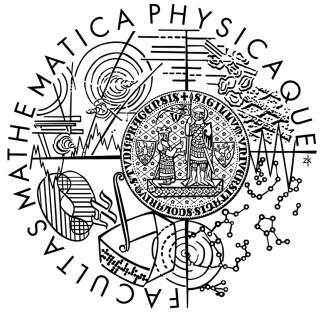
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Jana Kučerová

Interaktivní důkazy

Katedra algebry

Vedoucí bakalářské práce: Doc. RNDr. Jiří Tůma, DrSc.
Studijní program: Matematika
Studijní obor: Matematické metody informační bezpečnosti

2006

Děkuji doc. RNDr. Jiřímu Tůmovi, DrSc. za vedení mé práce a za jeho cenné připomínky.

Prohlašuji, že jsem svou bakalářskou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 2. 8. 2006 Jana Kučerová

Obsah

Název práce: Interaktivní důkazy
Autor: Jana Kučerová
Katedra (ústav): Katedra algebry
Vedoucí bakalářské práce: Doc.RNDr. Jiří Tůma,DrSc.
e-mail vedoucího: tuma@karlin.mff.cuni.cz

Abstrakt: Předložená práce se věnuje kryptografickým protokolům pro interaktivní důkazy. Protože tyto protokoly jsou jedny ze složitějších, jsou v první části práce popsána některá jednodušší kryptografická schémata, která jsou později využita jako stavební prvky těchto protokolů. V této práci jsou interaktivní důkazy nejprve definovány jako výpočet spojené dvojice interaktivních Turingových strojů. Později je uvedena souvislost takto definovaného interaktivního důkazu s interaktivním důkazem znalosti dokazovatelova tajemství v identifikačních protokolech. Na příkladu několika identifikačních protokolů je ukázáno, jakým způsobem lze rozhodnout, zda se jedná o důkaz s nulovou znalostí nebo zda je daný protokol prokazatelně bezpečný.

Klíčová slova: kryptografický protokol, Turingův stroj, prokazatelná bezpečnost, důkaz s nulovou znalostí

Title: Interactive proofs
Author: Jana Kučerová
Department: Department of Algebra
Supervisor: Doc.RNDr. Jiří Tůma,DrSc.
Supervisor's e-mail address: tuma@karlin.mff.cuni.cz

Abstract: The main topic of the present work is cryptographic protocols for interactive proofs. Because of complexity of these protocols, at first we describe some cryptographic elements, which we will use later as building blocs of interactive proof protocols. In this work, we define interactive proofs as a joint computation of a linked pair of two interactive Turing machines. Later, we explain the relation between this definition and an interactive proof of knowledge of a prover's secret in identification protocols. We present examples of several identification protocols and demonstrate how we can determine, if the given protocol is provably secure or if it is a zero-knowledge proof.

Keywords: cryptographic protocol, Turing machine, provable security, zero-knowledge proof

Kapitola 1

Úvod

Kryptografické protokoly se používají k řešení různých problémů, jako je např. výměna klíče pro symetrické šifrovací schéma, autentizace uživatele nebo elektronické volby. Tato práce se zabývá protokoly pro interaktivní důkazy, zejména pro důkazy identity uživatele (neboli autentizaci).

Identita se v identifikačních protokolech prokazuje důkazem znalosti určitého tajemství, které zná pouze daný uživatel. Aby tomu tak zůstalo i po proběhnutí protokolu, je vhodné, aby o tomto tajemství neunikla žádná netriviální informace, která by mohla vést k jeho odhalení. To zajišťují tzv. důkazy s nulovou znalostí. Jak se ale ukáže, některé důkazy s nulovou znalostí jsou nevhodné z jiných důvodů — neposkytují sice žádnou netriviální informaci o uživatelově tajemství, ale nezaručují bezpečnost z jiného hlediska.

V této práci nejprve uvedeme matematické základy nutné k pochopení dalšího textu. Dále uvedeme jednoduchá kryptografická schémata, která později použijeme jako stavební prvky identifikačních protokolů. Přes definici interaktivních důkazů pomocí interaktivních Turingových strojů se dopracujeme k identifikačním protokolům, které zhodnotíme z hlediska prokazatelné bezpečnosti a také u nich rozhodneme, zda se jedná o důkazy s nulovou znalostí.

Kapitola 2

Protokol

Protokol je přesně definovaná posloupnost kroků, zahrnující dva nebo více účastníků navržená k vykonání určité úlohy, např. výměny klíče pro symetrickou šifru, autentizaci uživatele, apod. Každý krok protokolu musí být vykonaný a pořadí kroků musí být dodržené. Další požadavky na protokol jsou:

- vzájemné odsouhlasení — každý účastník musí být obeznámený s celým protokolem a musí s ním souhlasit,
- jednoznačnost — každý krok musí být dobře definovaný, aby nemohlo dojít k nedorozumění,
- úplnost — musí být přesně vymezené chování každého účastníka v každé situaci.

Kryptografický protokol je protokol využívající kryptografii. Úlohou kryptografie v protokolech je zabránit nebo odhalit případný podvod a zabránit účastníkům nebo třetí straně získat víc informací, než protokol vymezuje. Úroveň bezpečnosti kryptografického protokolu je maximálně tak velká, jako je bezpečnost jím použitého kryptosystému, může však být mnohem nižší.

V této práci se budeme zabývat zejména indentifikačními protokoly. Ve čtvrté kapitole uvedeme nekterá jednodušší schémata, která budeme používat v dalších kapitolách při konstrukci složitějších schémat.

Kapitola 3

Matematické základy

V dalším textu budeme používat následující označení:

m	otevřený text
c	šifrový text
$GCD(a, b)$	největší společný dělitel čísel a a b
$\mathtt{P}(A)$	pravděpodobnost jevu A

Symbolem $\mathbb{Z}_n$ budeme označovat aditivní Abelovu grupu přirozených čísel modulo n a symbolem $\mathbb{Z}_n^*$ budeme označovat multiplikativní Abelovu grupu všech prvků $\mathbb{Z}_n$, které jsou s n nesoudělné.

Definice 3.1. Nechť g je funkce $g : \mathbb{N} \to \mathbb{R}$. Říkáme, že funkce $f : \mathbb{N} \to \mathbb{R}$ *má složitost* $\mathcal{O}(g)$ (nebo že leží v $\mathcal{O}(g)$), pokud existuje $c > 0$ takové, že pro všechna $n \in \mathbb{N}$ platí

$$f(n) < c \cdot g(n).$$

3.1 Problémy z teorie čísel, jednosměrné funkce

Bezpečnost mnohých kryptografických protokolů je založena na výpočetní nezvládnutelnosti některých matematických problémů. Pojem výpočetní nezvládnutelnost znamená, že v současnosti není znám žádný algoritmus, který by dokázal daný problém vyřešit v polynomiálním čase. O výpočetně nezvládnutelných problémech říkáme, že jsou *těžké*. Jsou to například problém faktorizace složených čísel, problém nalezení diskrétního logaritmu nebo problém nalezení druhé odmocniny v modulární aritmetice. Tyto problémy nyní přiblížíme.

3.1.1 Faktorizace

Definice 3.2. *Problém faktorizace přirozených čísel* je následující: Pro dané $n \in \mathbb{N}$ nalezněte jeho prvočíselný rozklad, tj. vyjádření $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$, kde p_i jsou po dvou různá prvočísla a $e_i \geq 1$ pro všechna $i = 1, \dots, k$.

Zatímco vynásobení dvou celých čísel a a b má časovou složitost $\mathcal{O}(|a| \cdot |b|)$, faktorizace složeného čísla je považována za těžký problém. Mezi nejlepší známé faktorizační algoritmy patří kvadratické síto, číselné síto a algoritmy založené na teorii o eliptických křivkách.

Poznámka 3.3 (Důležitá). V kryptologii se omezujeme na faktorizaci čísel, která jsou součinem dvou různých náhodných (a ve většině případů přibližně stejně velkých) prvočísel. V celém následujícím textu budou p a q značit prvočísla a n bude značit součin dvou různých přibližně stejně velkých lichých prvočísel, $n = pq$.

Současný rekord ve faktorizaci čísla, které je součinem dvou přibližně stejně velkých prvočísel, pochází z května roku 2005, kdy bylo pomocí několika paralelně zapojených počítačů faktorizováno 200-ciferné (663-bitové) číslo. Tato faktorizace trvala 18 měsíců a předpokládá se, že na jednom počítači by trvala 55 let. Pro asymetrické šifry (viz. oddíl 3.3) využívající obtížnosti faktorizace je proto nejčastější doporučovanou délkou modulu n 1024 bitů, což odpovídá 308-cifernému číslu v dekadickém zápise.

3.1.2 Modulární odmocnina

Definice 3.4. *Číslo* v nazveme *kvadratickým reziduem modulo* n, jestliže existuje číslo $u \in \mathbb{Z}_n^*$, pro které platí $v = u^2 \bmod n$.

Množinu všech kvadratických reziduí modulo n označujeme QR_n. Lze snadno dokázat, že množina QR_n je multiplikativní Abelova grupa.

Definice 3.5. *Problém nalezení modulární odmocniny* je následující: Pro dané kvadratické reziduum v modulo n nalezněte $u \in \mathbb{Z}_n^*$ takové, že platí

$$v = u^2 \bmod n.$$

Problém nalezení druhé odmocniny modulo číslo n, které je součinem dvou prvočísel, $n = pq$, úzce souvisí s výše uvedeným problémem faktorizace čísla n; přesněji řečeno, je s ním ekvivalentní. Pojem „ekvivalentní“ zde znamená, že jsme-li schopni vyřešit jeden z těchto problémů se složitostí $\mathcal{O}(g)$, pak jsme schopni (s pomocí algoritmu na řešení prvního problému) se složitostí $\mathcal{O}(g)$ vyřešit i druhý z nich. Algoritmus na faktorizaci n pomocí algoritmu na hledání odmocniny modulo n najdeme v oddílu 6.3. Na počítání odmocniny z kvadratického rezidua modulo prvočíslo p existují rychlé algoritmy (nejlepší z nich má časovou náročnost $\mathcal{O}((\log p)^3)$); nalézt je můžeme např. v knize [7] na straně 100. Ve stejné knize na straně 102 nalezneme algoritmus na hledání druhé odmocniny modulo číslo n, známe-li jeho rozklad.

3.1.3 Problém diskrétního logaritmu

Definice 3.6. Nechť G je cyklická grupa řádu n. Nechť α je generátor grupy G a nechť $\beta \in G$. *Diskrétní logaritmus* β *o základu* α, označovaný $\log_\alpha \beta$, je přirozené číslo x, $0 \leq x \leq n-1$ takové, že $\beta = \alpha^x \bmod n$.

Definice 3.7. *Problém nalezení diskrétního logaritmu (DLP)* je následující: Pro dané prvočíslo p, generátor α grupy $\mathbb{Z}_p^*$ a prvek $\beta \in \mathbb{Z}_p^*$ nalezněte celé číslo x,

$$0 \leq x \leq p-2$$

takové, že platí $\alpha^x \equiv \beta \pmod p$.

Problém nalezení diskrétního logaritmu je, podobně jako předchozí dva problémy, považován za těžký. Více se o tomto problému můžeme dočíst např. opět v knize [7] na straně 103; najdeme v ní také další těžké matematické problémy.

