Introduction to Zero-Knowledge Proofs

Distributed Lab

August 22, 2024

Plan

1. Introduction
   - Classical Proofs
   - Goal of the course
2. Relations. Languages. NP Statements.
   - Language of true statements. Examples.
3. Interactive Proofs
   - Interactive Proof System

Introduction

Classical Proofs

- First proofs you have probably encountered were **geometry proofs**.
- You were given **axioms** and you can prove certain **statements** x using them.
- The proof π is a sequence of logical steps that lead from axioms to the statement. Essentially, you have a witness w that proves the statement.
- Your teacher is the **verifier** $\mathcal{V}$ who checks your proof, while you are the **prover** $\mathcal{P}$.
- This is a **classical proof** and in a sense, it is a **non-interactive proof**.

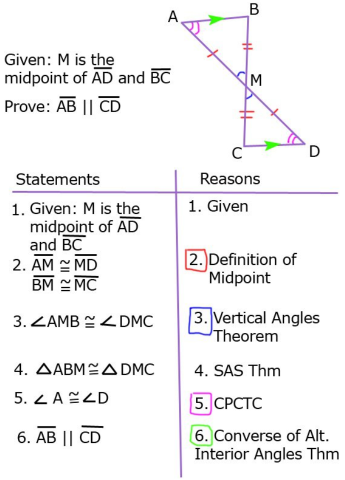


Figure: Geometry proof.

Motivation

Note

However, we cannot use such proofs in the digital world.

- Proofs must be verified by computers. Therefore, we need to develop **mathematic framework** to be able to program them.
- This leads to the question: what is **statement**? What is **proof**? What is **witness**? How to formally define them?
- We need to formalize these concepts.

Figure: Hmm. . .

The most basic setting

- We have a **prover** $\mathcal{P}$ and a **verifier** $\mathcal{V}$.
- Prover $\mathcal{P}$ wants to prove some statement x to the verifier.
- Prover $\mathcal{P}$ has a **witness** w that contains all necessary information to prove the statement x. He sends π as a proof.
- Verifier $\mathcal{V}$ wants to be convinced that the statement x is true.

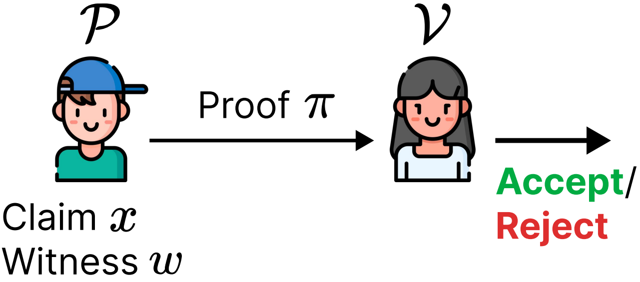


Figure: Typical setup for cryptographic proofs.

The Goal of SNARKs, STARKs etc.

We will try to solve the following problems:

- **Completeness:** If x is true, π proofs the statement.
- **Soundness:** If x is false, the prover $\mathcal{P}$ should not be able to convince the verifier $\mathcal{V}$ via any π^*.
- **Zero-knowledge:** π does not reveal anything about w.
- **Argument of knowledge:** Sometimes, the prover $\mathcal{P}$ should convince the verifier $\mathcal{V}$ that besides x is true, he **knows** the witness w.
- **Succinctness:** The proof should be short, ideally polylogarithmic in the size of the statement ($\pi = \text{polylog}(|x|)$) + fast verification.
- **Arithmetization:** We need to convert the statement x into some algebraic form + make it relatively universal.

Note

SNARK, STARK, etc. will solve these problems!

Example to demonstrate the goal

Example

Given a hash function $H : \{0,1\}^* \to \{0,1\}^\ell$, $\mathcal{P}$ wants to convince $\mathcal{V}$ that he knows the preimage $x \in \{0,1\}^*$ such that $H(x) = y$.

- **Zero-knowledge:** The prover $\mathcal{P}$ does not want to reveal *anything* about the pre-image x to the verifier $\mathcal{V}$.
- **Argument of knowledge:** Proving y has a pre-image is useless. $\mathcal{P}$ must show he **knows** $x \in \{0,1\}^*$ s.t. $H(x) = y$.
- **Succinctness:** If the hash function takes n operations to compute, the proof should be **much** shorter than n operations. **State-of-art**: size is $\text{polylog}(n) = O((\log n)^c)$. Verification time is also typically polylogarithmic (or even $O(1)$ in some cases).

