Katedra Informatiky
Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Bezznalostné dôkazy

(Bakalárska práca)

Lukáš Hlavička

Vedúci: doc. RNDr. Martin Stanek, PhD.

Bratislava, 2008

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím uvedených zdrojov.

.....

Ďakujem môjmu bakalárskemu vedúcemu doc. RNDr. Martinovi Stanekovi PhD. za odborné vedenie, cenné rady a pripomienky. Ďakujem mojej rodine za obrovskú podporu počas môjho štúdia.

Abstrakt

Názov práce: Bezznalostné dôkazy
Autor: Lukáš Hlavička
Vedúci práce: doc. RNDr. Martin Stanek PhD.
Práca sa zaoberá bezznalostnými dôkazmi. Snaží sa podať ucelený prehľad modelov bezznalostných dôkazov, pričom sa snaží prezentovať najdôležitejšie výsledky v tejto oblasti za takmer štvrťstoročie od ich zavedenia. V prvej časti prezentujeme neformálne ideu bezznalostných dôkazov, ktorú formalnejšie rozoberáme v druhej časti. V poslednej časti sa venujeme variantom bezznalostných dôkazov s orákulom, ako aj resetovateľnosti jednotlivých účastníkov komunikácie.
Kľúčové slová: zero-knowledge, interaktívny dôkaz, Arthur-Merlin games

Obsah

Časť I

1 Úvod

Táto práca sa bude zaoberať bezznalostnými dokazovacími systémami. Jej cieľom je zosumarizovať poznatky a priniesť jednoliaty celistvý pohľad na problematiku zero-knowledge dôkazov a dokazovacích systémov[1]. Prínosom tejto bakalárskej práce by malo byť uľahčenie štúdia danej problematiky. Logicky je rozčlenená na tri časti:

- **1. časť.** V prvej časti neformálne vysvetlíme princíp, využitie a základné vlastnosti zero-knowledge dôkazov. Tieto vlastnosti budú v závere demonštrované na niekoľkých jednoduchých príkladoch prevzatých z prác iných autorov (Ali-Babova jaskyňa, Rubikova kocka, Fiat-Shamirov protokol,...), ako aj nami vytvorenými príkladmi (Skúška z kryptológie, Dôkaz identity založený na RSA,...). Cieľom tejto časti je vybudovať predstavu o zero-knowledge bez formalnej výstavby teórie z dôvodu ľahšieho pochopenia ďalších častí tejto práce.

- **2. časť.** V druhej časti sa zameriame na formálnu výstavbu zero-knowledge od definície interaktívnych dokazovacích systémov, cez formy definícií zero-knowledge (pomocný vstup, interaktívne Turingove stroje, ...) až po vzťahy jednotlivých tried zero-knowledge (PZK,SZK,CZK) a kompozíciu zero-knowledge. V tejto časti uvedieme príklad výpočtového zero-knowledge protokolu pre G3C a súčasne s využitím vlastnosti NP-úplnosti jazyka 3CSAT vytvoríme analogický protokol pre tento jazyk.

- **3. časť.** V tretej časti sa budeme bližšie venovať neinteraktívnym zero-knowledge dokazovacím systémom, modelom zero-knowledge protokolov s orákulom a resetovateľným zero-knowledge protokolom.

1.1 Motivácia

Dôkazy sú z pohľadu matematiky statické objekty, ktoré určujú pravdivosť a neodškriepiteľnosť tvrdenia, ktoré dokazujú. V bežnom živote sú však dôkazy dynamické (napr. súdne pojednávanie, politické alebo filozofické debaty), pri ktorých sa oponent snaží napadnúť dôkaz[2], ktorý sa prijíma ako pravdivý[2] vtedy, ak oponent zlyhá.

Rovnako aj zero-knowledge dôkazy patria medzi dynamické dôkazy. V tomto dôkaze vystupujú dve entity:

- **Dokazovateľ** (D, Danka, Prover) chce dokázať overovateľovi tvrdenie bez toho, aby mu prezradil akúkoľvek informáciu okrem pravdivosti tvrdenia.

- **Overovateľ** (O, Oto, Verifier) sa pýta dokazovateľa istý vopred (prípade na začiatku protokolu) určený počet otázok, ktorých cieľom je zistiť, či dokazované tvrdenie je pravdivé.

[1] Namiesto slovenského prekladu budeme používať anglické slovo z dôvodu kompatibility s literatúrou.

[2] V tomto slova zmysle by bolo vhodnejšie použiť pojem *argument* namiesto pojmu dôkaz resp. dostatočne presvedčivý namiesto *pravdivý*.

V dôkaze tohto typu sa dokazovateľ pokúša presvedčiť overovateľa o pravdivosti tvrdenia, ale tak, aby mu neprezradil žiadnu dodatočnú informáciu okrem pravdivosti tvrdenia. Tento dôkaz prebieha prostredníctvom kôl, pričom v každom kole je pravdepodobnosť, že v tomto kole dokazovateľ presvedčí overovateľa o nepravdivom tvrdení menšia ako 1. Musí však platiť, že pravdepodobnosť akceptácie pravdivého tvrdenia je väčšia ako pravdepodobnosť akceptácie nepravdivého tvrdenia. Overovateľ si určí maximálnu pravdepodobnosť svojho omylu, pričom táto pravdepodobnosť môže byť ľubovolne malé kladné číslo. Na základe tejto pravdepodobnosti si vypočíta[3] potrebný počet kôl dôkazu na to, aby pravdepodobnosť, že ho dokazovateľ presvedčí o nepravdivom tvrdení bola menšia, ako ním zvolená maximálna pravdepodobnosť omylu. Ak dokazovateľ úspešne absolvuje všetky kolá dôkazu, overovateľ akceptuje dokazované tvrdenie.Ak však dokazovateľ neúspešne absolvuje aspoň jedno kolo dôkazu, overovateľ už zamietne tvrdenie ako nepravdivé.

Poznámka 1.1.1 *V prípade interaktívneho (resp. zero-knowledge) dôkazu je možné pojem „dôkaz“ zameniť za pojem „protokol“. Oba tieto pojmy sú v danom kontexte ekvivalentné. V ďalšom texte budeme tieto dva pojmy zamieňať, v závislosti od kontextu. Ak budeme hovoriť o teoretických aspektoch, budeme používať pojem „dôkaz“, zatiaľ čo pri demonštrovaní týchto dôkazov v praktických situáciách budeme používať pojem „protokol“.*

Zero-knowledge dôkazy boli v dobe svojho objavenia unikátnou a v istom slova zmysle aj kontroverznou myšlienkou. Je to dôkaz, ktorý dokazuje tvrdenie, ale pritom overovateľovi neprezradí žiadnu informáciu okrem platnosti tvrdenia, ktoré dokazuje. To znamená, že všetko čo je efektívne vypočítateľné z dôkazu, je taktiež efektívne vypočítateľné z pravdivého tvrdenia. Teda možno povedať, že tento protokol je simulovateľný (existuje simulátor). Tieto dôkazy nemajú len teoretickú hodnotu v teórií výpočtovej zložitosti, ale majú aj využitie v praxi, najmä v kryptografii, ako aj v testovaní rôznych vlastností kryptografických protokolov (napr. zachovanie bezpečnosti pri viacnásobnom spustení protokolu). V kryptografii sú zero-knowledge protokoly využívané na výnutenie správania sa účastníkov podľa predpísaného protokolu.

Pre ilustráciu využitia v praxi načrtneme nasledujúcu situáciu: Užívateľ sa chce prihlásiť na vzdialený server poskytujúci nejakú službu (napr. ftp server). Triviálne by to mohol spraviť týmto protokolom:

- Užívateľ (dokazovateľ) pošle svoje unikátne heslo serveru.
- Server (overovateľ) porovná prijaté heslo s tým, ktoré má uložené vo svojej databáze a podľa výsledku porovnania akceptuje alebo odmietne užívateľa ako oprávneného užívateľa.

Keď sa užívateľ pokúsi pripojiť k tomuto serveru, útočník túto komunikáciu odchytí, predstiera že on je týmto serverom a pošle požiadavku na autentifikáciu užívateľa. Ten odošle svoje heslo, ktoré útočník zachytí. Po zachytení prihlasovacích údajov vyhlási chybu komunikácie a ukončí protokol. Takto má k dispozícii prihlasovacie údaje, vďaka ktorým sa môže úspešne prihlásiť na spomínaný server a využívať jeho možnosti pre vlastnú potrebu v rámci prístupových práv poškodeného užívateľa. Podobná situácia nastáva aj v prípade

[3] poprípade dostane ako parameter

používania rovnakých hesiel na rôzne servery (čo je v praxi relatívne bežné). V tomto prípade má útočník prostredníctvom kompromitovaného serveru k dispozícii databázu používateľov aj s ich prístupovými heslami, čo môže použiť pri snahe o kompromitáciu ďalších prostriedkov.

Jedným z možných riešení tohto problému je použiť na prihlasovanie zero-knowledge protokol. Server (ale aj útočník) sa pri tomto riešení nedozvedia konkrétne prihlasovacie údaje, ale iba to či daný užívateľ pozná, alebo nepozná heslo. To znamená, že bezpečnosť (z pohľadu toho, že sa neprihlási niekto, kto nepozná heslo) je zachovaná s tým, že server (resp. útočník) nezískajú žiadnu informáciu okrem toho, či užívateľ pozná, alebo nepozná heslo.

Vo všeobecnosti sa pomocou zero-knowledge dôkazov dokazuje platnosť tvrdenia zodpovedajúceho príslušnosti slova na vstupe do nejakého vopred definovaného jazyka (jazyk trojofarbiteľných grafov, jazyk splniteľných formúl,...), alebo tzv. promise problému[4]. Dokazovateľ potom dokazuje príslušnosť slova na vstupe do dokazovaného jazyka. Ten možno chápať ako vopred definovanú množinu vstupov, ktoré overovateľ akceptuje. Keďže doteraz sme nikde nekládli časové obmedzenie pre dokazovateľa, tento mohol byť výpočtovo neobmedzený. V praktických aplikáciach zero-knowledge dôkazov najmä pre potreby kryptografie je z hladiska efektivity a praktickej využiteľnosti nutné, aby dokazovateľ v týchto aplikáciách pracoval v pravdepodobnostnom polynomiálnom čase aj to s nie veľmi vysokým stupňom polynómu a konštantami. V praxi sa preto používa modifikovaná verzia pôvodného protokolu, pri ktorej dokazovateľ na pomocnom vstupe dostane pomocnú informáciu. Vďaka nej mu stačí na dokázanie, že tvrdenie na vstupe patrí do dokazovaného jazyka pravdepodobnostný polynomiálny čas. O pomocnom vstupe potom môžeme uvažovať ako o súkromnom kľúči, a o štandardnom vstupe ako o verejnom kľúči.

Zero-knowledge protokoly sú z pohľadu kryptografie rodina protokolov spĺňajúca nasledujúce vlastnosti :

- *Overovateľ nemôže z protokolu získať žiadnu informáciu (okrem pravdivosti tvrdenia), ktorú by si nemohol vypočítať sám bez účasti dokazovateľa na protokole.Overovateľ nemôže získať z komunikácie žiadnu informáciu(s výnimkou pravdivosti tvrdenia), aj za predpokladu, že sa nedrží protokolu.*

  Táto vlastnosť je v definícii zero-knowledge protokolov reprezentovaná existenciou simulátora.

- *Ak vstup nepatrí do dokazovaného jazyka potom je pravdepodobnosť, že ho overovateľ akceptuje zanedbateľná.*

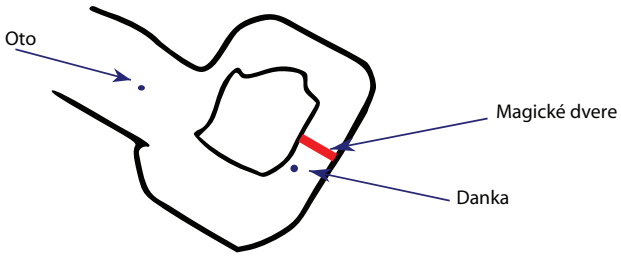
  Túto vlastnosť je možné dosiahnuť pomocou techník znižovania pravdepodobnosti opakovaním. Pre ľubovolnú pred začiatkom komunikácie danú hodnotu $p\,(p > 0)$ sa dá zaručiť, že pravdepodobnosť akceptovania dôkazu overovateľom, ak vstup nepatrí do jazyka je menšia ako p.

- *Overovateľ nemôže predstierať dokazovateľa pre iného overovateľa.*

  Ak sa overovateľ pokúsi dokázať tvrdenie na vstupe pre inú inštanciu protokolu[5] tak uspeje s rovnakou pravdepodobnosťou ako ktorýkoľvek iný neoprávnený užívateľ. Mnohé zero-knowledge protokoly sú založené na

[4] definíciu uvedieme neskôr v 3. časti tejto práce

[5] napríklad na inom servri



Obrázok 1: Alibabova jaskyňa.

princípe výberu otázky zo strany overovateľa[6], pri ktorom si overovateľ vyberie jeden z vopred definovaných podproblémov, ktorého riešenie dokazovateľ odhalí. Tieto medziproblémy sú volené tak, aby na to aby sa overovateľ „naučil“ dôkaz potreboval vedieť riešenie všetkých (alebo takmer všetkých).

- *Overovateľ môže akceptovať každé slovo patriace do dokazovaného jazyka.*

  Ak označíme L dokazovaný jazyk, potom pre $\forall x \in L$ existuje dôkaz, ktorý overovateľa prinúti akceptovať.

1.2 Príklady zero-knowledge

1.2.1 Ali-babova jaskyňa

Jedným z príkladov na zero-knowledge protokol je Ali-Babova jaskyňa [25].

Predstavme si dve chodby, ktoré sú na jednom konci spojené magickými dverami, ktoré sa otvoria každému, kto pozná zaklínadlo. Danka chce dokázať Otovi, že pozná zaklínadlo, ale nechce mu ho prezradiť. Preto spolu vymyslia nasledujúcu hru:

- Oto sa otočí a Danka vojde do jednej chodby.
- Oto sa pozrie a povie, z ktorej chodby má Danka prísť.

Ak Danka pozná zaklínadlo, vždy bude môcť prísť správnou chodbou, pretože podľa potreby si bude schopná otvoriť magické dvere a prejsť do druhej chodby. Ak ho však nepozná, potom na začiatku s pravdepodobnostou $\frac{1}{2}$ uhádne chodbu, ktorú určí Oto v druhom kroku hry. Ak ju Oto požiada, aby vyšla tou chodbou, ktorú si tipla, výjde požadovanou chodbou. Problém pre ňu nastane, ak ju Oto požiada aby vyšla z chodby opačnej. Danka nevie heslo, a preto si nemôže otvoriť magické dvere. Oto v tomto prípade zistí, že Danka nepozná tajné heslo, pretože inak by bola schopná vyhovieť jeho požiadavke.

Uvažujme situáciu, ak by Oto za účelom presvedčiť niekoho ďalšieho nahral na kameru celý priebeh tejto hry.(Nahrávka zodpovedá pohľadu Ota a teda nezachytáva informáciu do ktorej chodby Danka vchádza.) To by mu však nepomohlo. On by bol presvedčený o tom, že Danka pozná zaklínadlo,ale ak by sa pokúsil pomocou tejto nahrávky dokázať niekomu ďalšiemu, že Danka pozná zaklínadlo, neuspel by. Táto nahrávka by totiž mohla byť falošná, pretože Oto

[6] tento princíp vysvetlíme na praktickom príklade v závere kapitoly

s Dankou sa mohli dohodnúť na spolupráci, pri ktorej by boli vopred dohodnutí na poradí chodieb, do ktorých by Danka postupne vchádzala. Takže Oto nemôže presvedčiť nikoho ďalšieho, že Danka pozná heslo k magickým dverám.

Prirodzenou otázkou pri tomto protokole je či nasledujúca modifikácia tohto protokolu taktiež nezodpovedá zero-knowledge protokolu:

- Oto zastane pred vchodom do chodieb a požiada Danku, aby vstúpila do jednej a vyšla z druhej chodby.

Tento protokol je zdanlivo ešte lepší v zmysle, že Oto vie okamžite rozhodnúť, či ho Danka klame. To je síce naozaj prednosťou tohto protokolu, ale má aj nevýhodu v tom, že už nie je zero-knowledge protokolom. Pri zero-knowledge protokoloch vyžadujeme aby boli simulovateľné. Rovnako ako v predchádzajúcom protokole predpokladajme, že Oto celý priebeh tohoto protokolu nahrá na videokazetu. Táto videokazeta by už mohla presvedčiť niekoho ďalšieho za predpokladu, že by nebola zostrihaná (čo ale predpokladajme, že je relatívne jednoducho zistiteľné). Tento protokol teda nespĺňa jednu z vlastností zero-knowledge protokolu (Oto by mohol presvedčiť niekoho ďalšieho), čo však znamená, že nie je zero-knowledge protokolom.