Uvedené problémy souvisí s *jednosměrnými funkcemi* (je-li správný předpoklad o jejich výpočetní nezvládnutelnosti). Tuto nepřesnou formulaci objasníme v následujícím odstavci. Jednosměrná funkce $f : X \to Y$ je taková, že je snadné spočítat její hodnotu $f(x)$ pro libovolný vstup $x \in X$ (t.j. existuje polynomiální algoritmus, který má pro vstup x výstup $f(x)$), ale známe-li pouze hodnotu $y \in Y$, pak nejsme schopni v polynomiálním čase najít prvek $x \in X$ takový, že $f(x) = y$. Následuje formální definice obtížnosti invertovatelnosti funkce.

Definice 3.8. Říkáme, že funkce $f : X \to Y$ je *obtížně invertovatelná*, jestliže každý pravděpodobnostní polynomiální algoritmus (t.j. program pravděpodobnostního polynomiálního Turingova stroje (viz. definici 5.6)) na hledání inverze hodnoty $y = f(x)$, $x \in X$, může uspět pouze se zanedbatelnou (v $|y|$) pravděpodobností. Posloupnost $\{s_n\}_{n\in\mathbb{N}}$ se nazývá zanedbatelná v n, jestliže pro každý polynom $p(\cdot)$ a každé dostatečně velké n platí $s_n < 1/p(n)$.

Nyní objasníme souvislost jednosměrných funkcí s uvedenými těžkými problémy. V případě problému faktorizace čísla $n = pq$ je funkcí $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ násobení, které má polynomiální časovou složitost vzhledem k velikosti součinitelů p a q. Inverzní funkcí f^{-1} je nalezení prvočíselného rozkladu čísla n. V případě problému nalezení modulární odmocniny z čísla $v = u^2 \bmod n$ je funkcí $f : \mathbb{Z}_n^* \to QR_n$ umocnění prvku $u \in \mathbb{Z}_n^*$ na druhou modulo n a inverzní funkcí f^{-1} je při znalosti rezidua v nalezení u. Konečně v případě problému nalezení diskrétního logaritmu $x = \log_\alpha \beta$ v grupě $\mathbb{Z}_p^*$ je funkcí $f_\alpha : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ umocnění generátoru α grupy $\mathbb{Z}_p^*$ na x-tou modulo p a funkcí f_α^{-1} je při znalosti β nalezení čísla x.

3.2 Problémy z teorie složitosti

Zde uvedeme problém grafového izomorfismu.

Definice 3.9. Dva grafy $G_1 = (V_1, E_1)$ a $G_2 = (V_2, E_2)$ se nazývají *izomorfní*, pokud existuje bijekce π z množiny vrcholů V_1 do množiny vrcholů V_2 taková, že pro každé dva vrcholy $u, v \in V_1$ platí $(u, v) \in V_1$, právě když $(\pi(u), \pi(v)) \in V_2$.

Definice 3.10. *Problém grafového izomorfismu* je následující: Rozhodněte, zda jsou dva dané grafy G_1 a G_2 izomorfní.

Vygenerovat dva izomorfní grafy je snadné (stačí „přejmenovat“ vrcholy). O rozhodování, zda jsou dva dané grafy izomorfní se však předpokládá, že jde o těžký problém. V případě grafového izomorfismu nejde o jednosměrnou funkci. Proto jej nelze použít jako základ asymetrických šifer (viz. oddíl 3.3). Ale lze jej použít např. v identifikačních schématech (viz. kapitolu 6), kde dokazovatelovým tajemstvím je bijekce π z definice 3.9.

3.3 Prokazatelná bezpečnost

Šifry založené na obtížnosti matematických problémů, z nichž některé jsme zmínili v předchozí části, se nazývají *asymetrické* nebo *šifry s veřejným klíčem*. Zatímco

u symetrických šifer[1] musí odesílatel i příjemce své klíče chránit, aby je nezískala neoprávněná osoba, u asymetrických šifer se nechává v tajnosti pouze dešifrovací klíč a šifrovací klíč se zveřejňuje. Dešifrovacímu klíči se říká *soukromý* nebo *tajný klíč* a šifrovacímu klíči se říká *veřejný klíč*. Kdokoliv pak může zašifrovat zprávu příjemcovým veřejným klíčem a může si být (téměř) jistý, že si ji bude schopen přečíst pouze její oprávněný příjemce. Soukromý a veřejný klíč tvoří tzv. *klíčový pár*.

Při hodnocení bezpečnosti asymetrických schémat však nejde jen o neinvertovatelnost příslušné jednosměrné funkce. K některým šifrám totiž byly nalezeny útoky s menší časovou složitostí, než jakou má invertování dané jednosměrné funkce. Chceme-li tedy ukázat, že útok na šifru má složitost alespoň takovou jako výpočet inverze příslušné jednosměrné funkce, můžeme použít některou z metod tzv. *prokazatelné bezpečnosti (provable security)*. Jednou z nich je *redukcionistická bezpečnost (reductionist security)*. Budeme ji používat v následujícím tvaru[2]:

Definice 3.11 (Redukcionistická bezpečnost asymetrických šifrovacích schémat)**.** Kdokoliv, kdo je schopen v daném kryptosystému v polynomiálním čase k šifrovému textu c najít otevřený text m (pomocí algoritmu A), je schopen s využitím algoritmu A v polynomiálním čase vyřešit příslušný matematický problém.

Je ovšem třeba si uvědomit, že tato metoda zaručuje pouze odolnost vůči útokům určitého typu (který si v každém konkrétním případě pokusu o důkaz prokazatelné bezpečnosti musíme stanovit) a nebere v úvahu možné nedostatky fyzické implementace. Útoky na fyzickou implementaci kryptografických protokolů se nazývají *útoky pomocí postranních kanálů (side-channel attacks)*. Při nich se využívá např. výpočetní čas, elektromagnetické záření, vnucené chyby nebo chybová hlášení.

To, že je protokol redukcionisticky bezpečný, může paradoxně znamenat snadnou prolomitelnost systému. Např. Rabinův kryptosystém (viz. oddíl 4.4) lze takto prolomit pomocí tzv. *útoku s výběrem zprávy (chosen-ciphertext attack)*, který je rovněž uveden v oddílu 4.4.

3.4 Použitá tvrzení a algoritmy

Věta 3.12 (Čínská věta o zbytcích)**.** *Nechť $n_1, n_2, \ldots, n_k$ jsou kladná po dvou nesoudělná celá čísla a nechť $r_1, r_2, \ldots, r_k$ jsou libovolná celá čísla. Pak existuje právě jedno $N \pmod{n_1 n_2 \ldots n_k}$ takové, že*

$$N \equiv r_i \pmod{n_i}, \quad \forall i = 1, \ldots, k.$$

Důkaz nalezneme např. v knize [1] na straně 255. □

[1] Symetrické šifry jsou takové, že funkční vztah f algoritmu pro šifrování E_K a algoritmu pro dešifrování D_K, $D_K = f(E_K)$, i jeho inverze f^{-1} jsou snadno odvoditelné. U symetrických šifer se pro šifrování i dešifrování používá stejný klíč K.

[2] Idea redukce spočívá v tom, že můžeme ukázat, že ze složitosti jednoho problému $\mathcal{P}_1$ plyne složitost jiného problému $\mathcal{P}_2$ (nebo ekvivalentně z jednoduchosti problému $\mathcal{P}_2$ plyne jednoduchost problému $\mathcal{P}_1$) tak, že kdokoliv, kdo má algoritmus na řešení $\mathcal{P}_2$ (s časovou složitostí $\mathcal{O}(g)$) jej může použít k řešení $\mathcal{P}_1$ s relativně malým zvětšením úsilí (tedy opět s časovou složitostí $\mathcal{O}(g)$). V takovém případě říkáme, že se $\mathcal{P}_1$ redukuje na $\mathcal{P}_2$.

Následující jednoduchý důsledek čínské věty o zbytcích použijeme v oddílu 4.4 věnovaném Rabinovu kryptosystému.

Důsledek 3.13. *Nechť n je součin dvou různých lichých prvočísel, $n = pq$, a nechť $v \in QR_n$. Pak existují právě čtyři navzájem různé druhé odmocniny z v modulo n.*

Důkaz. Z předpokladu věty, že $v \in QR_n$, je zajištěna existence $u \in \mathbb{Z}_n^*$ takového, že $v = u^2 \bmod n$. Označme $v_p := v \bmod p$ a $v_q := v \bmod q$ a ukažme, že existují právě dvě různé opdmocniny z v_p modulo p a právě dvě různé odmocniny z v_q modulo q. Postačí, když se zaměříme na případ odmocniny z v_p modulo p; v případě odmocniny z v_q modulo q by byl postup obdobný. Využijeme známého faktu, že polynom stupně s nad tělesem T má v T nejvýše s kořenů. Na hledání kvadratického rezidua z v_p modulo p můžeme totiž nahlížet jako na hledání kořenů polynomu $u_p^2 - v_p$ nad tělesem $\mathbb{Z}_p$. Čísla $u_p, p - u_p$, kde $u_p := u \bmod p$, jsou zřejmě kořeny tohoto polynomu.

Kdyby platilo $u_p = p - u_p$, pak by bylo $p = 2u_p$, což by byl spor s lichostí prvočísla p.

Tím jsme dokázali existenci právě dvou odmocnin z v_p modulo p a právě dvou odmocnin z v_q modulo q (označme $u_q := u \bmod q$). Jejich „vzájemnou kombinací“ získáme čtyři čísla, která označíme u_1, u_2, u_3, u_4. Platí pro ně následující kongruence:

$$\begin{array}{ll} u_1 \equiv u_p \pmod p, & u_1 \equiv u_q \pmod q, \\ u_2 \equiv u_p \pmod p, & u_2 \equiv -u_q \pmod q, \\ u_3 \equiv -u_p \pmod p, & u_3 \equiv u_q \pmod q, \\ u_4 \equiv -u_p \pmod p, & u_4 \equiv -u_q \pmod q. \end{array} \tag{3.1}$$

Podle čínské věty o zbytcích (3.12) je každé z čísel u_1, u_2, u_3, u_4 jednoznačně určené modulo n. Zbývá ukázat, že čísla u_1, u_2, u_3, u_4 jsou navzájem různá modulo n a že se jedná o druhé odmocniny z u modulo n.

Pokud by některá z čísel u_1, u_2, u_3, u_4 byla shodná modulo n, byla by shodná také modulo p a modulo q, což zřejmě neplatí.

Nyní ukážeme, jakým způsobem lze spočítat čísla u_1, u_2, u_3, u_4 a že se jedná o druhé odmocniny z v modulo n.