Note

But first, let us start with the basics.

Relations. Languages. NP Statements.

Language

Definition (Relation)

Given two sets $\mathcal{X}$ and $\mathcal{Y}$, the **relation** is $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$.

- $\mathcal{X}$ is typically a set of **statements**.
- $\mathcal{Y}$ is a set of **witnesses**.

Definition (Language of true statements)

Let $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ be a relation. We say that a statement $x \in \mathcal{X}$ is a **true** statement if $(x, y) \in \mathcal{R}$ for some $y \in \mathcal{Y}$, otherwise the statement is called **false**. We define by $\mathcal{L}_{\mathcal{R}}$ (the language over relation $\mathcal{R}$) the set of all true statements, that is:

$$\mathcal{L}_{\mathcal{R}} = \{x \in \mathcal{X} : \exists y \in \mathcal{Y} \text{ such that } (x, y) \in \mathcal{R}\}.$$

Language Example #1: Semiprimes

Example (Product of Two Primes (Semiprimes))

Claim: number $n \in \mathbb{N}$ is the product of two prime numbers $w = (p, q) \in \mathbb{N} \times \mathbb{N}$. The **relation** is given by:

$$\mathcal{R} = \{(n, p, q) \in \mathbb{N}^3 : n = p \cdot q \text{ where } p, q \text{ are primes}\}$$

In this particular case, the **language of true statements** is defined as

$$\mathcal{L}_{\mathcal{R}} = \{n \in \mathbb{N} : \exists w = (p, q) \text{ are primes such that } n = p \cdot q\}$$

- **Valid witness #1:** $n = 15 \in \mathcal{L}_{\mathcal{R}}$. Witness: $w = (3, 5)$.
- **Invalid witness:** $n = 16 \notin \mathcal{L}_{\mathcal{R}}$. There is no valid witness.
- **Valid witness #2:** $n = 50252009 \in \mathcal{L}_{\mathcal{R}}$. Witness: $w = (5749, 8741)$.

Question: Is $n = 27$ a true statement? What about $n = 26$?

Language Example #2: Square Root

Reminder

$\mathbb{Z}_N^\times = \{x \in \mathbb{Z}_N : \gcd\{x, N\} = 1\}$. **Example:** $\mathbb{Z}_{10}^\times = \{1, 3, 7, 9\}$

Example

Claim: number $x \in \mathbb{Z}_N^\times$ is a **quadratic residue** modulo N:
$(\exists w \in \mathbb{Z}_N^\times) : \{x \equiv w^2 \pmod{N}\}$ (w is **modular square root** of x).
Relation: $\mathcal{R} = \{(x, w) \in (\mathbb{Z}_N^\times)^2 : x \equiv w^2 \pmod{N}\}$
Language: $\mathcal{L}_\mathcal{R} = \{x \in \mathbb{Z}_N^\times : \exists w \in \mathbb{Z}_N^\times \text{ such that } x \equiv w^2 \pmod{N}\}$.
Examples for $N = 7$:

- $4 \in \mathcal{L}_\mathcal{R}$ since $5^2 \equiv 4 \pmod{7}$.
- $3 \notin \mathcal{L}_\mathcal{R}$ since there is no valid witness for 3.

Question: Is $x = 1$ a true statement for $N = 5$? What about $x = 4$?

NP Statements: Demonstration

Well. . . We are simply going to send witness w to the verifier $\mathcal{V}$ and he will check if the statement is true (meaning, whether $x \in \mathcal{L}_{\mathcal{R}}$).

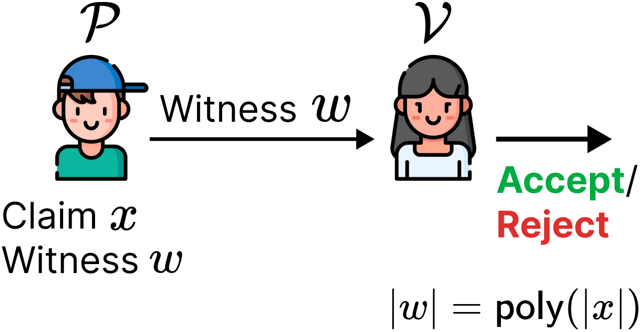


Figure: Typical setup for cryptographic proofs.

