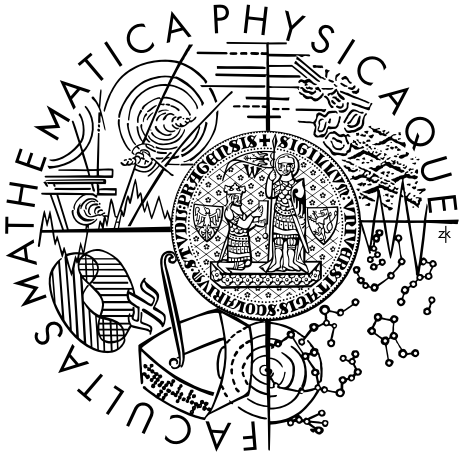
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Jiří Pavlů

Algoritmy dokazující prvočíselnost

Katedra algebry

Vedoucí bakalářské práce: doc. RNDr. Jan Šťovíček, Ph.D.

Studijní program: Matematika

Studijní obor: Matematické metody informační bezpečnosti

Praha 2015/2016

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Algoritmy dokazující prvočíselnost

Autor: Jiří Pavlů

Katedra: Katedra algebry

Vedoucí bakalářské práce: doc. RNDr. Jan Šťovíček, Ph.D., Katedra algebry

Abstrakt: Cílem práce je seznámit čtenáře s různými algoritmy pro dokazování prvočíselnosti spolu s použitím některých těchto algoritmů v praxi. Práce je zaměřena na Goldwasser-Killianův test, jehož výstupem je certifikát, který je možné rychle ověřit. Aby bylo možné tomuto testu porozumět, obsahuje práce úvod do teorie eliptických křivek, na nichž je test založen. Práce také ukazuje, proč tvoří sčítání na eliptické křivce grupu, jak se tato grupa konstruuje a jak těchto znalostí využít pro tvorbu algebraického vzorce pro výpočet součtu dvou bodů.

Klíčová slova: dokazování prvočíselnosti, Goldwasser-Killianův algortimus, eliptické křivky

Title: Algortihms for proving primality

Author: Jiří Pavlů

Department: Department of Algebra

Supervisor: doc. RNDr. Jan Šťovíček, Ph.D., Department of Algebra

Abstract: The goal of the thesis is introducing the reader to some of the algorithms for proving primality along with practical usage of some of these algorithms. The main objective of the thesis is a presentation of Goldwasser-Killian primality test, which can be used to produce primality certificates, which can be verified very quickly. For better understanding of the test the thesis also includes an introduction to elliptic curves, which are the basis of the test. The thesis also shows how is a group of points on elliptic curves constructed and how to use this information for construction of algebraic formula for a sum of two points on a curve.

Keywords: primality proving, Goldwasser-Killian algorithm, elliptic curves

Chtěl bych poděkovat svému vedoucímu bakalářské práce doc. RNDr. Janu Šťovíčkovi, Ph.D., nejen za tipy na literaturu, ale i za jeho pomoc s uchopením tématu, za jeho ochotu a za dobře kladené otázky, které mě vždy přinutily zamyslet se nad tím, zda to, co mi přišlo jako jasné, je skutečně tak jasné, jako se mi na první pohled zdálo.

Obsah

Úvod

Prvočísla zajímala matematiky už ve starém Řecku. Už Eratosthenés z Kyrény sepsal algoritmus, který najde všechna prvočísla od 1 do N se složitostí $O(N \log(\log(N)))$. Od 17. století se začaly objevovat efektivnější způsoby, jak rozlišit prvočísla od čísel složených. Měly ale tu nevýhodu, že nebyly schopny dokázat, že nějaké číslo je prvočíslo. Takové algoritmy pak mohly o čísle buď říci: „toto číslo je určitě složené“, nebo „o tomto čísle se mi nepovedlo prokázat, že je složené“. Prvočíselnost se pak „dokazovala“ tak, že se takové testy několikrát opakovaly, a pokud se nepovedlo ukázat, že je číslo složené, předpokládalo se, že je to prvočíslo.

Tato práce se ale bude zabývat algoritmy, které jsou schopné prvočíselnost skutečně dokázat – a to co nejrychleji. Sice bychom vždy mohli použít Eratosthenovo síto, ale pro důkaz jediného prvočísla je tento algoritmus příliš pomalý. Navíc u těchto algoritmů budeme požadovat, aby jejich správnost nebyla závislá na nějakém nedokázaném tvrzení. Protože už Rabin publikoval test, který je schopen rychle zjistit, zda zadané číslo je či není prvočíslo, ale správnost tohoto algoritmu závisí na nedokázané Riemannově domněnce. Naopak nám nebude vadit, pokud nějaký algoritmus bude schopen rychle prokázat prvočíselnost jen u čísel speciálního tvaru, neboť i takové algoritmy mohou mít velký význam pro teorii. Největší důraz ale bude kladen na Goldwasser-Killianův algoritmus, který navíc pro dokázané prvočíslo vystaví certifikát, který je možné nezávisle (a rychle) ověřit, a tak zjistit, zda algoritmus třeba neudělal chybu, a že číslo, jehož prvočíselnost jsme chtěli prokázat, je skutečně prvočíslem.

Cílem celé práce bude poskytnout čtenáři přehled o různých algoritmech dokazujících prvočíselnost, a představit mu a pomoci mu pochopit teorii, na níž stojí Goldwasser-Killianův algoritmus. Práce předpokládá, že čtenář se nemusí orienovat v oblasti projektivní geometrie nebo eliptických křivek, a proto se mu tyto znalosti snaží alespoň v omezené formě poskytnout.

První kapitola bude přehledem některých takových zajímavých prvočíslených testů, spolu s jejich historickým významem a možným využitím.

Ve druhé kapitole se pak čtenář seznámí s projektivní geometrií, díky které bude moci lépe pochopit na čem stojí Goldwasser-Killianova test.

Třetí kapitola se pak bude zabývat přímo eliptickými křivkami. Oproti předchozí kapitole zde bude kladen větší důraz na početní stránku věci.

Konečně ve čtvrté kapitole bude představen Goldwasser-Killianův algoritmus. Ukáže se, jak využít jeho výstup k tomu, aby bylo možné prokázat, že číslo, jehož prvočíselnost jsme chtěli ověřit, je skutečně prvočíslem. Dále bude ukázáno, že pokud platí jistá domněnka, pak Goldwasser-Killianův test běží rychle.

1. Prvočíselné testy

V této kapitole se seznámíme s různými prvočíselnými testy, které jsou schopny prvočíselnost daného čísla (často jen speciálního tvaru) prokázat. Také se zde setkáme s pojmy jako uvsložitost, nebo že algoritmus běží v čase $O(\log(N))$. Tyto pojmy jsou hezky vysvětlené v knize [Sta11], kde se s nimi čtenář, pokud je již nezná, může podrobněji seznámit.

1.1 Naivní test

Tento test spočívá v pouhém zkoušení všech možných dělitelů. Nechť máme číslo N, o kterém chceme rozhodnout, zda je prvočíslo, nebo ne. Pak nám stačí zkusit podělit jej se zbytkem všemi čísly menšími než N. Což nám dává složitost polynomiální (zhruba lineární) v N. Dělit se zbytkem totiž umíme v polynomiálním čase. Abychom ovšem nějaký prvočíselný test označili za rychlý, budeme po něm požadovat, aby běžel v čase polynomiálním v délce zápisu N, tedy polynomiální v $\log(N)$.