Aby byly splněny vztahy (3.1), musí existovat celá čísla $c_1, d_1, x_1, y_1, \ldots, c_4, d_4, x_4, y_4$ taková, že

$$\begin{array}{l} u_1 = (u_p(1 - c_1 p) + u_q(1 - d_1 q)) \bmod n = (u_p x_1 q + u_q y_1 p) \bmod n, \\ u_2 = (u_p(1 - c_2 p) - u_q(1 - d_2 q)) \bmod n = (u_p x_2 q - u_q y_2 p) \bmod n, \\ u_3 = (-u_p(1 - c_3 p) + u_q(1 - d_3 q)) \bmod n = (-u_p x_3 q + u_q y_3 p) \bmod n, \\ u_4 = (-u_p(1 - c_4 p) - u_q(1 - d_4 q)) \bmod n = (-u_p x_4 q - u_q y_4 p) \bmod n. \end{array} \tag{3.2}$$

Tedy $c_i p + x_i q = 1$ a $y_i p + d_i q = 1$ pro všechna $i = 1, \ldots, 4$. Tato čísla získáme např. pomocí rozšířeného Euklidova algoritmu 3.21. Je tedy zřejmé, že můžeme položit

$$c_1 = c_2 = c_3 = c_4 = y_1 = y_2 = y_3 = y_4 =: c$$

a

$$d_1 = d_2 = d_3 = d_4 = x_1 = x_2 = x_3 = x_4 =: d.$$

Pak se vztahy (3.2) zjednoduší na

$$\begin{aligned} u_1 &= (u_p dq + u_q cp) \bmod n, \quad & u_2 &= (u_p dq - u_q cp) \bmod n, \\ u_3 &= (-u_p dq + u_q cp) \bmod n, \quad & u_4 &= (-u_p dq - u_q cp) \bmod n. \end{aligned} \tag{3.3}$$

Nyní ukážeme, že číslo u_1 je druhá odmocnina z v modulo n. Pro u_2, u_3 a u_4 by byl postup obdobný. Zřejmě existují celá čísla l_1, l_2, l_3 taková, že

$$u_1 = (u_p dq + u_q cp) \bmod n = (u + l_1 p)dq + (u + l_2 q)cp + l_3 pq.$$

Umocníme-li číslo u_1 na druhou modulo n, po snadných úpravách získáme rovnost

$$\begin{aligned} u_1^2 \bmod n = ((u + l_1 p)dq + (u + l_2 q)cp + l_3 pq) \bmod n = u^2(c^2p^2 + d^2q^2) \bmod n = \\ = u^2(1 - 2cdpq) \bmod n = u^2 \bmod n = v, \end{aligned}$$

kde jsme využili vztahu $cp + dq = 1$, z kterého plyne $1 = (cp + dq)^2 = c^2p^2 + +2cdpq + d^2q^2$. □

Poznámka 3.14. Předchozí důkaz dává návod, jak nálezt všechny čtyři druhé odmocniny z kvadratického rezidua modulo n, známe-li druhé odmocniny z v_p modulo p a z v_q modulo q (jak už je uvedeno výše, algoritmus na jejich hledání nalezneme v knize [7] na straně 100), a to pomocí vztahů (3.3). Algoritmus 3.22 je speciálním případem algoritmu odvozeného z tohoto důkazu.

Poznámka 3.15. Všimněme si, že v důsledku 3.13 předpokládáme $v \in QR_n$, z čehož plyne $u \in \mathbb{Z}_n^*$, a tedy $GCD(u, n) = 1$. Kdyby $u \in \mathbb{Z}_n \setminus \{0\}$ bylo soudělné s n, pak by bez újmy na obecnosti $GCD(u, n) = p$. Potom by z důkazu předchozího důsledku plynula existence pouze dvou různých druhých odmocnin z $v = u^2$ modulo n, neboť by bylo $u_p = 0$.

Definice 3.16. *Eulerova funkce* $\varphi : \mathbb{N} \to \mathbb{N}$ je zobrazení, které pro $n \in \mathbb{N}$ udává počet přirozených čísel menších než n, která jsou s n nesoudělná.

K výpočtu hodnoty Eulerovy funkce lze použít následující vztah:

$$\varphi(n) = n \prod_{p|n} (1 - \frac{1}{p}),$$

kde součin bereme přes všechna navzájem různá prvočísla, která dělí n.

Příklad 3.17. *Použitím předchozí formule spočtěme hodnotu Eulerovy funkce čísla 36:*

$$\varphi(36) = \varphi(2^2 3^2) = 36(1 - \frac{1}{2})(1 - \frac{1}{3}) = 12.$$

Věta 3.18 (Malá Fermatova věta)**.** *Nechť n je libovolné přirozené číslo a nechť a je přirozené číslo nesoudělné s n. Pak platí*

$$a^{\varphi(n)} \equiv 1 \ (mod\ n).$$

Důkaz této věty nalezneme např. v knize [1] na straně 254. □

Poznámka 3.19. Předchozí větě se také někdy říká Fermat-Eulerova věta.

Algoritmus 3.20 (Euklidův algoritmus)**.**
VSTUP: Dvě nezáporná celá čísla a a b, $a \geq b$.
VÝSTUP: Největší společný dělitel čísel a a b.
POSTUP:

1. Dokud $b > 0$, dělej následující:

   Polož $r := a \bmod b$, $a := b$, $b := r$.

2. Vrať a.

Algoritmus 3.21 (Rozšířený Euklidův algoritmus)**.**
VSTUP: Dvě nezáporná celá čísla a a b, $a \geq b$.
VÝSTUP: $d = GCD(a, b)$ a čísla x, $y \in \mathbb{Z}$ splňující $ax + by = d$.
POSTUP:

1. Polož $x_1 := 0,\ x_2 := 1,\ y_1 := 1,\ y_2 := 0$.

2. Dokud $b > 0$, dělej následující:

   (a) Polož $q := a \text{ div } b$, $r := a - qb$, $x := x_2 - qx_1$, $y := y_2 - qy_1$.

   (b) Polož $a := b$, $b := r$, $x_2 := x_1$, $x_1 := x$, $y_2 := y_1$, $y_1 := y$.

3. Polož $d := a$, $x := x_2$, $y := y_2$ a vrať (d, x, y).

Algoritmus 3.22 (Výpočet druhých odmocnin modulo $n = pq$, kde p a q jsou prvočísla taková, že $p \equiv 3 \pmod 4$ a $q \equiv 3 \pmod 4$)**.**
VSTUP: Přirozené číslo $c \in \mathbb{Z}_n$.
VÝSTUP: Čtyři druhé odmocniny z c modulo n.
POSTUP:

1. Použitím rozšířeného Euklidova algortimu (3.21) najdi čísla $a,\ b \in \mathbb{Z}$, pro která platí $ap + bq = 1$.

2. Polož $r := c^{(p+1)/4} \bmod p$.

3. Polož $s := c^{(q+1)/4} \bmod q$.

4. Polož $x := (aps + bqr) \bmod n$.

5. Polož $y := (aps - bqr) \bmod n$.

6. Vrať $(\pm x \bmod n, \pm y \bmod n)$.

Poznámka 3.23. Algoritmy 3.20 a 3.21 mají časovou složitost $\mathcal{O}(\log^2(max\{a, b\}))$, algoritmus 3.22 má časovou složitost $\mathcal{O}(\log^3(max\{a, b\}))$.

Kapitola 4

Stavební prvky protokolů

U symetrických šifer se pro šifrování i dešifrování používá jeden klíč K. Šifrovací transformaci u symetrické šifry proto označujeme $E_K(.)$ a dešifrovací transformaci $D_K(.)$. U asymetrických šifer se používá klíčový pár — veřejný klíč K_1 a soukromý klíč K_2. Šifrovací transformaci u asymterické šifry označujeme $E_{K_1}(.)$ a dešifrovací transformaci $D_{K_2}(.)$.

4.1 Hašovací funkce

Definice 4.1. *Hašovací funkce (hash function)* $H : M \to \{0,1\}^k$ (kde $k \in \mathbb{N}$ je pevně dané) je funkce, která pro libovolný vstup $m \in M$ vrací výstup h zvaný *hodnota haše (hash value)* (nebo krátce *haš*), t.j. $h = H(m)$, splňující následující podmínky:

1. pro libvolné $m \in M$ je snadné spočítat $H(m)$,
2. pro dané h je výpočetně náročné najít m takové, že $h = H(m)$,
3. je výpočetně náročné najít dvě zprávy m a m' takové, že

$$H(m) = H(m').$$

Vlastnost 2. z předchozí definice se nazývá *jednosměrnost* a vlastnosti 3. říkáme *bezkoliznost*[1].

Úlohou hašovací funkce je vytvořit jakýsi otisk zprávy. V kryptografii se nejčastěji používá u digitálních podpisů a k zajištění integrity zpráv. Zajištění integrity zprávy znamená, že pokud s šifrovou zprávou $c = E_{k_1}(m)$ pošleme příjemci také haš $h = H(m)$, pak v případě, že nějaký útočník zprávu c modifikuje na zprávu c', příjemce tuto skutečnost pozná, pokud si po dešifrování $D_{k_2}(c') = m'$ spočítá $H(m') = h' \neq h$.

[1]Rozlišujeme dva druhy kolizí. Zmíněná kolize se nazývá *kolize prvního řádu*. Schopnost hledat kolize druhého řádu znamená, že k libovolné dané zprávě m jsme schopni nalézt zprávu m' takovou, že $H(m) = H(m')$.

4.2 Digitální podpis

Definice 4.2. *Digitální podpisové schéma (digital signature scheme)* je schéma používající metody asymetrické kryptografie určené k autentizaci zprávy. Jedná se o dva komplementární algoritmy — jeden pro podpis a jeden pro ověření jeho pravosti. Podepisovací algoritmus (ve kterém se používá soukromý klíč) budeme značit **Sig** a ověřovací **Ver** (v něm se používá veřejný klíč). Digitální podpis musí mít následující vlastnosti:

- *Autentizace (authenticity)*: Lze prokázat, že dokument skutečně podepsal jeho autor.
- *Integrita (integrity)*: Zpráva nemůže být změněná se zachováním platnosti podpisu.
- *Nepopiratelnost (non-repudiation)*: Osoba, která dokument podepsala, to nemůže později popřít.

Digitálním podpisem nazýváme výslednou posloupnost bitů.

Digitální podpisové schéma se většinou používá ve spojení s hašovací funkcí[2]. Důvodem je to, že výpočet podpisu pro rozsáhlé dokumenty trvá (oproti výpočtu haše) příliš dlouho. Algoritmus **Sig** digitálního podpisu se aplikuje na výslednou haš.

Výstup ověřovacího algoritmu pro podpis $s = \mathbf{Sig}(m)$ zprávy m označíme $\mathbf{Ver}(s, m) = 1$, je-li podpis v pořádku, a $\mathbf{Ver}(s, m) = 0$, jde-li o podvržený podpis.

4.3 RSA

RSA je asymetrická šifra. Popis šifry:

Zvolíme náhodně dvě různá přibližně stejně velká prvočísla p a q. Položíme $n := pq$, zvolíme celé e takové, že $1 < e < \varphi(n)$ a $GCD(e, \varphi(n)) = 1$, a pomocí rozšířeného Euklidova algoritmu (3.21) spočteme $d = e^{-1} \bmod \varphi(n)$. GCD značí funkci „největší společný dělitel“ a φ je Eulerova funkce (viz. definici (3.16)), $\varphi(n) = (p-1)(q-1)$. Veřejným klíčem je pár (n, e) a soukromým klíčem je číslo d (případně trojice (p, q, d)).