NP Statements

Definition (P Language)

Problem is in the **P** class if exists a polytime algorithm checking $x \in \mathcal{L}$.

Definition (NP Language)

A language $\mathcal{L}_{\mathcal{R}}$ belongs to the **NP** class if there exists a polynomial-time verifier $\mathcal{V}$ such that the following two properties hold:

- **Completeness:** If $x \in \mathcal{L}_{\mathcal{R}}$, then there is a witness w such that $\mathcal{V}(x, w) = 1$ with $|w| = \text{poly}(|x|)$. Essentially, it states that true claims have *short* proofs.
- **Soundness:** If $x \notin \mathcal{L}_{\mathcal{R}}$, then for any w it holds that $\mathcal{V}(x, w) = 0$. Essentially, it states that false claims have no proofs.

Theorem

Any **NP** *problem has a zero-knowledge proof.*

Question (aka Motivation)

But can we do better?
Sending witness is. . . Weird. . .

Figure: Hmm. . . #2

Interactive Proofs

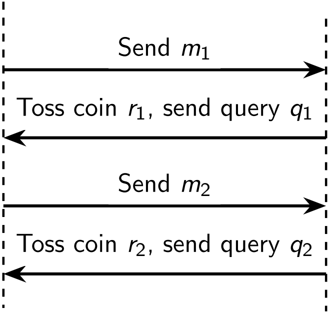
Solution!

We add two more ingredients:

- **Interaction:** instead of **passively** receiving the proof, the verifier $\mathcal{V}$ can **interact** with the prover $\mathcal{P}$ by sending **challenges** and receiving **responses**.
- **Randomness:** $\mathcal{V}$ can send random coins (challenges) to the prover, which $\mathcal{P}$ can use to generate responses.

Prover $\mathcal{P}$
Comp. Unbounded

Verifier $\mathcal{V}$
Probabilistic Poly-Time (PPT)



Quadratic Residue Interactive Proof

Problem Statement

- **Statement:** $x \in \mathcal{L}_{\mathcal{R}}$ where our **language** is defined as:

$$\mathcal{L}_{\mathcal{R}} = \{x \in \mathbb{Z}_N^{\times} : \exists w \in \mathbb{Z}_N^{\times} \text{ such that } x \equiv w^2 \pmod{N}\}$$

- **Witness:** $w =$ modular square root of x.

How does $\mathcal{P}$ and $\mathcal{V}$ interact? Consider the figure below.

$\mathcal{P}$

I know w s.t.
$w^2 = x \pmod{N}$

1. Sample r from $\mathbb{Z}_N$ uniformly
2. Send $a = r^2 \pmod{N}$

$\mathcal{V}$

Is x indeed a quadr. residue?

Quadratic Residue Interactive Proof

$\mathcal{P}$

I know w s.t.
$w^2 = x \pmod N$

1. Sample r from $\mathbb{Z}_N$ uniformly
2. Send $a = r^2 \pmod N$

- If I gave you the square root of a and ax, you would be convinced that the claim is true, but you learn the witness w.
- Instead, I will send you either r or rw, but you are to choose!

$\mathcal{V}$

Is x indeed a quadr. residue?

Quadratic Residue Interactive Proof

$\mathcal{P}$

I know w s.t.
$w^2 = x \pmod N$

1. Sample r from $\mathbb{Z}_N$ uniformly
2. Send $a = r^2 \pmod N$

- If I gave you the square root of a and ax, you would be convinced that the claim is true, but you learn the witness w.
- Instead, I will send you either r or rw, but you are to choose!

$\mathcal{V}$

Is x indeed a quadr. residue?

Ok, I choose random bit b

Quadratic Residue Interactive Proof

$\mathcal{P}$

I know w s.t.
$w^2 = x \pmod N$

1. Sample r from $\mathbb{Z}_N$ uniformly
2. Send $a = r^2 \pmod N$

- If I gave you the square root of a and ax, you would be convinced that the claim is true, but you learn the witness w.
- Instead, I will send you either r or rw, but you are to choose!

