Zero-knowledge 101

Gabriele Vanoni

Politecnico di Milano

5 Dicembre 2017

13 Dicembre 2013, Sivio Micali al Politecnico di Milano

Dimostrazioni interattive e dimostrazioni a conoscenza zero

L'importanza dell'interazione

Un docente deve convincere i propri studenti della validità di un teorema. Ha almeno due possiblità:

- Presentare una dimostrazione in **classe**.
- Scrivere una dimostrazione nelle proprie dispense e lasciarla da fare a **casa**.

Qual è la differenza? Le due dimostrazioni devono essere uguali?

Intuitivamente in classe il docente può esporre una **prova** che sfrutti l'**interazione** con gli studenti, che potranno fare domande nei passaggi critici, così da rendergli più **semplice** il compito.

La dimostrazione stampata invece è come se dovesse già contenere in anticipo le risposte a tutti i possibili dubbi degli studenti.

Una dimostrazione interattiva per NonIso

Supponiamo che **P**(rover) voglia convincere **V**(erifier) che due grafi G_1 e G_2 non sono isomorfi.
V non può fidarsi della parola di **P**, per cui chiede a **P** di giocare al seguente gioco.

1. **V** sceglie con probabilità uniforme un grafo tra G_1 e G_2 senza comunicarlo a **P**.
2. **V** operando una permutazione casuale dei nomi dei vertici genera un grafo H isomorfo a quello selezionato.
3. **V** spedisce H a **P**.
4. **P** comunica a **V** se H è stato generato da G_1 o G_2.

Se **P** mente o non conosce la risposta ha probabilità $\frac{1}{2}$ di vincere.
Se il gioco viene ripetuto n volte, $\left(\frac{1}{2}\right)^n$. Altrimenti se conosce la risposta ed è onesto **P** vince con probabilità 1 .

Sistemi di prove interattivi

Definizione ([Wig17])

Un sistema di prove interattivo per un insieme S è un gioco con due giocatori, **V** che esegue una strategia polinomiale probabilistica e **P** che esegue una strategia computazionalmente illimitata tale che le seguenti proprietà siano verificate:

- **Completezza**: dopo aver interagito con **P** sull'input comune x, **V** accetta sempre se $x \in S$.
- **Correttezza**: dopo aver interagito con **P** sull'input comune x, **V** rifiuta con probabilità almeno $\frac{1}{2}$ se $x \notin S$.

Chiamiamo IP (*interactive polynomial time*) la classe contenente tutti gli insiemi S per cui è possibile costruire un sistema di dimostrazioni interattivo.

La classe di complessità IP

Chiaramente NP $\subseteq$ IP, dal momento che per ogni $S \in$ NP, **P** può fornire a **V** il **certificato** che prova l'appartenenza di x a S, che deve esistere per definizione di NP.
È possibile però dare una caratterizzazione più precisa della classe IP attraverso il seguente

Teorema ([Sha92])

IP = PSPACE.

L'**interazione** dunque aggiunge effettivamente **potere computazionale**.

Conoscenza zero

Nei **protocolli crittografici** interattivi non vorremmo rivelare i nostri **segreti**, perché non possiamo fidarci della controparte.
Vorremmo cioè una dimostrazione che non comunicasse altro che la **validità** dell'enunciato stesso.

Definizione

Una dimostrazione interattiva è (computazionalmente) a conoscenza zero se l'insieme $\mathcal{V}$ delle trascrizioni delle esecuzioni del protocollo è (computazionalmente) indistinguibile dall'insieme $\mathcal{S}$ delle possibili simulazioni di tale protocollo.
Chiamiamo CZK la classe contenente tutti gli insiemi S per cui è possibile costruire un sistema di dimostrazioni interattivo computazionalmente a conoscenza zero.

Un protocollo interattivo per 3-Coloring

Consideriamo il problema 3-Coloring, ovvero se sia possibile assegnare a un grafo $G = (V, E)$ una colorazione dei vertici $\psi : V \to \{1, 2, 3\}$ tale che se $\{v_i, v_j\} \in E$ allora $\psi(v_i) \neq \psi(v_j)$.