Šifrování je umocňování otevřené zprávy $m \in \mathbb{Z}_n$ na veřejný exponent e modulo n:

$$c = m^e \bmod n.$$

Dešifrování je umocňování šifrového textu c na soukromý exponent d modulo n:

$$m = c^d \bmod n.$$

Ukážeme platnost poslední rovnosti:

$$c^d \bmod n = (m^e \bmod n)^d \bmod n = m^{ed} \bmod n = m^{1 \bmod \varphi(n)} \bmod n =$$
$$= m^{k\cdot\varphi(n)+1} \bmod n = m \cdot m^{k\cdot\varphi(n)} \bmod n \overset{\text{(V3.18)}}{=} m \cdot 1 \bmod n = m.$$

[2]Existuje i tzv. *schéma s obnovou zprávy* (neboli RSA podepisovací schéma), kde se podepisuje celý dokument, ale to nebudeme uvažovat

Máme-li dáno n, y a e a hledáme x takové, že $y = x^e \bmod n$, říkáme, že řešíme tzv. *RSA problém*. Předpokládá se, že RSA problém je ekvivalentní problému faktorizace modulu n, a tedy že jde o těžký problém.

4.4 Rabinův kryptosystém

Rabinův kryptosystém je podobný systému RSA. Jediný rozdíl je ten, že namísto čísla e nesoudělného s $\varphi(n)$ je veřejným exponentem číslo 2.

Přesněji, zvolíme náhodně dvě různá přibližně stejně velká prvočísla p a q, $p \equiv 3 \pmod 4$, $q \equiv 3 \pmod 4$, a položíme $n := pq$. Veřejným klíčem je modulus n a soukromým klíčem je dvojice (p, q). Při šifrování umocňujeme otevřený text $m \in \mathbb{Z}_n$ na druhou modulo n,

$$c = m^2 \bmod n. \tag{4.1}$$

Při dešifrování najdeme pomocí algoritmu 3.22 všechny čtyři druhé odmocniny z c modulo n (viz. důsledek 3.13). Ve velmi málo pravděpodobném případě, že by $GCD(m, n) > 1$, by existovaly pouze dvě druhé odmocniny z c modulo n (viz. poznámku 3.15). Takto získané zprávy označíme m_1, m_2, m_3 a m_4. Jako zprávu m vybereme tu, která dává smysl. Jednoznačnosti otevřeného textu můžeme docílit také tím, že před zašifrováním připojíme na konec zprávy m nějaký konstantní řetězec znaků. Po dešifrování potom vybereme tu zprávu m_i, která končí tímto řetězcem.

Zatímco o RSA problému se pouze předpokládá, že je ekvivalentní faktorizaci modulu, o invertování Rabinovy funkce (4.1) to lze dokázat.

Tvrzení 4.3. *Jsme-li schopni faktorizovat modulus n v polynomiálním čase, jsme schopni (pomocí algoritmu na faktorizaci modulu n) v polynomiálním čase invertovat Rabinovu funkci, a naopak, jsme-li schopni v polynomiálním čase invertovat Rabinovu funkci, jsme schopni (pomocí algoritmu na invertování Rabinovy funkce) v polynomiálním čase faktorizovat modulus n.*

Důkaz. Způsob, jakým lze při znalosti faktorů p, q modulu n invertovat Rabinovu funkci, dává např. algoritmus 3.22, který běží v čase $\mathcal{O}(\log^3(.))$. Zbývá dokázat druhou implikaci, totiž že kdokoliv, kdo je schopen z šifrového textu c najít otevřený text m v polynomiálním čase, je také schopen v polynomiálním čase faktorizovat n.

Invertovat Rabinovu funkci znamená nalézt některou ze čtyř odmocnin z c modulo n. Jsou to $\pm m \pmod n$, $\pm \alpha m \pmod n$, kde $\alpha \equiv 1 \pmod p$ a $\alpha \equiv -1 \pmod q$. Důvodem platnosti těchto kongruencí je to, že

$$c = m^2 \bmod n \qquad \text{a} \qquad c = \alpha^2 m^2 \bmod n,$$

a tedy

$$m^2 \equiv \alpha^2 m^2 \pmod n,$$

z čehož plyne

$$\alpha^2 \equiv 1 \pmod n \quad \Rightarrow \quad \alpha^2 - 1 \equiv 0 \pmod n.$$

Tedy existuje nějaké $\gamma \in \mathbb{Z}$ takové, že

$$(\alpha - 1)(\alpha + 1) = \gamma pq.$$

Nyní mohou nastat dva případy:

1. Buď $\alpha - 1 = \sigma p$ a $\alpha + 1 = \varrho q$, kde $\gamma = \sigma\varrho$, $\sigma, \varrho \in \mathbb{Z}$, a tedy platí $\alpha \equiv 1 \pmod{p}$ a $\alpha \equiv -1 \pmod{q}$,

2. nebo $\alpha - 1 = \tau pq$ (resp. $\alpha + 1 = \tau pq$), kde τ dělí γ, což znamená, že $\alpha \equiv 1 \pmod{n}$ (resp. $\alpha \equiv -1 \pmod{n}$), ale odsud plyne, že existují pouze dvě druhé odmocniny z kvadratického rezidua x modulo n, což je ve sporu s důsledkem 3.13.

S pravděpodobností 1/2 bude inverzí Rabinovy funkce $\pm\alpha m \pmod{n}$. V tom případě vydělením spočítáme $\pm\alpha \bmod n$ a pak rychle (se složitostí $\mathcal{O}(\log^2(.))$) faktorizujeme n pomocí Euklidova algoritmu (3.20), neboť $GCD(n, \alpha - 1) = p$. Pokud je inverzí Rabinovy funkce $\pm m$, provedeme tento postup s jiným m. Pravděpodobnost, že po k krocích se nám nepodaří faktorizovat n je 2^{-k}, což se limitně blíží k nule. □

Nyní dokážeme redukcionistickou bezpečnost (viz. oddíl 3.3) Rabinova kryptosystému.

Tvrzení 4.4 (Redukcionistická bezpečnost Rabinova ryptosystému)**.** *Kdokoliv, kdo je schopen v polynomiálním čase (vzhledem k velikosti modulu n) k šifrovému textu c najít otevřený text m ($c = m^2 \bmod n$), je schopen (pomocí algoritmu na hledání otevřeného textu m) v polynomiálním čase faktorizovat n.*

Důkaz. Ten, kdo je schopen najít otevřený text m k šifrovému textu c, musí být schopen najít všechny čtyři odmocniny z c modulo n, protože kterákoliv z nich může být hledaným otevřeným textem m. To znamená, že zná příslušné α (viz. důkaz předchozího tvrzení), a tedy je schopen faktorizovat n v čase $\mathcal{O}(\log^2(n))$. □

Tím je tedy dokázaná redukcionistická bezpečnost Rabinova systému. Jak jsme ale uvedli v oddíle 3.3, plyne odsud jeho zranitelnost útokem s výběrem zprávy. V článku [5] je uveden následující útok:

Předpokládejme, že by útočník (Eva) nějakým způsobem přiměl vlastníka soukromého klíče (p, q) (Alici), aby pro něj dešifroval nějakou šifrovou zprávu, kterou si útočník vybere. Eva si vybere zprávu $m \in \mathbb{Z}_n$ a pošle Alici šifrový text $c = m^2 \bmod n$. Alice pošle Evě (nějakou[3]) druhou odmocninu z c. S pravděpodobností 1/2 to bude $\pm\alpha m$. V takovém případě Eva snadno faktorizuje n (viz. důkaz předchozího tvrzení) a bude schopná dešifrovat všechny zprávy v tomto systému. Pokud jí Alice pošle $\pm m$, pak Eva provede stejný útok s jiným m. Po k pokusech je pravděpodobnost úspěšného útoku $1 - 2^{-k}$.

Jak může Eva docílit toho, aby jí Alice poslala dešifrované zprávy? Jedním z takových případů může být zhroucení systému, kdy se Eva jeví jako oprávněný uživatel, který se snaží obnovit ztracená data.

Je tedy vidět, že stejná vlastnost, která dává Rabinovu systému redukcionistickou bezpečnost, vede k jeho totálnímu kolapsu při konfrontaci s jiným typem

[3]Zde je třeba si uvědomit, že m není smysluplný text, ale (většinou binární) číslo, které vznikne zakódováním daného otevřeného textu. Proto Alice nepošle Evě nutně číslo m. Dekódování získaných odmocnin z šifrového textu by totiž pro ni bylo zbytečně časově náročné.

útočníka. V praxi je ovšem takový útok obtížně realizovatelný. Ze strany Alice by vyžadoval velkou míru spolupráce a hlouposti.

Kapitola 5

Turingovy stroje

Definice týkající se Turingových strojů v této a příští kapitole jsou převzaty z textů [2] a [4].

Základní model *(jednopáskového) Turingova stroje (TM)* se skládá z *hlavy* a oboustranně nekonečné *pásky* rozdělené na políčka. Políčka mohou být prázdná, nebo nést symboly z nějaké konečné množiny. Hlava může číst a přepisovat obsah políčka, na kterém se nachází a pohybovat se po pásce vlevo a vpravo. Dále Turingův stroj obsahuje vnitřní paměť, která může nabývat konečného počtu *stavů*.

Na začátku výpočtu je páska prázdná až na konečnou souvislou posloupnost políček. Jejich obsahu říkáme *vstup*.

Matematicky je Turingův stroj definován takto:

Definice 5.1. *Turingův stroj* je čtveřice $T = (\Sigma, Q, P, q_s)$, kde

- Σ je neprázdná konečná množina symbolů,
- Q je neprázdná konečná množina stavů,
- P je libovolná množina $P \subseteq Q \times \Sigma \times Q \times \Sigma \times \mathcal{M}$ zvaná *program* Turingova stroje. $\mathcal{M}$ je množina pohybů $\{\leftarrow, \rightarrow, -\}$, kde $\leftarrow$ znamená pohyb o jedno políčko doleva, $\rightarrow$ o jedno políčko doprava a $-$ znamená, že hlava zůstane na aktuálním políčku,
- $q_s \in Q$ je *počáteční stav* stroje.

Libovolný prvek $(q, s, q', s', m) \in P$ se nazývá *instrukce*.

Instrukci Turingova stroje můžeme interpretovat takto: Pokud jsi ve stavu q a čteš symbol s, změň ho na symbol s', přejdi do stavu q' a proveď pohyb m. Má-li stroj pro každý vnitřní stav (kombinaci stavu q a symbolu s) jedinou instrukci, nazývá se *deterministický*. V opačném případě se nazývá *nedeterministický*.

Definice 5.2. Množinu symbolů Σ nazýváme *abeceda*. Množinu konečných posloupností symbolů z abecedy Σ značíme Σ^*. Prvky množiny Σ^* nazýváme *slova*. Libovolnou množinu slov nazýváme *jazyk*.

Pro teorii interaktivních důkazů budeme potřebovat také definici vícepáskového Turingova stroje. Vícepáskový Turingův stroj je podobný jednopáskovému, ale má k nezávislých pásek, kde k je konstanta. Každá z pásek má svoji vlastní čtecí a zapisovací hlavu. Instrukce stroje jsou závislé na všech k symbolech čtených hlavami na jednotlivých páskách. k-páskový TM má však pouze jeden stav. Protože množina stavů Q může být libovolná, můžeme ji chápat například jako $Q = Q_1 \times Q_2 \times \cdots \times Q_k$. Vícepáskové Turingovy stroje lze nahradit jednopáskovými. Důvod používání vícepáskových TM je zejména ten, že oproti jednopáskovým TM umožňují přehlednější a rychlejší výpočet.