1.2.2 Oriešky

Danka tvrdí, že má šiesty zmysel. Aby to dokázala navrhne túto hru: Na stole sú tri orechové škrupinky. Danka sa otočí a Oto vloží do jednej z nich guličku tak, aby Danka po otočení späť nevidela v ktorej škrupinke je guličká. Danka sa otočí späť a povie v ktorej škrupinke je guličká.

V prípade, že má Danka skutočne šiesty zmysel, potom vždy vie v ktorej škrupinke je guličká (teda vie vždy správne vybrať v ktorej škrupinke je guličká). Ak však nemá šiesty zmysel, potom sa môže iba pokúsiť uhádnuť, kde je ukrytá guličká. Toto sa jej podarí s pravdepodobnosťou $\frac{1}{3}$ v jednom kole hry. Ale Otovi nestačí pravdepodobnosť $\frac{1}{3}$, že sa pomýli a uverí Danke keď by ho klamala. Oto môže túto pravdepodobnosť znižovať prostredníctvom počtu kôl tejto hry.

1.2.3 Rubikova kocka

O niečo zložitejší protokol možno demonštrovať na príklade Rubikovej kocky [25]. Danka chce dokázať Otovi, že vie poskladať Rubikovu kocku, ale nechce mu prezradiť spôsob ako poskladať kocku z aktuálnej rozloženej pozície. Preto navrhne nasledovnú hru:

- Oto dá Danke rozloženú Rubikovu kocku.
- Danka vytvorí z pôvodnej pozície novú[7] a vráti Otovi.
- Oto má na výber z dvoch možností:
  - Chce vidieť riešenie z nového stavu Rubikovej kocky.
  - Chce vidieť, či bol druhý krok urobený korektne. (Chce vedieť, ako Danka došla k novému stavu kocky z pôvodného stavu.)

[7] nie však vyriešenú

Ak Danka vie poskladať pôvodnú Rubikovu kocku, tak vie ľahko odpovedať na obe Otove otázky.Ak sa Oto spýta na riešenie nového stavu, potom ju Danka vie vyriešiť vďaka skladaniu permutácií[8]. Označme ξ permutáciu častí kocky, ktorá by viedla k vyriešeniu pôvodnej pozície a π permutáciu z pôvodnej pozície do novej, potom Danka si vie spočítať $\pi^{-1} \circ \xi$(kde $\circ$ označuje skladanie permutacií), ktorú odhalí Otovi. Ak sa Oto spýta na korektnosť druhého kroku, Danka mu jednoducho odhalí permutáciu π. Tu je dobré si uvedomiť že nejde o postupnosť krokov (k rovnakej pozícii kocky je možné prísť rôznymi cestami vďaka konštrukcii kocky), pretože v opačnom prípade by ju Oto po predvedení týchto krokov dokázal poskladať aj sám.

Ak ju Danka poskladať nevie, tak si musí tipnúť otázku, ktorú jej Oto položí v ďalšom kroku a na základe toho sa rozhodnúť, či spraví korektne druhý krok, alebo bude vychádzať z už poskladanej Rubikovej kocky (nevráti mu tú jeho, ale najakú inú,ktorú má pripravenú v zálohe). V každom kole hry sa jej to podarí uhádnuť s pravdepodobnosťou $\frac{1}{2}$.Teda po n kolách je pravdepodobnosť, že sa Danke podarí oklamať Ota $(\frac{1}{2})^n$. Danka však v každom kole musí zvoliť iný stav[9] vrátenej kocky, pretože ak Oto zistí, že mu vrátila dvakrát rovnakú pozíciu[10], spýta sa na komplementárnu otázku a je si schopný vďaka obom odpovediam vypočítať riešenie sám[11] nasledujúcim spôsobom:

Oto pozná vďaka odpovediam na obe otázky permutácie $\pi^{-1} \circ \xi$ a π. Aby zistil ξ (teda ako poskladať Rubikovu kocku z pôvodnej pozície) mu stačí spočítať $\pi \circ \pi^{-1} \circ \xi$.

1.2.4 Fiatov-Shamirov dôkaz identity (FS protokol)

FS protokol je model autentifikačného protokolu v modeli s verejným klúčom, ktorý demonštruje využitie zero-knowledge protokolu v praxi [29].

V tomto protokole sa Danka snaží dokázať Otovi, že na druhej strane komunikačnej linky je skutočne ona. Fiatov-Shamirov dôkaz identity sa skladá z dvoch častí:

Inicializácia: Danka vygeneruje $n = p.q$ (p, q sú veľke prvočisla), zvolí si $s \in Z_n^*$ ($s \in \{1, 2, ..., n-1\}$) a vypočíta $v = s^2 \bmod n$. Hodnoty p, q, s sú súkromným klúčom a v, n sú verejným kľúčom[12].

Po inicializácii prebehne k kôl protokolu, kde jedno kolo protokolu vyzerá nasledovne:

1. Danka si náhodne zvolí $r \in Z_n^*$, vypočíta $x \equiv r^2 \bmod n$ a pošle x Otovi.
2. Oto zvolí náhodne $b \in \{0, 1\}$ a pošle b Danke.
3. Danka vypočíta $y = r \cdot s^b \bmod n$ a pošle y Otovi.
4. Oto overí, či $y^2 \equiv x \cdot v^b (\bmod n)$. Ak táto kongruencia platí, potom (ak toto nie je posledné) spustí ďalšie kolo protokolu. Ak je toto kolo posledné,

[8] Skladanie Rubikovej kocky sa dá modelovať pomocou grupy permutácií.

[9] Musí mu vrátiť taký stav, z ktorého sa Oto nevie dostať do žiadnej už videnej alebo koncovej konfigurácie.

[10] alebo pozíciu ktorú už videl, resp. do ktorej sa vie dostať z niektorej už videnej

[11] Za predpokladu, že Danka vie kocku poskladať

[12] Predpokladáme, že Oto pozná hodnoty v, n a vie, že prislúchajú Danke.

potom akceptuje Danku ako oprávneného uživateľa. Ak táto kongruencia neplatí, potom ukončí protokol a pokladá Danku za neoprávneného užívateľa.

V praxi existuje veľa modifikácií tohoto protokolu. Asi najznámejšou modifikáciou je protokol bezznalostného dôkazu znalosti (Feigeov-Fiatov-Shamirov protokol) [11].

Zero-knowledge protokoly sa delia podľa toho, či dokazujú znalosť alebo identitu dokazovateľa delia na dôkazy znalosti a dôkazy identity. Z predchádzajúcich príkladov sú dôkazmi znalosti napr. Ali-babova jaskyňa a Rubikova kocka. Príkladom na dôkaz identity je napr. Fiatov-Shamirov dôkaz identity.

1.2.5 Dôkaz identity založený na RSA

Ďalším príkladom na možnosť využita zero-knowledge protokolu v praxi je autentifikačný protokol založený na systéme RSA.

RSA Šifrovací systém RSA je založený na predpoklade, že faktorizácia je „ťažký“ problém, teda že pre veľké n je ťažke nájsť jeho rozklad na prvočísla.

Inicializácia systému:

- Zvolíme dve dostatočne veľké prvočísla p a p ($p \neq q$). Vypočítame $n = p.q$.
- Vyberieme prirodzené číslo e také, že $1 \leq e \leq \varphi(n)$ a nsd$(e, \varphi(n))$=1, kde $\varphi(n) = (p-1).(q-1)$ je Eulerova funkcia a nsd$(e, \varphi(n))$ je najväčší spoločný deliteľ e a $\varphi(n)$.
- Vypočítame d , také že $e.d \equiv 1 \,(\mathrm{mod}\varphi(n))$.

Verejným kľúčom sú hodnoty e, n a súkromným kľúčom je hodnota d. Priestor otvorených aj šifrových textov je množina $Z_n = \{0, 1, ..., n-1\}$.

Šifrovanie/Dešifrovanie:

- Šifrovanie: $E(m) = m^e \bmod n$.
- Dešifrovanie: $D(c) = c^d \bmod n$.

Rovnako ako vo Fiat-Shamirovom dôkaze identity, aj tu sa Danka snaží dokázať Otovi, že na druhej strane komunikačnej linky je skutočne ona (teda, že pozná zodpovedajúci súkromný kľúč). V inicializačnej fáze protokolu si vygeneruje inštanciu RSA s verejným kľúčom e, n a súkromným kľúčom d. Protokol potom prebieha behom k kôl protokolu. Jedno kolo protokolu prebieha nasledovne:

- Danka si náhodne zvolí $r \in Z_n^*$ a pošle Otovi r.
- Oto si zvolí náhodne $l \in Z_n^*$, vypočíta $t = |l - r|$ a $x = l^e \bmod n$ a pošle x Danke.
- Danka spočíta $l \equiv x^d \bmod n$, potom spočíta $t' = |l - r|$ a pošle t'_0 (posledný[13] bit t') Otovi.

[13] najmenej významný

- Oto overí či $t_0 = t'_0$.

Ak Danka pozná zodpovedajúci súkromný kľúč, potom dokáže poslať v poslednom kroku korektné t'_0. Ak však zodpovedajúci súkromný kľúč nepozná, môže sa iba pokúsiť uhádnuť t'_0. Aby sme ukázali, že tento protokol spĺňa vlastnosť zero-knowledge stačí nám už len ukázať, že je simulovateľný. Simulátor sa bude držat protokolu až do okamihu, keď má poslať Otovi svoje t'_0. V tomto okamihu si náhodne zvolí t'_0 a pošle ho Otovi. V prípade, že $t_0 \neq t'_0$ simulátor zmaže správy z posledného kola.

Poznámka 1.2.1 *Vo všeobecnosti má simulátor prístup k Otovej stratégii buď priamo (pozná algoritmus, podľa ktorého sa riadi Oto) alebo využíva Ota (resp. jeho algoritmus) ako orákulum. V oboch týchto prípadoch má možnosť zmazať správy z posledného kola a vrátiť sa do stavu pred uskutočnením interakcie v tomto kole. Viac informácii o oboch týchto typoch simulátorov ako aj ich využite možno nájsť v práci [3].*

1.2.6 Skúška z kryptológie

V škole sa na skúške z kryptológie píše test s $n = 20$ otázkami. Ku každej otázke je $o = 5$ odpovedí, z ktorých je vždy práve jedna správna. Testom študent prejde, ak zodpovie správne na všetky otázky. Vyučujúci navrhne nasledujúci protokol:

- Študent si rovnomerne náhodne zvolí permutáciu poradia otázok. Tieto otázky v tomto poradí pošle zašifrované pomocou commitmentov vo forme $<$*commitment číslo otázky,commitment odpoveď*$>$.

  Poznámka 1.2.2 *Commitment správy je analógia posielania zamknutých skriniek. V takejto konštrukcii platia dve podmienky:*

  1. *Príjemca nie je schopný zistiť obsah skrinky bez získania kľúča od odosielateľa.*
  2. *Odosielateľ nie je po odoslaní skrinky schopný zmeniť jej obsah.*

- Vyučujúci si rovnomerne náhodne zvolí $x \in \{1, 2, ..., n\}$ a pošle x študentovi.

- Študent odokryje x-tú poslanú otázku aj s odpoveďou a všetky čísla otázok.

- Vyučujúci overí, či boli odoslané všetky čísla otázok, a či odpoveďou na x-tú odoslanú otázku v prvom kroku bola odokrytá odpoveď.

Vyučujúci akceptuje, že študent urobil skúšku, ak študent pošle počas behu protokolu správne odpovede na všetky výzvy a pravdepodobnosť, že neodpovedal na všetky otázky správne je menšia alebo rovná ako $\frac{1}{p}$ $(p \geq 1)$. Na základe toho si vypočíta zodpovedajúci počet kôl protokolu $k = \min\{l | l \in \mathbf{N} : (\frac{1}{n} \cdot \frac{o-1}{o})^l \leq \frac{1}{p}\}$.

Ak študent vie správne odpovede na všetky otázky testu, vie odpovedať správne na každú otázku. Ak študent nevie správne odpovede aspoň na l otázok, potom ho vyučujúci v jednom kole odhalí s pravdepodobnosťou $\frac{l}{n} . \frac{o-1}{o}$ (pretože stále je tu istá pravdepodobnosť, že uhádne správne odpovede).

Simulátor pre tento protokol by fungoval nasledovne:

- Ku každej otázke si simulátor tipne odpoveď a odošle commitmenty v požadovanom tvare.
- Po prijatí x od vyučujúceho odkryje zodpovedajúce commitmenty a odošle vyučujúcemu.
- V prípade, že ním hádaná odpoveď nie je správna zmaže všetky správy z posledného kola.

1.2.7 Mah-jang

Danka chce dokázať Otovi, že vie vyriešiť (každú pozíciu) Mah-jang, ale nechce mu ukázať postup, ako poskladať momentálne rozloženie kameňov.

Mah-jang Mah-jang je verzia hry Mah-jong (upravená na demonštráciu zero-knowledge protokolu) v ktorej sú nasledujúce pravidlá:

- Kamene sú rozostavené pyramídovito so základňou tvaru štvorca na hracej doske.
- Je možné odoberať iba dvojice kameňov, ktoré majú rovnaké piktogramy a nie sú v rámci pyramídy prekryté iným kameňom a súčasne je vedľa nich aspoň z jednej strany prázdna pozícia.
- Mah-Jang je vyriešený, ak sú vyššie uvedeným spôsobom odobraté všetky kamene.

Preto sa dohodnú na nasledujej hre:

- Oto dá Danke rozložený Mah-Jang (nejaké konkrétne rozloženie kameňov).
- Danka ho upraví, ale môže jednotlivé kamene nielen odoberať, ale aj pridávať a takýto upravený Mah-Jang potom vráti Otovi.
- Oto má na výber z dvoch možností:
  - Chce vidieť riešenie upraveného stavu Mah-Jangu.
  - Chce vidieť, či bol druhý krok urobený korektne. (Chce vedieť, ako Danka došla k upravenému stavu Mah-Jangu z pôvodného stavu.)

Tento protokol je podobný protokolu s Rubikovou kockou. Oba tieto protokoly sú založené na rovnakom princípe. Napriek tomu, že tento protokol je podobný protokolu pre Rubikovu kocku, sám nie je zero-knowledge protokolom. Pretože, zatiaľ čo v prípade rubikovej kocky sú v každom kroku k dispozícii všetky transformácie (všetky možné rotácie), v prípade tohto protokolu nie sú k dispozicii všetky možné transformácie v každom kroku. Teda v istých konfiguráciách rozloženia pyramídy by existovalo príliš málo možností na ďalší ťah na to, aby mohol existovať simulátor so zodpovedajúcimi vlastnosťami.

Časť II

2 Interaktívne dôkazy

Neformálne povedané, interaktívny dokazovací systém je protokol s dvomi účastníkmi, v ktorom sa dokazovateľ snaží presvedčiť overovateľa o pravdivosti tvrdenia x. Pravdivosť v tomto prípade znamená príslušnosť vstupu x do jazyka. Toto tvrdenie je pravdivé, ak $x \in L$ a nepravdivé ak $x \notin L$.

V tejto kapitole uvedieme najprv viacero definícií a výsledkov založených na týchto definíciách, ako aj výsledky popisujúce interaktívne dokazovacie systémy z pohľadu výpočtovej zložitosti.

Interaktívne dôkazy sú dôležité aj z pohľadu zero-knowledge dôkazov, pretože zero-knowledge dôkazy sú istým rozšírením definície interaktívnych dôkazov. Teda každý zero-knowledge dôkaz je interaktívnym dôkazom, čo ale opačne už neplatí.

Definícia 2.0.1 *Interaktívny dokazovací systém (O,D) pre jazyk L je interaktívny protokol medzi O (overovateľ) a D (dokazovateľ) taký, že O je časovo obmedzený pravdepodobnostným polynomialnym časom a platí:*

- *Kompletnosť:* $\forall x \in L$ *overovateľ akceptuje s pravdepodobnosťou* 1 *na vstupe* x.
- *Nepriestrelnosť : Existuje polynóm* $p(.)$ *taký, že pre* $\forall x \notin L$ *a všetky potenciálne stratégie* D^* *overovateľ akceptuje s pravdepodobnosťou* $\leq \frac{1}{p(|x|)}$ *na vstupe* x.

Trieda jazykov majúcich interaktívny dokazovací systém je označená **IP**.

Definícia 2.0.2 *Interaktívny dokazovací systém (O,D) pre jazyk L je interaktívny protokol medzi O (overovateľ) a D (dokazovateľ) taký, že overovateľ je časovo obmedzený pravdepodobnostným polynomialnym časom a platí:*

- *Kompletnosť:* $\forall x \in L$ *overovateľ akceptuje s pravdepodobnosťou* $\geq \frac{2}{3}$ *na vstupe* x.
- *Nepriestrelnosť: Pre* $\forall x \notin L$ *a všetky potenciálne stratégie* D^* *overovateľ akceptuje s pravdepodobnosťou* $\leq \frac{1}{3}$ *na vstupe* x.

