Zero-knowledge Proofs

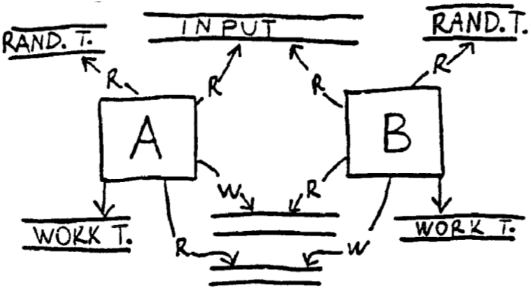
Yiannis Tselekounis

February 23, 2012

Basic concepts

- An efficient method of communicating a proof (interactively).
- Interactive proofs: The Prover and the Verifier exchange messages regarding a Theorem T.
- The Prover wants to convince the Verifier that T is True.
- e.g. for Hamiltonian graph the proof might be the Hamiltonian tour.
- What if we don't want to release so much knowledge?
- Computational complexity measure of knowledge.
- Language classification according to the amount of additional knowledge.
- Zero-knowledge :)

Interactive proof-systems



Interactive pair of Turing machines (A,B):

i. A and B share the same input tape,

ii. B's write-only communication tape is A's read-only tape and vice versa.

Interactive proofs are "easier" than NP-proofs.

Formally

Definition 1

Let $L \subseteq \{0,1\}^*$ be a language and (P, Vr) an IPTM, where P (prover) has infinite power and Vr (verifier) is polynomial time. (P, Vr) is an interactive proof-system for L if

i. *Completeness: for all $x \in L$, is we supply (P, Vr) with x, Vr halts and accepts with probability at least $1 - \frac{1}{n^k}$, for each k and n sufficiently large ($n = |x|$).*

ii. *Validity: for all $x \notin L$, and all ITMs P, Vr accepts with probability at most $\frac{1}{n^k}$, for each k and n sufficiently large.*

Example 1 (QR)

- Let $\mathbb{Z}_m^* = \{x \mid 1 \leq x \leq m,\ (x, m) = 1\}$. An element $\alpha \in \mathbb{Z}_m^*$ is a quadratic residue if $\alpha = x^2 \mod m$ for some $x \in \mathbb{Z}_m^*$.
- Let $L = \{(m, x) \mid x \in \mathbb{Z}_m^* \text{ is a quadratic nonresidue}\}$.
- The prover (P) computes the factorization of m and sends it to the verifier (Vr).
- Interactive proof-system:
    1. Vr chooses $r_i \in \mathbb{Z}_m^*$, for $1 \leq i \leq n$, randomly, $n = |m|$.
    2. For each i, she flips a coin:
       - heads $\rightarrow t_i = r_i^2 \mod m$,
       - tails $\rightarrow t_i = x \cdot r_i^2 \mod m$,
    3. Vr sends $t_1, \ldots, t_n$ to P.
    4. P using his power, finds which of the t_i are quadratic residues, i.e. P finds Vr's coin tosses.

Example 2 (Graph non-isomorphism)

- $G(Vr, E)$ and $H(V, F)$ are isomorphic $\leftrightarrow \exists \pi \in Perm(V)$ s.t. $(u, v) \in E$ iff $(\pi(u), \pi(v)) \in F$.
- Construct a random isomorphic $H(V, F)$ copy of $G(V, E)$: $\pi \in_R Perm(V)$ and $F = \{(\pi(u), \pi(v)) : (u, v) \in E\}$.
  Interactive proof-system for input $G_1(V, E_1)$, $G_2(V, E_2)$:
  i. Vr: chooses $\alpha_i \in_R \{1, 2\}, 1 \leq i \leq n$. Sends $H_i(V, F_i)$ s.t. H_i is a random isomorphic copy of G_{α_i}.
  ii. P: sends $\beta_i \in \{1, 2\}$ s.t. $H_i(V, F_i)$ is isomorphic to $G_{\beta_i}(V, E_{\beta_i})$.
  iii. Vr: if $\alpha_i = \beta_i$ accepts, else rejects.

Knowledge Complexity

- Question: Which communications convey knowledge?
- Answer: Those that transmit the output of an unfeasible computation.
- Question: How much knowledge should be communicated to prove theorem T?
- Answer: Enough to verify that T is true. Usually, much more (recall the preceding examples).
- We want to measure the additional knowledge that is being sent from the prover to the verifier.

Knowledge Complexity (Formally)

Definition 2

Let (P, Vr) *be an* $IPTM$, I *the set of its inputs and* $f : \mathbb{N} \to \mathbb{N}$, *non decreasing.* A *communicates at most* $f(n)$ *bits of knowledge to* Vr *if there exists* PPT *machine* M, *such that for all* PPT *algorithms* D, *the ensembles* $M[\cdot]$ *and* $(P, Vr)[\cdot]$ *are at most* $p = 1 - \frac{1}{2^{f(n)}}$ *distinguishable, i.e.*

$$|\Pr[D(M[x]) = 1] - \Pr[D((P, Vr)[x]) = 1]| < p + \frac{1}{|x|^k}.$$

We say that P *communicates at-most* $f(n)$ *bits of knowledge if for all polynomial-time* ITM*'s* Vr', P *communicates at most* $f(n)$ *bits of knowledge to* Vr'.

