Traditional proofs and complexity classes

Task: prove that $x \in \mathcal{L}$.

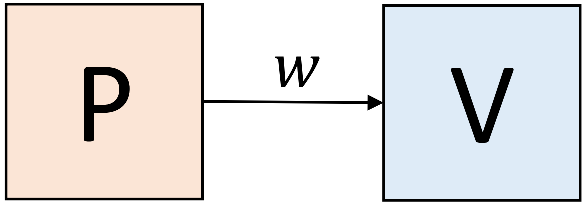
$\mathcal{L} \in$ **P** $\longleftrightarrow$ $x \in \mathcal{L}$ is obvious and needs no proof!



Efficient checks that $x \in \mathcal{L}$

$\mathcal{L} \in$ **NP** $\longleftrightarrow$ $x \in \mathcal{L}$ has a proof which is easy to check

w may be hard to find



Efficient checks that $(x, w) \in \mathcal{R}_{\mathcal{L}}$

$x \notin \mathcal{L} \Rightarrow (x, w) \notin \mathcal{R}_{\mathcal{L}}$

Interactive algorithms

Definition: Functions, Turing machines, circuits

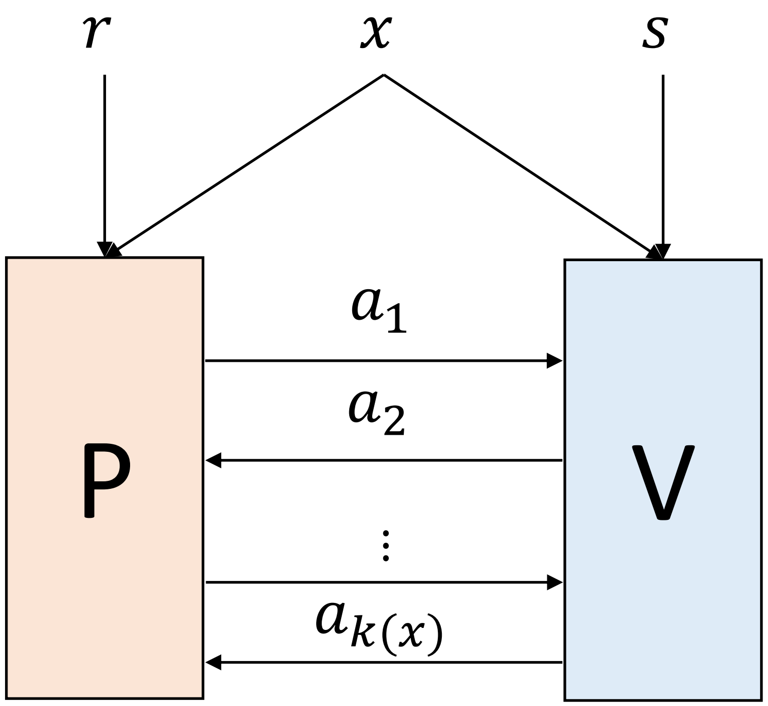
Let $P, V : \{0,1\}^* \to \{0,1\}^*$. Let $k : \{0,1\}^* \to \mathbb{N}$.

A $k(x)$-*move interaction* between P and V

on input $x \in \{0,1\}^*$ is defined by:

- $a_1 = P(x, r)$
- $a_2 = V(x, a_1, s)$
- $a_3 = P(x, a_1, a_2, r)$

  $\vdots$
- $a_{k(x)} = V\left(x, a_1, \dots, a_{k(x)-1}, s\right)$



The interaction is denoted $\langle P(r), V(s)\rangle(x) = a_{k(x)}$. The result

The *transcript* is $(x, a_1, \dots, a_{k(x)})$.

The complexity class **IP**

Definition:

P need not be efficient!

$\mathcal{L} \in$ **IP** if $\exists$ P and V with V efficient (polynomial time in $|x|$) satisfying:

- *Completeness*

$\forall x \in \mathcal{L}, \Pr_{r,s}[\langle P(r), V(s)\rangle(x) = 1] \geq {}^3/_4$

"True statements usually accepted"

- *Soundness*

$\forall x \notin \mathcal{L}, \forall\, P^*, \Pr_{r,s}[\langle P^*(r), V(s)\rangle(x) = 1] \leq {}^1/_2$

"False statements usually rejected"

If so, we say that (P, V) is an *interactive proof system* for $\mathcal{L}$.

Any constants with a gap define the same complexity class

Agenda

- Definitions of complexity classes and IPs ✓
- **IP for graph non-isomorphism – increased proving power**
- IP for graph isomorphism – counterintuitive properties