Formálně definujeme k-páskový Turingův stroj takto:

Definice 5.3. *k-páskový Turingův stroj* je čtveřice $T = (\Sigma, Q, P, q_s)$, kde

- Σ je neprázdná konečná množina symbolů,
- Q je neprázdná konečná množina stavů,
- program P je libovolná množina $P \subseteq Q \times \Sigma^k \times Q \times \Sigma^k \times \mathcal{M}^k$, kde $\mathcal{M}$ je množina pohybů $\{\leftarrow, \rightarrow, -\}$,
- $q_s \in Q$ je *počáteční stav* stroje.

Neobejdeme se ani bez definice interaktivního Turingova stroje:

Definice 5.4 (Interaktivní Turingův stroj)**.** *Interaktivním Turingovým strojem (ITM)* nazýváme deterministický sedmipáskový Turingův stroj s následujícími páskami:

- jedna *veřejná vstupní* páska pouze ke čtení,
- jedna *soukromá vstupní* páska pouze ke čtení,
- jedna *pracovní* páska pro čtení i zápis,
- jedna *výstupní* páska pouze pro zápis,
- dvojice *komunikačních* pásek — jedna pouze pro zápis (tu nazýváme *výstupní komunikační* páska) a druhá pouze ke čtení (tu nazýváme *vstupní komunikační* páska),
- a jedna *stavová* páska pro čtení i zápis sestávají z jediného políčka, které může obsahovat 0 nebo 1 a na začátku obsahuje 0.

Každému ITM je přiřazen samostatný bit $\sigma \in \{0, 1\}$ nazývaný jeho *identitou*. O ITM řekneme, že je *aktivní*, pokud je obsah jeho stavové pásky roven jeho identitě. V opačném případě říkáme, že je *nečinný*. Je-li stroj nečinný, nemění se jeho stav, polohy hlav na jednotlivých páskách, ani obsah jeho přepisovatelných pásek.

Obsah veřejné vstupní pásky se nazývá *veřejný vstup*, obsah soukromé vstupní pásky se nazývá *náhodný vstup* a obsahu výstupní pásky po skončení výpočtu stroje říkáme *výstup*. Obsah napsaný na přepisovatelnou komunikační pásku během časového úseku, kdy je stroj aktivní, se nazývá *zpráva poslaná* v tomto časovém úseku. Obsah přečtený ze čtecí komunikační pásky během aktivního úseku se nazývá *zpráva přijatá* v tomto časovém úseku.

Při výpočtu nikdy neuvažujeme jediný interaktivní stroj, ale dvojici ITM, které spolu sdílejí některé pásky. Zpráva poslaná jedním strojem je přijatou zprávou druhého stroje. Aktivní stroj se stane nečinným přepsáním obsahu stavové pásky a tím se druhý stroj (s opačnou identitou) stane aktivním. Výpočet takové dvojice strojů sestává ze zpráv, které stroje střídavě posílají jeden druhému v závislosti na jejich počátečním (veřejném) vstupu, náhodných vstupech a na dosavadních přijatých zprávách.

Definice 5.5 (Spojený výpočet dvou ITM)**.** Říkáme, že dva interaktivní stroje jsou *spojené*, pokud

- mají opačnou identitu,
- sdílejí veřejnou vstupní pásku,
- sdílejí stavovou pásku,
- vstupní komunikační páska jednoho stroje je výstupní komunikační páskou druhého stroje a naopak.

Zdůrazněme, že zbylé pásky těchto strojů (t.j. soukromá vstupní, pracovní a výstupní páska) jsou různé.

Spojený výpočet spojené dvojice ITM na veřejném vstupu x je posloupnost párů lokálních konfigurací každého ze strojů. V každém páru lokální konfigurace je jeden stroj aktivní a druhý nečinný.

Jak už bylo uvedeno, když aktivní stroj dokončí svůj dílčí výpočet, přepíše obsah stavové pásky na opačnou hodnotu. Tím se stane nečinným a druhý stroj se stane aktivním. Jestliže jeden ze strojů dokončí svůj výpočet, když je obsah stavové pásky roven jeho identitě, pak říkáme, že oba stroje dokončily výpočet.

Definice 5.6. *Pravděpodobnostní Turingův stroj* je nedeterministický Turingův stroj spolu s funkcí $\mathtt{P} : (Q \times \Sigma \times Q \times \Sigma \times M) \to [0,1]$ takovou, že pro každé $(q,s) \in Q \times \Sigma$ platí

$$\sum_{(q',s',m) \in Q \times \Sigma \times M} \mathtt{P}(q,s,q',s',m) = 1,$$

kde Q je množina stavů, Σ množina symbolů a M množina pohybů.

Předpokládejme, že všechny možné interakce spojených strojů A a B na každém veřejném vstupu skončí po konečném počtu kroků. Symbolem $(A,B)(x)$ označujeme výstup stroje B při interakci se strojem A na veřejném vstupu x, kde náhodný vstup na soukromé vstupní pásce každého ze strojů je vybrán rovnoměrně a nezávisle z množiny všech nekonečných posloupností bitů. Z této nekonečné posloupnosti je při (konečném) výpočtu přečten (a má význam) pouze konečný prefix. Význam soukromé vstupní pásky můžeme chápat tak, že daný stroj má pro každou dvojici $(q,s) \in Q \times \Sigma$ dvě možné instrukce (q,s,q'_1,s'_1,m_1) a (q,s,q'_2,s'_2,m_2) a každou z nich provede s pravděpodobností $1/2$ (v závislosti na bitu čteném ze soukromé vstupní pásky). Interaktivní Turingův stroj je tedy pravděpodobnostní.

Definice 5.7 (Složitost výpočtu interaktivního stroje)**.** Říkáme, že interaktivní *stroj B má časovou složitost $t : \mathbb{N} \to \mathbb{N}$*, jestliže pro každý interaktivní stroj A a každý vstup x platí, že stroj B při interakci se strojem A na vstupu x vždy (t.j. nezávisle na obsahu jeho soukromé vstupní pásky a soukromé vstupní pásky stroje A) skončí výpočet během $t(|x|)$ kroků.

Zdůrazněme, že právě definovaná časová složitost je funkcí pouze vstupu x a je tedy nezávislá na zprávách, které stroj B přijímá.

Kapitola 6

Interaktivní důkazy

Interaktivní důkaz zahrnuje dvě výpočetní úlohy — tvorbu důkazu a ověřování jeho správnosti. Tyto úlohy vykonávají dva různí účastníci zvaní *dokazovatel (prover)* a *ověřovatel (verifier)*. Těmito účstníky jsou dva spojené interaktivní Turingovy stroje. Dokazovatele označujeme P a ověřovatele V. Jejich interakce je parametrizovaná jejich společným veřejným vstupem, který reprezentuje dokazované tvrzení.

V následující definici je výstupem ověřovatele jeho rozhodnutí, zda důkaz přijme, nebo ne. Výstup 1 znamená přijetí důkazu a výstup 0 znamená jeho odmítnutí.

Definice 6.1. Dvojici spojených interaktivních Turingových strojů (P, V) říkáme *interaktivní důkazový systém jazyka* L, jestliže jsou splněny následující tři podmínky:

- **Efektivita:**

  stroj V pracuje v polymoniálním čase (vzhledem k délce vstupu x),

- **Úplnost:**
$$\mathtt{P}[(P, V)(x) = 1 \mid x \in L] \geq \varepsilon,$$

- **Spolehlivost:** pro každý interaktivní stroj P^* platí
$$\mathtt{P}[(P^*, V)(x) = 1 \mid x \notin L] \leq \delta,$$

kde ε a δ jsou konstanty splňující

$$\varepsilon \in (\frac{1}{2}, 1], \qquad \delta \in [0, \frac{1}{2}).$$

Poznámka 6.2. Všimněme si, že v definici (6.1) neklademe žádné omezující podmínky na časovou složitost výpočtu stroje P. Dále zdůrazněme, že podmínka pro spolehlivost se vztahuje ke všem možným dokazovatelům P^*, zatímco podmínka pro úplnost pouze k danému dokazovateli P.

Poznámka 6.3. Pro účely protokolů pro interaktivní důkazy přidáme každému ITM jednu soukromou *pomocnou vstupní pásku* pouze ke čtení. Obsah každé z nich bude souviset s vnitřní konfigurací příslušného stroje před začátkem běhu protokolu. Dokazovateli tato páska umožní efektivní implementaci jeho úkolu.

Pod pojmem jazyk si můžeme představit např. grafový izomorfismus. Vrátíme-li se k definici jazyka (5.2), pak v tomto případě bude abecedou grafy, slovy budou dvojice izomorfních grafů a veřejným vstupem bude dvojice grafů, o kterých chceme rozhodnout, zda jsou izomorfní.

Dále rozebereme několik protokolů pro interaktivní důkazy z hlediska definice (6.1) a z hlediska dokazatelné bezpečnosti (viz. oddíl 3.3). V protokolech se dokazovateli zpravidla říká Peggy a ověřovateli Viktor. Toho se budeme držet i my.

Účastníky protokolu pro interaktivní důkaz mohou být dvě osoby, osoba a stroj nebo dva stroje. Typicky jsou od sebe fyzicky vzdáleni, takže výpočet na sdílených páskách spojených interaktivních strojů z definice (5.5) musíme chápat tak, že si vzájemně posílají výsledky jednotlivých výpočtů. Veřejným vstupem x bývá v protokolech číslo (nebo množina čísel), o kterém dokazovatel tvrdí, že pro ně zná řešení nějakého těžkého matematického problému, z nichž některé jsme uvedli v kapitole 3. Tomuto řešení říkáme *dokazovatelovo tajemství*. Jazykem L bývá množina všech čísel, pro které dokazovatel zná řešení příslušného problému. Dokazovatel tedy dokazuje, že pro veřejný vstup x zná řešení nějakého těžkého matematického problému; nemusí to ovšem znamenat, že má algoritmus na řešení tohoto problému. Například pokud je protokol založen na obtížnosti hledání diskrétního logaritmu v grupě $\mathbb{Z}_p^*$ s generátorem α, pak dokazovatel dokazuje, že pro konkrétní veřejný vstup $x \in \mathbb{Z}_p^*$ zná diskrétní logaritmus $\log_\alpha x$ (ne však, že tento logaritmus umí obecně spočítat pro libovolné x), tedy že x patří do jazyka, jímž je množina všech čísel z grupy $\mathbb{Z}_p^*$, pro něž dokazovatel zná diskrétní logaritmus. Dokazovatel získá veřejné x tak, že si zvolí $y \in \{0, 1, \ldots, p-2\}$ a umocní generátor α na y-tou modulo p; $x := \alpha^y \bmod p$. Ověřovateli potom poskytne x a snaží se mu dokázat, že zná diskrétní logaritmus $y = \log_\alpha x$. Dokazovatel si může zvolit libovolný počet (který označíme k) čísel $y_i \in \{0, 1, \ldots, p-2\}$, $i = 1, 2, \ldots, k$, a na každé z nich umocnit generátor α; označme $x_i := \alpha^{y_i} \bmod p$, $i = 1, 2, \ldots, k$. Ke každému takto získanému x_i dokazovatel zná jeho diskrétní logaritmus y_i — tato x_i tedy tvoří jazyk L.

Interaktivní protokoly jsou obvykle formy *výzva-opověď (challenge-response)*, což znamená, že oveřovatel posílá dokazovateli výzvy a ten na ně reaguje odpověďmi.

