

Zero-knowledge pro NP

Pokud existují OWP (jednosměrné permutace), pak každý jazyk v NP má ZK (zero-knowledge) interaktivní důkaz.

Konstrukce pro jeden NP-úplný jazyk je postačující

Idea Protocol

pro L_{3COL} (obarvení grafu třemi barvami)

Společný vstup: graf G

Proverův vstup: validní obarvení G třemi barvami

1) Prover zvolí permutaci $\pi \leftarrow S_3$ barev, obarví G pomocí toho obarvení (po permutaci), zakryje graf pomocí kalíšků

2) Verifier vybere hranu grafu $e \in E$ a pošle ji proverovi

3) Ten odkryje kalíšky na hraně

4) Verifier akceptuje, pokud je e obarvena různými barvami

1)–4) opakujeme $|V||E|$ krát

Důkaz správnosti:

Soundness Pokud graf není obarvitelný třemi barvami, verifier zvolí špatně obarvenou hranu s pravděpodobností aspoň $\frac{1}{|E|}$

Pravděpodobnost podvodu při $|V||E|$ opakováních je $(1 - \frac{1}{|E|}) \approx e^{-|V|} = e^{-n}$.

Completeness Verifier akceptuje s pravděpodobností 1 pro $x \in L_{3COL}$.

Co kdyby prover měl nějaké kalíšky, se kterými si může dělat, co chce?

Verifier vidí dvě náhodné různé barvy, je to zero-knowledge. Nepozná, jestli mu prover lže.

Kalíšky = commitment schemes

Prover musí být vždy schopen to otevřít jen na tu hodnotu, která v karbičce/kalíšku byla.

Protokol mezi Sender a Receiver – má dvě fáze

1) commit – sender pošle receiverovi commitment c pro hodnotu v

2) reveal – S pošle R decommitment r pro hodnotu v

Obrázek:

S	R
$\text{Comm}_r(v), c \rightarrow \text{commit}$	
$v, r \rightarrow \text{reveal } c = \text{Comm}_r(v)$	

Definition: (commitment scheme)

PPT algoritmus Comm nazýváme commitment scheme, pokud pro polynom l platí:

1) **Binding** $\forall n \in \mathbb{N}, v_0, v_1 \in \{0, 1\}^n, r_0, r_1 \in \{0, 1\}^{l(n)}$ platí:

$$\text{Comm}(v_0, r_0) \neq \text{Comm}(v_1, r_1)$$

tzv. Perfectly-binding

2) **Hiding** $\forall PPT$ distinguishera D existuje negligible ε tž. $\forall n \in \mathbb{N}, r_0, r_1 \in \{0, 1\}^n$

$$|\Pr_{r \in \{0, 1\}^{l(n)}} [D(\text{Comm}(r_0, r)) = 1] - \Pr[D(\text{Comm}(r_1, r)) = 1]| \leq \varepsilon$$

přes náhodné mince D .

Tvrzení pokud existují OWP, pak existují schémata pro commitment.

Důkaz: konstrukce pro commitment pro jeden bit

(pomocí hybridního argumentu lze rozšířit pro libovolně mnoho bitů)

Nechť f je OWPs hardcore bitem H . Pak můžeme definovat commitment (b, r) jako $f(r) || b \oplus h(r)$

Binding: Z konstrukce na základě toho, že f je permutace

Hiding: stejný důkaz jako v konstrukci PRG z OWP

Commitment je důležitý stavební prvek kryptografických protokolů

Protokol pro L_{3COL}

Společný vstup: $G = (V, E) \mid |V| = n$

Proverův vstup: witness $y(c_1, \dots, c_n), c_i \in \{1, 2, 3\}, c'_i = \pi(c_i)$

1) Prover zvolí $\pi \leftarrow S_3$. Pro $i \in \{1, \dots, n\}$ pošle verifierovi $\text{Comm}_{r_i}(c'_i)$

2) Verifier zvolí $(i, j) \in E$ a pošle (i, j) P

3) P pošle V (c'_i, r_i) a (c'_j, r_j)

4) V akceptuje, pokud $c'_i \neq c'_j$

1)–4) opakujeme $n|E|$ krát

Tvrzení: protokol je zero-knowledge interaktivní důkaz pro $|_{COL}$, pokud comm je commitment scheme.