Trieda jazykov majúcich interaktívny dokazovací systém je označená **IP**.

V definícii 2.0.2 je možné namiesto konštánt $\frac{1}{3}$ resp. $\frac{2}{3}$ použiť $\frac{1}{2} - \varepsilon$ resp. $\frac{1}{2} + \varepsilon$ pre $\varepsilon \in (0, \frac{1}{2})$.

Veta 2.0.1 *Definície 2.0.1 a 2.0.2 sú ekvivalentné.*

Dôkaz tohto tvrdenia možno nájsť v práci [13]. Tvrdenie poukazuje na fakt, že požiadavka na úplnú kompletnosť (ak $x \in L$, overovateľ akceptuje s pravdepodobnosťou 1) nezmení triedu jazykov majúcich interkatívny protokol. Ak však požadujeme kompletnú nepriestreľnosť (ak $x \notin L$, overovateľ akceptuje s pravdepodobnosťou 0), potom je trieda **IP** redukovaná na triedu **NP**.

Z definícii vyplýva, že pravdepodobnosť akceptácie $x \notin L$ možno stlačiť ľubovolne blízko k nule opakovaním behu protokolu.

V prípade definície 2.0.1 je to nasledovne: V každom kole protokolu overovateľ akceptuje s pravdepodobnosťou menšou ako $\frac{1}{p(|x|)}$ pre $x \notin L$. Označme túto pravdepodobnosť q (táto pravdepodobnosť je určite menšia ako 1). Po n kolách protokolu je pravdepodobnosť, že overovateľ akceptuje $x \notin L$ menšia ako q^n. A keďže $q < 1$, potom pre $n \to \infty$ ide tento výraz k nule. Pravdepodobnosť, že overovateľ zamietne $x \notin L$ v k-tom kole protokolu je $p_k = q^{(k-1)} \cdot (1 - q)$. Potom pravdepodobnosť toho, že overovateľ zamietne $x \notin L$ najneskôr v k-tom kole protokolu je $s_k = \sum_{i=1}^{k} (p_k)$. Platí, že pre $k \to \infty$ konverguje $s_k \to 1$.

V prípade definície 2.0.2 sa to dá ukázať podobne, lenže v tomto prípade sa na dôkaz využíva Chernoffova nerovnosť.

Interaktívne dokazovacie systémy boli zavedené v práci [21]. Nezávisle boli predstavené hry Arthura a Merlina (Arthur-Merlin games) v práci [2], ktoré sú špecialnym prípadom interaktívnych dokazovacích systémov, v ktorom musí overovateľ poslať všetky svoje náhodné bity hneď ako ich vygeneruje (preto sú často označované aj ako *public-coin* dokazovacie systémy). V ďalších prácach bolo ukázané, že tento špeciálny prípad je rovnako silný ako interaktívny dokazovací systém vo všeobecnom prípade [22]. To pre interaktívne dokazovacie systémy znamená, že kladenie náhodných otázok je rovnako silné ako kladenie otázok s úmyslom (zo strany overovateľa) nachytať dokazovateľa. Tento poznatok je významný pri konštrukcii a overovaní vlastností interaktívnych dokazovacích systémov, avšak neplatí aj pre zero-knowledge dôkazy.

Neizomorfizmus grafov. Klasickým príkladom na interaktívne dokazovacie systémy (ktorý má však zásadný význam z pohľadu výpočtovej zložitosti) je neizomorfizmus dvoch grafov. Inak povedané problém komplementárny k izomorfizmu dvoch grafov.

Protokol:

Vstup: Dva grafy G_0 a G_1

- Overovateľ náhodne zvolí permutáciu vrcholov π na $V(G_1)$ (kde $V(G_1)$ označuje množinu vrcholov grafu G_1), náhodný bit b, vypočíta $H = \pi(G_b)$ a pošle H dokazovateľovi.
- Dokazovateľ určí bit a, pre ktorý sú G_a a H izomorfné a pošle hodnotu a overovateľovi.
- Overovateľ skontroluje či $a = b$.

Ak sú vstupné grafy neizomorfné, dokazovateľ určí hodnotu a a overovateľ akceptuje s pravdepodobnosťou 1. Ak sú izomorfné dokazovateľ si môže hodnotu a iba tipnúť a teda overovateľ akceptuje s pravdepodobnosťou $(\frac{1}{2})^n$ po n behoch protokolu. Overovateľ akceptuje, ak v každom kole protokolu v treťom kroku platí $a = b$, inak neakceptuje.

Argumenty. Argumenty sú variant interaktívnych dokazovacích systémov, pri ktorom sa neberú v podmienke nepriestrelnosti do úvahy všetky cesty ako

oklamať overovateľa, ale iba všetky výpočtovo efektívne stratégie. Definícia argumentu je rovnaká ako definícia A interaktívneho dokazovacieho systému až na podmienku nepriestrelnosti.

Definícia 2.0.3 *Argumentom (O,D) pre jazyk L je interaktívny protokol medzi O (overovateľ) a D (dokazovateľ) taký, že O je časovo obmedzený pravdepodobnostným polynomialnym časom a platí:*

- *Kompletnosť:* $\forall x \in L$ *overovateľ v interakcii s D akceptuje s pravdepodobnosťou* 1 *na vstupe* x.
- *Nepriestrelnosť: Existuje polynóm* $p(.)$ *a pre* $\forall x \notin L$ *a všetky potenciálne stratégie D* implementovateľné v pravdepodobnostnom polynomiálnom čase overovateľ akceptuje s pravdepodobnosťou* $\leq \frac{1}{p(|x|)}$ *na vstupe* x.

Interaktívne argumenty (alebo iba argumenty) sú často označované aj ako výpočtovo nepriestrelné dôkazy[14]. V interaktívnych dôkazoch môže dokazovateľ oklamať overovateľa[15] iba so zanedbateľnou pravdepodobnosťou bez ohľadu na jeho výpočtovú silu. Pri argumentoch môže dokazovateľ oklamať overovateľa iba so zanedbateľnou pravdepodobnosťou len ak predpokladáme, že podvádzajúci dokazovatelia sú výpočtovo obmedzený pravdepodobnostným polynomiálnym časom.

V praxi sa využívajú argumenty v prípadoch, keď sú argumenty efektívnejšie implementovateľné a je dôležitejšia výpočtová efektivita ako (silnejšia) bezpečnosť.

2.1 Vlastnosti IP

Definícia 2.1.1 *Nech L je jazyk. Potom hovoríme, že* $L \in$ **BPP** *ak existuje pravdepodobnostný polynamiálny algoritmus A, pre ktorý na všetkých vstupoch x platí:*

1. *Ak* $x \in L$, *potom* $\Pr[A \text{ akceptuje } x] \geq \frac{2}{3}$.
2. *Ak* $x \notin L$, *potom* $\Pr[A \text{ akceptuje } x] \leq \frac{1}{3}$.

Veta 2.1.1 **BPP** $\subseteq$ **IP**.

Dôkaz. Vetu dokážeme konštrukciou protokolu pre každý jazyk $L \in$ **BPP** na ktorom ukážeme, že spĺňa podmienky definície **IP** s obojstrannou chybou[16].

Nech $L \in$ **BPP**, O(Overovateľ) je algoritmus A z definície **BPP** a D je algoritmus, ktorý na každom vstupe vráti 0.

Protokol :

1. D pošle 0.
2. O vezme vstup x a overí , či $x \in L$ a na základe výsledku sa rozhodne, či akceptuje.

[14] computational—sound proof systems

[15] presvedčiť overovateľa, aby akceptoval $x \notin L$

[16] Ako sme uviedli vyššie tieto dve definície popisujú rovnako bohatú rodinu jazykov.

Teraz už len potrebujeme ukázať, že tento protokol spĺňa vlastnosť kompletnosti a nepriestrelnosti.

- *Kompletnosť* aj *nepriestrelnosť* sú splnené triviálne, ak si uvedomíme ako je definovaný overovateľ a fakt, že dokazovateľ pošle na začiatku kola protokolu iba konštantný bit 0.[17]

□

Poznámka 2.1.1 *V dôkaze nebolo vôbec nutné uvažovať dokazovateľa, pretože neovplyvňuje overovateľa. Táto konštrukcia protokolu je však názornejšia a aj intuitívne spĺňa predstavu o interaktívnom dôkaze.*

Definícia 2.1.2 *Nech R je relácia na $\{0,1\}^* \times \{0,1\}^*$, potom hovoríme, že R je* **NP-relácia** *ak platí:*

- *Existuje rozhodovací deterministický algoritmus A pracujúci v polynomiálnom čase, ktorý rozpoznáva R. (Teda existuje deterministický polynomiálny algoritmus, ktorý pre danú dvojicu (x,y), $x,y \in \{0,1\}^*$ rozhodne, či $(x,y) \in R$.)*
- $\exists$ *polynóm* $p(\cdot)$ $\forall (x,y) \in R : |y| \leq p(|x|)$.

Definícia 2.1.3 *Nech L je jazyk. Potom hovoríme, že $L \in$* **NP** *ak existuje* **NP-relácia** *R taká, že pre každé $x \in \{0,1\}^*$ platí:*

$$x \in L \Leftrightarrow \exists y \in \{0,1\}^* : (x,y) \in R \tag{1}$$

y nazývame svedok x pre R.

Táto definícia triedy **NP** je odlišná od bežne používanej definície pre túto triedu cez nedeterministické Turingove stroje, avšak dá sa ľahko ukázať, že tieto dve definície popisujú tú istú triedu jazykov **NP**.

Veta 2.1.2 **NP** $\subseteq$ **IP**.

Dôkaz. Nech L $\in$ **NP** a nech R_L je zodpovedajúca **NP-relácia**.
Protokol:

- Dokazovateľ nájde y: $(x,y) \in R_L$ a pošle y Overovateľovi.
- Overovateľ overí, či $(x,y) \in R_L$.

V tomto protokole sú dokazovateľ aj overovateľ deterministickí a overovateľ sa nikdy nemýli. Overovateľ pracuje v deterministickom polynomiálnom čase, lebo podľa definície stačí deterministický polynomiálny čas na overenie, či $(x,y) \in R_L$.

Stači už len ukázať *kompletnosť* a *nepriestrelnosť*:

- *Kompletnosť* : je dokázaná existenciou svedka y, pretože z definície platí: $\forall x \in L : \exists y : (x,y) \in R_L$. Svedka y je dokazovateľ schopný nájsť vďaka svojej neobmedzenej výpočtovej sile (napr. úplným preberaním).

[17] Je dobré si uvedomiť, že dokazovateľ žiadnym spôsobom neovplyvňuje overovateľa.

- *Nepriestrelnosť* je dokázaná vlastnosťou $\forall x \notin L \rightarrow \forall y : (x, y) \notin R_L$ (Overovateľ nikdy neakceptuje, pretože neexistuje y také, že $(x, y) \in R_L$).

□

Poznámka 2.1.2 *Jedine pre jazyky z* **NP** *existujú interaktívne dôkazy s deterministickým dokazovateľom aj overovateľom [33].*

Definícia 2.1.4 *Nech L je jazyk. Potom hovoríme, že* $L \in$ **co-NP** *ak* $L^c \in$ **NP**.

Veta 2.1.3 co-NP$\subseteq$ **IP**.

Dôkaz je uvedený v práci [16].

Veta 2.1.4 IP=**PSPACE**.

Dôkaz. S dôkazom prišiel ako prvý Shamir [30]. Jadro dôkazu je tvrdenie uvedené vyššie [16].□

2.2 Alternatívne modely interaktívnych dôkazov

2.2.1 IDS a Turingove stroje

Formálne možno interaktívny dokazovací systém definovať pomocou formálnej konštrukcie špeciálneho typu turingových strojov – tzv. Interkatívnych turingových strojov (ITM).

Definícia 2.2.1 *Interaktívny turingov stroj (ITM) je deterministický 8 páskový Turingov stroj, ktorého pásky pozostávajú z:*

1. *vstupnej pásky iba na čítanie. Obsah vstupnej pásky nazývame* **vstup**.
2. *náhodnej pásky iba na čítanie (náhodný generátor). Obsah náhodnej pásky nazývame* **náhodný vstup**.
3. *pracovnej pásky na čítanie aj zápis.*
4. *výstupnej pásky iba na zapisovanie. Obsah výstupnej pásky nazývame* **výstup**.
5. *komunikačnej pásky iba na čítanie. Obsah komunikačnej pásky iba na čítanie nazývame* **správa prijatá v danom kole**.
6. *komunikačnej pásky iba na zápis. Obsah komunikačnej pásky iba na zapisovanie nazývame* **správa odoslaná v danom kole**.
7. *stavovej pásky obsahujúcej práve jedno pole so zapísanou hodnotou 0.*
8. *pomocnej vstupnej pásky iba na čítanie. Obsah pomocnej vstupnej pásky iba na čítanie nazývame* **pomocný vstup**.

Každý ITM ma pridelený polarizačný bit $b \in \{0, 1\}$ *(identita ITM). Hovoríme že ITM je aktívny, ak je jeho identita rovnaká ako hodnota zapísaná na stavovej páske Turingovho stroja, inak hovoríme, že je neaktívny. Ak je neaktívny nemení sa jeho stav, obsah zapisovateľných pások, ani pozícia hláv na jednotlivých páskach. Kolom protokolu nazývame činnosť stroja ITM medzi dvomi po sebe idúcimi zmenami na stavovej páske.*

Definícia 2.2.2 *Hovoríme, že dva Interaktívne turingove stroje sú spojené ak platí:*

- *Majú opačné identity.*
- *Zdieľajú rovnaku vstupnú a stavovú pásku.*
- *Komunikačná páska na zapisovanie prvého je komunikačnou páskou na čítanie druhého a opačne.*
- *Ostatné pásky sú rozdielne.*

Poznámka 2.2.1 *$\langle A(y), B(z)\rangle(x)$ označujeme náhodnú premennú reprezentujúcu komunikáciu*[18] *A po interakcii s B na vstupe x za predpokladu, že náhodný vstup je zvolený rovnomerne náhodne, a A (resp. B) má pomocný vstup y (resp. z).*

Definícia 2.2.3 *Usporiadaná dvojica (O, D) je interaktívny dokazovací systém pre jazyk L, ak platí:*

- *(O, D) sú spojené Interaktívne turingove stroje.*
- *O pracuje v pravdepodobnostnom polynomálnom čase.*
- *Kompletnosť: $\forall x \in L$ existuje reťazec y taký, že pre všetky $z \in \{0,1\}^*$ platí:*
$$\Pr(\langle O(z), D(y)\rangle(x) = 1) \geq \frac{2}{3} \tag{2}$$
- *Nepriestrelnosť: $\forall x \notin L$, každý Interaktívny turingov stroj D^* a každé $y, z \in \{0,1\}^*$ platí:*
$$\Pr(\langle O(z), D^*(y)\rangle(x) = 1) \leq \frac{1}{3} \tag{3}$$

Tento model je formálnejším modelom zápisu definície interaktívnych dôkazov ako sme uviedli v úvode druhej kapitoly.

Označenie 2.2.1 *Triedu jazykov majúcich interaktívny dokazovací systém označujeme* $\mathbf{IP_{ITM}}$.

Model interaktívnych dôkazov v modeli s ITM a podľa definícií z úvodu druhej kapitoli sú ekvivalentné. Tento fakt by bolo možné ukázať cez rovnosť tried jazykov **IP** a $\mathbf{IP_{ITM}}$. Táto rovnosť by sa ukázala prostredníctvom rovnosti s triedou jazykov **PSPACE**. Pretože platia tvrdenia $\mathbf{IP} = \mathbf{PSPACE}$ ([29]) a $\mathbf{IP_{ITM}} = \mathbf{PSPACE}$ ([16]), potom platí aj tvrdenie $\mathbf{IP} = \mathbf{IP_{ITM}}$.

[18] Transcript komunikácie obsahujúci postupnosť obsahov komunikačných pások po jednostlivých kolách protokolu.

