Knowledge and Interactive Proofs

Ekene Ezeunala

July 3rd, 2022

Outline

1 Introduction

2 A Theory of Knowledge

3 Encryption

4 Interactive Proofs

5 References

One-Way Functions

Definition (One-Way Functions)

A function $f : \{0,1\}^* \to \{0,1\}^*$ is called *one-way* if the following two conditions hold:

- *easy to evaluate*: There exists a polynomial-time algorithm A such that $A(x) = f(x)$ for every $x \in \{0,1\}^*$.
- *hard to invert*: For every family of polynomial size circuits $\{C_n\}$, every polynomial p, and all sufficiently large n,

$$\Pr[C_n(f(x)) \in f^{-1}(f(x))] < \frac{1}{p(n)},$$

where the probability is taken uniformly over all the possible choices of $x \in \{0,1\}^n$.

Computational indistinguishability

Definition

We say that $X = \{X_\alpha\}_{\alpha \in S}$ and $Y = \{Y_\alpha\}_{\alpha \in S}$ are *computationally indistinguishable* if for every family of polynomial-size circuits $\{D_n\}$, every polynomial p, all sufficiently large n and every $\alpha \in \{0,1\}^n \cap S$,

$$|\Pr[D_n(X_\alpha) = 1] - \Pr[D_n(Y_\alpha) = 1]| < \frac{1}{p(n)}$$

where the probabilities are taken over the relevant distribution (i.e., either X_n or Y_n).

Knowledge

- We think of knowledge here as a required enabler—knowledge is only regarded as such when it is necessarily actionable.
- Quantifying the knowledge inherent in a message $m \equiv$ quantifying how much easier it becomes to compute some new function given m.

Example

Suppose, given a one-way function f, Alice sends the message m to Bob, where

$$m := \underbrace{\text{the preimage}}_{n \text{ times}} \text{ of } 0.$$

- m is deterministic and has a short description.
- But Bob may not be able to reproduce this message himself because it is computationally expensive to do so.

Notion

The amount of knowledge conveyed in a message can be quantified by considering the running time and size of a Turing machine that generates the message.

Non-deterministic *m* and zero knowledge

Suppose now that Alice randomly draws her message from a probability distribution.

- It then suffices to consider the complexity of a Turing machine that similarly samples m from a computationally indistinguishable distribution.

Notion

Alice conveys zero knowledge to Bob if Bob can sample from a distribution of messages that is computationally indistinguishable from the distribution of messages that Alice would send.

Knowledge and Encryption

Definition (Security, informal)

An encryption scheme is *secure* if the encrypted message doesn't allow an eavesdropper to compute any new (effectively computable) function about the plaintext message wrt to what she would have computed without the encrypted message.

Definition (Zero-Knowledge Encryption)

A private-key encryption scheme $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is a *zero-knowledge encryption scheme* if there exists a probabilistic polynomial time (ppt) simulator algorithm S such that $\forall$ non-uniform ppt D, $\exists$ a negligible function $\epsilon(n)$ such that $\forall m \in \{0,1\}^n$, D distinguishes the following distributions with probability at most $\epsilon(n)$:

- $\{k \leftarrow \mathrm{Gen}(1^n) : \mathrm{Enc}_k(m)\}$
- $\{S(1^n)\}$.

Security of encryption schemes

Theorem

Let $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *be an encryption scheme such that* $\mathrm{Gen}, \mathrm{Enc}$ *are both p.p.t, and there exists a polynomial-time machine* M *such that for every* n, $M(n)$ *outputs a message in* $\{0,1\}^n$. *Then* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *is secure iff it is zero-knowledge.*

Proof outline

Security implies ZP.

If we could extract knowledge from the encrypted message, then we'd be able to distinguish between encryptions of two different messages m and m'. Suppose $\exists$ simulator $S(1^n)$: pick $m \in \{0,1\}^n$, $k \leftarrow \mathrm{Gen}(1^n)$, $c \leftarrow \mathrm{Enc}_k(m)$, and output c.

Assume $\exists$ n.u. p.p.t. distinguisher D and polynomial $p(n)$ such that for ∞ n, $\exists m_n'$ such that D distinguishes $\{k \leftarrow \mathrm{Gen}(1^n) : \mathrm{Enc}_k(m_n)\}$ and $\{S(1^n)\}$ with probability $p(n)$. As

$$\{S(1^n)\} = \{k \leftarrow \mathrm{Gen}(1^n); m_n' \leftarrow M(1^n) : \mathrm{Enc}_k(m_n')\},$$

such m_n and m_n' exist; this contradicts the security of $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$. □

ZP implies security.