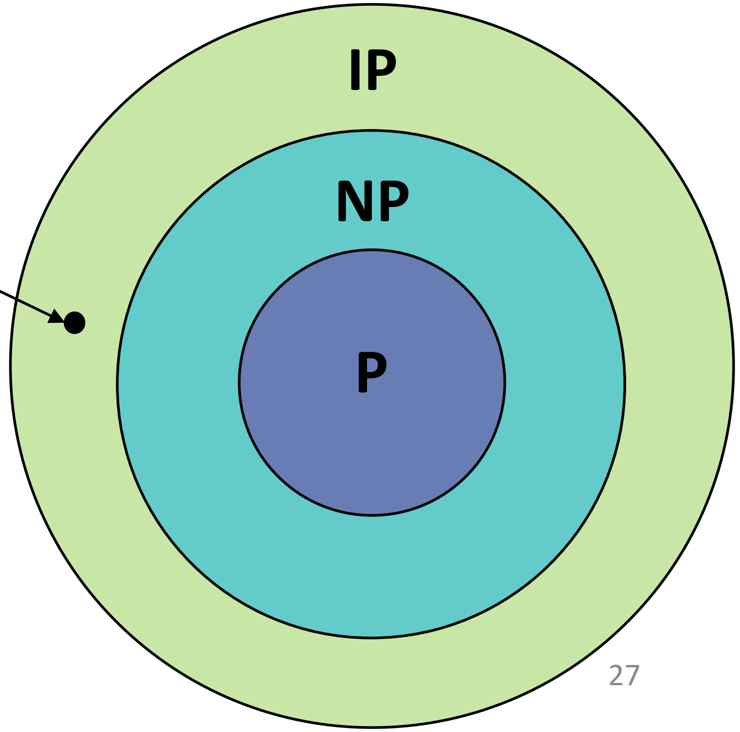
Graph Non-Isomorphism Problem

Definitions:

- Two graphs G_0 and G_1 on vertex set $[n]$ are *isomorphic* (written $G_0 \cong G_1$) if there exists a permutation $\pi : [n] \to [n]$ such that $G_0 = \pi(G_1)$.

π relabels vertices

- $\mathcal{L}_{GNI} \coloneqq \left\{ \begin{matrix} (G_0, G_1) : G_0, G_1, \text{vertex set } [n] \\ G_0 \ncong G_1 \end{matrix} \right\}$



$\exists$ an interactive proof for $\mathcal{L}_{GNI}$

$\mathcal{L}_{GNI} \notin$ **NP** (as far as we know)

Evidence that **NP** $\neq$ **IP**.

Interactive proof for $\mathcal{L}_{GNI}$

$P(G_0, G_1)$

Test $H \cong G_0$ or $H \cong G_1$ by computing all permutations of H

$V(G_0, G_1)$

$b \leftarrow_\$ \{0,1\}$

$\pi \leftarrow_\$ \mathrm{Perm}(n)$

$H = \pi(G_b)$

$P \xleftarrow{H} V$

$P \xrightarrow{B \text{ s.t. } H \cong G_B} V$

Output $(B == b)$

Security analysis: Traditional mathematical proofs of completeness and soundness

Completeness: $G_0 \ncong G_1$

- $(H \cong G_0) \wedge (H \cong G_1) \Rightarrow G_0 \cong G_1$ #
- So $H \cong G_b$ and $H \ncong G_{1-b}$
- P must choose $B = b$ so V accepts

Always accepts

Soundness: $G_0 \cong G_1$

- $\pi(G_0), \pi(G_1)$ identically distributed
- B, b independent
- $\Pr[B = b] = {}^1/_2 \leq {}^1/_2$

Agenda

- Definitions of complexity classes and IPs ✓
- IP for graph non-isomorphism – increased proving power ✓
- **IP for graph isomorphism – counterintuitive properties**

Transfer of knowledge

$$\sqrt{2} \notin \mathbb{Q}$$

1. Suppose $\sqrt{2} = \frac{a}{b} \in \mathbb{Q}$, $\gcd(a, b) = 1$
2. $\sqrt{2}b = a$ so $2b^2 = a^2$
3. $2 \mid a^2$ so $2 \mid a$
4. $\exists c$ with $a = 2c$
5. $2b^2 = 4c^2$ so $b^2 = 2c^2$
6. $2 \mid b^2$ so $2 \mid b$
7. Contradiction to $\gcd(a, b) = 1$

Transfers knowledge from P to V

You can now generalise to $\sqrt{3}, \sqrt{5}, \ldots$

Can we avoid this?

Hard to produce if you didn't know it already

Easy to produce if you did know it already

knowledge gained

no knowledge gained

Now easy to produce and generalise

Working definition of algorithmic knowledge

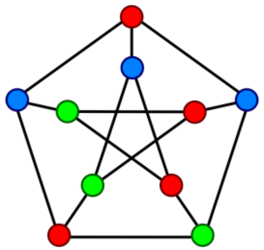
Informal Definition:

IP definition: P unbounded, V efficient

An algorithm's *knowledge* is anything that it can compute efficiently.

Examples:	Every-day	Algorithmic
1. Your birthday	Yes	Yes
2. 100th decimal digit of $\frac{\text{your birth month}}{\text{your birth year}} \cdot \sqrt{2}$	No	Yes
3. 43rd digit of π	Not usually	Yes
4. 3-colouring of a large graph	No	Probably not

NP-complete



Zero-knowledge

No knowledge gained by verifiers who could already produce the proof by themselves.

V may not follow the protocol

Definition:

Let (P, V) be an IP for $\mathcal{L}$.

everything V sees

- The *verifier's view* is $\mathrm{View}_V^P = (x, s, a_1, \dots, a_{k(x)})$.

expected probabilistic polynomial time

- (P, V) is *perfect zero-knowledge* if $\forall$ efficient V^*, $\exists$ efficient simulator S such that $\forall x \in \mathcal{L}$, we have $\left\{\mathrm{View}_{V^*}^P\right\} = \{S(V^*, x)\}$.

equal as probability distributions

- If so, (P, V) is a *perfect zero-knowledge proof (perfect ZKP)*.

Life lessons

- Interaction is more powerful. You can achieve more by asking questions and having discussions.

- Don’t have conversations that you can already simulate in your head – you won’t learn anything!

Next time:

- Variations on soundness and zero-knowledge.
- Zero-knowledge for an NP-complete problem.