Snadno si ale uvědomíme, že jsme ve skutečnosti počítali spoustu zbytků po dělení zbytečně. Existuje-li totiž rozklad $N = AB$, pak nutně A nebo B musí být menší nebo rovno $\sqrt{N}$. Tedy nám bude stačit provést $O(\sqrt{N})$ dělení.

Dále si lze uvědomit, že všechna prvočísla (kromě 2 a 3) jsou tvaru $6k \pm 1$ pro $k \in \mathbb{N}$. Čísla tvarů $6k, 6k+2$ a $6k+4$ jsou totiž sudá, a čísla tvaru $6k+3$ jsou dělitelná třemi. Zbývají jen čísla tvaru $6k+1$ a $6k+5$ – což je ale stejné jako $6k-1$. Díky tomu můžeme můžeme tento test zase o něco urychlit. Ovšem asymptotická složitost se nám už nezlepší.

1.2 Lucas-Lehmer-Rieslerův test

Tento test slouží k určení prvočislnosti čísel tvaru $C = k2^n - 1$, kde k,$n \in \mathbb{N}$ a $2^n > k$. Test funguje následovně: definujeme $\forall i \in \mathbb{N} : u_i = u_{i-1}^2 - 2$. Číslo C je prvočíslo právě když, $C|u_{n-2}$, pro vhodně zvolenou počáteční hodnotu u_0. Tu můžeme volit následovně:

$$u_0 = \begin{cases} 4 & \text{pro } n \text{ liché a k=1,} \\ 5778 & \text{pro } n \equiv 0 \bmod 4 \text{ nebo } n \equiv 3 \bmod 4 \text{ a } k = 3 \\ (2+\sqrt{3}^k) + (2-\sqrt{3}^k) & \text{pokud } 3 \nmid C \text{ a } k \equiv 1 \bmod 6 \text{ nebo } k \equiv 1 \bmod 5. \end{cases}$$

Pokud nenastává ani jeden z těchto případů, musí se u_0 určit složitějším způsobem. Jeden z takových způsobů je popsán v [RÖ94]. Teorii, která stojí za tímto testem, můžeme najít například v článku [Leh30].

Tento test je zobecněním původního Lucas-Lehmerova testu, který (dosti podobně) zkoumá jen taková C, že $C = 2^p - 1$, a p je prvočíslo.

Výhodou tohoto testu je jeho rychlost a determinističnost. Snadno si totiž uvědomíme, že tento test běží v polynomiálním čase vzhledem k délce vstupu. Test spočívá ve spočtení hodnoty u_0, kterou umíme spočítat rychle (alespoň některé případy jsou uvedeny zde, případně v odkazovaných článcích), a v $O(n) =$

$O(log(C))$ modulárních násobeních. O modulárním násobení pak víme, že má polynomiální složitost.

Jeho nevýhodou pak je, že jej lze použít jen na čísla ve speciálním tvaru. Největší nám známá prvočísla byla ale ověřena variantou tohoto testu.

1.3 Prothův test

Tento test slouží k určení prvočislnosti čísel tvaru $C = k2^n + 1$, kde k,$n \in \mathbb{N}$, k liché a $2^n > k$. Test funguje následovně: pokud C je tvaru $k2^n + 1$ a existuje takové číslo a, že $a^{(C-1)/2} \equiv 1 \bmod p$, pak C je prvočíslo. Navíc pokud je Jacobiho symbol $(\frac{a}{C}) = -1$, pak se implikace v tomto testu stává ekvivalencí. Jacobiho symbol je pak definován takto $(\frac{a}{C}) = \prod_{i=1}^{n}(a^{\frac{p_i-1}{2}} \bmod p_i)^{e_i}$, kde $C = \prod_{i=1}^{n} p_i^{e_i}$ a p_i jsou prvočísla. Navíc má některé dobré vlastnosti, které jej umožňují spočítat rychle.

Tento výsledek můžeme najít už v [Pro78] (i když ten nepoužívá Jacobiho symbol, ale pojem kvadratického nezbytku).

Tento test vznikl jako zobecnění jistého Pépinova testu, který je podobný, ale funguje jen pro čísla tvaru $C = 2^{2^n} + 1$. Běží v očekávaném polynomiálním čase - počítáme $a^{(C-1)/2} \bmod p$, což pomocí binárního algoritmu umíme v polynomiálním čase. Zbývá jen najít vhodné a, kterých je ale zhruba 50%.

Motivací pro původní verzi testu, bylo rychlé testování prvočíselnosti tzv. Fermatových čísel – to jsou právě čísla tvaru $2^{2^n} + 1$. Ale vzhledem k tomu, jak rychle tato čísla rostou, byl k tomuto účelu použit jen málokrát (testy dalších Fermatových čísel jsou zatím nad naše výpočetní možnosti).

1.4 Pocklingtonův test

Tento test využívá Malé Fermatovy věty.

Věta 1 (Malá Fermatova)**.** *Nechť p prvočíslo. Pak pro každé číslo $a \in \mathbb{N}$ takové, že* GCD(a,p) $= 1$ *platí, že $a^{(p-1)} \equiv 1 \bmod p$.*

Test funguje takto: nechť máme číslo N, které chceme otestovat. Vezmeme $x \in \mathbb{Z}_N$ a zjistíme, zda $x^{N-1} \equiv 1 \bmod N$. Pokud ne, pak je dle Malé Fermatovy věty N složené. Pokud kongruence platí, pak postupujeme následovně - najdeme co největšího prvočíselného dělitele p čísla $N-1$, jehož exponent v prvočíselném rozkaldu $N-1$ označíme jako a. Dále spočítáme $x^{(N-1)/p} \bmod N$. Pokud je toto různé od jedné, pak spočítáme GCD($x^{(N-1)/p} - 1$,N). Pokud toto není rovno jedné, pak máme dělitele N, a N je složené.

Pokud je ovšem GCD($x^{(N-1)/p} - 1$,N) $= 1$, pak lze ukázat, že N je prvočíslo, nebo lze pro N alespoň omezit množinu možných dělitelů. Vidíme totiž, že řád x v grupě $\mathbb{Z}_n^*$ musí být násobkem p^a, označíme jej r. Víme, že $p^a|r$. Nechť q je nějaký prvočíselný dělitel N. Pak platí (z Čínské věty o zbytcích), že $x^r \equiv 1 \bmod q$. Protože x je nesoudělné s N, musí být nesoudělné i s q, a tedy (dle Malé Fermatovy věty) $x^{q-1} \equiv 1 \bmod q$. Tedy platí, že $r|(q-1)$. A z transitivity dělitelnosti platí, že $p^a|(q-1)$. Tedy každý dělitel N musí být tvaru $kp^a + 1$.

Pokud jsme tedy nalezli p^a dostatečně velké (větší než $\sqrt{N}$), nebude existovat žádný dělitel N menší, než $\sqrt{N}$. Pokud ne, pak se nám alespoň povedlo hodně snížit počet čísel, kterými musíme vyzkoušet podělit. Navíc můžeme proceduru zopakovat s dalšími děliteli $N-1$ a dostat tak silnější požadavky na možné dělitele N.