Suppose $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is zero-knowledge, but $\exists$ a n.u. p.p.t. distinguisher D and a polynomial $p(n)$ such that for ∞ n, $\exists$ m_n^1 and m_n^2 such that D distinguishes

- $H_n^1 = \{k \leftarrow \mathrm{Gen}(1^n) : \mathrm{Enc}_k(m_n^1)\}$
- $H_n^2 = \{k \leftarrow \mathrm{Gen}(1^n) : \mathrm{Enc}_k(m_n^2)\}$

with probability $p(n)$. Define the hybrid dist. $H_n^3 = \{S(1^n)\}$. By the hybrid lemma, D distinguishes between the pairwise adjacent distributions with probability $\frac{1}{2p(n)}$ for infinitely many n; this is a contradiction. □

Motivation

- Why proofs? To convince other people something is true.

Motivation

- Why proofs? To convince other people something is true.
- Interactive proofs—randomized and dynamic

Motivation

- Why proofs? To convince other people something is true.
- Interactive proofs—randomized and dynamic
  - prover: has unlimited computational ability; verifier: only operates in PPT

Motivation

- Why proofs? To convince other people something is true.
- Interactive proofs—randomized and dynamic
    - prover: has unlimited computational ability; verifier: only operates in PPT
    - multi-round randomized protocol

Motivation

- Why proofs? To convince other people something is true.
- Interactive proofs—randomized and dynamic
  - prover: has unlimited computational ability; verifier: only operates in PPT
  - multi-round randomized protocol
  - allow the prover to convince the verifier of the validity of a true assertion

Motivation

- Why proofs? To convince other people something is true.
- Interactive proofs—randomized and dynamic
    - prover: has unlimited computational ability; verifier: only operates in PPT
    - multi-round randomized protocol
    - allow the prover to convince the verifier of the validity of a true assertion
    - prevent any malicious prover strategy from fooling the verifier

Motivation

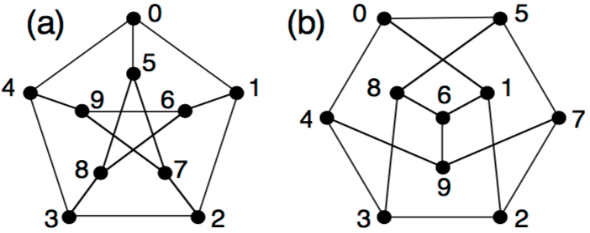
- Why proofs? To convince other people something is true.
- Interactive proofs—randomized and dynamic
    - prover: has unlimited computational ability; verifier: only operates in PPT
    - multi-round randomized protocol
    - allow the prover to convince the verifier of the validity of a true assertion
    - prevent any malicious prover strategy from fooling the verifier
- "Informal" idea of the interaction:

$$\text{“Tricky” questions} + \text{“Convincing” responses} \implies \text{Valid statement}$$

Illustration

Definition

Two undirected graphs G and H are isomorphic if they are identical except for a permutation of the nodes.



Problem (ISO)

Given two graphs G and H, are they isomorphic?

Complexity

- $\mathrm{ISO} \in \mathrm{NP}$?—yes
- $\mathrm{ISO} \in \mathrm{P}$?—unknown
- $\mathrm{ISO} \in \mathrm{NP{-}complete}$?—unknown
- $\overline{\mathrm{ISO}} \in \mathrm{NP}$?—unknown
    - That is, given two graphs, the question of whether they are not isomorphic is not known to be in NP.

Remarks

1. $\mathrm{ISO} \in \mathrm{NP}$, *so a prover can convince a poly-time verifier that G and H are isomorphic (if true).*
2. *Regardless of* $\overline{\mathrm{ISO}} \in \mathrm{NP}$, *a prover can still convince a verifier that G and H are not isomorphic (if true), provided the prover and the (probabilistic) verifier can interact with each other.*

$\mathcal{IP}$—Informal Model

- Imagine Victor wants to show that two graphs are not isomorphic; to solve graph non-isomorphism $\overline{\mathrm{ISO}}$.

$\mathcal{IP}$—Informal Model

- Imagine Victor wants to show that two graphs are not isomorphic; to solve graph non-isomorphism $\overline{\text{ISO}}$.
- However, Victor only exists in probabilistic polynomial time.

$\mathcal{IP}$—Informal Model

- Imagine Victor wants to show that two graphs are not isomorphic; to solve graph non-isomorphism $\overline{\mathrm{ISO}}$.
- However, Victor only exists in probabilistic polynomial time.
- So he enlists the help of Patty and Peggy, both of whom have unlimited computational ability, to do this.

$\mathcal{IP}$—Informal Model

- Imagine Victor wants to show that two graphs are not isomorphic; to solve graph non-isomorphism $\overline{\mathrm{ISO}}$.
- However, Victor only exists in probabilistic polynomial time.
- So he enlists the help of Patty and Peggy, both of whom have unlimited computational ability, to do this.
- Patty and Peggy have other things going on, so they're not the most devoted to solving this problem. Victor knows this, so he doesn't trust that they indeed solved it.