Důkaz: completeness + soundness stejné (díky binding)

Obrázek:

P	V
cm_1, \dots, cm_n	$\rightarrow \text{commit}$
	$\leftarrow (i, j)$
$(c'_i, r_i), (c'_j, r_j)$	$\rightarrow \text{reveal } c = \text{Comm}_r(v)$

Simulátor:

$S(G, Z)$:

1) vyber náhodnou hranu (i_s, j_s) grafu a dvě náhodné barvy za podmínky, že jsou různé
definujme $c'_k = 1$ pro $k \notin \{i, j\}$

2) zkonstruuj commitment cm_i pro všechny c'_i

emuluj $V^*(x, z, cm_1, \dots, cm_n)$

Nechť odpověď V^* je (i, j)

3) Pokud $(i, j) = (i_s, j_s)$, pak otevři cm_{i_s} a cm_{j_s} a vrať odpovídající $\text{view}_{V^*} = (X, Z, r^*, cm_1, \dots, cm_n, (c'_{i_s}, r_{i_s}), (c'_{j_s}, r_{j_s}))$
V opačném případě se vrať na 1)

4) Pokud neuspějeme ani po $n|E|$ iteracích, vrať **fail**

Simulátor pouze pro jednu iteraci protokolu

(pro všechny iterace ze sekvenční kompozice Z_k)

Zbývá ukázat, že $\forall \text{PPT } D$ existuje $\text{negl. } \varepsilon$ tž. $\forall x \in L_{3COL}, y \in R_{3COL}(x), z \in \{0, 1\}^*$:

$$|\Pr[D(\text{view } V^*(P(x, y) \leftrightarrow V^*(x, z))) = 1] - \Pr[D(S(x, z)) = 1]| \leq \varepsilon(n)$$

Pro spor předpokládejme, že existuje PPT D , který rozliší $\{\text{view } (P(x, y) \leftrightarrow V^*(x, z))\}$ a $S(x, z)$
s pravděpodobností $\geq \frac{1}{p(n)}$ pro polynom p a nekonečně mnoho $x \in L_{3COL}, y \in R_{3COL}(x)$ a $z \in \{0, 1\}^*$

Hybridní simulátory

$S'(x, z, y)$: postupuje jako S kromě volby c'_{i_s} a c'_{j_s}

spočítá cm_{i_s} a cm_{j_s} jako $P[c'_{i_s} = \pi(c_{i_s})]$ a $P[c'_{j_s} = \pi(c_{j_s})]$ pro $\pi \leftarrow S_3$

distribuce $\{S(x, z)\}$ a $\{S'(x, z, y)\}$ jsou identické

$S''(x, z, y)$: postupuje jako S , commitment pro obarvení kompletně jako prover

pokud se V^* zeptá na (i, j) různé od (i_s, j_s) , pak iteruje znovu a případně vrátí **fail**

pokud S'' nevrátí **fail**, pak jsou distribuce $\{view V^*(P(x, y) \leftrightarrow V^*(x, z))\}$ a $\{S''(x, zy)\}$ identické jsou-li (i, j) a (i_S, j_S) nezávislé, pak S'' vrátí **fail** s pravděpodobností $\leq (1 - \frac{1}{|E|})^{n|E|} \approx e^{-n} \leq \frac{1}{2p(n)}$

D rozliší tyto distribuce s pravděpodobností $\leq \frac{1}{2p(n)}$

pro distribuce S' a S'' platí, že D je rozlišuje s ppstí alespoň $\frac{1}{2p(n)}$

$S \equiv S'$ (ppst pro rozlišení $\geq \frac{1}{2p(n)}$) S'' (pravděpodobnost pro rozlišení $\leq \frac{1}{2p(n)}$) (P, V^*) – schematicky to, co je napsáno výše

Pro alespoň jeden z hybridů budeme mít přechod, který nám umožní prolomit hiding toho commitmentu

Shrnutí: mezi S' a S'' lze zkonstruovat polynomiálně ($q(n)$) mnoho hybridů, které se liší v jednom commitmentu, z toho musí existovat dvojice hybridů, které lze rozlišit s ppstí alespoň $\frac{1}{q(n)p(n)}$, tedy spor s hiding property commitmentu \square

Aplikace

ZeroCash – anonymní varianta BTC na základě ZK
 $coin = (r, sn, cm)$, kde $cm = Comm_r(sn)$ (serial number)

na ledger umístíme transakci TX_{Mint}
 $cm; 1BTC \rightarrow POOL-BEC$

pro utracení zveřejníme sn a dokážeme pomocí ZK
znám r tž. $cm = Comm_r(sn)$ je v $CMList$

tx_{spent}
 $sn, \pi, 1ZEC = 1BTC$

máme tzv. ZK-SNARK = ZK-sufficient non-interactive argument of knowledge
 π – důkaz má konstantní délku a je neinteraktivní
– veřejně ověřit!
pro aplikaci výše můžeme dokazovat, že je Listem pro CMTree s kořenem Root
2014 – ZK-verifikace transakce u BTC

Předpoklady

dělíme na :

- Falsifiable** – lze falsifikovat (neexistuje algoritmus na...)
- Non-falsifiable** – „pro každý vstup existuje algoritmus...“
- False** – můžeme pro tento předpoklad dokázat, co chceme :)