Zbývá zjistit, co by se stalo, pokud by $x^{(N-1)/p} \bmod N$ bylo rovno jedné. Dá se ukázat, že se dá postupovat velmi podobným postupem s obdobnou diskuzí. O něco více je tento postup vysvětlen (navíc s aplikacemi v různých speciálních případech) v [Poc16].

Výhodou tohoto testu je, že jej lze použít na více vstupů, než předešlé dva. Navíc, pokud u nějakého čísla neprokáže (a zároveň nevyvrátí) prvočíselnost, pak dá alespoň omezující podmínku na tvar potenciálních dělitelů.

Jeho nevýhodou pak je, že musíme alespoň částečně faktorizovat $N-1$, což obecně neumíme v polynomiálním čase. Navíc u čísla, které chceme otestovat, nelze tak snadno jako v předchozích případech zjistit, zda test prvočíselnost skutečně prokáže.

1.5 AKS test

Tento test je historicky prvním testem, který splňuje najednou následující: běží v polynomiálním čase, lze jej použít pro jakékoli číslo, je deterministický a jeho správnost nezávisí na žádném nedokázaném tvrzení.

Myšlenka testu je založena na následujícím lemmatu:

Lemma 2. *Nechť $a \in \mathbb{Z}$, $N \in \mathbb{N}$, $2 \leq N$ a navíc* GCD(a,N) $= 1$. *Pak N je prvočíslo, právě když platí tato modulární rovnost polynomů: $(x+a)^N = x^N + a \bmod N$.*

Důkaz. Nechť N prvočíslo, pak $\binom{N}{i} = 0 \bmod N$, výraz na levé straně upravíme pomocí binomické věty, a vidíme, že lemma platí. Nechť N složené, q prvočíslo, $q^k \mid N$, $q^{k+1} \nmid N$. Pak $q^k \nmid \binom{N}{q}$ a $q \nmid a^{n-q}$. Koeficient příslušející x^q je tedy nenulový, rovnost tedy nenastává, a lemma platí. Kongruenci $a^N = a$ pak dostáváme z věty 1

□

Toto lemma již dává nutnou a postačující podmínku pro prvočíselnost, ale takový test by měl exponenciální složitost. Je totiž nutné umocnit polynom na N a poté spočítat zbytek po dělení číslem N pro $N+1$ koeficientů.

Pro snížení výpočetní náročnosti lze celou rovnost z lemmatu brát modulo $x^r - 1$, pro nějaké $r < N$. Cenou za to ovšem je, že ekvivalence z lemmatu se stává jen implikací, protože takto oslabenou podmínku mohou splňovat i některá složená čísla N.

Autoři algoritmu v [AKS04] ukazují, že pro vhodně zvolené (a nepříliš velké) r platí, že pokud nějaké číslo N splňuje rovnost z lemmatu modulo $x^r - 1$, pak již musí jít nutně o mocninu prvočísla. Díky tomu lze dosáhnout jak polynomiální složitosti vyhodnocení podmínky testu, tak toho, že pokud číslo projde testem (s tím, že je nutné ověřit, zda se nejedná o mocninu), pak musí být nutně prvočíslem, a zároveň žádná složená čísla testem neprojdou.

2. Něco projektivní geometrie

Cílem této kapitoly je poskytnout čtenáři lehké seznámení s pojmy projektivní geometrie, které se využijí v kapitole o eliptických křivkách. Eliptické křivky se používají v Goldwasser-Killianově testu prvočíselnosti.

Tato kapitola velmi čerpá z knihy [Ful], hlavně pak z kapitol 3 a 5, ve kterých může čtenář nalézt souvislosti, které zde nejsou uvedeny, protože pochopit ideu fungování Goldwasser-Killiannova algoritmu lze i bez nich.

2.1 Motivace a základní definice

Představme si, že chceme zkoumat průsečíky různých křivek v rovině, a zároveň nechceme řešit takové speciální případy, jakými jsou například rovnoběžky, nebo křivky, které se k sobě asymptoticky blíží. Chceme, aby se každé dvě přímky (i rovnoběžky) protly v jednom bodě. Chceme dát nějaký rozumný význam tomu, že se dvě křivky „protnou v nekonečnu“. Budeme si tedy chtít rozšířit klasickou afinní rovinu[1] tak, aby se takové křivky skutečně protly v nějakém bodě.

Ztotožníme tedy každý bod z afinní roviny se souřadnicemi (x,y) s bodem $(x,y,1)$ v trojrozměrném afinním prostoru. Tedy každý bod z roviny nám bude určovat právě jednu přímku procházející bodem (0,0,0) v prostoru. Takto jsou určeny všechny přímky procházející bodem (0,0,0) až na ty, které leží v rovině $z = 0$. Ty pro nás budou reprezentovat „body v nekonečnu“.

Definice 1 (Projektivní rovina)**.** Projektivní rovinou nad tělesem $\mathbb{F}$*, psáno* $\mathbb{P}^2(\mathbb{F})$*, případně* $\mathbb{P}^2$*, budeme rozumět množinu všech přímek, procházejících počátkem souřadnic v trojrozměrném afinním prostoru dimenze 3.*

Prvky $\mathbb{P}^2(\mathbb{F})$ budeme nazývat *body*. Abychom se v projektivním prostoru nějak rozumně vyznali, budeme si na něm chtít zavést souřadnice. Nechť $A \in \mathbb{P}^2(\mathbb{F})$, je určen bodem $A_{af} = (x,y,z)$ v afinním prostoru. Pak určíme *homogenní souřadnice* bodu A, jako $(x : y : z)$. Všimneme si, že homogenní souřadnice jsou určeny jednoznačně, až na nějaký λ-násobek, kde $\lambda \in \mathbb{F}^*$. Budeme proto za body považovat třídy ekvivalence souřadnic.

Vidíme, že ačkoli nejsou homogenní souřadnice bodu určeny jednoznačně, víme alespoň, zda je jeho i-tá souřadnice nulová, či nenulová. Navíc, pokud je souřadnice z nenulová, pak jsou zjevně jednoznačně určeny podíly x/z a y/z. Obdobně pokud je nenulové x nebo y.

Dále se nám bude hodit uvažovat množinu všech „bodů v nekonečnu“ (tedy bodů s nulovou souřadnicí z) jako přímku. Označíme ji H_∞ – všimneme si, že pokud vezmeme dva různé body z H_∞, a povedeme jimi přímku (v afinním prostoru si můžeme představit rovinu určenou dvěma přímkami), dostaneme celé H_∞. (Intuitivně jde o všechny směry v rovině.)

Příklad. Motivací pro projektivní rovinu byly průniky rovinných křivek v nekonečnu. Uvažujme křivky dané rovnicemi $y = x$ a $y^2 = x^2 + 1$. Z dobrých důvodů, převedeme druhou rovnici do tvaru $y^2 = x^2 + z^2$ (tomuto procesu budeme říkat

[1] Pokud se čtenář nesetkal s pojmem afinního prostoru, pak si může představit klasický Euklidův prostor, který je mu jistě dobře znám.

homogenizace) . Pak vidíme, že obě tyto rovnice jsou splněny i v bodě $(1:1:0)$. To je ten bod v nekonečnu, ve kterém se tyto dvě křivky protnou.