2.2.2 Merlin-Arthur Games

Ak $L \in \mathbf{NP}$ potom podľa definície existuje pre L zodpovedajúca **NP-relácia** R_L, ktorá je rozpoznávaná nejakým deterministickým Turingovím strojom A pracujúcom v polynomiálnom čase. Teda pre $\forall x \in L$ existuje y také, že $(x, y) \in R_L$, kde y je dôkaz, že $x \in L$. A je deterministickým overovateľom pre jazyk L. Ak umožníme A, aby pracoval v znáhodnenom polynomiálnom čase, získame rozšírenie triedy **NP** na triedu **MA** (Merlin-Arthur). Výpočtovo neobmedzený Merlin (dokazovateľ) sa snaží presvedčiť Arthura (overovateľ, ktorý je obmedzený pravdepodobnostným polynomiálnym časom) o tom, že slovo na vstupe patrí do dokazovaného jazyka. Ak je vstup skutočne z jazyka donúti Merlin Arthura akceptovať vo väčšine prípadov. Ak však vstup nepatrí do jazyka, potom Arthur vo väčšine prípadov nebude akceptovať. Merlin sa v tomto modeli dozvie Arthurove náhodné bity okamžite po ich vygenerovaní. Merlin však nevie, aký bude nasledujúci reťazec náhodných bitov vygenerovaných Arthurom.

Definícia 2.2.4 *Nech L je jazyk. Potom hovoríme, že $L \in \mathbf{MA}$, ak existuje relácia $S(x, y, z)$ rozpoznateľna v polynomiálnom čase a konštanta $c \in \mathbf{R}$ taká, že pre každé x s dĺžkou n platí:*

$$x \in L \rightarrow \exists \mathbf{y} \Pr_{z \in \{0,1\}^*}[S(x, y, z) = 1] \geq \frac{3}{4}$$

$$x \notin L \rightarrow \forall \mathbf{y} \Pr_{z \in \{0,1\}^*}[S(x, y, z) = 1] \leq \frac{1}{4}$$

kde $|y| = |z| \leq n^c$

To znamená, že ak $x \in L$, potom existuje Merlinov dôkaz (y), ktorý presvedčí Arthura (s náhodným reťazcom z) s pravdepodnosťou aspoň $\frac{3}{4}$. A ak $x \notin L$, potom každý Merlinov dôkaz (y) presvedčí Arthura najviac s pravdepodobnosťou $\frac{1}{4}$.

Redukcia chýb pre MA. Častokrát nám nestačia pravdepodobnosti chýb určené v definícií a preto sa snažíme redukovať pravdepodobnosť chybnej akceptácie alebo odmietnutia vstupu. Nech $L \in \mathbf{MA}$, potom je možné znížiť chybu protokolu rozpoznávajúceho L tak, že pravdepodobnosť chybnej akceptácie vstupu (slovo nepatrí do jazyka, ale Arthur akceptuje) bude klesať exponenciálne rýchlo k nule.

2.2.3 Arthur-Merlin Games

AM je varianta MA, pri ktorej ako prvý v protokole začína Arthur.

Definícia 2.2.5 *Nech L je jazyk. Potom hovoríme, že $L \in \mathbf{AM}$, ak existuje relácia $S(x, y, z)$ rozpoznateľna v polynomiálnom čase a konštanta $c \in \mathbf{R}$ taká, že pre každé x s dĺžkou n platí:*

$$x \in L \rightarrow \Pr_{z \in \{0,1\}^*}[\exists \mathbf{y} : S(x, y, z) = 1] \geq \frac{3}{4}$$

$$x \notin L \rightarrow \Pr_{z \in \{0,1\}^*}[\forall \mathbf{y} : S(x, y, z) = 1] \leq \frac{1}{4}$$

kde $|y| = |z| = n^c$

Vzťah medzi triedami MA a AM popisuje nasledujúca veta [26]:

Veta 2.2.1 $\mathbf{MA} \subseteq \mathbf{AM}$.

2.3 Hierarchia tried zložitostí IP a AM

Označme $\mathbf{IP}[k]$ resp. $\mathbf{AM}[k]$ triedu jazykov, pre ktorá existuje **IP** resp. **AM** protokol s najviac k kolami.

Z definície **IP** vyzerá, že $\mathbf{IP}[2]$ je bohatšia trieda jazykov ako $\mathbf{AM}[2]$, pretože je relatívne jednoduché simulovať $\mathbf{AM}[2]$ pomocou $\mathbf{IP}[2]$, ale je vo všeobecnosti netriviálne simulovať $\mathbf{IP}[2]$ pomocou $\mathbf{AM}[2]$. Riešenie vzťahu medzi jednotlivými triedami rieši nasledujúce tvrdenie [27]:

Veta 2.3.1 *Goldwasser-Sipser:*
$\forall k \in \mathbf{N} : \mathbf{AM}[k] = \mathbf{IP}[k]$.

Ďalšie dve tvrdenia týkajúce sa vzťahu medzi jednotlivými triedami zložitosti dokázali Shamir a Babai:

Veta 2.3.2 *Shamir:*
Nech poly označuje polynomiálny počet kôl protokolu v závislosti na dlžke vstupu. Potom platí:

$$\mathbf{IP}[poly] = \mathbf{AM}[poly] = \mathbf{PSPACE}$$

a pre každú konštantu $k \in \mathbf{N}$; $k > 0$ *platí:*

$$\mathbf{AM}[k] = \mathbf{AM}[2] = \mathbf{AM}$$

Veľmi významným dôsledkom tohto tvrdenia je fakt, že ak povolíme polynomiálne veľa kôl protokolu, tak Artur-Merlin games sú rovnako silné ako interaktívne protokoly.

Veta 2.3.3 *Babai:*
Pre každú konštantu $k \in \mathbf{N}$; $k > 0$ *platí:*

$$\mathbf{AM}[k] = \mathbf{IP}[k] = \mathbf{IP}[2] = \mathbf{AM}$$

Poznámka 2.3.1 *Pre* **AM** *a* **IP** *je z pohľadu počtu kôl v literatúre zavedené rozličné označenie. Zatiaľ čo pre* **IP** *platí* $\mathbf{IP} = \mathbf{IP}[\mathbf{poly}]$, *pre* **AM** *platí* $\mathbf{AM} = \mathbf{AM}[\mathbf{2}]$.

3 Bezznalostné dokazovacie systémy

Zero-knowledge protokoly sú protokoly, ktoré dokazujú tvrdenie (alebo ekvivalentne príslušnosť slova do jazyka akceptovaného overovateľom) bez toho, aby počas komunikácie bola rozhraním overovateľ:dokazovateľ prenesená akákoľvek informácia, okrem pravdivosti tvrdenia. Hoci v bežnej praxi sa snažíme naučiť čo najviac, v prípade zero-knowledge je to práve naopak. Využitie tohto zdanlivého paradoxu je v mnohých oblastiach teoretickej aj praktickej informatiky. Teória zero-knowledge dôkazov otvára nové možnosti pre kryptografiu. V kryptografii je bežna požiadavka na limitovanie množstva informácií, ktoré

sú poskytnuté členovi zabezpečenej komunikácie. V mnohých kryptografických schémach (schéma na zdieľanie tajomstva, autorizačné a autentifikačné protokoly,...) je nutné taktiež prinútiť členov komunikácie, aby sa držali vopred dohodnutého protokolu, ale dovolili im zachovať si súkromné bity (resp. ďalšie súkromné informácie[19]) v tajnosti. Práve tu príde vhod nástroj, ktorý umožní overiť platnosť tvrdenia, ale nie je z neho možné vyextrahovať ani za pomoci zneužívania protokolu žiadnu dodatočnú informáciu.

Pri definícií zero-knowledge (ZK) považujeme overovateľa za potenciálneho útočníka, ktorý sa snaží získať informácie z protokolu od dokazovateľa (ktorý sa drží predpísaného protokolu). Našou snahou je, aby žiadna výpočtovo efektívna[20] stratégia (protokolu sa nepridŕžajúceho) overovateľa nebola schopná získať od dokazovateľa dodatočné informácie. Intuitivne overovateľ v takomto protokole nemá byť po skončení interakcie schopný urobiť hocičo, čo nebol schopný urobiť pred začatím interakcie. Túto intuíciu formalizujeme požiadavkou na existenciu simulátora, ktorý bez prístupu k dokazovateľovi dokáže simulovať výstup (nečestného) overovateľa po skončení interakcie s dokazovateľom. Existencia takéhoto simulátora však znamená, že ak útočník dokáže po interakcii s dokazovateľom vykonať nejakú akciu, tak je túto akciu schopný vykonať aj bez interakcie s dokazovateľom spustením simulátoru.

Oproti klasickým interaktívnym dôkazom majú všetky známe zero-knowledge dôkazy[21] dve hlavné odlišnosti:

- *Skryté náhodné bity:* Overovateľ (a v závislosti od protokolu aj dokazovateľ) má vlastné skryté náhodné bity.
- *Výpočtová zložitosť:* Dokazovateľ využíva pri dôkaze výpočtovú zložitosť nejakého iného problému (napr. existencia jednosmernej funkcie, predpoklad, že faktorizácia je ťažký problém).

Prirodzenou otázkou je, či sú tieto vlastnosti nevyhnutné. Vzniklo viacero modelov snažiacich sa túto otázku zodpovedať. Jedným z nich (neinteraktívne zero-knowledge dôkazy) sa budeme zaoberať v tretej časti tejto práce.

3.1 Definície

Rozšírením definície pre interaktívne dokazovacie systémy možno získať zero-knowledge protokol nasledovne:

Nerozoznateľnosť. V ďalšom texte je jedným z kľúčových pojmov pojem nerozoznateľnosti distribúcií dvoch náhodných vektorov.

Prvým typom nerozoznateľnosti distribúcii dvoch postupností náhodných premenných je perfektná nerozoznateľnosť.

Definícia 3.1.1 *Hovoríme, že distribúcie postupností náhodných premenných sú perfektne nerozoznateľné, ak sú identické.*

[19] napr. súkromné kľúče

[20] V prípade, že by sme umožnili aj výpočtovo neefektívne, tak by si mohol mohol overovateľ takmer všetko vypočítať sám. Výnimku by tvorili reťazce, ktoré by očakával v interakcii a nedokázal by výpočtovo zistiť, ktorý reťazec je ten správny aj za predpokladu vyskúšania všetkých reťazcov.

[21] Máme na mysli netriviálne (poznámka 3.2.1) zero-knowledge dôkazy ako boli definované v práci [21].

O niečo slabší je typ nerozoznateľnosti distribúcií dvoch postupností náhodných premenných pri ktorom nepožadujeme identické rozdelenie týchto dvoch distribúcií, ale iba nerozoznateľnosť distribúcii postupností náhodných premenných vzhľadom na isté štatistické testy.

Definícia 3.1.2 *Nech X, Y sú náhodné premenné.*

$$\Delta(X,Y) = \frac{1}{2} \sum_{\alpha \in \{0,1\}^*} |\Pr[X = \alpha] - \Pr[Y = \alpha]| \tag{4}$$

$\Delta(X,Y)$ nazývame štatistická odchýlka.

Definícia 3.1.3 *Hovoríme, že funkcia $f : \mathbf{N} \to \langle 0,1 \rangle$ je zanedbateľná, ak klesá rýchlejšie ako prevrátená hodnota ľubovolného polynómu. Teda pre každý polynóm $p(\cdot)$ platí:*

$$\exists m \in \mathbf{N}\ \forall n \geq m : f(n) \leq \frac{1}{p(n)} \tag{5}$$

Definícia 3.1.4 *Hovoríme, že postupnosti náhodných premenných $\{X_k\}_{k \in \mathbf{N}}$ a $\{Y_k\}_{k \in \mathbf{N}}$, kde k je premenná, sú štatisticky blízke ak $\Delta(X_k, Y_k)$ je zanedbateľná funkcia.*

Definícia 3.1.5 *Hovoríme, že distribúcie postupností náhodných premenných sú takmer perfektne (štatisticky) nerozoznateľné, ak sú štatisticky blízke.*

Ak upustíme aj od podmienok štatistickej nerozoznateľnosti a vyžadujeme iba, aby distribúcie postupností náhodných premenných boli nerozoznateľné pomocou výpočtovo efektívnych algoritmov, získame výpočtovú nerozoznateľnosť distribúcií postupností náhodných premenných.

Definícia 3.1.6 *Nech $S \subseteq \{0,1\}^*$. Hovoríme, že postupnosti náhodných premenných $X = \{X_\alpha\}_{\alpha \in S}$ a $Y = \{Y_\alpha\}_{\alpha \in S}$ sú výpočtovo norozoznateľné (nerozlíšiteľné), ak pre každý pravdepodobnostný polynomiálny algoritmus D, každý polynóm $p(\cdot)$, dostatočne dlhé n a $\alpha \in \{0,1\}^{poly(n)} \cap S$ platí:*

$$|\Pr(D(\alpha, X_\alpha) = 1) - \Pr(D(\alpha, Y_\alpha) = 1)| \leq \frac{1}{p(|n|)} \tag{6}$$

Definícia 3.1.7 *Hovoríme, že distribúcie postupností náhodných premenných sú nerozoznateľné, ak sú perfektne, takmer perfektne, alebo výpočtovo nerozoznateľné.*

Definícia 3.1.8 *Nech (O, D) je interaktívny dokazovací systém pre jazyk L. Hovoríme, že (O, D) je zero-knowledge protokol pre L práve vtedy, ak pre každého pravdepodobnostného polynomiálneho O^*, existuje simulátor S^*, taký že platí:*

- *S^* beží na znáhodnenom polynomiálnom Turingovom stroji a simuluje dokazovateľa D v komunikácií s O^* (vzniká simulovaný protokol (O^*, S^*)).*
- *$\forall x \in L$ k-tice[22] $(a_1, a_2, ..., a_k)$ a $(b_1, b_2, ..., b_k)$ reprezentujúce komunikáciu medzi (O^*, D) resp. (O^*, S^*) sú nerozoznateľné vzhľadom na náhodné bity v komunikácií (O^*, D) resp. (O^*, S^*).*

[22] postupnosti náhodných premenných

Poznámka 3.1.1 *V definícií 3.1.8 je dobré si uvedomiť, že (O, S) je iba simulovaný protokol. Teda v skutočnosti nedochádza k interakcii medzi overovateľom O a simulátorom S, ale simulátor S má prístup k overovateľovi O ako k znáhodnenému orákulu, alebo priamo k algoritmu overovateľa vrátane jeho náhodných bitov.*

Definícia v 3.1.8 neberie do úvahy pomocný vstup, ktorý sa môže útočník pokúsiť použiť na to, aby získal dodatočné informácie. Preto definícia 3.1.8 nespĺňa predpoklady na to, aby zodpovedajúce zero-knowledge dôkazy mohli byť použité ako subprotokoly v rámci väčších protokolov a taktiež nie sú uzavreté na kompozíciu (dokonca ani sekvenčnú). Tieto nedostatky je možné odstrániť zahrnutím pomocného vstupu do definície zero-knowledge protokolu.

Definícia 3.1.9 *Nech (O, D) je interaktívny dokazovací systém. Hovoríme,že (O, D) je zero-knowledge dokazovacím systémom v modeli s pomocným vstupom na vstupe z S, kde $S \subseteq \{0,1\}^*$ ak pre každého pravdepodobnostného polynomiálneho O^* a každý polynóm $p(\cdot)$ existuje pravdepodobnostný polynomiálny algoritmus C^* taký, že distribúcie nasledujúcich dvoch postupností náhodných premenných sú nerozoznateľné:*

- $\{(O^*(z), D)(x)\}_{x \in S, z \in \{0,1\}^{p(|x|)}}$ *(Transkript komunikácie O^* s D na vstupe x, pričom O^* má pomocný vstup z.)*
- $\{C^*(x, z)\}_{x \in S, z \in \{0,1\}^{p(|x|)}}$ *(Výstup C^* s pomocným vstupom z na vstupe x.)*

C^* sa nazýva simulátor.

Označenie 3.1.1 *Triedu jazykov majúcich zero-knowledge dokazovací systém označujeme* **ZK**.

Poznámka 3.1.2 *Reťazec x z definícií vyššie je často označovaný ako „dokazované tvrdenie“.*

Podľa typu nerozoznateľnosti v definícii vyššie hovoríme o:

- *perfektnom* zero-knowledge protokole, ak je nerozoznateľnosť perfektná.

  Označenie 3.1.2 *Triedu jazykov majúcich perfect zero-knowledge dokazovací systém označujeme* **PZK**.

- *štatistickom* zero-knowledge protokole, ak je neroznateľnosť takmer perfektná. (Štatistický zero-knowledge protokol býva často označovaný aj ako almost-perfect zero-knowledge protokol.)

  Označenie 3.1.3 *Triedu jazykov majúcich štatistický zero-knowledge dokazovací systém označujeme* **SZK**.

- *výpočtovom* zero-knowledge protokole, ak je nerozoznateľnosť výpočtová.

  Označenie 3.1.4 *Triedu jazykov majúcich výpočtový zero-knowledge dokazovací systém označujeme* **CZK**.

Poznámka 3.1.3 *Najslabším typom nerozoznateľnosti, o ktorom sa uvažuje v prípade zero-knowledge protokolov je výpočtová nerozoznateľnosť. Preto platí:* **CZK**=**ZK**.