Komunikace účastníků (posílání zpráv) probíhá pomocí veřejného komunikačního kanálu. Předpokládáme, že tento kanál je nezabezpečený, což znamená, že případný útočník může zachytit (odposlechnout) posílané zprávy, číst si je a případně je i pozměňovat. Těmto útokům nelze zabránit. Pomocí různých kryptografických metod lze však snížit jejich úspěšnost.

Dalším problémem je to, že se Viktor může z komunikace s Peggy dozvědět nějakou netriviální informaci o jejím tajemství — něco navíc, co nepotřebuje ke svému přesvědčení, že Peggy skutečně zná dokazovatelovo tajemství. Tento problém řeší protokoly s nulovou znalostí (viz. definici (6.4)).

Také dokazovatel může chtít získat nějakou výhodu. Například se může snažit o to, aby s co největší pravděpodobností bylo výsledkem protokolu $(P, V)(x) = 1$, i kdyby ve skutečnosti $x \notin L$. Takový dokazovatel se nazývá *podvádějící dokazovatel (cheating prover)* a označuje se $\widetilde{P}$; v protokolech mu budeme říkat Eva. Ověřovateli, který se snaží získat nějaké netriviální informace o dokazovatelově tajemství, říkáme *nečestný ověřovatel (dishonest verifier)* a označujeme jej $\widetilde{V}$. Dokazovatel, resp. ověřovatel, který dodržuje chování specifikované v protokolu,

se nazývá *čestný (honest) dokazovatel*, resp. *čestný ověřovatel*.

Každý dokazovatel i ověřovatel, ať už čestný, nebo ne, musí dodržovat syntax komunikačního rozhraní, protože její nedodržování by bylo bezprostředně odhaleno druhou stranou. Mohou tedy podvádět pouze ve svých soukromých výpočtech a v tom, jaká data posílají.

Další užitečnou vlastností interaktivních důkazů (která ovšem není podmínkou pro interaktivnost důkazu) je **nulová znalost (zero-knowledge)**, což v podstatě znamená, že cokoliv, co je ověřovatel schopen efektivně spočítat po proběhnutí protokolu, byl schopen efektivně spočítat i před ním.

Pravděpodobnostní polynomiální Turingův stroj je pravděpodobnostní Turingův stroj (viz. definici 5.6), který vždy skončí svůj výpočet po polynomiálním počtu kroků (vzhledem k délce vstupu).

Definice 6.4. Interaktivní důkazový systém (P, V) jazyka L se nazývá *důkaz s dokonale nulovou znalostí (perfect zero-knowledge proof)* pro jazyk L, jestliže pro každý interaktivní pravděpodobnostní polynomiální Turingův stroj V^* existuje (obyčejný) pravděpodobnostní polynomiální Turingův stroj M^* takový, že pro každé $x \in L$ platí:

- S pravděpodobností nejvýše $1/2$ bude výstupem stroje M^* speciální symbol $\perp$. V tom případě řekneme, že výpočet stroje M^* selhal.
- Platí
$$\{M^*(x) \mid M^*(x) \neq \perp\}_{x \in L} \sim \{(P, V^*)(x)\}_{x \in L},$$
kde $M^*(x)$ je výstup stroje M^* na vstupu x, $(P, V^*)(x)$ je výstup stroje V^* po interakci se strojem P na vstupu x. Symbol $\sim$ označuje stejné rozdělení náhodných veličin.

Stroj M^* se nazývá *dokonalý simulátor* pro interakci stroje V^* se strojem P.

Předchozí definice říká, že pro *každý* stroj V^* (nejen pro V) interagující se strojem P existuje dokonalý simulátor M^*. Tento simulátor dokáže simulovat interakci V^* s P, i když nemá přístup ke stroji P. Z toho je vidět, že stroj V^* nezíská od stroje P žádnou informaci o dokazovatelově tajemství (až na to, že $x \in L$), protože stejný výstup může být vygenerován bez přístupu k P. Uvědomme si, že zde již není výstupem ověřovatele pouze jeho rozhodnutí o přijetí nebo nepřijetí důkazu. Výstupem (stroje V^* po interakci se strojem P a stroje M^*) může být například celá komunikace (tedy odeslané a přijaté zprávy) strojů P a V^*.

Zdůrazněme, že stroj M^* dokáže simulovat interakci strojů P a V^* pouze na vstupech $x \in L$ (na vstupech $x \notin L$ se může chovat libovolně).

Nejběžnější použití interaktivních důkazových systémů je v identifikačních schématech k prokázání dokazovatelovy identity, neboli *autentizaci*. Předpokládáme totiž, že pouze dokazovatel zná dokazovatelovo tajemství.

Nejjednodušší identifikační schéma je následující: Peggy prokazuje svou identitu tajným heslem. Jediná zpráva v tomto protokolu je toto heslo, které Peggy pošle Viktorovi. Viktor má uloženou haš Peggyina hesla (pokud by měl heslo v otevřené podobě, mohl by je použít k vydávání se za Peggy), kterou srovná s haší doručeného hesla a přijme Peggyinu identitu, právě když se tyto dvě haše shodují. Toto schéma zřejmě splňuje požadavky pro interaktivní důkaz,

ale má tu nevýhodu, že kdokoliv, kdo během komunikace získal Peggyino heslo (odposlechem nebo jako oprávněný příjemce), je může použít a vydávat se za Peggy.

6.1 Identifikace založená na kryptografii s veřejným klíčem

Označme Peggyin tajný klíč sk a veřejný klíč pk. Zde je veřejným vstupem klíč pk, jazykem „množina všech veřejných klíčů příslušných k Peggyinu soukromému klíči“ a dokazovatelovým tajemstvím klíč sk. Peggy prokazuje svoji identitu následovně:

1. Viktor náhodně vybere zprávu m, zašifruje ji veřejným klíčem pk a šifrový text (výzvu) $c = E_{pk}(m)$ pošle Peggy.

2. Peggy dešifruje c tajným klíčem sk a výsledek (odpověď) $m' = D_{sk}(c)$ pošle Viktorovi.

3. Viktor přijme Peggyinu identitu, právě když $m = m'$.

Tento protokol je efektivní, protože veškerý Viktorův výpočet spočívá ve vygenerování a zašifrování zprávy m (složitost tohoto výpočtu závisí na konkrétní použité asymetrické šifře, ale jistě je polynomiální, protože jinak by byla daná šifra prakticky nepoužitelná) a jednoduchém porovnání $m \stackrel{?}{=} m'$ (které má konstantní složitost $\mathcal{O}(1)$).

Úplnost tohoto protokolu je splněna s $\varepsilon = 1$ (předpokládáme-li, že nedošlo k chybě při přenosu zpráv).

Spolehlivost plyne z toho, že podvádějící dokazovatel, který zná pouze veřejný klíč a šifrový text, by musel uhodnout otevřený text. To však nastane s pravděpodobností $\delta = 1/|M|$, kde M je prostor všech otevřených zpráv. Toto schéma má však následující nedostatek: kdyby podvádějící Viktor místo zašifrované náhodné zprávy poslal zprávu určenou Peggy, kterou dříve zachytil v její komunikaci s jiným účastníkem a kterou nemohl dešifrovat, protože nezná její soukromý klíč, Peggy by mu zprávu dešifrovala a Viktor by získal příslušný otevřený text.

Přidáním jednoho kroku na začátek tohoto protokolu lze zabránit právě zmíněnému *útoku s opakováním zprávy*, a to pomocí tzv. *jednorázového čísla* (v anglické literatuře se používá název *nonce* jako zkratka „number used once“). Jednorázové číslo N_P je číslo, které si Peggy na začátku běhu protokolu náhodně zvolí a pošle je Viktorovi. Stejně jako v základní verzi protokolu, Viktor vybere náhodnou zprávu m, za kterou připojí obdržené jednorázové číslo N_P, toto spojení zašifruje a pošle Peggy výsledný šifrový text $E_{pk}(m||N_P)$. Peggy zprávu dešifruje a zkontroluje, jestli je na konci její jednorázové číslo. Pokud ano, pošle Viktorovi zprávu, kterou získala dešifrací (ale už bez N_P); pokud ne, ukončí komunikaci s Viktorem. Viktor opět Peggyin důkaz přijme právě když se zpráva, kterou od Peggy obdržel, shoduje se zprávou m.

Pokud Peggy vybírá svoje jednorázové číslo z dostatečně velkého rozmezí, pak je pravděpodobnost toho, že by je Viktor uhodl, zanedbatelná. Protože Peggy volí na začátku každého protokolu vždy nové náhodné jednorázové číslo, může si být

jistá, že šifrový text, který jí Viktor poslal, byl vytvořen až po tom, co mu poslala číslo N_P.

Další způsob obrany proti útoku s reprodukcí zprávy je ten, že Viktor připojí ke své zprávě m aktuální časový údaj a pošle Peggy zašifrované toto spojení. Peggy text dešifruje a získanou zprávu pošle Viktorovi zpět pouze v případě, že rozdíl jejího systémového času a času, který jí poslal Viktor, je menší než předem zvolená konstanta κ. K tomu je však nutná synchronizace jejich systémových časů.

Při hodnocení tohoto schématu z hlediska prokazatelné bezpečnosti je třeba brát ohled na konkrétní použitou asymetrickou šifru. Lze použít např. šifru RSA uvedenou v kapitole 4. Rabinův systém lze použít pouze v případě zajištění jednoznačnosti dešifrování, což lze například i použitím zmíněného jednorázového čísla.

Při použití šifry RSA nemůže být o redukcionistické bezpečnosti tohoto protokolu řeč, neboť se zatím nepodařilo prokázat ekvivalenci mezi RSA problémem a problémem faktorizace modulu n.

Při použití Rabinova kryptosystému se zajištěním jednoznačnosti dešifrování jde o redukcionisticky bezpečný protokol. Aby byl podvádějící dokazovatel schopen správně odpovědět na Viktorovu výzvu, musel by buď uhodnout m (což je pro dostatečně velký prostor otevřených zpráv M prakticky nemožné), nebo by musel být schopen m spočítat, neboli prolomit Rabinův kryptosystém. Ten je ale, jak jsme uvedli v oddílu 4.4 redukcionisticky bezpečný. Odsud plyne redukcionistická bezpečnost identifikačního protokolu založeného na Rabinově kryptosystému.

Tvrzení 6.5. *Identifikační protokol založený na kryptografii s veřejným klíčem je (nezávisle na konkrétní použité asymetrické šifře) důkaz s dokonale nulovou znalostí.*

Důkaz. Výše jsme ukázali, že jde o interaktivní důkaz. Dále je třeba popsat činnost simulátoru M^*. Jediné, čím může ověřovatel ovlivnit podobu posílaných zpráv, je volba zprávy m. Simulátor M^* si na začátku zvolí zprávu m (náhodně, pokud simuluje komunikaci dokazovatele P s čestným ověřovatelem V^*, a speciálně, pokud simuluje komunikaci s podvádějícím dokazovatelem V^*). Komunikace simulovaná strojem M^* má následující podobu:

1. Ověřovatel pošle dokazovateli šifrovou zprávu $c = E_{pk}(m)$.

2. Dokazovatel pošle ověřovateli zprávu m (aniž by musel dešifrovat c).

Simulátor M^* nikdy neselže a vzhledem k tomu, že podoba zpráv posílaných stroji P a V^* závisí pouze na volbě zprávy m, výstup stroje M^* jistě bude patřit do stejného rozdělení jako výstup stroje P po interakci se strojem V^*. □

6.2 Fiat-Shamirovo identifikační schéma

Toto schéma je založeno na obtížnosti hledání odmocniny modulo n, které je součinem dvou prvočísel.