2.2 Algebraické množiny v projektivní rovině

V projektivní rovině budeme chtít pracovat s takovými množinami bodů, které mají nějaké hezké algebraické vlastnosti – dají se vyjádřit polynomiální algebraickou rovnicí. Takovým hezkým množinám budeme říkat algebraické.

Definice 2 (Nula polynomu)**.** *O bodu $A \in \mathbb{P}^2(\mathbb{F})$ řekneme, že je* nulou polynomu *$p \in \mathbb{F}[x,y,z]$, pokud pro všechny možné homogenní souřadnice $(x:y:z)$ bodu A platí, že $p(x,y,z) = 0$. Pak píšeme: $p(A) = 0$.*

Definice 3 (Homogenní polynom)**.** *Jako* stupeň *monomu definujeme součet mocnin všech neznámých v monomu (tedy monom $x^2y^3z^5$ má stupeň $2+3+5=10$). Nechť p je polynom z $\mathbb{F}[x,y,z]$. Pak o p řekneme, že je* homogenní*, respektive, že je to* forma*, pokud jsou všechny jeho monomy stejného stupně. Dále definujeme* stupeň *formy jako stupeň libovolného jejího monomu.*

Všimneme si, že pokud pro homogenní polynom h platí, že pokud pro nějaké homogenní souřadnice $(x:y:z)$ bodu A je $h(x,y,z) = 0$, pak již nutně $p(A) = 0$. Uvědomíme si, že pro homogenní souřadnice bodu A tvaru $(\lambda x:\lambda y:\lambda z)$ stačí, poté co dosadíme za neznámé, vytknout z celého polynomu vhodnou mocninu λ a máme převedeno na původní případ.

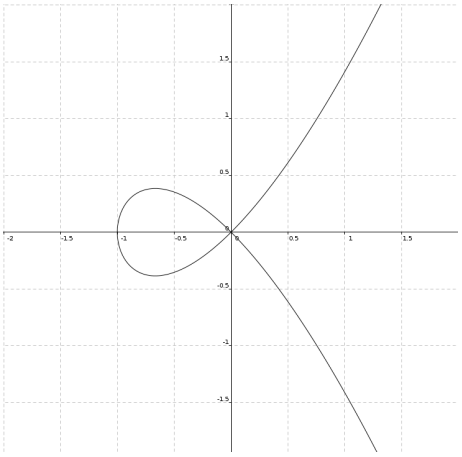
Definice 4 (Projektivní algebraická množina)**.** *Pro libovolnou množinu polynomů S z $\mathbb{F}[x,y,z]$ definujeme $V(S) = \{P \in \mathbb{P}^2 : P$ je nulou každého polynomu z $S\}$. O množině $M \subset \mathbb{P}^2$ řekneme, že je algebraická, pokud $M = V(S)$ pro nějakou množinu polynomů S.*

Budeme také chtít nějaký způsob, jak převádět situace z afinního do projektivního prostoru a naopak. K tomu nám budou sloužit procesy homogenizace a dehomogenizace.

Definice 5 (Homogenizace a dehomogenizace)**.** *Nechť F je libovolný polynom z $\mathbb{F}[x,y]$, pro jehož monomy platí, že jejich maximum jejich stupňů je rovno d. Pak označíme F^* takový polynom z $\mathbb{F}[x,y,z]$, který dostaneme předpisem $F^*(x,y,z) = z^dF(\frac{x}{z},\frac{y}{z})$. Zobrazení $F \to F^*$ budeme říkat homogenizace. Naopak pro libovolný homogenní polynom $H \in \mathbb{F}[x,y,z]$ označíme jako H_* takový polynom z $\mathbb{F}[x,y]$, který dostaneme z H dosazením $z = 1$. Zobrazení $H \to H_*$ pak budeme říkat dehomogenizace.*

Homogenizace pak v podstatě znamená, že pomocí proměnné z doplníme všechny členy polynomu na stejný, nejnižší možný stupeň.

Definice 6 (Projektivní algerbraická křivka)**.** *Nechť $f \neq 0$ je forma. Pak* projektivní algebraickou křivkou *nazveme algebraickou množinu $V(\{f\})$.* Stupněm *algebraické křivky nazveme stupeň formy, kterou dostaneme z f tak, že vezmeme všechny členy rozkladu f na prvočinitele v první mocnině a roznásobíme je. Křivkám stupně 3 budeme říkat* kubické*.*



Obrázek 2.1: Příklad kubické křivky, která má v bodě (0,0) singulární bod „typu uzel“.

Lze ukázat, že si odpovídají projektivní algebraické křivky a bezčtvercové (tedy ty které obsahují všechny své faktory jen v první mocnině) homogenní nekonstantní polynomy, a proto nás budou zajímat jen ty.

Definice 7 (Projektivní změna souřadnic)**.** *Nechť Z je bijektivní zobrazení $\mathbb{P}^2$, definované předpisem: $Z(x : y : z) = (f_1(x,y,z) : f_2(x,y,z) : f_3(x,y,z))$, kde f_1, f_2, f_3 jsou lineární formy. Pak o Z řekneme, že je to* projektivní změna souřadnic*.*

Projektivní změna souřadnic má mnoho dobrých vlastností – například zachovává algebraičnost množin a díky tomu umožňuje přesunout bod, který nás zajímá do počátku souřadnic, což může některé věci velmi usnadnit.

Definice 8 (Singulární bod)**.** *Nechť bod P leží na křivce F, pak řekneme, že bod P je singulární, pokud platí, že*

$$\frac{\partial F}{\partial x}(P) = \frac{\partial F}{\partial y}(P) = \frac{\partial F}{\partial z}(P) = 0.$$

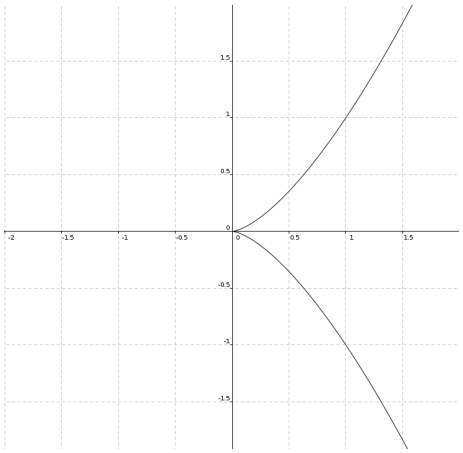
V opačném případě řekneme, že P je nesingulární

Dále definujeme násobnost bodu P na křivce jako nejmenší m takové, že alespoň jedna parciální derivace řádu m v bodě P je nenulová. Násobnost bodu P na křivce F budeme označovat $m_P(F)$.