PZK a SZK v sebe vyžadujú omnoho silnejšie predpoklady ako výpočtový ZK, a preto je podmienka zero-knowledge protokolu (poďla definície zero-knowledge protokolu) zmysluplná bez ohľadu na výpočtovú silu overovateľa. (Podvádzajúci overovateľ sa môže pokúsiť získať znalosti z protokolu za použitia dodatočného výpočtového výkonu. Štatistické a perfect ZK garantujú,že zvýšenie výpočtového výkonu mu v tomto nepomôže [33].)

Hoci sa definícia so zahrnutím pomocného vstupu výrazne líši od prvej definície bez pomocného vstupu, väčšina zero-knowledge protokolov sú zero-knowledge protokoly v modeli s pomocným vstupom. Jedinou výnimkou sú protokoly zostrojené na demonštráciu rozdielu medzi týmito definíciami a protokoly majúce iba tzv. Black-box simulátor[23] [17].

3.2 Hierarchia tried zero-knowledge

Triedy zložitostí zero-knowledge tvoria hierarchiu vzhľadom na inklúziu. Najmenšou triedou je **PZK**. Vzťahy jednotlivých tried popisujú nasledujúce tvrdenia:

Veta 3.2.1 $\mathbf{BPP} \subseteq \mathbf{PZK}$

Poznámka 3.2.1 *Trieda* **BPP** *je trieda, ktorá má tzv. triviálne zero-knowledge dôkazy, pretože ak jazyk* $L \in \mathbf{BPP}$*, tak overovateľ je vďaka tomu, že má k dispozícii pravdepodobnostný polynomiálny čas schopný overiť príslušnosť vstupu do tohto jazyka sám bez účasti dokazovateľa na protokole.*

Veta 3.2.2 $\mathbf{PZK} \subseteq \mathbf{SZK}$

Dôkaz. Tu si stačí uvedomiť že ak sú distribúcie náhodných vektorov identické, potom sú aj štatisticky blízke.□

Veta 3.2.3 $\mathbf{CZK} \subseteq \mathbf{IP}$

Dôkaz. Trieda majúca **ZK** protokol je definovaná pomocou interaktívnych dokazovacích systémov a teda nemôže presiahnuť triedu **IP**. **CZK** je podľa definície podmnožinou **ZK**. Teda $\mathbf{CZK} \subseteq \mathbf{IP}$.□

Veta 3.2.4 $\mathbf{SZK} \subseteq \mathbf{CZK}$

Podľa [33] je trieda $\mathbf{CZK} = \mathbf{IP}$. Trieda **SZK** definovaná pomocou interkatívnych dôkazov a teda platí $\mathbf{SZK} \subseteq \mathbf{CZK}$.

Veta 3.2.5 $\mathbf{BPP} \subseteq \mathbf{PZK} \subseteq \mathbf{SZK} \subseteq \mathbf{CZK} \subseteq \mathbf{IP}$

Dôkaz. Toto tvrdenie je dokazané vetami vyššie.□

Bolo ukázané [33], že štatistické a perfektné zero-knowledge majú veľa spoločných vlastností, avšak zatiaľ ostáva otvoreným problémom či platí: $\mathbf{PZK} = \mathbf{SZK}$.

Ak súčasne predpokladáme, že existuje jednosmerná funkcia, potom možno dokázať ďalšie tvrdenia ohľadne vzťahov s ďalšími triedami zložitosti.

[23] Simulátory, ktoré používajú overovateľa ako orákulum.

3.3 Výpočtové zero-knowledge

V tejto kapitole najprv ukážeme príklad výpočtového zero-knowledge protokolu a potom ukážeme konštruktívný dôkaz ako za pomoci zero-knowledge protokolu pre 3-ofarbiteľnosť grafu skonštruujeme zero-knowledge protokol pre všetky jazyky z **NP**. V závere kapitoly naznačíme spôsob ako skonštruovať zero-knowledge protokoly pre všetky jazyky z **IP** a rovnako predpokladajúc existenciu bit-commitment schémy ukážeme ďalšie vzťahy medzi triedami zložitostí z predchádzajúcej kapitoly.

Definícia 3.3.1 *Hovoríme, že funkcia* $f : \{0,1\}^* \to \{0,1\}^*$ *je jednosmerná, ak platí:*

1. *(Ľahko spočítateľná) Existuje deterministický polynomiálny algoritmus* A *taký, že* $\forall x \in \{0,1\}^* : A(x) = f(x)$.
2. *(Ťažko invertovateľná) Pre každý pravdepodobnostný polynomiány algoritmus* A'*, každý polynóm* p *a dostatočne veľké* n *platí:*

$$\Pr[A'(f(x)) \in f^{-1}(f(x))] \leq \frac{1}{p(n)} \tag{7}$$

pričom pravdepodobnosť je braná rovnomerne cez $x \in \{0,1\}^n$ *a všetky náhodné bity algoritmu* A'.

3.3.1 Zero-knowledge pre G3C

Definícia 3.3.2 *Hovoríme, že graf G(V,E) je 3-ofarbiteľný ak existuje funkcia* $f : V \to \{1,2,3\}$ *,*$\forall (i,j) \in E : f(i) \neq f(j)$.

Definícia 3.3.3 $\mathbf{G3C} = \{G|$ *graf* $G(V,E)$ *je 3-ofarbiteľný* $\}$

Protokol: Samotný protokol vyžaduje existenciu tzv. perfect-binding commitment schémy.

Perfect-binding commitment schéma. Commitment schéma je dvojfázový protokol s dvomi účastníkmi (A, B), ktorý umožnuje účastníkovi A poslať účastníkovi B digitálnu podobu uzamknutej skrinky obsahujúcej reťazec[24] $x \in \{0,1\}^*$. V prvej fáze protokolu (*fáza commitmentu*) je zafixovaná konkrétna hodnota x a odoslaná účastníkovi B takým spôsobom, že B o hodnote x nemá žiadne (alebo takmer žiadne) informácie (*utajenie*). V druhej fáze protokolu (*fáza odokrytia*) je hodnota x odokrytá účastníkovi B. Účastník B dokáže overiť, či odokrytá hodnota je rovnaká ako hodnota odoslaná v prvej fáze protokolu (*záväznosť*). V prípade, ak má účastník pravdepodobnosť 0, že vo fáze odokrytia pošle iný reťazec ako vo fáze commitmentu a účastník B to nezistí, potom hovoríme o commitment schéme s perfektnou záväznosťou (perfect-binding commitment schéma).

- **Štandardný vstup:**

  Graf $G(V,E)$ s očíslovanými vrcholmi ($V = \{1,2,..,n\}$, kde n je počet vrcholov).

[24] v prípade fyzickej skrinky je napríklad napísaný na konečnom kuse papiera

- **Pomocný vstup pre dokazovateľa:**

  3-zafarbenie vrcholov grafu $\psi : V \rightarrow \{1, 2, 3\}$.

Kolo Protokolu:

- Dokazovateľ zvolí rovnomerne náhodne permutáciu π prvkov $\{1, 2, 3\}$. Pre každý vrchol i $(i \in \{1, 2, ..., n\})$ pošle overovateľovi commitment hodnoty $\pi(\psi(i))$.
- Overovateľ si zvolí rovnomerne náhodne hranu $e \in E$ a pošle ju dokazovateľovi.
- Dokazovateľ odokryje overovateľovi ofarbenie i, j pre hranu $e = (i, j)$ z commitmentu odoslaného v prvom kroku.
- Overoveteľ skontroluje, či sú odokryté hodnoty pre i a j rôzne.

Dokazovateľ je v tomto protokole implementovateľný v pravdepodobnostnom polynomiálnom čase, a vždy presvedčí overovateľa za predpokladu, že má k dispozícií korektné ofarbenie grafu na vstupe. Ak však graf na vstupe nie je 3-ofarbiteľný, potom akokoľvek sa dokazovateľ zachová, overovateľ neakceptuje s pravdepodobnosťou aspoň $\frac{1}{|E|}$ (ak neexistuje korektné 3-ofarbenie grafu, potom aspoň jedna hrana musí mať rovnaké ofarbenie na oboch koncoch).

Teraz je potrebné iba ukázať že existuje simulátor s požadovanými vlastnosťami na preukázanie vlastnosti zero-knowledge.

Konštrukcia simulátora: Simulátor nezávisle rovnomerne náhodne vyberie farbu pre každý vrchol a zodpovedajúce commitmenty odošle overovateľovi. Hoci simulátor odosiela overovateľovi značne odlišný náhodný vektor od toho, ktorý by odoslal dokazovateľ, overovateľ je výpočtovo obmedzený, a preto nie je schopný rozlíšiť distribúcie týchto dvoch náhodných premenných. Ak by overovateľ dokázal rozlíšiť tieto dve distribúcie, potom by to znamenalo že vie rozlíšiť aj bity odoslané vo fáze commitmentu. To by však znamenalo spor s predpokladom existencie bit-commitmentu.

Ak sa overovateľ spýta na korektne ofarbenú hranu, potom simulátor odkryje hodnoty, pošle ich overovateľovi a skončí.Ak sa spýta na nekorektne ofarbenú hranu, simulátor skončí s chybou. Podľa [17] chyba sa vyskytne s pravdepobnosťou približne $\frac{1}{3}$, a ak simulátor neohlási chybu potom je prepis komunikácie medzi ním a overovateľom výpočtovo nerozlíšiteľný od skutočnej komunikácie. Celý dôkaz možno nájsť v práci [16].

3.3.2 Výpočtový zero-knowledge protokol pre 3-CSAT

Definícia 3.3.4 *3-CSAT je množina splniteľných formúl v konjunktívnom normálnom tvare, pričom každá z konjugovaných formúl obsahuje najviac dve disjunkcie.*

Výpočtový zero-knowledge protokol pre 3-CSAT zostrojíme podobne ako pre G3C. V konštrukcii by bolo možné využiť fakt, že oba jazyky sú NP-úplné (dokonca by nám stačilo, že jazyk G3C je NP-úplný). Použili by sme polynomiálnu redukciu 3-CSAT na G3C a spustili protokol pre G3C. Tento spôsob je možné použiť na vytvorenie výpočtových zero-knowledge protokolov pre celú triedu **NP**.

My najprv vytvoríme výpočtový zero-knowledge protokol pre 3-CSAT priamo a potom ukážeme ako využiť existenciu výpočtového zero-knowledge protokolu pre nejaký NP-úplný problém na vytvorenie výpočtového zero-knowledge protokolu pre všetky jazyky z **NP**.

Perfect-binding commitment schéma. V protokole pre 3-CSAT rovnako ako v protokole pre G3C predpokladáme existenciu perfect-binding commitment schémy, pretože v prípade, že by sme tento predpoklad nepotrebovali aspoň pri jednom NP-úplnom jazyku, potom by sme ho nepotrebovali pri žiadnom NP-úplnom jazyku (dokonca by sme tento predpoklad nepotrebovali pri žiadnom jazyku $L \in \mathbf{NP}$ ako uvidíme v ďalšej časti). V takom prípade by sme využili fakt, že všetky NP-úplné jazyky sú navzájom medzi sebou polynomiálne redukovateľné. Nech L_0 je NP-úplný jazyk, pre ktorý existuje výpočtový zero-knowledge protokol a pre ktorý nepotrebujeme predpoklad existencie perfect-binding commitment schémy. Potom pre všetky L NP-úplné vytvoríme výpočtový zero-knowledge nasledovne:

1. L polynomiálne redukujeme (označme túto redukciu f) na L_0 tak, že platí: $\forall x : x \in L \Leftrightarrow f(x) \in L_0$
2. Spustíme výpočtový zero-knowledge protokol pre L_0.

Vďaka tomu, že na transformáciu potrebujeme len polynomiálny čas, a že zachováva „príslušnosť resp. neprísluš nosť do jazyka“ je táto konštrukcia korektná.

Potom by bolo z teoretického hľadiska vhodnejšie využívať takéto konštrukcie, nakoľko by sme nemuseli brať do úvahy predpoklad existencie jednosmernej funkcie. (Otázka existencie jednosmernej funkcie je zatiaľ jedným z otvorených problémov teoretickej informatiky.)

Definícia 3.3.5 *(Pravdivostné) ohodnotenie premenných formuly* $\varphi(x_1, x_2, \ldots, x_n)$ *je zobrazenie* ψ*:* $X \to \{0, 1\}$*, ktoré každej premennej z množiny* $X = \{x_1, x_2, \ldots, x_n\}$ *priradí pravdivostnú hodnotu* 0 *alebo* 1*.*

Definícia 3.3.6 *Hovoríme, že formula* $\varphi(x_1, x_2, \ldots, x_n)$ *je splniteľná, ak existuje také ohodnotenie* ψ *jej premenných, že* $\varphi(\psi(x_1), \psi(x_2), \ldots, \psi(x_n)) = 1$*.*

Protokol:

- **Vstup:**

  Formula $\varphi(x_1, x_2, ..., x_n)$ v konjunktívnom normálnom tvare, pričom každá z konjunkcií má najviac tri literály.

- **Pomocný vstup pre dokazovateľa:**

  Ohodnotenie ψ premenných $x_1, x_2, \ldots, x_n$ také, že $\varphi(\psi(x_1), \psi(x_2), \ldots, \psi(x_n)) = 1$.

Označme k počet konjugovaných podformúl formuly φ.

Kolo protokolu:

- Dokazovateľ zvolí náhodnú permutáciu α konjugovaných podformúl a náhodnú permutáciu β pomenovania premenných formuly, a takúto upravenú formulu ϖ pošle overovateľovi spolu s commitmentmi ohodnotení jednotlivých premenných v tvare $Com(\beta(x_i) = v)$ $v \in \{0, 1\}$ $\forall i \in \{1, 2, \ldots, n\}$.

- Overovateľ si náhodne zvolí $t \in \{0, 1, 2, \ldots, k\}$ a pošle t dokazovateľovi.
- Dokazovateľ podľa prijatého t vykoná:
  - Ak $t = 0$, potom dokazovateľ pošle overovateľovi permutácie α a β.
  - Ak $t \in \{1, 2, \ldots, k\}$, potom dokazovateľ odokryje overovateľovi ohodnotenia premenných t-tej konjugovanej podformuly od začiatku.
- Overovateľ overí:
  - Ak $t = 0$, či $\varpi = \alpha(\varphi(\beta(x_1), \beta(x_2), \ldots, \beta(x_n)))$.
  - Ak $t > 0$, potom overí či t-ta konjugovaná podformula s odkrytým ohodnotením premenných je pravdivá.

Ak dokazovateľ pozná korektné ohodnotenie premenných, potom vždy presvedčí overovateľa. Ak však formula nie je splniteľná, potom v jednom kole neuspeje aspoň s pravdepodobnosťou $\frac{1}{k+1}$, pretože aspoň jedna konjugovaná podformula formuly φ nie je splniteľná a overovateľ sa spýta s rovnakou pravdepodobnosťou na ohodnotenie t-tej podformuly aj na konštrukciu formuly ϖ. Môže sa teda rozhodnúť pre jednu z alternatív:

- Ak predpokladá, že sa ho overovateľ spýta na vytvorenie formuly ϖ z formuly φ, potom korektne vytvorí formulu ϖ.
- Ak predpokladá, že sa ho overovateľ spýta na najakú konjugovanú podformulu, potom si nevytvorí korektne formulu ϖ, ale vytvorí novú formulu tak, že bude mať dĺžku k[25] a bude splniteľná[26].

Aj v tomto prípade je možné znížiť chybu nesprávneho akceptovania overovateľom pomocou viacerých kôl protokolu.