Vogliamo che **P**, in possesso di ψ, dimostri a **V** che un grafo G è colorabile con tre colori a conoscenza zero cioè senza rivelare ψ.

1. **P** seleziona uno schema di commitment $C(x, r)$ e una permutazione casuale π di $\{1, 2, 3\}$.
2. **P** manda a **V** i commitment $c_i = C(\pi(\psi(v_i)), r_i)$ per ogni $v_i \in V$.
3. **V** seleziona un lato $\{v_i, v_j\} \in E$ e lo manda a **P**.
4. **P** rivela a **V** $\pi(\psi(v_i))$ e $\pi(\psi(v_j))$.
5. **V** verifica che $\pi(\psi(v_i)) \neq \pi(\psi(v_j))$.

3-Coloring ∈ CZK

- **Completezza.** Risulta chiaro che se **P** fornisce una colorazione valida ed esegue correttamente il protocollo di commitment allora **V** accetta con probabilità uno.
- **Correttezza.** Se **P** sta mentendo (cioè se G non è colorabile con tre colori), allora c'è almeno un lato $\{v_i, v_j\}$ di G tale che $\psi(v_i) = \psi(v_j)$. **V** allora ha probabilità almeno $\frac{1}{|E|}$ di selezionare un lato colorato in maniera scorretta e quindi probabilità al più $1 - \frac{1}{|E|}$ di accettare. Se il protocollo viene ripetuto n volte questa probabilità può essere fatta diventare piccola a piacere, per cui la probabilità di rifiuto diventa $1 - \left(1 - \frac{1}{|E|}\right)^n$ che tende a uno per $n \to \infty$.
- **Conoscenza zero.** Se lo schema di commitment è computazionalmente occultante allora l'ovvia simulazione e il protocollo effettivo saranno computazionalmente indistinguibili.

La classe di complessità CZK

Considerando che 3-COLORING è NP-completo è immediato affermare il seguente

Teorema ([WMG86])

Sotto l'ipotesi di esistenza di funzioni one-way $\mathsf{NP} \subseteq \mathsf{CZK}$.

La caratterizzazione completa di CZK è invece fornita dal seguente

Teorema ([BOGG+88])

Sotto l'ipotesi di esistenza di funzioni one-way $\mathsf{CZK} = \mathsf{IP}$.

È possibile dunque assumendo **ipotesi standard** in crittografia fornire dimostrazioni a conoscenza zero per ogni problema appartenente alla classe PSPACE.

Protocolli sigma e dimostrazioni di conoscenza

I protocolli sigma

I **protocolli sigma** sono protocolli applicativi a **conoscenza zero** composti dalle seguenti fasi:

1. **P**: Commitment
2. **V**: Challenge
3. **P**: Response
4. **V**: (Verification)

In questo modo è semplice implementare protocolli di **identificazione**, di **firma** e **dimostrazioni a conoscenza zero non interattive**, a costo di considerare **V** onesto.

Il protocollo di identificazione di Schnorr

Supponiamo che **P** voglia **dimostrare** a **V** di **conoscere** il logaritmo discreto x di y in base g in $\mathbb{Z}_q$, dove q è un primo grande. **P** usa allora il seguente **protocollo**, che gli permette di non rivelare x:

1. commitment: **P** manda a **V** $r = g^k$, con k estratto uniformemente da $\mathbb{Z}_q$.
2. challenge: **V** manda a **P** e estratto uniformemente da $\mathbb{Z}_q$.
3. response: **P** manda a **V** $s = k + x \cdot e \pmod{q}$.
4. verification: **V** verifica che $r = g^s \cdot y^{-e}$.