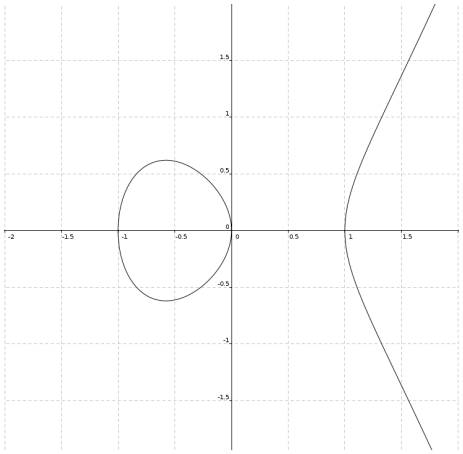
Singulární body jsou ty, ve kterých křivka protíná sama sebe, nebo ty, ve kterých má křivka „špičku“. Pokud nějaká křivka obsahuje jen nesingulární body, pak o ní řekneme, že je *nesingulární*. Jak takové body vypadají si ukážeme na příkladě afinních eliptických křivek a sice na obrázcích 2.1, 2.2 a 2.3.

Definice 9. *Nechť $E = V(\{e\})$ a $F = V(\{f\})$ jsou eliptické křivky. Pak pokud jsou e a f mají společného dělitele stupně* 1*, pak řekneme, že křivka generovaná tímto dělitelem je* společnou komponentou *E a F.*

Dále se nám bude hodit ztotožňovat projektivní rovinné křivky s formami, které je určují. Tedy zápis FG, kde $F = V(\{f\})$ a $G = V(\{g\})$ bude značit křivku určenou formou fg. Tedy $FG = V(\{fg\})$. Obdobně pro sčítání. Navíc



Obrázek 2.2: Příklad kubické křivky, která má v bodě (0,0) singulární bod „typu špička“.



Obrázek 2.3: Příklad kubické křivky, která nemá žádný singulární bod – je to eliptická křivka.

budeme dále požadovat, aby výsledný polynom byl bezčtvercový. Pokud ne, pak jeho faktory budeme brát v první mocnině.

Také budeme chtít nějak vyjádřit, „jak moc si jsou křivky blízko“ v jejich společném bodě. Následující definici, lze nalézt například v knize [Ful], v kapitole 3.

Definice 10 (Stupeň dotyku)**.** *Jako* stupeň dotyku křivek F a G v bodě P *označíme funkci, kterou budeme značit* $\mathrm{I}(P,F\cap G)$*, a která má následující vlastnosti:*

1. $\mathrm{I}(P,F\cap G)\in\mathbb{N}_0$*, pokud se* F *a* G *v* P *protínají,* $\mathrm{I}(P,F\cap G)=\infty$ *právě, když* P *leží ve společné komponentě* F *a* G*.*
2. $\mathrm{I}(P,F\cap G)=0$*, právě když* $P\notin F\cap G$*.*
3. *Nechť* Z *je projektivní změna souřadnic. Pak* $\mathrm{I}(P,F\cap G)=\mathrm{I}(Z(P),Z(F)\cap Z(G)$*.*
4. $\mathrm{I}(P,F\cap G)=\mathrm{I}(P,G\cap F)$
5. $\mathrm{I}(P,F\cap G)\geq m_P(F)m_P(G)$*, kdy rovnost nastává právě, když* F *a* G *nemají v* P *společnou tečnu.*
6. *Pokud* $F=\prod F_i$ *a* $G=\prod G_j$*, pak* $\mathrm{I}(P,F\cap G)=\sum_{i,j}\mathrm{I}(P,F_i\cap G_j)$
7. $\mathrm{I}(P,F\cap G)=\mathrm{I}(P,F\cap(G+AF))$*, pro libovolné* $A\in\mathbb{F}[x,y,z]$*.*

První vlastnost říká, že pokud mají dvě křivky společnou komponentu, tak si na ní „blíže už být nemohou“. Druhá vlastnost říká, že stupeň dotyku „vidí“ všechny průniky. Pátá vlastnost vyjadřuje to, že pokud pokud dvě křivky mají v nějakém bodě společnou tečnu, tak „si v něm jsou blíže“, než kdyby tomu tak nebylo.

Pomocí stupně dotyku je možné definovat inflexní body.

Definice 11. *Nechť* C *je křivka,* $P\in C$*,* L *je tečna k* C *v* P*. Pak o bodu* P *řekneme, že je inflexní, pokud* $\mathrm{I}(P,C\cap L)\geq 3$*.*

Projektivní prostory jsme zaváděli i proto, abychom měli jistotu, že se nám libovolné dvě přímky protnou v jednom bodě. Bézoutova věta (jejíž důkaz lze najít například v knize [Ful], v páté kapitole), kterou nyní vyslovíme, ale říká něco dokonce mnohem silnějšího.

Věta 3 (Bézoutova)**.** *Nechť* F *stupně* n *a* G *stupně* m *jsou projektivní rovinné křivky bez společné komponenty. Pak* $\sum_{P\in F\cap G}\mathrm{I}(P,F\cap G)=mn$*.*

2.3 Noetherova podmínka a sčítání na kubických křivkách

Nyní si budeme chtít definovat sčítání na kubických křivkách a ukázat, že tvoří grupu. Toto sčítání je použito v Goldwasser-Killianově testu. Tato podkapitola čerpá z páté kapitoly knihy [Ful]. Ta je také doporučena čtenáři, pokud by chtěl hlouběji proniknout do teorie, která se k tomuto tématu váže.

Nechť máme bod $P\in\mathbb{P}^2$. Pak označíme $\mathcal{O}_P(\mathbb{P}^2)$ množinu všech racionálních funkcí na $\mathbb{P}^2$ definovaných v P.

Definice 12 (Průnikový cyklus)**.** *Nechť F, G jsou rovinné křivky bez společné komponenty. Pak* průnikovým cyklem křivek F a G *nazveme formální součet* $F \bullet G = \sum_{P \in \mathbb{P}^2} \mathrm{I}(P,F \cap G)P$. *Jeho stupněm pak nazveme* $\sum_{P \in \mathbb{P}^2} \mathrm{I}(P,F \cap G)$. *Řekneme, že průnikový cyklus* $\sum q_P P$ *je větší než* $\sum r_P P$, *pokud pro každé* $P \in \mathbb{P}^2$ *platí, že* $q_P \geq r_P$.

Pokud je F stupně m a G stupně n, pak z Bézoutovy věty vidíme, že stupeň průnikového cyklu F a G je nejvýše mn.

Dále je pro průnikové cykly platí $F \bullet GH = F \bullet G + F \bullet H$ a $F \bullet (G + AF) = F \bullet G$, pokud A je forma stupně $deg(G) - deg(F)$.

Bude nás zajímat následující situace. Máme křivky F,G,H a jejich průnikové cykly $H \bullet F$ a $G \bullet F$. Lze najít křivku B tak, aby $H \bullet F - G \bullet F = B \bullet F$? Stačí najít takové formy A,B, aby $H = AF + BG$. V tom případě totiž $H \bullet F = (AF + BG) \bullet F = BG \bullet F = B \bullet F + G \bullet F$. Tuto situaci popisuje Noetherova věta.