3.3.3 Výpočtové zero-knowledge protokoly pre NP

Za rovnakého predpokladu, za akého existuje výpočtový zero-knowledge protokol pre 3-ofarbiteľnosť grafu, existuje výpočtový zero-knowledge aj pre všetky jazyky L z **NP**. Jadro konštrukcie spočíva v nasledujúcom tvrdení:

Veta 3.3.1 *Jazyk 3-ofarbiteľnosti je NP-úplný.*

Ak je nejaký jazyk L_0 NP-úplný, potom z definície musí pre každé $L \in \mathbf{NP}$ existovať polynomiálny algoritmus C taký, že pre $\forall x$ platí: $x \in L_0 \leftrightarrow C(x) \in L$. Teda všetky jazyky z **NP** sú polynomiálne transformovateľné na L_0. Teda ak máme výpočtový zero-knowledge protokol pre nejaký NP-úplný jazyk, potom vďaka tejto vlastnosti vieme akoby preniesť vlastnosť byť zero-knowledge protokolom prostredníctvom tejto vlastnosti na všetky jazyky z **NP** nasledovne:

Je daný jazyk $L \in \mathbf{NP}$, pre ktorý chceme zostrojiť výpočtový zero-knowledge protokol s tým, že poznáme výpočtový zero-knowledge protokol pre nejaký NP-úplný jazyk L_0. V konštrukcii protokolu pre L postupne vykonáme:

1. Jazyk L polynomiálne redukujeme na L_0.

[25] okrem toho musía mať s pôvodnou formulou rovnaký počet konjugovaných podformúl s dvomi resp. jednou resp. žiadnou disjunkciou

[26] takúto formulu možno zostrojiť v lineárnom čase

2. Spustíme protokol pre L_0.

Teda v našom prípade jazyk L, pre ktorý chceme vytvoriť výpočtový zero-knowledge protokol najprv redukujeme (v polynomiálnom čase) na jazyk 3-ofarbiteľnosti grafu a následne spustíme (hore uvedený) výpočtový zero-knowledge protokol pre 3-farbiteľnosť grafu. Tento spôsob je podobný ako postup uvedený vyššie na dokázanie faktu, že ak nepotrebujeme predpoklad perfect-binding commitment schémy pre jeden NP-úplný jazyk, potom ho nepotrebujeme pre žiadny NP-úplný jazyk. Týmto spôsobom získame výpočtový zero-knowledge protokol pre všetky jazyky z **NP**. Toto je podstatnou časťou dôkazu tvrdenia:

Veta 3.3.2 *Ak existuje jednosmerná funkcia, potom* $\mathbf{NP} \subseteq \mathbf{ZK}$. *Naviac dokazovateľ je implementovateľný v pravdepodobnostnom polynomiálnom čase za predpokladu, že na pomocnom vstupe dostane zodpovedajúceho NP-svedka.*

Toto tvrdenie má význam najmä pre konštrukciu kryptografických protokolov.

Vo väčšine prípadov využívame NP-úplnosť nejakého jazyka na odvodenie zložitosti tohto problému, zatiaľ čo v tomto prípade využívame NP-úplnosť na konštrukciu výpočtových zero-knowledge protokolov pre celú triedu **NP**.

Za rovnakých predpokladov, aké sme požadovali pre konštrukciu výpočtových zero-knowledge protokolov pre triedu **NP** (existencia jednosmernej funkcie) je možné ukázať existenciu výpočtových zero-knowledge protokolov pre celú triedu **IP** [6].

3.4 Štatistické zero-knowledge dôkazy

Štatistické zero-knowledge dôkazy vyžadujú, aby distribúcie postupností náhodných premenných z definície zero-knowledge dôkazu[27] boli štatisticky blízke. Táto podmienka je silnejšia ako podmienka výpočtovej nerozoznateľnosti, pri ktorej nám stačí, aby tieto distribúcie neboli rozoznateľné výpočtovo efektívnymi (pravdepodobnostnými polynomiálnymi) algoritmami.

Štatistické zero-knowledge dôkazy poskytujú teoretickú bezpečnosť pre overovateľa aj dokazovateľa. Preto, ak existujú konštrukcie pre výpočtový aj štatistický zero-knowledge dôkaz pre konkrétny jazyk L, je z hľadiska bezpečnosti vhodnejšie uprednostniť štatistický zero-knowledge dôkaz pred výpočtovým.

Za štandardných predpokladov (existencia jednosmernej funkcie) sa dá ukázať, že pre celú triedu **NP** existujú štatistické zero-knowledge argumenty [23]. Súčasne však za rovnakých predpokladov pravdepodobne neexistujú štatistické zero-knowledge dôkazy pre celú triedu **NP** [17, 12]. Predpokladá sa, že trieda jazykov majúcich štatistický zero-knowledge dôkaz (**SZK**) je netriviálnou nadmnožinou triedy **BPP**. Súčasne platí, že $\mathbf{SZK} \subseteq \mathbf{AM} \cap \mathbf{co\text{-}AM}$ [33].

Trieda **SZK** obsahuje (podobne ako trieda **NP**) tzv. SZK-úplné jazyky. SZK-úplné jazyky sú také jazyky pre ktoré platí, že všetky ostatné jazyky patriace do triedy **SZK** sú na ne polynomiálne redukovateľné. Dva najznámejšie SZK-úplné problémy sú problémy [33, 24]:

- Problém štatistickej vzdialenosti (Statistical difference) dvoch distribúcii náhodných premenných.
- Problém vzdialenosti entropii (Entropy difference) dvoch distribucii náhodných premenných.

[27] definícia 3.1.8 resp. definícia 3.1.9

3.5 Kompozícia zero-knowledge protokolov

Z pohľadu aplikovateľnosti zero-knowledge protokolov ako podprotokolov v rámci bezpečnostných protokolov je dôležitou otázkou otázka zachovania vlastností zero-knowledge protokolu v prípade rôznych typov kompozícií. Úvahy sa týkajú týchto druhov kompozícií:

- sekvenčná kompozícia zero-knowledge protokolov
- paralelná kompozícia zero-knowledge protokolov
- súbežná kompozícia zero-knowledge protokolov

Ak predpokladáme aplikáciu zero-knowledge protokolu v praxi je prirodzenou požiadavkou žiadať jeho znovupoužiteľnosť. Protokol, ktorý by bolo možné spustiť iba raz tak, aby sa zachovala jeho bezpečnosť by nemal takmer žiadne uplatnenie v praxi. Z toho dôvodu už nie je dostatočné uvažovať o bezpečnosti zero-knowledge protokolu iba v rámci jedného behu, ale je nutné uvažovať o bezpečnosti kompozícií zero-knowledge protokolov.

Poznámka 3.5.1 *Znovupoužiteľnosť za rôznych predpokladov reprezentuje vyššie uvedené typy kompozícií. Ak predpokladáme, že v každom okamihu bude môcť existovať iba jedna inštancia protokolu, a žiadna ďalšia nemôže začať, kým táto jedna neskončí, potom nám stačí uvažovať o bezpečnosti sekvenčnej kompozície zero-knowledge protokolov. Naopak, ak predpokladáme, že súčasne môže bežať viacero inštancii protokolu, potom už musíme uvažovať o bezpečnosti paralelnej (popr. súbežnej) kompozície zero-knowledge protokolov.*

Čestný užívateľ sa drží predpísaného protokolu pre každý beh protokolu, ale potenciálny útočník sa môže pokúsiť koordinovať činnosti v rámci viecerých behov protokolu za účelom oklamať overovateľa. Na princípe koordinácie činnosti útočníka v jednotlivých behoch protokolu sú založené útoky na kompozíciu (v jednom behu bezpečných) zero-knowledge protokol.

3.5.1 Sekvenčná kompozícia zero-knowledge protokolov

Definícia 3.5.1 *Hovoríme, že zero-knowledge protokol (O, D) je sekvenčný zero-knowlege protokol (uzavretý vzhľadom na sekvenčnú kompozíciu), ak sú vlastnosti zero-knowledge protokolu zachované aj po polynomiálnom počte (vzhľadom na dĺžku vstupu) spustení protokolu (O, D) sekvenčne.*

V tomto prípade je protokol spustený polynomiálne veľa krát, pričom každé spustenie protokolu nasleduje po ukončení predchádzajúceho behu protokolu (teda v žiadnom okamihu nie sú spustené dva behy protokolu súčasne).

Veta 3.5.1 *Nech (O, D) je výpočtový zero-knowledge dokazovací systém podľa definície 3.1.8. Potom (O, D) nie je uzavretý vzhľadom na sekvenčnú kompozíciu.*

Dôkaz tohto tvrdenia možno nájsť v práci [19]. V dôkaze sa využíva fakt, že dokazovateľ je výpočtovo neobmedzený. Zatiaľ neexistuje dôkaz v prípade ak je dokazovateľ výpočtovo obmedzený pravdepodobnostným polynomiálnym

časom a rovnako zatiaľ neexistuje dôkaz podobného tvrdenia pre štatistické a perfektné zero-knowledge dôkazy.

Zatiaľ čo verzia výpočtového zero-knowledge protokolu bez pomocného vstupu nezachováva vlastnosti výpočtového zero-knowledge protokolu v prípade sekvenčného spúštania, verzia výpočtového zero-knowledge protokolu so zahrnutím pomocného vstupu už zachováva vlastnosti výpočtového zero-knowledge protokolu v prípade sekvenčného spúštania.

3.5.2 Paralelná kompozícia zero-knowledge protokolov

Definícia 3.5.2 *Hovoríme, že zero-knowledge protokol* (O, D) *je paralelný zero-knowlege protokol (uzavretý vzhľadom na paralelnú kompozíciu), ak sú vlastnosti zero-knowledge protokolu zachované aj po polynomiálnom počte (vzhľadom na dĺžku vstupu) paralelných spustení protokolu* (O, D).

V tomto prípade je spustených polynomiálne veľa behov protokolu súčasne a bežia rovnakou rýchlosťou. Teda prepokladáme dokonale synchronizovaný model komunikácie v ktorom i-ta správa všetkých ešte neskončených behov protokolu je prijatá skôr ako je odoslaná $i+1$ správa v ktoromkoľvek z ešte aktívnych behov protokolu.

Vo všeobecnosti nie sú zero-knowledge protokoly (aj v prípade úvah s pomocným vstupom) uzavreté na paralelnú kompozíciu. Príklad zero-knowledge protokolu, ktorý nezachováva vlastnosti zero-knowledge protokolu v prípade paralelného spustenia možno nájsť v prácach [17] a [6].

Hlavnou motiváciou pre štúdium paralelnej kompozície zero-knowledge protokolov bola snaha o efektívnu redukciu chyby overovateľa (akceptácie $x \notin L$[28]) vzhľadom na počet kôl. K tomuto cieľu sa však podarilo dostať iným spôsobom ako prostredníctvom paralelnej kompozície. V súčasnosti sú známe iné spôsoby ako konštruovať zero-knowledge protokoly s konštantným počtom kôl (a ľubovolne malou kladnou chybou) [18].

Napriek tomu, že zero-knowledge dôkazy nie sú vo všeobecnosti uzavreté na paralelnú kompozíciu, za štandardných predpokladov (existencia jednosmernej funkcie, predpoklad že faktorizácia je ťažký problém a pod.) existuju zero-knowledge protokoly pre triedu **NP** uzavreté vzhľadom na paralelnú kompozíciu. Naviac tieto protokoly maju konštantný počet kôl [15].

3.5.3 Súbežná kompozícia zero-knowledge protokolov

Definícia 3.5.3 *Hovoríme, že zero-knowledge protokol* (O, D) *je súbežný zero-knowlege protokol (uzavretý vzhľadom na súbežnú kompozíciu), ak sú vlastnosti zero-knowledge protokolu zachované aj po polynomiálnom počte (vzhľadom na dĺžku vstupu) súbežných spustení protokolu* (O, D).

Súbežná kompozícia spája a zovšeobecňuje sekvenčnú a paralelnú kompozíciu. V tomto prípade je spustených polynomiálne veľa behov protokolu v ľubovolnom čase a bežia ľubovolnou rýchlosťou. Teda predpokladáme asynchrónny model komunikácie (na rozdiel od synchrónnej ako v prípade paralelnej kompozície). Pri súbežnej kompozícii možno uvažovať o dvoch modeloch:

- Čisto asynchrónny model.

[28] L je dokazovaný jazyk

- Asynchrónny model s časovačom.

Ukázalo sa [28, 17], že za štandardných predpokladov je možné skonštruovať súbežne zero-knowledge protokoly pre celú triedu **NP**.

Časť III

4 Modely s orákulom

4.1 Neinteraktívne zero-knowledge dôkazy

Všetky doteraz spomínané protokoly mali určitý (či už konštantný [15] alebo polynomiálny (kapitola 3.3.1) resp. logaritmický [10]) počet kôl v ktorých prebiehal dôkaz. Výskum neinteraktívnych zero-knowledge dôkazov bol motivovaný najmä snahou obmedziť interakciu medzi overovateľom a dokazovateľom, pretože v mnohých aplikáciách zero-knowledge protokolov by bolo vhodné, aby zero-knowledge protokol pozostával iba z jednej správy odoslanej dokazovateľom overovateľovi, ktorý by na základe tejto správy rozhodol, či dôkaz akceptuje, alebo neakceptuje. Ukázalo sa však [20], že takéto priame neinteraktívne zero-knowledge protokoly existujú iba pre jazyky z triedy **BPP**. Ak však zmeníme model a umožníme overovateľovi aj dokazovateľovi pristupovať k zdieľanému rovnomerne náhodne generovanému reťazcu, potom existujú v tomto modeli neinteraktívne zero-knowledge aj pre jazyky, o ktorých sa predpokladá, že sú mimo triedy **BPP**. Ak umožníme, aby tento referenčný reťazec bol generovaný pravdepodobnostným polynomiálnym algoritmom, ktorý na vstupe dostane dokazované tvrdenie (resp. dĺžku dokazovaného tvrdenia) dostávame model s pomocou (resp. model s verejným parametrom). V týchto zosilnených modeloch je potom možné ukázať viacero vlastností zero-knowledge protokolov, ako aj vzájomné súvislosti medzi interaktívnymi a neinteraktívnymi zero-knowledge dôkazmi.

Z praktického hľadiska je možné neinteraktívne zero-knowledge dôkazy využívať napríklad na vytváranie šifrových systémov odolných voči útoku s možnosťou voľby otvoreného textu[29], alebo na implementáciu podpisových schém.

Neinteraktívne zero-knowledge dôkazy možno ilustrovať v nasledujúcom príklade:

Príklad Dvaja matematici D a O boli spolu svedkami nejakej náhodnej udalosti. Matematik D potom ide na cestu okolo sveta, pričom pokračuje vo svojom matematickom bádaní. Vždy keď dokáže nové tvrdenie, napíše list matematikovi O, v ktorom dokáže platnosť tohto tvrdenia v zero-knowledge. V tomto prípade ide nutne o neinteraktívnu (jednosmernú) komunikáciu od matematika D k O, pretože aj keby chcel D odpovedať na otázky matematika O, nemôže, pretože tento nemá fixnú adresu [8].

Neinteraktívne zero-knowledge dôkazy sú variantom zero-knowledge dôkazu, v ktorom je definovaný dôveryhodný tretí účastník protokolu (tzv. dealer). Dealer vygeneruje referenčný reťazec, ktorý majú k dispozícii overovateľ aj dokazovateľ. Dokazovateľ potom vygeneruje a odošle overovateľovi jedinú správu (závislú na referenčnom reťazci) na základe ktorej sa overovateľ rozhodne, či akceptuje alebo neakceptuje dokazovateľov dôkaz. Overovateľ v tomto protokole neposiela dokazovateľovi žiadnu správu počas dokazovania.

[29] V tomto type type útoku si útočník môže nechať šifrovacím orákulom (resp. užívateľom v praktických aplikáciách) zašifrovať ním zvolenú spravu

Existuje viacero modelov neinteraktívnych zero-knowledge dokazovacích systémov v závislosti od spôsobu generovania referenčného reťazca:

- *Model so spoločným reťazcom*[30] *(crs).* Referenčný reťazec je rovnomerne náhodný reťazec polynomiálnej dĺžky v závislosti od vstupu.
- *Model s verejným parametrom (public parameter model).* Tento model je prirodzeným rozšírením modelu *crs*. Referenčný reťazec je generovaný pravdepodobnostným polynomiálnym algoritmom T so vstupom 1^n, kde n je dĺžka dokazovaného tvrdenia[31] x.
- *Model s pomocou (Help model).* Ďalším zovšeobecnením modelu s verejným parametrom je model s pomocou. V tomto prípade je referenčný reťazec generovaný pravdepodobnostným polynomiálnym algoritmom T, ktorý na vstup dostane dokazované tvrdenie x.

4.1.1 Model crs

Definícia 4.1.1 *Nech $\delta \in \{0,1\}^*$. Neinteraktívny zero-knowledge dokazovací systém v modeli crs pre jazyk L je zero-knowledge dokazovací systém v ktorom dostanú overovateľ aj dokazovateľ reťazec δ na pomocnom vstupe, a v ktorom je odoslaná iba jedna správa π od dokazovateľa D k overovateľovi O, pričom $\pi = D(x, \delta)$ a reťazec δ je generovaný rovnomerne náhodne na $\{0,1\}^n$, kde $n = p(|x|)$ a $p(|x|)$ je polynóm dĺžky vstupného slova x.*

Označenie 4.1.1 *Triedu jazykov majúcich neinteraktívny zero-knowledge dokazovací systém v modeli crs označujeme* $\mathbf{NIZK^{crs}}$ *alebo* **NIZK**.

4.1.2 Model s verejným parametrom

Definícia 4.1.2 *Nech $\delta \in \{0,1\}^*$. Neinteraktívny zero-knowledge dokazovací systém v modeli s verejným parametrom pre jazyk L je zero-knowledge dokazovací systém v ktorom dostanú overovateľ aj dokazovateľ reťazec δ na pomocnom vstupe, a v ktorom je odoslaná iba jedna správa π od dokazovateľa D k overovateľovi O, pričom $\pi = D(x, \delta)$ a reťazec δ je generovaný pravdepodobnostným polynomiálnym algoritmom so vstupom 1^n,kde $n = |x|$.*

Označenie 4.1.2 *Triedu jazykov majúcich neinteraktívny zero-knowledge dokazovací systém v modeli s verejným parametrom označujeme* $\mathbf{NIZK^{pub}}$.