IP is the class of languages that possess an interactive proof system.

Knowledge Complexity (Cont.)

Definition 3

Let L be a language, (P, Vr) an interactive proof-system for L and $f : \mathbb{N} \rightarrow \mathbb{N}$ non decreasing. L has knowledge complexity $f(n)$ if, when restricting the inputs of (P, Vr) to the strings in L, P communicates at most $f(n)$ bits of knowledge (we denote this by $L \in KC(f(n))$).

- We concentrate only on the "yes-instances". If $x \in L$, Vr is convinced with overwhelming probability.
- Vr possesses the text of the entire computation.
- This text verifies that $x \in L$ and does not contain more than $f(n)$ bits of additional knowledge.
- If $L \in KC(0)$, then the text of the entire computation is irrelevant for any other purpose.

Languages in KC(0)

- Every language in P, RP, BPP.
- Let $n = p_1^{h_1} \cdots p_k^{h_k}$. Then $n \in BL$ if the number of different p_is congruent to $3 \mod 4$ is even.
- $L = \{(y, m) \mid y \text{ is a quadratic non-residue } \mod m\}$.

Graph non-isomorphism

Question

Is the interactive proof system for graph non-isomorphism zero-knowledge?

Graph non-isomorphism

Question

Is the interactive proof system for graph non-isomorphism zero-knowledge?

Answer

No: the verifier may use the prover in order to test to which of G_1, G_2 is a third graph G_3 isomorphic.

Graph non-isomorphism

Question

Is the interactive proof system for graph non-isomorphism zero-knowledge?

Answer

No: the verifier may use the prover in order to test to which of G_1, G_2 is a third graph G_3 isomorphic.

Solution

Let the verifier first prove to the prover that he knows an isomorphism between his query H and one of the input graphs.

Zero-knowledge for Graph isomorphism

Interactive proof-system for input $G_1(V, E_1)$, $G_2(V, E_2)$ (one round):

i. P: chooses $\pi \in_R Perm(V)$ and sends $H(V, F)$ (for G_1).
   Recall that $(\pi(u), \pi(v)) \in F$ iff $(u, v) \in E_1$.

ii. Vr: sends $\alpha \in_R \{1, 2\}$.

iii. P: if $\alpha \notin \{1, 2\}$ then halt, if $\alpha = 1$ then send π, else send $\pi\phi^{-1}$.
   ϕ denotes the isomorphism between G_1, G_2.

iv. Vr: if the received permutation is not an isomorphism between G_α and H then reject, else continue.

- The above system is an IP system for GI. The previous steps are executed n times ($n = |V|$).
- Zero-knowledge: Vr can generate random isomorphic copies of G_1, G_2 by himself.
- Do we achieve zero-knowledge if Vr deviates from the protocol?

Zero-knowledge for Graph isomorphism (Cont.)

Theorem 1

The previous protocol consitutes a zero-knowledge interactive proof system for Graph Isomorphism.

Zero-knowledge for Graph 3-Colorability

Definition 4

We consider a secure encryption scheme as a PPT algorithm f, that on input x and internal coin tosses r, outputs an encryption $f(x, r)$.

Interactive proof-system for input $G(V, E)$ (one round):

i. P: chooses $\pi \in_R Perm(\{1, 2, 3\})$ and random r_v's. Computes $R_v = f(\pi(\phi(v)), r_v)$ for all $v \in V$ and sends $R_1, \ldots, R_n$ to Vr.

ii. Vr: sends $e \in_R E$ to P.

iii. P: If $e \in E$, send $(\pi(\phi(u), r_u))$, $(\pi(\phi(v), r_v))$ to Vr. If $e \notin E$ stop.

iv. Vr: If $R_u = f(\pi(\phi(u)), r_u)$, $R_v = f(\pi(\phi(v)), r_v)$, $\pi(\phi(u))$, $\pi(\phi(v)) \in \{1, 2, 3\}$, then continue, else reject and stop.

The previous steps are executed m^2 times ($m = |E|$).

Zero-knowledge for Graph 3-Colorability (Cont.)

Theorem 2

If $f(\cdot, \cdot)$ is a secure probabilistic encryption, then the above protocol is a zero-knowledge interactive proof system for 3-colourability.

Further Results

Theorem 3

If $f(\cdot,\cdot)$ is a secure probabilistic encryption, then every NP language has a zero-knowledge interactive proof system.

Theorem 4

If there exists a secure probabilistic encryption, then every language in NP has a zero-knowledge interactive proof system in which the prover is a probabilistic polynomial-time machine that gets an NP proof as auxiliary input.

Theorem 5

If there exists a secure probabilistic encryption, then for every fixed k, every language in $IP(k)$ has zero-knowledge proof systems.

Bibliography

- S Goldwasser, S Micali, and C Rackoff. 1985. The knowledge complexity of interactive proof-systems. In Proceedings of the seventeenth annual ACM symposium on Theory of computing (STOC '85). ACM, New York, NY, USA, 291-304.
- Oded Goldreich, Silvio Micali, and Avi Wigderson. 1986. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In Proceedings of the 27th Annual Symposium on Foundations of Computer Science (SFCS '86). IEEE Computer Society, Washington, DC, USA, 174-187.

Thank you!