Definice 13 (Noetherova podmínka)**.** *Řekneme, že Noetherova podmínka je splněna v P (vzhledem k F, G a H), pokud $H_* \in (F_*,G_*) \subset \mathcal{O}_P(\mathbb{P}^2)$*

Věta 4 (Max Noether)**.** *Nechť F,G a H jsou projektivní rovinné křivky. Nechť navíc F a G nemají společnou komponentu. Pak existují A, B formy, $deg(A) = deg(H) - deg(F), deg(B) = deg(H) - deg(G)$, že je splněna rovnice $H = AF + BG$, právě když je Noetherova podmínka v každém bodě $P \in F \cap G$.*

Tvrzení 5. *Nechť F,G,H jsou křivky, $P \in F \cap G$. Pak Noetherova podmínka je splněna v P, pokud platí některé z následujících:*

1. *F a G se kříží v P (tedy nedotýkají - nemají v P společnou tečnu), a $P \in H$.*
2. *P je nesingulární bod na F a $\mathrm{I}(P,H \cap F) \geq \mathrm{I}(P,G \cap F)$.*

Z tohoto tvrzení nám plyne užitečný důsledek.

Důsledek. Nechť F,G,H jsou křivky. Pokud platí některé následujících:

1. F a G se protínají v $deg(F) \cdot deg(G)$ různých bodech nebo
2. F a G se protínají v jen nesingulárních bodech a $H \bullet F \geq G \bullet F$,

pak existuje křivka B, že $H \bullet F = G \bullet F + B \bullet F$.

Lemma 6. *Nechť C, C', C'' jsou kubické křivky a C navíc ireducibilní. Nechť $C \bullet C' = \sum_{i=1}^{9} P_i$, kde P_i jsou nesingulární body. Nechť dále $C \bullet C'' = \sum_{i=1}^{8} P_i + Q$. Pak $P_9 = Q$.*

Důkaz. Nechť L je přímka procházející P_9, neprocházející Q. Pak $L \bullet C = P_9 + R + S$, kde R,$S \in C$. Pak $LC'' \bullet C = \sum_{i=1}^{8} P_i + Q + P_9 + R + S = \sum_{i=1}^{9} P_i + Q + R + S = C' \bullet C + Q + R + S$. Z důsledku 2 předchozího tvrzení plyne, že musí existovat přímka M, že $M \bullet C = Q + R + S$ (stačí brát $C = F$, $C' = G$, $LC'' = H$ a $B = M$). Přímka procházející body R a S je ale L. Tedy $L = M$ a $Q = P_9$.

□

Definice 14 (Sčítání na kubické křivce)**.** *Nechť C je nesingulární kubická křivka. Pak pro každé dva body P,$Q \in C$ existuje jediná přímka L taková, že $C \bullet L = P + Q + R$, pro nějaké $R \in C$ (Bézoutova věta). (Pokud $P = Q$, pak je L tečna k C v P.) Definujme $\phi : C \times C \to C$ tak, že $\phi(P,Q) = R$. Dále zvolme $O \in C$. Definujeme sčítání na kubické krřivce následovně: $P + Q = \phi(O,\phi(P,Q))$ (Bod O „uměle“ přidáváme proto, abychom pro sčítání měli neutrální prvek).*

Věta 7. *Body na kubické křivce tvoří s výše definovaným sčítáním abelovskou grupu. Bod O je pak vzhledem ke sčítání neutrálním prvkem.*

Důkaz. Komutativitu sčítání dostaneme snadno z toho, že přímka L, která je určená body P a Q, je jimy určena nezávisle na jejich pořádí.

Důkaz asociativity je těžší. Nechť P,Q,$R \in C$. Dále nechť $L_1 \bullet C = P+Q+S'$, $L_2 \bullet C = S+R+T'$, $L_3 \bullet C = O+U+U'$, $M_1 \bullet C = O+S+S'$, $M_2 \bullet C = Q+R+U'$ a $M_3 \bullet C = P + U + T''$. Platí $(P + Q) + R = \phi(O,T')$, $P + (Q + R) = \phi(O,T'')$. Stačí tedy ukázat, že $T = T'$. Definujme $C' = L_1L_2L_3$, $C'' = M_1M_2M_3$. $T = T'$ pak plyne z lemmatu.

Mějme body $O, P \in C$. Pak $\phi(O,P) = S$ je třetí bod na přímce určené body O a P, $P + O = \phi(O,S)$, a tedy jde o třetí bod na přímce určené body O a S. Musí tedy jít o bod P. O je proto vzhledem ke sčítání neutrálním prvkem.

Opačným prvkem je pak bod S z důkazu existence neutrálního prvku.

□

2.3.1 Eliptické křivky

Definice 15 (Projektivní eliptická křivka)**.** *Jako* projektivní eliptickou křivku *označíme projektivní algebraickou křivku stupně 3, která je nesingulární.*

Pokud bychom pracovali nad algebraicky uzavřeným tělesem, jehož charakteristika není 2 ani 3, pak lze dokázat, že nad takovým tělesem má každá eliptická křivka právě 9 inflexních bodů. Vhodnými substitucemi a projektivními změnami souřadnic je možné docílit toho, že v nekonečnu je jeden z inflexních bodů, a tečna k němu je H_∞. Tedy z věty 3 plyne, že je to jediný bod křivky v nekonečnu. Ten je dobré volit jako bod O z definice sčítání na eliptické křivce.

Lze také dokázat, že pro afinní eliptické křivky nad takovým tělesem, které dostaneme dehomogenizací jejich projektivních protějšků, lze převést polynom, jímž je daná eliptická křivka zadaná, do tvaru $y^2 - x^3 - Ax - B$.

V Goldwasser-Killianově algoritmu budeme pracovat s takovými eliptickými křivkami, které jsou ale definované nad konečným tělesem.

3. Úvod do eliptických křivek

Cílem této kapitoly je poskytnout čtenáři základní informace o eliptických křivkách, které potřebuje k tomu, aby pochopil, jak funguje Goldwasser-Killianův algoritmus pro ověření prvočíslenosti.

3.1 Základní definice

Definice 16 (Weierstrassův tvar)**.** *Nechť* $\mathbb{F}$ *je těleso,* $char(\mathbb{F}) \neq 2,3$*. Pak* eliptickou křivkou ve Weierstrassově tvaru *nazveme uspořádanou dvojici* (A,B) *takovou, že* $A,B \in \mathbb{F}$*, a navíc* $4A^3 + 27B^2 \neq 0$*.*

Podmínka $4A^3 + 27B^2 \neq 0$ je zde proto, abychom měli zajištěno, že naše křivka bude nesingulární. Lze totiž ukázat, že v singulárním bodě má křivka násobný kořen (například lze změnou souřadnic převést singulární bod do počátku souřadnic a zderivovat - bod $x = 0$ bude kořenem derivace křivky a tedy násobným kořenem křivky). A pokud platí, že $4A^3 + 27B^2 = 0$, pak se lze přesvědčit, že jedna hodnota z $x = \pm\sqrt{A/3}$ je násobným kořenem křivky.

Dále budeme uvažovat jen eliptické křivky ve Weierstrassově tvaru.