4.1.3 Model s pomocou

Definícia 4.1.3 *Nech $\delta \in \{0,1\}^*$. Neinteraktívny zero-knowledge dokazovací systém v modeli s pomocou pre jazyk L je zero-knowledge dokazovací systém v ktorom dostanú overovateľ aj dokazovateľ reťazec δ na pomocnom vstupe, a v ktorom je odoslaná iba jedna správa π od dokazovateľa D k overovateľovi O, pričom $\pi = D(x, \delta)$ a reťazec δ je generovaný pravdepodobnostným polynomiálnym algoritmom, ktorý na vstupe dostane dokazované tvrdenie x.*

Označenie 4.1.3 *Triedu jazykov majúcich neinteraktívny zero-knowledge dokazovací systém v modeli s pomocou označujeme* $\mathbf{NIZK^h}$.

[30] Common random string model

[31] Teda algoritmus T nepozná samotné dokazované tvrdenie x, iba jeho dĺžku

Podobne ako pri zero-knowledge protokoloch (kapitola 3) možeme definovať jednotlivé podtriedy $\mathbf{NIZK^h}$:

- Perfektné $\mathbf{NIZK^h}$ ($\mathbf{NIPZK^h}$).
- Štatistické $\mathbf{NIZK^h}$ ($\mathbf{NISZK^h}$).
- Výpočtové $\mathbf{NIZK^h}$ ($\mathbf{NICZK^h}$).

Analogicky sú definované triedy neinteraktívnych zero-knowledge protokolov pre model s verejným parametrom ($\mathbf{NIPZK^{pub}}$, $\mathbf{NISZK^{pub}}$, $\mathbf{NICZK^{pub}}$) a model *crs* ($\mathbf{NIPZK^{crs}}$, $\mathbf{NISZK^{crs}}$, $\mathbf{NICZK^{crs}}$ resp. $\mathbf{NIPZK}$, $\mathbf{NISZK}$, $\mathbf{NICZK}$).

Pomerne ľahko vidieť, že platí pre jednotlivé triedy neinteraktívnych dôkazov nasledujúce tvrdenie:

Veta 4.1.1 $\mathbf{NIZK^{crs}} \subseteq \mathbf{NIZK^{pub}} \subseteq \mathbf{NIZK^h}$

Podrobný dôkaz analogického tvrdenia pre interaktívne zero-knowledge dôkazy v modeloch s orákulom uvedieme v ďalšej kapitole.

Hlavnou výhodou modelov s pomocou a s verejným parametrom pred *crs* je ľahšia konštruovateľnosť neinteraktívnych zero-knowledge protokolov z jednoduchších objektov, ako sú jednosmerné funkcie, alebo tzv. instance-dependent commitmemnt schémy [24].

Ukázalo sa však [8], že aj v modeli *crs* platí tvrdenie:

Veta 4.1.2 *Nech jazyk* $L \in \mathbf{NP}$*, potom existuje neinteraktívny zero-knowledge protokol v modeli crs pre jazyk* L*.*

Z hľadiska konštrukcie neinteraktívnych zero-knowledge dôkazov je dôležité aj nasledujúce tvrdenie ([33]):

Veta 4.1.3 *Tieto dve tvrdenia sú ekvivalentné:*

- $\mathbf{SZK} = \mathbf{NISZK}$
- $\mathbf{NISZK}$ *je uzavretá na komplement.*

Napriek tomu však zostáva otvoreným problémom, či $\mathbf{SZK} = \mathbf{NISZK}$.

4.2 Interaktívne zero-knowledge dôkazy

V tejto časti budeme uvažovať o troch modeloch zovšeobecnených verzii zero-knowledge dôkazov z definície bez pomocného vstupu z časti 3.1. V týchto modeloch majú overovateľ aj dokazovateľ prístup k referenčnému reťazcu vygenerovanom dôveryhodným tretím účastníkom protokolu (dealer) rovnako ako pri neinteraktívnych dôkazoch. Pridanie orákula (vo všetkých troch modeloch) umožnuje redukovať počet kôl zero-knowledge protokolu a teda vytvárať efektívnejšie (vzhľadom na počet kôl) zero-knowledge protokoly.

Pre tieto modely definujeme trojicu algoritmov O, D, T, kde O označuje overovateľa, D označuje dokazovateľa a T označuje dealera. Overovateľ a dokazovateľ dostanú na vstupe dokazované tvrdenie x a reťazec δ vygenerovaný dealerom a spustia interaktívny protokol, na konci ktorého overovateľ akceptuje, alebo neakceptuje.

Jednotlivé modely sú pre interaktívne zero-knowledge dôkazy s orákulom definované analogicky ako pri neinteraktívnych zero-knowledge dôkazoch s orákulom.

4.2.1 Model s pomocou

Model v ktorom je referenčný reťazec δ generovaný pravdepodobnostným polynomiálnym algoritmom, ktorý na vstupe dostane dokazované tvrdenie x sa nazýva model s pomocou.

Definícia 4.2.1 *Interaktívny dokazovací systém v modeli s pomocou pre jazyk L je trojica pravdepodobnostných algoritmov (O, D, T) (pričom O a T sú výpočtovo obmedzené polynomiálnym časom) taká, že platí:*

- *Kompletnosť:* $\forall x \in L : \Pr[(O, D, T) = accept] \geq \frac{2}{3}$, *pričom pravdepodobnosť je braná cez náhodné bity O,D a T.*
- *Nepriestrelnosť:* $\forall D^* \forall x \notin L : \Pr[(O, D^*, T) = accept] \leq \frac{1}{3}$, *pričom pravdepodobnosť je braná cez náhodné bity O,D^* a T.*
- *Zero-knowledge: Existuje pravdepodobnostný polynomiálny algoritmus S taký, že $\{\langle O, D, T\rangle(x)\}_x$ a $\{S(x)\}_x$ sú nerozoznateľné pre všetky $x \in L$.*

pričom algoritmus T dostane na vstupe dokazované tvrdenie x.

Poznámka 4.2.1 *$\langle O, D, T\rangle(x)$ označuje transkript komunikácie v protokole.*

Označenie 4.2.1 *Triedu jazykov majúcich interaktívny zero-knowledge dokazovací systém v modeli s pomocou označujeme* $\mathbf{ZK^h}$.

Poznámka 4.2.2 *Triedy* $\mathbf{CZK^h}$, $\mathbf{SZK^h}$ *a* $\mathbf{PZK^h}$ *definujeme podobne ako pri neinteraktívnych zero-knowledge dôkazoch v modeli s pomocou v závislosti od typu nerozoznateľnosti.*

Aj napriek dodaniu pomoci do modelu zero-knowledge dôkazov sú jazyky triedy $\mathbf{ZK^h}$ rozpoznateľné v polynomiálnom priestore a teda platí:

Veta 4.2.1 $\mathbf{ZK^h} \subseteq \mathbf{PSPACE}$.

Dôkaz. Najprv ukážeme, že $\mathbf{ZK^h} \subseteq \mathbf{IP}$. Pretransformujeme dôkaz v modeli s pomocou nasledovne: Overovateľ simuluje dealera a v prvej správe odošle referenčný reťazec δ. Táto transformácia nezachováva vo všeobecnosti vlastnosť zero-knowledge, pretože aj čestný overovateľ sa dozvie náhodné bity dealera, ale zachováva vlastnosti kompletnosť a nepriestrelnosť. Teda platí, že $\mathbf{ZK^h} \subseteq \mathbf{IP}$. Podľa [30] platí $\mathbf{IP} = \mathbf{PSPACE}$. Z toho však vyplýva, že $\mathbf{ZK^h} \subseteq \mathbf{PSPACE}$. □

4.2.2 Model s verejným parametrom

Model v ktorom je referenčný reťazec δ generovaný pravdepodobnostným polynomiálnym algoritmom, ktorý na vstupe dostane reťazec 1^n, kde $n = |x|$ sa nazýva model s verejným parametrom.

Definícia 4.2.2 *Interaktívny dokazovací systém v modeli s verejným parametrom pre jazyk L je trojica pravdepodobnostných algoritmov (O, D, T) (pričom O a T sú výpočtovo obmedzené polynomiálnym časom a T na vstupe nedostane dokazované tvrdenie x iba reťazec $1^{|x|}$) taká, že platí:*

- *Kompletnosť:* $\forall x \in L : \Pr[(O, D, T) = accept] \geq \frac{2}{3}$, *pričom pravdepodobnosť je braná cez náhodné bity* O,D *a* T.
- *Nepriestrelnosť:* $\forall D^* \forall x \notin L : \Pr[(O, D^*, T) = accept] \leq \frac{1}{3}$, *pričom pravdepodobnosť je braná cez náhodné bity* O,D^* *a* T.
- *Zero-knowledge: Existuje pravdepodobnostný polynomiálny algoritmus* S *taký, že* $\{\langle O, D, T\rangle(x)\}_x$ *a* $\{S(x)\}_x$ *sú nerozoznateľné pre všetky* $x \in L$.

pričom algoritmus T *dostane na vstupe dĺžku dokazovaného tvrdenia* x.

Označenie 4.2.2 *Triedu jazykov majúcich interaktívny zero-knowledge dokazovací systém v modeli s verejným parametrom označujeme* $\mathbf{ZK^{pub}}$.

Poznámka 4.2.3 *Triedy* $\mathbf{CZK^{pub}}$, $\mathbf{SZK^{pub}}$ *a* $\mathbf{PZK^{pub}}$ *definujeme podobne ako pri neinteraktívnych zero-knowledge dôkazoch v modeli s verejným parametrom v závislosti od typu nerozoznateľnosti.*

Rovnako ako pri modeli s pomocou platí nasledujúce tvrdenie:

Veta 4.2.2 $\mathbf{ZK^{pub}} \subseteq \mathbf{PSPACE}$.

4.2.3 Model crs

Posledným modelom je *crs*.V tomto modeli je referenčný reťazec δ generovaný rovnomerne náhodne na $\{0,1\}^n$, kde $n = |x|$.

Definícia 4.2.3 *Interaktívny dokazovací systém v modeli crs pre jazyk* L *je trojica* (O, D, δ) *(pričom* O *je výpočtovo obmedzený polynomiálnym časom a* δ *je referenčný retazec generovaný rovnomerne náhodne na* $\{0,1\}^n$, *kde* $n = |x|$ *) taká, že platí:*

- *Kompletnosť:* $\forall x \in L : \Pr[(O, D, \delta) = accept] \geq \frac{2}{3}$, *pričom pravdepodobnosť je braná cez náhodné bity* O,D *a* δ.
- *Nepriestrelnosť:* $\forall D^* \forall x \notin L : \Pr[(O, D^*, \delta) = accept] \leq \frac{1}{3}$, *pričom pravdepodobnosť je braná cez náhodné bity* O,D^* *a* T.
- *Zero-knowledge: Existuje pravdepodobnostný polynomiálny algoritmus* S *taký, že* $\{\langle O, D, \delta\rangle(x)\}_x$ *a* $\{S(x)\}_x$ *sú nerozoznateľné pre všetky* $x \in L$.

pričom reťazec δ *je generovaný rovnomerne náhodne na* $\{0,1\}^n$.

Poznámka 4.2.4 $\langle O, D, \delta\rangle(x)$ *označuje transkript komunikácie v protokole.*

Označenie 4.2.3 *Triedu jazykov majúcich interaktívny zero-knowledge dokazovací systém v modeli crs označujeme* $\mathbf{ZK^{crs}}$.

4.3 Vlastnosti

V tejto kapitole sa už budeme zaoberať najmä vlastnosťami a vzťahmi s ostatnými triedami pre model s pomocou. V tomto modeli môže už intuitívne dealer najviac pomôcť rozšíriť triedu jazykov majúcich zero-knowledge dôkaz daného typu. Túto intuíciu potvrdzuje aj nasledujúca veta:

Veta 4.3.1 $\mathbf{ZK^{crs}} \subseteq \mathbf{ZK^{pub}} \subseteq \mathbf{ZK^{h}}$

Dôkaz. Najprv ukážeme inklúziu $\mathbf{ZK^{crs}} \subseteq \mathbf{ZK^{pub}}$. Nech (O, D, δ) je interaktívny zero-knowledge dôkaz v modeli *crs* pre jazyk L. Nech (O', D', T) je interaktívny zero-knowledge dôkaz v modeli *pub* taký, že platí $O = O'$,$D = D'$ a $T(|x|)$, kde x je dokazované tvrdenie pracuje takto:

- Vezme $\gamma = \delta$ resp. rovnomerne náhodne vygeneruje reťazec dĺžky n na $\{0,1\}^*$.
- Pošle reťazec γ na vstup O a D.

Takto vytvorený protokol (O', D', T) zachováva kompletnosť aj nepriestrelnosť triviálne, pretože overovateľ aj dokazovateľ sú totožný s pôvodným protokolom a na vstupe dostanú rovnako ako v pôvodnom protokole x a rovnomerne náhodný reťazec dĺžky n. Ostáva už len ukázať, že existuje simulátor S' pre protokol (O', D', T) podľa definície interaktívnych zero-knowledge dôkazov v modeli s vrejným parametrom. Platí však že $S' = S$, kde S je simulátor pre protokol (O, D, δ), pretože pre overovateľa a dokazovateľa je to z ich pohľadu rovnaký protokol a distribúcia náhodných bitov T je rovnaká ako distribúcia reťazca δ. Z toho vyplýva, že $\mathbf{ZK^{crs}} \subseteq \mathbf{ZK^{pub}}$.

Dôkaz pre inklúziu $\mathbf{ZK^{pub}} \subseteq \mathbf{ZK^{h}}$ je analogický.Dealer pre protokol pre model s pomocou bude ignorovať dokazované tvrdenie x a bude pracovať iba s n, kde $n = |x|$. □

Pre model s pomocou bol hlavný prúd záujmu vo výskume porovnanie s klasickými triedami zero-knowledge protokolov. Bolo objavených viacero výsledkov týkajúcich sa rôznych tried zero-knowledge, pričom veľká pozornosť je venovaná najmä štatistickým zero-knowledge dôkazom:

Veta 4.3.2 $\mathbf{SZK} = \mathbf{SZK^{h}}$.

Prekvapivý a dôležitý výsledok bol publikovaný v práci [24]:

Veta 4.3.3 $\mathbf{SZK} = \mathbf{NISZK^{h}}$.

To znamená, že pomoc (teda referenčný reťazec generovaný pravdepodobnostným polynomiálnym algoritmom, ktorý na vstupe dostanú overovateľ aj dokazovateľ) dokáže nahradiť interakciu (potenciálne až polynomiálne veľa kôl protokolu). Z týchto dvoch výsledkov už vyplýva:

Veta 4.3.4 $\mathbf{SZK^{h}} = \mathbf{NISZK^{h}}$.

5 Resetovateľnosť účastníkov ZK protokolu

Výskum bezpečnosti sa v posledných rokoch zaoberá nielen bezpečnosťou samotných protokolov, ale uvažuje aj o situácii, keď niektorý z účastníkov protokolu môže v ktoromkoľek kroku protokolu vrátiť[32] niektorého iného účastníka protokolu do stavu v ktorom bol na začiatku protokolu vrátane všetkých jeho náhodných bitov[33].

Význam resetovateľných zero-knowledge protokolov. Význam výskumu zero-knowledge protokolov s možnosťou vrátiť niektorého účastníka do stavu v ktorom bol na začiatku protokolu je v dvoch rovinách:

- V teoretickej rovine je zaujímavá otázka, či zero-knowledge protokoly (ale aj všeobecne kryptografické protokoly) zachovávajú svoju bezpečnosť aj v prípade, že účastníci protokolu nemôžu využiť v každom behu protokolu nanovo vygenerované náhodné bity.
- V praxi je často nemožné a/alebo neželané generovať za behu nové náhodné bity či už z dôvodu efektivity, nedostatku entropie[34], alebo rôznych iných dôvodov.

Na základe toho, ktorý z účastníkov je resetovaný uvažujeme o dvoch prípadoch:

- *Overovateľ môže resetovať dokazovateľa.* Protokoly, ktoré zostávajú zero-knowledge protokolmi aj v prípade možnosti resetovať dokazovateľa sa nazývajú resetovateľné zero-knowledge protokoly (rZK).
- *Dokazovateľ môže resetovať overovateľa.* Protokoly, ktoré zostávajú zero-knowledge protokolmi aj v prípade možnosti resetovať overovateľa sa nazývajú zero-knowledge protokoly s resetovateľnou nepriestrelnosťou (rsZK[35]).