Mějme uzavřenou společnost účastníků, v jejichž středu je důvěryhodná autorita (trusted party), které budeme říkat Trent. Trent vygeneruje dvě různá přibližně stejně velká prvočísla p a q a položí $n := pq$. Faktory p a q si ponechá

v tajnosti a modulus n poskytne všem účastníkům (zveřejní). Každému účastníkovi také přidělí jedinečné identifikační číslo Id. Každý účastník si zvolí tajné $v \in \mathbb{Z}_n^*$, spočítá $u := v^2 \bmod n$ a pošle u Trentovi. Trent ověří, jestli platí $GCD(u, n) = 1$, což je ekvivalentní $GCD(v, n) = 1$[1]. Pokud čísla u všech účastníků splní tuto podmínku, Trent ke každému Id zveřejní příslušné u a dále už se procesu nezúčastňuje.

Pokud se Peggy chce autentizovat vůči Viktorovi, pošle mu svoje Id a Viktor si k němu najde příslušné u. Veřejným vstupem jsou v tomto schématu modulus n a číslo u, dokazovatelovým tajemstvím je číslo v a jazykem je množina všech čísel z QR_n (viz. definici 3.4), jejichž modulární odmocninu Peggy zná.
Protokol probíhá následovně:

1. Peggy náhodně zvolí $r \in \mathbb{Z}_n^*$, položí $a := r^2 \bmod n$ a pošle a Viktorovi.

2. Viktor náhodně zvolí $e \in \{0, 1\}$ a pošle e Peggy.

3. Peggy spočítá $b := rv^e \bmod n$ a pošle b Viktorovi.

4. Viktor přijme Peggyinu identitu, právě když $b^2 \equiv au^e \pmod{n}$.

V tomto protokolu jsou poslány tři zprávy. První zprávou je *závazek (commitment)* Peggy, že zná druhou odmocninu z a, druhou zprávou je Viktorova výzva e a třetí zpráva je Peggyina odpověď na Viktorovu výzvu.

Všimněme si, že kdyby Peggy dvakrát zvolila stejné $r \in \mathbb{Z}_n^*$, ať už v komunikaci s Viktorem nebo s jiným účastníkem, hrozilo by jí prozrazení tajemství v. Pokud by se tak stalo v komunikaci s Viktorem, který by si zaznamenával hodnoty a, které mu Peggy poslala a příslušné výzvy e, které on poslal jí, pak by jí jako výzvu poslal „to druhé e" a poté jednoduše vydělením příslušných b získal v. Kdyby poslala stejné a jinému účastníkovi a tímto účastníkem byl útočník, který zachytil dřívější komunikaci Peggy s Viktorem, pak by podobně jako Viktor zvolil komplementární e a spočítal by si v. Pokud by druhým účastníkem nebyl útočník, pak by s pravděpodobností $1/2$ zvolil komplementární e a odposlouchávající útočník by opět snadno spočítal Peggyino tajemství.

Fiat-Shamirovo schéma je zřejmě efektivní, neboť veškerý Viktorův výpočet spočívá ve volbě čísla $e \in \{0, 1\}$ (se složitostí $\mathcal{O}(1)$) a výpočtu b^2 (se složitostí $\mathcal{O}(|b|^2)$), výpočtu au^e (se složitostí $\mathcal{O}(|a| \cdot |u|)$) a porovnání $b^2 \stackrel{?}{=} au^e$ (se složitostí $\mathcal{O}(1)$).

Úplnost Fiat-Shamirova schématu je splněna, neboť pokud $b = rv^e$, pak $b^2 = au^e$, a stejně jako u předchozího schématu $\varepsilon = 1$.

Spolehlivost: Podvádějící dokazovatel Eva může přesvědčit Viktora o znalosti v, pokud udělá toto:

1. Eva náhodně zvolí $r \in \mathbb{Z}_n^*$ a $\tilde{e} \in \{0, 1\}$, položí $a := r^2 u^{-\tilde{e}} \bmod n$ a pošle a Viktorovi.

2. Viktor zvolí $e \in \{0, 1\}$ a pošle e Evě.

[1]Pokud by to neplatilo, pak by se největší společný dělitel u a n rovnal některému z faktorů p nebo q, a tím by byla ohrožena spolehlivost schématu, protože účastník, který by znal faktory modulu, by si uměl spočítat tajemství v každého jiného účastníka. V tom případě by musel Trent zvolit jiné n a účastníci jiná u.

3. Eva pošle r Viktorovi.

Jestliže $e = \tilde{e}$, pak Viktor přijme Evin důkaz. To nastane s pravděpodobností 1/2. Tato pravděpodobnst je nejvyšší možná. Pokud by totiž pravděpodobnost Evina úspěchu byla $> 1/2$, neboli $\mathtt{P}[(\widetilde{P}, V)(v) = 1 \mid v \notin L] > 1/2$ (jazykem L je zde množina všech čísel z QR_n, k nimž Eva zná druhou odmocninu modulo n), znamenalo by to, že Eva zná nějaké a, pro které dokáže správně zodpovědět obě výzvy (pro $e = 0$ i $e = 1$). V následujícím tvrzení ukážeme, že Fiat-shamirovo schéma je redukcionisticky bezpečné (viz. oddíl 3.3).

Tvrzení 6.6. *Kdokoliv, kdo je schopen pro alespoň jeden závazek ve Fiat-Shamirově schématu správně odpovědět na obě výzvy, je schopen faktorizovat n.*

Důkaz. Nechť tedy Eva dokáže pro závazek a správně odpovědět na obě výzvy $e = 0$ i $e = 1$. Znamená to, že umí spočítat b_1, b_2 taková, že $b_1^2 \equiv a \pmod{n}$ a $b_2^2 \equiv au \pmod{n}$. Z toho plyne, že umí spočítat druhou odmocninu v z kvadratického rezidua u, $v = b_2/b_1 \bmod n$. Tedy má algoritmus A, jehož výstupem při vstupu $u \in QR_n$ je druhá odmocnina z u. Algoritmus A pak může použít k faktorizaci n (viz. oddíl 6.3). □

Toto schéma není spolehlivé, neboť neexistuje žádné $\delta \in [0, \frac{1}{2})$, pro které by byla splněna 3. podmínka z definice 6.1.

t-násobným opakovaním protokolu snížíme pravděpodobnost úspěšného podvodu ze strany dokazovatele na $\delta = 2^{-t}$. Takové schéma už tedy podmínku spolehlivosti splňuje. Při používání tohoto protokolu nehrozí čestnému dokazovateli nebezpečí podvrhnutí zprávy jako u předchozího protokolu založeného na kryptografii s veřejným klíčem, protože Viktor vybírá svoje výzvy z malé množiny $\{0, 1\}$. Jediná informace, kterou Viktor od Peggy získá, je, že Peggy zná druhou odmocninu z u.

Tvrzení 6.7. *Fiat-Shamirovo identifikační schéma je důkaz s dokonale nulovou znalostí (viz. definici 6.4).*

Důkaz. Výše jsme ukázali, že jde o interaktivní důkaz. Je třeba popsat činnost simulátoru M^*. Jediné, čím může ověřovatel ovlivnit běh protokolu, je volba bitu e. Simulátor si na začátku zvolí čísla r a e (číslo r volí náhodně a bit e volí v závislosti na tom, simuluje-li komunikaci ověřovatele P s čestným nebo s podvádějícím ověřovatelem V^* — náhodně pro čestného ověřovatele a v případě podvádějícího dokazovatele zvolí stejný bit jako stroj V^*). V simulaci stroje M^* má komunikace dokazovatele s oveřovatelem následující průběh:

1. Dokazovatel pošle ověřovateli číslo $a := r^2 u^{-e} \bmod n$.

2. Ověřovatel pošle dokazovateli číslo e zvolené na začátku.

3. Dokazovatel pošle ověřovateli číslo r.

Simulátor M^* nikdy neselže a vzhledem k tomu, že podoba zpráv posílaných stroji P a V^* závisí pouze na volbě čísel r a e, výstup stroje M^* jistě bude patřit do stejného rozdělení jako výstup stroje P po interakci se strojem V^*. □

6.3 Faktorizace $n = pq$ pomocí algoritmu na hledání druhé odmocniny z kvadratického rezidua

Jak jsem již uvedli v části 4.4, jednoduchým důsledkem čínské věty o zbytcích (věta 3.12) je toto tvrzení: Je-li n rovno součinu dvou lichých prvočísel p a q, pak existují čtyři druhé odmocniny z kvadratického rezidua x modulo n. Jsou to $\pm y$, $\pm\alpha y \pmod n$, kde $\alpha \equiv 1 \pmod p$ a $\alpha \equiv -1 \pmod q$. Zdůvodnění najdeme opět v části 4.4.

Eva může faktorizat n následujícím způsobem: Zvolí náhodné $y \in \mathbb{Z}_n^*$, položí $x := y^2 \bmod n$ a hodnotu x použije jako vstup algoritmu A. Jeho výstupem bude druhá odmocnina y' z kvadratického rezidua x. S pravděpodobností $1/2$ bude $y' = \pm\alpha y$. V tom případě Eva např. pomocí rozšířeného Euklidova algoritmu 3.21 spočítá $\alpha = \pm y/y'$ a použitím Euklidova algoritmu 3.20 faktorizuje n, neboť platí $GCD(n, \alpha - 1) = p$. V případě, že $y' = \pm y$, opakuje stejný postup s jiným y. Pravděpodobnost, že po k opakováních právě popsaného algoritmu se Evě nepodaří faktorizovat n, je rovna $1/2^k$, což se pro rostoucí k blíží k nule.

6.4 Rozšířené Fiat-Shamirovo identifikační schéma

Chceme-li zmenšit počet opakování Fiat-Shamirova protokolu (např. je-li komunikace časově nebo finančně náročná) se zachováním parametru δ, můžeme použít tzv. *rozšířené Fiat-Shamirovo identifikační schéma*. V něm si každý účastník místo jednoho $v \in \mathbb{Z}_n^*$ zvolí vektor $v = (v_1, \ldots, v_k)$ prvků této grupy. Dalším rozdílem oproti základní verzi Fiat-Shamirova identifikačního schématu je to, že je základní schéma t-krát zopakováno. Obdobně jako v předchozím schématu, modulus $n = pq$ a vektor $u = (v_1^2, \ldots, v_k^2)$ jsou zveřejněny a v je dokazovatelovým tajemstvím.

Na začátku protokolu důvěryhodná autorita Trent vygeneruje dvě různá přibližně stejně velká prvočísla p a q, která utají, a zveřejní modulus $n = pq$. Dále každému účastníkovi přidělí jedinečné identifikační číslo Id. Každý z účastníků si zvolí k náhodných čísel $v_1, v_2, \ldots, v_k \in \mathbb{Z}_n$ a položí $u = (u_1, \ldots, u_k) := (v_1^2, \ldots, v_k^2)$. Vektor u poté odešle Trentovi, který ověří, jestli $GCD(u_i, n) = 1$ pro všechna $i = 1, \ldots, k$ (vysvětlení nalezneme u předchozího protokolu). Stejně jako u základního Fiat-Shamirova schématu, pokud vektory u všech účastníků splní tuto podmínku, Trent ke každému Id zveřejní příslušné u a dále už se procesu nezúčastňuje.