Definice 17. *Nechť* $\mathbb{F}$ *je těleso,* $char(\mathbb{F}) \neq 2,3$*, a nechť* (A,B) *je eliptická křivka nad* $\mathbb{F}$*. Definujeme množinu bodů této eliptické křivky jako množinu uspořádaných dvojic* $(x,y) : x,y \in \mathbb{F}$ *takových, že* $y^2 = x^3 + Ax + B$*, ke kterým přidáme navíc bod* I *(„bod v nekonečnu“, bod* O *ze sčítání na eliptické křivce). Budeme pro ně používat označení* $E_{A,B}(\mathbb{F})$*, pokud* $\mathbb{F}$ *bude Galoisovo těleso s* q *prvky, pak budeme psát* $E_{A,B}(q)$*.*

3.2 Sčítání bodů na křivce

V minulé kapitole jsme ukázali, že sčítání na eliptické křivce nad tělesem tvoří grupu. Nyní si ukážeme, jak funguje v afinním případě, a k tomu algebraická pravidla pro výpočet souřadnic součtu dvou bodů.

Nechť dále $L = (x_l,y_l), M = (x_m,y_m) \in E_{A,B}(\mathbb{F})$. Součet bodů na eliptické křivce lze dobře definovat následovně:

Pro každý bod L na křivce definujeme $L + I = L$.

Pro křivky nad reálnými čísly můžeme použít geometrickou představu: Body L a M povedeme přímku, najdeme další průsečík s křivkou a ten zobrazíme osovou souměrností podle osy x. O tomto bodě prohlásíme, že jde o bod $L + M$.

Pokud mají body L a M opačnou souřadnici x, definujeme $L + M = I$. Pokud chceme sečíst bod L sám se sebou, pak jako přímku, na které budeme hledat další průsečík, zvolíme tečnu ke křivce v bodě L, a jako bod $L + L$ (budeme psát $2L$) označíme obraz dalšího průsečíku po osové symetrii vzhledem k ose x. Pokud bude mít přímka vedené body L a M jen dva průsečíky s křivkou, pak víme, že pak je v jednom z nich tato přímka tečnou ke křivce. A budeme tento bod brát jako třetí bod na přímce. Konečně, budeme-li chtít sečíst bod L sám se sebou, a tečna ke křivce v bodě L nebude mít s křivkou žádné další průniky, pak jako třetí průsečík na přímce zvolíme bod L.

Výše uvedená pravidla můžeme zapsat algebraicky, což nám dá algoritmus 1. Pro zvídavého čtenáře vysvětlíme proč. Proměnná s je zde směrnicí přímky dané body L a M. Tu dostaneme pro dva různé body snadno, a pokud sčítáme bod sám se sebou, pak lze vzorec pro směrnici dostat pomocí implicitního derivování. Jak lze odvodit vzorce pro bod (x_{res},y_{res}) si teď ukážeme v obecném případě. Uvažujme zápis přímky, jejíž průsečíky nás zajímají, v této formě: $y = s * x + b$. Víme, že eliptickou křivkou zadanou rovnicí: $y^2 = x^3 + Ax + B$ protíná právě ve třech bodech (při započtení násobností) a dva z nich známe. Do rovnice popisující křivku dosadíme za y a po úpravách dostaneme následující: $x^3 - sx^2 + (A - 2sb)x + B - b^2 = 0$. Viètovy vzorce ale říkají, že $s^2 = x_l + x_m + x_{res}$, a tedy dostáváme vzorec pro x_{res}. Ze vzorce pro křivku pak již snadno dopočítáme y_{res}.

Algorithm 1 Sčítací algoritmus pro body na eliptické křivce (*SČÍTEJ*)

VSTUP: (A,B) - eliptická křivka, $L = (x_l, y_l), M = (x_m, y_m)$ - body na (A,B)
switch [$(x_l,y_l),(x_m,y_m)$]
  case [$x_l = x_m$ **and** $y_l = -y_m$]
    return I
  case [$x_l = x_m$ **and** $y_l = y_m$]
    $s \leftarrow \frac{3x_l + A}{2y_l}$
  case [$x_l \neq x_m$]
    $s \leftarrow \frac{y_m - y_l}{x_m - x_l}$
end switch
$x_{res} = s^2 - x_l - x_m$
$y_{res} = s(x_l - x_{res}) - y_l$
return (x_{res},y_{res})

Dále definujeme qL, kde $q \in \mathbb{F}, L \in E_{A,B}(\mathbb{F})$, přirozeně jako opakované sčítání. To lze efektvině spočítat obdobou algoritmu pro binární mocnění:

$$qL = \begin{cases} I, & \text{pro } q = 0, \\ L, & \text{pro } q = 1, \\ (L + L) * q/2, & \text{pro } q \text{ sudé} \\ L + (q - 1)L, & \text{pro } q \text{ liché} \end{cases}$$

3.3 Sčítání nad $\mathbb{Z}_n$

Je zřejmé, že naše vzorce pro sčítání bodů na eliptické křivce můžeme použít nad libovolným okruhem $\mathbb{Z}_n$. Nicméně se může stát, že náš algoritmus v takovém případě selže - například nemáme zajištěnou existenci inverzních prvků. To ovšem nutně znamená, že n musí být složené! Dále si všimneme, že námi definované sčítání se chová rozumně, pokud budeme brát souřadnice bodů modulo p, kde $p|n$.

Definice 18. *Nechť $p > 3$, $p|n$. Pro každé $x \in \mathbb{Z}_n$ definujeme x_p jako přirozenou projekci x do $\mathbb{Z}_p$ – budeme brát koeficienty ze $\mathbb{Z}_n$ modulo p. Pro bod $L = (x,y) \in E_{A,B}(\mathbb{Z}_n)$ definujeme $L_p = (x_p,y_p) \in E_{A,B}(p)$. Dále definujeme $I = I_p$.*

Lemma 8. *Nechť $L,M \in E_{A,B}(\mathbb{F})$ Pokud je $L + M$ definováno, pak $(L + M)_p = L_p + M_p$.*

Důkaz. Lemma zjevně platí, pokud $L = I$ nebo $M = I$. Dále si všimneme, že pro každou racionální funkci R nad $\mathbb{Z}_n$ platí, že $(R(x_1,x_2...))_p = (R((x_1)_p,(x_2)_p...)_p$ mají-li obě strany smysl (z vlastností přirozené projekce). Sčítací algoritmus máme definován tak, že rozlišuje tři případy. Pokud jak $(L + M)_p$, tak $L_p + M_p$ spadají pod stejný případ, pak z předchozí úvahy víme, že lemma platí.

Zbývá jen rozmyslet si, co se stane, pokud bychom $L_p + M_p$ a $(L + M)_p$ museli řešit různě. V tom případě by nutně musela nastat jedna z následujících možností:

1) $x_l = x_m$, ale $y_l \neq \pm y_m$

2) $x_l \neq x_m$, ale $(x_l)_p = (x_m)_p$

První případ sčítací algoritmus neuvažuje, a proto v takovém případě není výstup definován. V druhém případě platí, že $p|(x_l - x_m)$, a sčítací algoritmus selže na hledání inverzu k $(x_l - x_m)$. To znamená, že $x_l - x_m$ a n jsou soudělné, a Euklidovým algoritmem můžeme nalézt dělitele. To by napovídalo, že lze eliptické křivky využít i k faktorizaci. A vskutku, v článku [Len87] autor popisuje algoritmus, který toto dokáže.