Validità del protocollo di Schnorr

- **Completezza.** Se effettivamente $x = \log_g y$ allora $g^s \cdot y^{-e} = g^k \cdot (g^x)^e \cdot y^{-e} = r \cdot y^e \cdot y^{-e} = r$. **V** dunque accetta certamente se **P** esegue correttamente il protocollo.
- **Correttezza.** Se **P** non conosce x ha probabilità $\frac{1}{q}$ di riuscire a convincere **V**, cioè la probabilità di estrarre uniformemente x.
- **Conoscenza zero.** **V** può simulare esecuzioni corrette del protocollo estrendo uniformemente e ed s, e calcolando $r = g^s \cdot y^{-e}$.
- **Correttezza speciale**. Date due trascrizioni del protocollo (r, e, s) e (r, e', s') possiamo calcolare $x = \frac{s'-s}{e'-e}$.

Prove di conoscenza

La **correttezza speciale** implica la "**conoscenza**" effettiva di x da parte di **P**, perché fornisce un **algoritmo** che permette di estrarlo. La dimostrazione interattiva diventa quindi una prova di conoscenza.

Definizione

Dato un linguaggio $\mathcal{L}$ in NP, e un'istanza $x \in \mathcal{L}$, una dimostrazione è detta di conoscenza a conoscenza zero se è a conoscenza zero ed esiste un algortmo E che permette usando **P** di estrarre il certificato c di appartenenza di x a $\mathcal{L}$.

Bibliografia I


Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway.
Everything Provable is Provable in Zero-Knowledge.
In *Advances in Cryptology — CRYPTO' 88*, Lecture Notes in Computer Science, pages 37–56. Springer, New York, NY, August 1988.


S Goldwasser, S Micali, and C Rackoff.
The Knowledge Complexity of Interactive Proof-systems.
In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 291–304, New York, NY, USA, 1985. ACM.

Bibliografia II

Oded Goldreich.
Zero-Knowledge twenty years after its invention.
Technical report, Electronic Colloquium on Computational Complexity (http://www.eccc.uni-trier.de/eccc/), Report No, 2002.

Adi Shamir.
IP = PSPACE.
J. ACM, 39(4):869–877, October 1992.

Michael Sipser.
Introduction to the Theory of Computation.
Cengage Learning, Boston, MA, third edition, June 2012.

Nigel P. Smart.
Cryptography Made Simple.
Springer, 2016.

Bibliografia III


Avi Wigderson.
Mathematics and Computation.
Draft, 2017.


Avi Wigderson, Silvio Micali, and Oded Goldreich.
Proofs that yield nothing but their validity and a methodology of cryptographic protocol design.
In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 174–187, Los Alamitos, CA, USA, 1986. IEEE Computer Society.

Appendice: gli schemi di commitment (informalmente)

Gli **schemi di commitment** permettono ad un agente **M** di **affidare** ad un agente **R** un messaggio facendo in modo che egli non possa leggerlo, per poi svelarne il contenuto in un momento successivo.

Se x è il messaggio e r un valore random un modo semplice di costruire uno schema di commitment è considerare $C(x, r) = H(x||r)$ dove H è una primitiva di **hash crittografico**.

In questo modo infatti **M** non può mentire in virtù della resistenza alla seconda preimmagine (**binding**) e R non può scoprire quale sia il messaggio grazie alla resistenza alla preimmagine (**hiding**).

Appendice: gli schemi di commitment (formalmente)

Definizione

Chiamiano schema di commitment un algoritmo pubblico $C(x, r)$ dove x è il messaggio e r un valore random. Se **M** è il mittente e **R** il ricevente il commitment consiste in **M** che calcola e spedisce $c = C(x, r)$ a **R** mentre il decommitment in **M** che svela i valori di x' ed r' e **R** che controlla che $C(x', r') = c$.

Definizione (Binding)

Uno schema di commitment $C(x, r)$ è detto (computazionalmente) vincolante se nessun avversario (computazionalmente limitato) noti x ed r può generare $x' \neq x$ ed r' tali che $C(x', r') = C(x, r)$.

Definizione (Concealing o Hiding)

Uno schema di commitment $C(x, r)$ è detto (computazionalmente) occultante se nessun avversario (computazionalmente limitato) può indovinare b dato $c = C(x_b, r)$ con b estratto uniformemente da $\{0, 1\}$ ed essendo a lui noti x_0 e x_1.