Autentizace Peggy vůči Viktorovi probíhá následovně:
Zopakuj následující t-krát:

1. Peggy náhodně zvolí $r \in \mathbb{Z}_n^*$, položí $a := r^2$ a pošle a Viktorovi.
2. Viktor náhodně zvolí $e := (e_1, \ldots, e_k) \in \{0, 1\}^k$ a pošle e Peggy.
3. Peggy spočítá $b := r \prod_{i=1}^{k} v_i^{e_i} \bmod n$ a pošle b Viktorovi.

4. Pokud $b^2 \not\equiv a\prod_{i=1}^{k} u_i^{e_i} \pmod{n}$, pak Viktor odmítne Peggyinu identitu a ukončí běh protokolu.

Toto schéma je efektivní, protože Viktorovým nejnáročnějším výpočtem je výpočet $a\prod_{i=1}^{k} u_i^{e_i}$, který má složitost $\mathcal{O}(|a| \cdot |u_1| \cdot \ldots \cdot |u_k|)$.

Úplnost: Pokud pravý dokazovatel Peggy i Viktor dodržují protokol, pak Viktor přijme Peggyin důkaz.

Spolehlivost: Podvádějící dokazovatel Eva přesvědčí Viktora o tom, že je Peggy, pokud správně uhádne Viktorovu výzvu e pro každou iteraci, t.j. pokud se jí podaří vybrat správný prvek z $\{0,1\}^{kt}$, což nastane s pravděpodobností 2^{-kt}. Kdyby pravděpodobnost jejího úspěchu byla větší než 2^{-kt} (pravděpodobnost bereme přes všechny výzvy e), znamenalo by to, že zná vektor $A = (a^1, \ldots, a^t)$ závazků a^j (jeden pro každou iteraci j, $1 \leq j \leq t$), pro který je schopna správně odpovědět na dvě různé Viktorovy výzvy $E = (e^1, \ldots, e^t)$ a $F = (f^1, \ldots, f^t)$, $E \neq F$. Protože jsou vektory E a F různé, existuje iterace j taková, že $e^j \neq f^j$. Eva umí správně odpovědět na obě výzvy $e := e^j$ a $f := f^j$ pro závazek $a := a^j$. To znamená, že dokáže spočítat b_1 a b_2 takové, že

$$b_1^2 \equiv a\prod_{i=1}^{k} u_i^{e_i} \pmod{n} \text{ a } b_2^2 \equiv a\prod_{i=1}^{k} u_i^{f_i} \pmod{n}.$$

Stejně jako u předešlého schématu z toho plyne, že Eva dokáže spočítat druhou odmocninu $v = b_2/b_1$ náhodného kvadratického rezidua $u = \prod_{i=1}^{k} u_i^{f_i - e_i}$. A to je spor s naším předpokladem, že počítání druhých odmocnin je nezvládnutelné bez znalosti faktorů p a q modulu n.

6.5 Schnorr-Okamotovo identifikační schéma

Na závěr uvedeme identifikační schéma (které je „vylepšením“ Schnorrova schématu založeného na obtížnosti hledání diskrétního logaritmu; viz. např. [7], str. 414), k jehož prolomení by útočníkovi nestačila ani případná znalost řešení problému diskrétního logaritmu.

Důvěryhodná autorita Trent přidělí každému účastníkovi P jedinečné identifikační číslo Id_P. Dále vygeneruje dvě prvočísla p a q taková, aby problém nalezení diskrétního logaritmu byl těžký v grupě $\mathbb{Z}_p^*$ (tedy aby velikost p byla přibližně 1024 bitů) a aby $q \mid (p-1)$ a q mělo zhruba 160 bitů. Trent dále vygeneruje veřejné parametry $\alpha_1, \alpha_2 \in \mathbb{Z}_p^*$ řádu q v grupě $\mathbb{Z}_p^*$ (např. $\alpha_i = \beta_i^{(p-1)/q}$, kde β_i jsou generátory $\mathbb{Z}_p^*$), a konečně zvolí tzv. bezpečnostní parametr t takový, aby platilo $2^t < q$, např. $t \geq 40$.

Každý účastník P si zvolí jako svůj tajný klíč dvojici $a_1, a_2 \in \mathbb{Z}_q$ a spočítá si svůj veřejný klíč $v := \alpha_1^{-a_1}\alpha_2^{-a_2} \bmod p$. P se identifikuje Trentovi „nekryptografickým“ způsobem a pošle mu číslo v. Trent k sobě „sváže“ v a Id_P svým podpisem $s := \mathbf{Sig}_{KT_S}(Id_P, v)$, kde KT_S je Trentův soukromý podepisovací klíč příslušný k veřejnému klíči pro ověřování podpisů, který označíme KT_V. Účastník P obdrží od Trenta certifikát $cert_P = (Id_P, v, s)$.

Protokol pro autentizaci Peggy vůči Viktorovi:

1. Peggy si náhodně zvolí (závazek) k_1, k_2, $1 < k_1, k_2 < q-1$, položí

$$\gamma := \alpha_1^{k_1}\alpha_2^{k_2} \bmod p$$

a pošle Viktorovi $cert_P, \gamma$.

2. Viktor ověří Trentův podpis na certifikátu $cert_P$, $\mathbf{Ver}_{KT_V}(Id_P, v, s) \stackrel{?}{=} 1$ a poté pošle Peggy náhodné r (výzvu), $1 < r < 2^t$.

3. Peggy zkontroluje, jestli $1 < r < 2^t$, spočítá

$$y_1 := k_1 + a_1 r \bmod q,$$

$$y_2 := k_2 + a_2 r \bmod q$$

a pošle Viktorovi (odpověď) y_1, y_2.

4. Viktor přijme Peggyinu identitu, právě když $\gamma = \alpha_1^{y_1}\alpha_2^{y_2}v^r \bmod p$.

Veškerý Viktorův výpočet zřejmě proběhne v polynomiálním čase. Schéma je tedy efektivní.

Schéma je i úplné (s $\varepsilon = 1$), neboť

$$\alpha_1^{y_1}\alpha_2^{y_2}v^r \bmod p = \alpha_1^{k_1+a_1 r \bmod q}\alpha_2^{k_2+a_2 r \bmod q}(\alpha_1^{-a_1}\alpha_2^{-a_2})^r \bmod p =$$

$$= \alpha_1^{k_1}\alpha_2^{k_2} \bmod p = \gamma,$$

protože α_1, α_2 jsou řádu q v $\mathbb{Z}_p^*$.

Nyní ukážeme, že protokol je i spolehlivý. Jediná možnost, jak by se mohl podvádějící dokazovatel, který nezná tajemství (a_1, a_2), pokusit předvěčit Viktora o jeho znalosti, je ta, že by se pokusil uhodnout r, a pak by ověřovateli poslal $\gamma = \alpha_1^{y_1}\alpha_2^{y_2}v^r \bmod p$ s libovolnými y_1, y_2. To však nastane s pravděpodobností $1/2^t$, která se pro rostoucí bezpečnostní parametr t blíží k nule. Protokol je tedy spolehlivý s $\delta = 1/2^t$.

Nyní dokážeme, že toto schéma má vlastnost zero-knowledge.

Tvrzení 6.8. *Schnorr-Okamotovo schéma je důkaz s dokonale nulovou znalostí.*

Důkaz. Skutečnost, že se jedná o interaktivní důkaz, jsem ukázali výše. Nyní popíšeme činnost simulátoru M^*. Ověřovatel ovlivňuje běh protokolu pouze volbou čísla r. Simulátor M^* si na začátku svého výpočtu zvolí čísla $r \in [2, 2^t - 1]$ a $y_1, y_2 \in [0, q-1]$ (čísla y_1, y_2 náhodně, číslo r v závislosti na tom, jestli simuluje komunikaci dokazovatele P a čestného nebo podvádějícího ověřovatele V^* — pro čestného ověřovatele náhodně, pro podvádějícího dokazovatele stejně jako simulovaný dokazovatel V^*). Komunikace simulovaná strojem M^* má následující podobu:

1. Dokazovatel pošle ověřovateli číslo $\gamma := \alpha_1^{y_1}\alpha_2^{y_2}v^r \bmod p$.

2. Ověřovatel pošle dokazovateli číslo r.

3. Dokazovatel pošle ověřovateli čísla y_1, y_2.

Simulátor M^* nikdy neselže a protože podoba zpráv posílaných stroji P a V^* závisí pouze na volbě čísel r, y_1 a y_2, výstup stroje M^* jistě bude patřit do stejného rozdělení jako výstup stroje P po interakci se strojem V^*. □

Na závěr ukážeme zmíněnou zajímavou vlastnost tohoto protokolu, totiž že útočníkovi, který by se snažil z veřejného klíče v získat soukromý klíč (a_1, a_2), by nepomohla ani znalost řešení problému diskrétního logaritmu.

Jistě existuje číslo $l \in \{1, 2, \ldots, q\}$ takové, že platí $\alpha_2 = \alpha_1^l \bmod p$. Pak můžeme vztah pro v vyjádřit jako $v = \alpha_1^{-a_1}(\alpha_1^l)^{-a_2} \bmod p = \alpha_1^{-a_1-la_2} \bmod p$. Předpokládáme, že útočník dokáže řešit problém diskrétního logaritmu, a tedy dokáže najít číslo $x \in \{0, 1, \ldots, p-2\}$, pro které platí $x = -\log_{\alpha_1} v$. Útočník tak získá modulární rovnici

$$x = a_1 + la_2 \bmod q$$

s neznámými a_1, a_2, jejímž řešením je nekonečně mnoho dvojic (a_1, a_2). Další rovnice, které má útočník k dispozici jsou

$$y_1 = k_1 + a_1 r \bmod q,$$

$$y_2 = k_2 + a_2 r \bmod q$$

s neznámými a_1, a_2, k_1, k_2.

Lze si snadno rozmyslet, že všechny další rovnice, které je útočník schopen získat, jsou lineární kombinací uvedených tří rovnic. Při dalším proběhnutí protokolu se při správné volbě čísel k_1, k_2 počet rovnic zvýší o dvě, stejně jako počet neznámých. Počet rovnic, které bude mít útočník k dispozici bude tedy stále o jedna menší než počet neznámých v těchto rovnicích. Jejich soustava bude tedy mít stále nekonečně mnoho řešení.

Literatura

[1] Delfs H., Knebl H., *Introduction to Cryptography — Principles and Applications*, Springer, Berlin, 2002, ISBN 3-540-42278-1

[2] Goldreich O., Foundations of Cryptography, Cambridge University Press, Cambridge, 2001, ISBN 0-521-79172-3

[3] Hojsík M., *Kryptografické protokoly pro elektronické volby*, bakalářská práce, Karlova Univerzita, Praha, 2004

[4] Holub Š., texty k přednášce Složitost pro kryptografii na MFF UK, http://www.karlin.mff.cuni.cz/∼holub/texty.htm

[5] Koblitz N., Menezes A. J., *Another Look at „Provable Security“*, http://eprint.iacr.org/2004/152.pdf, 2004

[6] Mao W., *Modern Cryptography — Theory and Practise*, Prentice Hall PTR, Prentice-Hall, Inc., New Jersey, 2004, ISBN 0-13-066943-1

[7] Menezes A., van Oorschot P., Vanstone S., *Handbook of Applied Cryptography*, CRC Press, Boca Raton, 1997, ISBN 0-8493-8523-7