$\mathcal{IP}$—Informal Model

- Imagine Victor wants to show that two graphs are not isomorphic; to solve graph non-isomorphism $\overline{\mathrm{ISO}}$.
- However, Victor only exists in probabilistic polynomial time.
- So he enlists the help of Patty and Peggy, both of whom have unlimited computational ability, to do this.
- Patty and Peggy have other things going on, so they're not the most devoted to solving this problem. Victor knows this, so he doesn't trust that they indeed solved it.
- If G and H are isomorphic, then Patty and Peggy can immediately present this isomorphism and convince Victor.

$\mathcal{IP}$—Informal Model

- Imagine Victor wants to show that two graphs are not isomorphic; to solve graph non-isomorphism $\overline{\mathrm{ISO}}$.
- However, Victor only exists in probabilistic polynomial time.
- So he enlists the help of Patty and Peggy, both of whom have unlimited computational ability, to do this.
- Patty and Peggy have other things going on, so they're not the most devoted to solving this problem. Victor knows this, so he doesn't trust that they indeed solved it.
- If G and H are isomorphic, then Patty and Peggy can immediately present this isomorphism and convince Victor.
- But what if they weren't isomorphic?

Scheme for $\mathcal{IP}$ Verifier—Informal Model

- Victor randomly & secretly picks G or H, permutes its nodes to get a new graph K, and sends K to Patty and Peggy, along with the question, “Which graph did I permute?”

Scheme for $\mathcal{IP}$ Verifier—Informal Model

- Victor randomly & secretly picks G or H, permutes its nodes to get a new graph K, and sends K to Patty and Peggy, along with the question, "Which graph did I permute?"
- K must be isomorphic to G or H.

Scheme for $\mathcal{IP}$ Verifier—Informal Model

- Victor randomly & secretly picks G or H, permutes its nodes to get a new graph K, and sends K to Patty and Peggy, along with the question, “Which graph did I permute?”
- K must be isomorphic to G or H.
- If they were non-isomorphic in the first place, then picking out G or H is trivial.

Scheme for $\mathcal{IP}$ Verifier—Informal Model

- Victor randomly & secretly picks G or H, permutes its nodes to get a new graph K, and sends K to Patty and Peggy, along with the question, “Which graph did I permute?”
- K must be isomorphic to G or H.
- If they were non-isomorphic in the first place, then picking out G or H is trivial.
- If they were, then it is essentially reduced to a coin flip between G and H.

Scheme for $\mathcal{IP}$ Verifier—Informal Model

- Victor randomly & secretly picks G or H, permutes its nodes to get a new graph K, and sends K to Patty and Peggy, along with the question, "Which graph did I permute?"
- K must be isomorphic to G or H.
- If they were non-isomorphic in the first place, then picking out G or H is trivial.
- If they were, then it is essentially reduced to a coin flip between G and H.
- If this process is repeated a large number of times, then the probability of malicious success is disastrous.

$\mathcal{IP}$—Formal Model

Definition (Interactive Proof)

A pair of interactive machines (P, V) is an interactive proof system for a language L if V is a ppt machine and the following properties hold:

- (*Completeness.*) For every $x \in L$, there exists a witness string $y \in \{0,1\}^*$ such that for every auxiliary string z:

$$\Pr[\mathrm{out}_V[P(x,y) \leftrightarrow V(x,z) = 1] = 1.$$

- (*Soundness.*) There exists some negligible function ϵ such that for all $x \notin L$ and for all prover algorithms P^*, and all auxiliary strings $z \in \{0,1\}^*$,

$$\Pr[\mathrm{out}_V[P^*(x) \leftrightarrow V(x,z)] = 0] > 1 - \epsilon(|x|).$$

The class of problems having interactive proof systems is denoted $\mathcal{IP}$.

Check-In

Suppose we change the model to allow the prover (Patty and Peggy) access to the verifier's (Victor's) random choices. Consider the $\mathcal{IP}$ protocol as described above. What language does it describe?

- $\{\langle G, H\rangle \mid G \neq H\}$
- $\{\langle G, H\rangle \mid G$ and H are not isomorphic$\}$
- $\{\langle G, H\rangle \mid G$ and H are any two graphs$\}$
- $\emptyset$

References

1. Knospe, Heiko. *A Course in Cryptography.* Vol. 40. American Mathematical Soc., 2019.
2. Goldreich, Oded. "A Short Tutorial of Zero-Knowledge." (2013): 28-60.
3. Thaler, Justin. "Proofs, arguments, and zero-knowledge." (2020).

Theorem

Ad-hoc anti-zero-knowledge protocols have been developed by most customer helpline services.

Proof.

THIS IS TRIVIAL, IMMEDIATELY OBVIOUS, AND IS LEFT AS AN EXERCISE FOR THE READER!!! □