$\mathcal{V}$

Is x indeed a quadr. residue?

Ok, I choose random bit b

- If $b=0$, send $z = r$
- If $b=1$, send $z = rw \pmod N$

Check if $z^2 = ax^b$

Quadratic Residue Interactive Proof: Analysis

Interactive Protocol

1. $\mathcal{P}$ samples $r \xleftarrow{R} \mathbb{Z}_N^\times$ and sends $a = r^2$ to $\mathcal{V}$.
2. $\mathcal{V}$ sends a random bit $b \in \{0, 1\}$ to $\mathcal{P}$.
3. $\mathcal{P}$ sends $z = r \cdot w^b$ to $\mathcal{V}$.
4. $\mathcal{V}$ accepts if $z^2 = a \cdot x^b$, otherwise it rejects.
5. Repeat $\lambda \in \mathbb{N}$ times.

Lemma

The protocol is **complete** *and* **sound**.

Completeness. If $b = 0$, then $z = r$ and thus $z^2 = r^2 = a$, check passes. If $b = 1$, then $z = rw$ and thus $z^2 = r^2w^2 = ax$, check passes.

Quadratic Residue Interactive Proof: Analysis

Soundness. The main reason why the protocol is sound is insribed in the theorem below.

Theorem

For any prover $\mathcal{P}^*$ with $x \notin \mathcal{L}_{\mathcal{R}}$, the probability of $\mathcal{V}$ accepting the proof is at most $1/2$.

Corollary. After repeating the protocol λ times, we have

$$\Pr[\mathcal{V} \text{ accepts after } \lambda \text{ rounds}] \leq \frac{1}{2^\lambda} = \mathsf{negl}(\lambda).$$

Thus, we showed both **completeness** and **soundness** of the protocol.

Interactive Protocol Definition

$\langle \mathcal{P}, \mathcal{V} \rangle(x)$ reads as "interaction between $\mathcal{P}$ and $\mathcal{V}$ on the statement x".

Definition

A pair of algorithms $(\mathcal{P}, \mathcal{V})$ is called an **interactive proof** for a language $\mathcal{L}_\mathcal{R}$ if $\mathcal{V}$ is a polynomial-time verifier and the following two properties hold:

- **Completeness:** For any $x \in \mathcal{L}_\mathcal{R}$, $\Pr[\langle \mathcal{P}, \mathcal{V} \rangle(x) = \text{accept}] = 1$.
- **Soundness:** For any $x \notin \mathcal{L}_\mathcal{R}$ and for any prover $\mathcal{P}^*$, we have

$$\Pr[\langle \mathcal{P}^*, \mathcal{V} \rangle(x) = \text{accept}] \leq \text{negl}(\lambda)$$

Definition

The class of interactive proofs (**IP**) is defined as:

$$\textbf{IP} = \{\mathcal{L} : \text{there is an interactive proof } (\mathcal{P}, \mathcal{V}) \text{ for } \mathcal{L}\}.$$

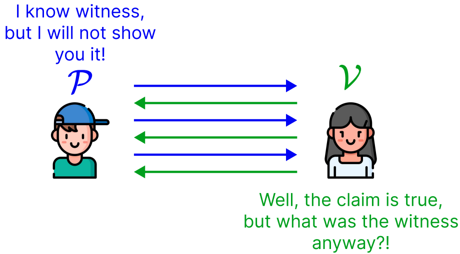
Zero-Knowledge Informal Definition

Definition

An interactive proof system $(\mathcal{P}, \mathcal{V})$ is called **zero-knowledge** if for any polynomial-time verifier $\mathcal{V}^*$ and any $x \in \mathcal{L}_{\mathcal{R}}$, the interaction $\langle \mathcal{P}, \mathcal{V}^* \rangle(x)$ gives nothing new about the witness w.

Definition

The pair of algorithms $(\mathcal{P}, \mathcal{V})$ is called a **zero-knowledge interactive protocol** if it is *complete*, *sound*, and *zero-knowledge*.



Thanks for your attention!