5.1 Resetovateľné zero-knowledge dôkazy

Na resetovateľný zero-knowledge dôkaz sa možno pozerať tak, že v klasickom zero-knowledge dôkaze pre jazyk L je zafixovaný vstup x, náhodné bity dokazovateľa r a svedka w (ak uvažujeme o jazyku $L \in \mathbf{NP}$). Overovateľ má prístup k polynomiálne veľa „klonom“ dokazovateľa D, pričom ich môže spúšťať v ľubovolnom poradí, môže ich spúšťať v rôznom čase, každý z nich má rovnaké náhodné bity r , vstup x aj svedka w, pričom jednotlivé inštancie dokazovateľa nemôžu spolu komunikovať.

Označenie 5.1.1 *Triedu jazykov majúcich resetovateľný zero-knowledge dôkaz označujeme* **rZK**.

[32] resetovať

[33] To znamená, že v ďalšom behu protokolu použije rovnaké náhodné bity ako v predchádzajúcom behu protokolu a nemôže tieto rôzne behy protokolu koordinovať.

[34] Napríklad v Linuxe známy */dev/random* sa po vyčerpaní entropie zasekne a čaká na jej doplnenie.

[35] resettably-sound zero-knowledge

Resetovateľné zero-knowledge dôkazy sú z pohľadu bezpečnosti[36] v súčasnosti najsilnejším modelom. Každý zero-knowledge dôkaz, ktorý je resetovateľným zero-knowledge dôkazom je aj uzavretý vzhľadom na súbežnú kompozíciu.

Všetky protokoly, ktoré sme uviedli v tejto práci sú triviálne rozbiteľné ak umožníme overovateľovi resetovať dokazovateľa. Uvažujme napríklad protokol pre $G3C$ z časti 3.3.1. Ak by sme v tomto protokole umožnili overovateľovi resetovať dokazovateľa tak by prostredníctvom tejto možnosti zistil kompletné ofarbenie grafu, pretože by sa v druhom kroku spýtal vždy na inú hranu a po odpovedi dokazovateľa ho resetoval. Takýmto spôsobom by získal 3-zafarbenie grafu (ak by existovalo).

Existujú techniky ktoré umožnia niektoré protokoly s istými ďalšími dobrými vlastnosťami pretransformovať tak, aby boli resetovateľné. Tieto techniky sú založené na využití pseudonáhodnej funkcie, ktorá vytvorí pseudonáhodné bity zo vstupu pozostávajúceho z náhodných bitov zafixovaných na začiatku protokolu a prijatej správy od overovateľa v predchádzajúcom kole prokolu. Zodpovedajúci resetovateľný zero-knowledge protokol pre $G3C$ možno nájsť v práci [9].

V tomto kontexte je zaujímava aj otázka veľkosti triedy **rZK**.

Označenie 5.1.2 *$DLP(k)$ označuje úlohu riešenia problému diskrétneho logaritmu pre inštancie dĺžky k.*

Pre nasledujúce tvrdenia popisujúce veľkosť triedy **rZK** potrebujeme okrem štandardných predpokladov aj silný a slabý DLP predpoklad.

- *Silný DLP predpoklad*: Úloha $DLP(k)$ nie je riešiteľná v čase 2^{k^ϵ} pre nejaké $\epsilon > 0$.
- *Slabý DLP predpoklad*: Úloha $DLP(k)$ nie je riešiteľná v pravdepodobnostnom polynomiálnom čase.

Veta 5.1.1 *Ak platí slabý DLP predpoklad, potom existuje resetovateľný zero-knowledge protokol s polynomiálnym počtom kôl pre každý jazyk $L \in$ **NP** [9].*

Tento výsledok je z praktického hladiska nepoužiteľný z dôvodu polynomiálneho počtu kôl. Pre praktické aplikácie je omnoho dôležitejší výsledok, ktorý však vyžaduje viacero predpokladov (silný DLP predpoklad a public-key model) a uvažuje iba o resetovateľných zero-knowledge argumentoch[37].

Public-key model. Public-key modelom rozumieme model v ktorom má každý účastník protokolu registrovaný svoj verejný kľúč v súbore prístupnom v ktoromkoľvek čase ktorýmkoľvek účastníkom protokolu. Jediným predpokladom je garancia, že tieto verejné kľúče boli registrované (uložené v súbore) pred začiatkom komunikácie. Na tento súbor[38] nie sú kladené žiadne ďalšie požiadavky a preto útočník sem môže umiestniť viacero verejných kľúčov (aj nekorektných, alebo takých, ku ktorým nepozná zodpovedajúci súkromný kľúč).

[36] vzhľadom na dokazovateľa v zmysle získania z protokolu dodatočnej informácie overovateľom

[37] definovaných analogicky ako pri zero-knowledge argumentoch

[38] Prístup k tomuto súboru môže byť implementovaný napríklad prostredníctvom viacerých identických serverov alebo prostredníctvom certifikátov verejného kľúča prideleného pred začiatkom komunikácie.

Veta 5.1.2 *Ak platí silný DLP predpoklad, potom existuje resetovateľný zero-knowledge argument v public-key modeli pre každý jazyk* $L \in \mathbf{NP}$.

Využitie resetovateľných zero-knowledge dôkazov. Vo všeobecnosti pre resetovateľné zero-knowledge dôkazy platí:

- *rZK* zväčšujú počet možností na implementáciu zero-knowledge protokolu pri zachovaní bezpečnosti.
- *rZK* garantujú zachovanie bezpečnosti aj pri súbežnej kompozícii v asynchrónnej sieti[39] (napr. Internet).
- *rZK* umožňujú vytvoriť bezpečnejšie identifikačné schémy ako predchádzajúce modely zero-knowledge dôkazov.

5.2 Zero-knowledge dôkazy s resetovateľnou nepriestrelnosťou

Prípad zero-knowledge dôkazov s resetovateľnou nepriestrelnosťou je analogický ako prípad resetovateľných zero-knowledge dôkazov s rozdielom, že overovateľ nemôže resetovať dokazovateľa, ale dokazovateľ môže resetovať overovateľa. Aj v tomto prípade sa na resetovatelnú časť (v tomto prípade overovateľa) dá pozerať ako na viacero klonov tejto inštancie s rovnakými náhodnými bitmi. Neformálne povedané interaktívny dôkaz alebo argument má resetovateľnú nepriestrelnosť, ak dokazovateľ dokáže presvedčiť overovateľa o nepravdivom tvrdení ($x \notin L$, ale overovateľ akceptuje toto tvrdenie ako pravdivé) len so zanedbateľnou pravdepodobnosťou aj v prípade, že môže resetovať overovateľa.

Keďže trieda jazykov majúcich zero-knowledge dôkazy s resetovateľnou nepriestrelnosťou je relatívne malá ($\mathbf{P/poly}$), výskum v tejto oblasti sa zameriava na zero-knowledge argumenty s resetovateľnou nepriestrelnosťou. Veľkosť tejto triedy popisuje tvrdenie ([4]):

Veta 5.2.1 *Ak existuje hašovacia funkcia odolná voči kolíziám, potom pre každý jazyk z NP existuje zero-knowledge argument s resetovateľnou nepriestrelnosťou.*

Poznámka 5.2.1 $\mathbf{P/poly}$ *je trieda jazykov rozpoznateľná v polynomiálnom čase s polynomiálne obmedzenou tzv. advice funkciou. Advice funkcia je pomocný vstup pre Turingov stroj ktorá nemá prístup k štandardnému vstupu iba k jeho dĺžke. V tomto prípade je veľkosť advice reťazca polybomiálne obmedzený vzhľadom na vstup.*

5.3 Resetovateľné zero-knowledge dôkazy s resetovateľnou nepriestrelnosťou

Prirodzenou otázkou je, či je možné zabezpečiť zároveň odolnosť overovateľa aj dokazovateľa proti resetovaniu. Je však zatiaľ otvoreným problémom, či jazyky mimo triedy **BPP** majú resetovateľný zero-knowledge argument, ktorý má aj vlastnosť resetovateľnej nepriestrelnosti.

[39] teda sú veľmi dobre prakticky využitelné

6 Promise problémy

V celej tejto práci, ale aj všeobecne v teoretickej informatike je jedným z najdôležitejších pojmov pojem jazyka. Jazyk je ľubovolná množina slov na vopred zvolenej abecede. Rozoznávame napr. jazyk všetkých grafov, ktoré majú hamiltonovu kružnicu, alebo jazyk všetkých prvočísel. Príslušnosť slova x do jazyka L je rozhodovací problém pri ktorom sú dve možnosti $x \in L$ a $x \notin L$. Zrejme platí $\{x|x \in \{0,1\}^* : x \in L\} \cup \{x|x \in \{0,1\}^* : x \notin L\} = \{x|x \in \{0,1\}^*\}$.

Promise[40] problémy sú zovšeobecnením rozhodovacieho problému príslušnosti slova x do jazyka L. Promise problém Π je dvojica disjunktných množín (Π_Y, Π_N) $(\Pi_Y, \Pi_N \subseteq \{\{0,1\}^*\})$, kde Π_Y označuje množinu YES inštancii a Π_N označuje množinu NO inštancii promise problému[41]. Výpočtový problém pre promise problém Π je určiť či dané x patrí do Π_Y alebo do Π_N, pričom máme sľúbené[42], že $x \in \Pi_Y \cup \Pi_N$. $x \in \Pi_Y \cup \Pi_N$ sa nazývajú inštancie Π. $x \notin \Pi_Y \cup \Pi_N$ sa nazývajú porušením sľubu.

Všetky tvrdenia a definície v tejto práci platia nielen pre jazyky, ale aj pre ich zovšeobecnenú verziu. Využitie promise problémov možno nájsť napríklad v prácach [33] a [24].

7 Záver

V tejto práci sme uviedli stručný prehľad o zero-knowledge dôkazoch. Našou snahou bolo názorne vysvetliť princíp fungovania zero-knowlege protokolov, ako aj vytvoriť prácu, ktorá by uviedla čitateľa do problematiky a súčasne ho nasmerovala v prípade záujmu o niektorú konkrétnu časť na zodpovedajúcu literatúru. V prípade hlbšieho záujmu odporúčame najmä štúdium [17] a [16]. Bližšie informácie o jednotlivých aspektoch problematiky, ako aj množstvo prác zaoberajúcich sa kryptografiou možno nájsť na webovskej adrese *http://eprint.iacr.org/complete/*.

[40] vyžívame anglické označenie *promised* namiesto slovenského prekladu *sľubné*

[41] Pri analógii s jazykom je Π_Y množina slov patriacich do jazyka a Π_N je množina slov nepatriacich do jazyka.

[42] v angličtine promised, z čoho pochádza aj názov tejto konštrukcie

Referencie

[1] S. Arora, B. Barak: *Computational Complexity: A modern approach*. http://www.cs.princeton.edu/theory/complexity/, 2008.

[2] L. Babai, S. Moran: *Arthur-Merlin games:A randomized proof system, and a hierarchy of complexity classes*. Journal of Computer and System Sciences, 36(2):254–276, 1988.

[3] B. Barak: *How to go beyond the black-box simulation barrier*. http://www.cs.princeton.edu/∼boaz/Papers/nonbb.ps, 2001.

[4] B. Barak, O. Goldreich, S. Goldwasser, Y. Lindell: *Resettably-Sound Zero-Knowledge and its Applications*. http://www.wisdom.weizmann.ac.il/∼oded/PS/rszk.ps, 2001.

[5] B. Barak, Y. Lindell, S. Vadhan: *Lower bounds for non-black-box zero knowledge*. http://www.cs.princeton.edu/∼boaz/Papers/zklower.pdf, 2006.

[6] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Hastad, J. Kilian, S. Micali, P. Rogaway: *Everything Provable is Provable in Zero-Knowledge*. In Crypto88, Springer-Verlag lecture notes in computer science (Vol. 403), pp. 37-56, 1990.

[7] M. Ben-Or, D. Gutfreund: *Trading Help for Interaction in Statistical Zero-Knowledge Proofs*. www-math.mit.edu/∼danny/pubs/help_interaction.ps.

[8] M. Blum, P. Feldman, S. Micali: *Non-Interactive Zero-Knowledge and Its Applications*. Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988). 103-112. 1988.

[9] R. Canetti, O. Goldreich, S. Goldwasser, S. Micali: *Resettable Zero-Knowledge*. http://www.wisdom.weizmann.ac.il/∼oded/p_cggm.html, 2000.

[10] R. Canetti, J. Kilian, E. Petrank, A. Rosen: *Black-box concurrent zero-knowledge requires* $\sim \Omega(\log n)$ *rounds*. http://people.csail.mit.edu/canetti/materials/ckpr02.ps, 2001.

[11] U. Feige, A. Fiat, A. Shamir: *Zero-knowledge proofs of identity*. Journal of Cryptology,1(2):pp. 77–94, 1988.

[12] L. Fortnow: *The complexity of perfect zero-knowledge*. In 19th ACM Symposium on the Theory of Computing, pp. 204-209, 1987.

[13] M. Furer, O. Goldreich, Y. Mansour, M. Sipser, S. Zachos: *On Completness and Soundness in Interactive Proof Systems*. Advances in Computing Research,pp. 429-442, 1989.

[14] O. Goldreich: *Computational Complexity: A Conceptual Perspective*. http://www.wisdom.weizmann.ac.il/∼oded/cc-book.html, 2008.

[15] O. Goldreich: *Concurrent Zero-Knowledge With Timing.* http://www.wisdom.weizmann.ac.il/∼oded/p_conc-zk.html, 2001 revisited 2005.

[16] O. Goldreich : *Foundation Of Cryptography Vol.2 Part 4.* http://www.wisdom.weizmann.ac.il/∼oded/PSBookFrag/part4N.ps, 1998.

[17] O. Goldreich: *Zero knowledge twenty years after its invention.* http://www.wisdom.weizmann.ac.il/∼oded/zk-tut02.html, 2002.

[18] O. Goldreich, A. Kahan: *How to Construct Constant-Round Zero-Knowledge Proof Systems for NP.* http://www.wisdom.weizmann.ac.il/∼oded/PS/zkAK.ps, 1996.

[19] O. Goldreich, H. Krawczyk: *On the Composition of Zero-Knowledge Proof Systems.* http://www.wisdom.weizmann.ac.il/∼oded/PS/zk-comp.ps, 1994.

[20] O. Goldreich, Y. Oren: *Definitions and properties of Zero-Knowledge proof systems.* http://www.wisdom.weizmann.ac.il/∼oded/PS/oren.ps, 1992.

[21] S. Goldwasser, S. Micali, C. Rackoff: *The knowledge complexity of interactive proof systems.*In Proceedings of the 17th ACM Symposium on Theory of Computing, pp. 291–304. ACM Press, April 1985.

[22] S. Goldwasser, M. Sipser : *Private Coins versus Public Coins in Interactive proof systems.* Advances in computing ressearch, a ressearch annual Vol.5 pp 73-90 , 1989.

[23] I. Haitner, M. Nguyen, S. J. Ong, O. Reingold, S. Vadhan: *Statistically Hiding Commitments and Statistical Zero-Knowledge Arguments from Any One-Way Function.* http://www.eecs.harvard.edu/ salil/papers/SHcommit-nov07.pdf, 2007.

[24] A. Chailloux, D. F. Ciocan, I. Kerenidis, S. Vadhan: *Interactive and Noninteractive Zero Knowledge are Equivalent in the Help Model.* http://eprint.iacr.org/2007/467.pdf, 2007.

[25] G. Jain: *Zero knowledge Proofs: A survey.* http://www.it.lut.fi/kurssit/03-04/010635000/luennot/29-Tmt-ek-2004_Zero-knowledge-proofs-survey.pdf

[26] V. Kabanets: *Lectures From Computatinal complexity lecture 18.* http://www.cs.sfu.ca/∼kabanets/cmpt710/lec18.pdf, 2003.

[27] V. Kabanets: *Lectures From Computatinal complexity lecture 19.* http://www.cs.sfu.ca/∼kabanets/cmpt710/lec19.pdf, 2003.

[28] R. Richardson, J. Kilian: *On the Concurrent Composition of Zero-Knowledge Proofs.* Springer Lecture Notes in Computer Science (vol. 1592), pp. 413-415.

[29] J. Rothe: *Some Facets of Complexity Theory and Cryptography : Five lectures Tutorial.* ACM Computing Surveys, vol. 34, no. 4, December 2002, pp. 34-46, 2002.

[30] A. Shamir: *IP=PSPACE*. The Journal of the ACM, 39(4):869-877, October 1992.

[31] M. Stanek: *Základy kryptológie*. http://www.dcs.fmph.uniba.sk/∼stanek/crypto/main2.pdf, 2004.

[32] D. Stinson: *Cryptography: Theory and Practice*.CRC Press, 1995.

[33] S. P. Vadhan: *Study of statistical Zero knowledge*. http://www.eecs.harvard.edu/∼salil/papers/phdthesis-abs.html, 1999.