Případy jako $y_l = \pm y_m$, ale $(y_l)_p \neq \pm(y_m)_p$ nemusíme uvažovat, protože $a = b$ implikuje $a_p = b_p$ a stejně $a = -b$ implikuje $a_p = -b_p$ (z vlastností přirozené projekce). Definici qL tedy doplníme o to, že pokud při běhu algoritmu nastane chyba, pak qL není definováno. To nám ale nevadí, protože díky tomu zjistíme, že n je složené.

□

3.4 Křivky jako grupy nad $\mathbb{Z}_p$

Dokázali jsme, že body na eliptické křivce tvoří abelovskou grupu. Lze dokázat, že každá konečná abelovská grupa je izomorfní nějaké grupě $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_n}$ pro nějaká $m_i \in \mathbb{N}$ taková, že $m_i|m_{i+1}$. Nás budou zajímat křivky s $2q$ body, kdy q je liché prvočíslo.

Tvrzení 9. *Nechť (A,B) eliptická křivka na tělesem $\mathbb{F}$. Nechť $|E_{A,B}(\mathbb{F})| = 2q$, kde q je liché prvočíslo. Pak $E_{A,B}(\mathbb{F})$ je izomorfní grupě $\mathbb{Z}_{2q}$.*

Důkaz. Děliteli čísla $2q$ jsou jen čísla: $1,2,q,2q$. Z důvodů velikosti grupy, a dělitelnosti čísla $2q$ tedy musí být izomorfní grupě $\mathbb{Z}_{2q}$.

□

Dále se budeme zajímat o velikost grup bodů na eliptické křivce. Lze si snadno uvědomit, že vzhledem k tomu, že body na křivce $E_{A,B}(p)$, máme definované jako uspořádané dvojice $(x,y), x,y \in \mathbb{Z}_p$ a k nim přidaný bod I, může být maximální velikost takové grupy $p^2 + 1$. Lze ale dokázat daleko lepší odhad. Lze jej společně s jeho důkazem najít ve čtvrté kapitole [Was08].

Věta 10 (Hasseho odhad)**.** *Nechť $E_{A,B}(p)$ je množina bodů eliptické křivky nad konečným tělesem s p prvky. Pak $||E_{A,B}(p)| - p - 1| \leq 2\sqrt{p}$.*

Dále se pro analýzu doby běhu algoritmu použije následující věta, která dává informaci o rozložení $|E_{A,B}(p)|$. Lze ji najít v lehce obměněné podobě v [Len87].

Věta 11 (Lenstrova)**.** *Nechť $p > 5$ prvočíslo. Nechť $S \subseteq [p+1-\lfloor\sqrt{p}\rfloor, p+1+\lfloor\sqrt{p}\rfloor]$. Pak pokud je křivka (A,B) vybrána náhodně s rovnoměrným rozdělením, pak pravděpodobnost*

$$P[|E_{A,B}(p)| \in S] > \frac{c}{\log(p)} * \frac{|S| - 2}{2\lfloor\sqrt{p}\rfloor + 1}.$$

Nyní vyslovíme větu, na které stojí Goldwasser-Killianův test. Díky ní lze redukovat problém prvočíselnosti čísla p, na problém prvočíselnosti čísla q, které je mnohem menší.

Věta 12 (Redukční)**.** *Nechť n je číslo, které je nesoudělné s 6. Nechť (A,B) je eliptická křivka, s koeficienty ze $\mathbb{Z}_n$, $L \neq I \in E_{A,B}(\mathbb{Z}_n)$. Potom pokud $qL = I$ pro nějaké prvočíslo $q > n^{1/2} + 2n^{1/4} + 1$, pak n je prvočíslo.*

Důkaz. Předpokládejme pro spor, že n je složené. Pak existuje jeho prvočíselný dělitel $p \neq 2,3$, $p < \sqrt{n}$. Navíc, pokud budeme uvažovat přirozenou projekci L_p bodu L, pak si uvědomíme, že $qL_p = I$. Tedy řád L_p musí dělit q. Protože q je prvočíslo, tak musí být dokonce rovný q. Ale projekce křivky (A,B) do $\mathbb{Z}_p$ může obsahovat maximálně $p + 2\sqrt{p} + 1 < q$ bodů. Což nám dává spor.

□

4. Goldwasser-Killianův test prvočíslenosti

V této kapitole se seznámíme s Goldwasser-Killianovým testem prvočíselnosti. Tato kapitola čerpá z článků [GK86] a [GK99], ve kterých autoři testu vysvětlují svůj přístup k řešení tohoto problému.

4.1 Popis algoritmu

Algoritmus se skládá ze tří hlavních částí - generování vhodné eliptické křivky o $2q$ bodech (kde q je prvočíslo), výběru vhodného bodu na křivce, a použití redukční věty k redukci problému.

Idea algoritmu je tedy následující: důkaz prvočíselnosti zadaného čísla p, se pomocí věty 12 převede na důkaz prvočíselnosti vhodně zvoleného čísla q, kdy platí, že $q \approx p/2$, a poté se rekurzivně pokračuje. Rekurze skončí ve chvíli, kdy je číslo q_i, na jehož prvočíselnost byl důkaz prvočíselnosti p redukován, tak malé, že jeho prvočíselnost může být rychle dokázána jiným algoritmem. Z jistého důvodu ale chceme, aby tato dolní mez byla vyšší než 37. Přičemž pokud algoritmus běží příliš dlouho, pak jej zastavíme a spustíme znovu. Schoofův algoritmus je pak podrobněji popsán například v [Sch85]. Tam také Schoof představuje (v praxi pomalý) algoritmus pro počítání druhé odmocniny modulo p.

Algorithm 2 Generování křivky (*GENERUJ*)

VSTUP: p (dokazované prvočíslo)
1: **repeat**
2: náhodně generuj křivky (A,B)
3: **until** $GCD(4A^3 + 27B^2,p) == 1$
4: pomocí Schoofova algoritmu spočítej $|E_{A,B}(p)|$
5: **if** $|E_{A,B}(p)| \equiv 1 \bmod 2$ **then**
6: **goto** 1.
7: **else**
8: $q \leftarrow |E_{A,B}(p)|/2$
9: **end if**
10: nech q projít $2k$ pravděpodobnostními prvočíslenými testy (kde p je $k-$bitové).
11: **if** q neprošlo testem **or** q je dělitelné 2 nebo 3 **then**
12: **goto** 1.
13: **else**
14: **return** ((A,B),q)
15: **end if**

Tímto algoritmem hledáme takovou křivku, že $|E_{A,B}| = 2q$, pro nějaké q prvočíslo. O čísle q pak chceme vědět, že jde o prvočíslo s dost vysokou pravděpodobností, a vždy po něm chceme, aby nebylo dělitelné 2 nebo 3.

Algorithm 3 Výběr bodu na křivce (*VYBER*)

VSTUP: p,q,(A,B)
1: **repeat**
2: náhodně vyber $x \in \mathbb{Z}_p$
3: **until** $z = x^3 + Ax + B$ je kvadratický zbytek modulo p
4: $y \leftarrow \sqrt{z}$ (náhodně zvol druhou odmocninu)
5: $L \leftarrow (x,y)$
6: **if** $qL \neq 1$ **then**
7: **goto** 1.
8: **else**
9: **return** L.
10: **end if**

Tímto algoritmem hledáme na zadané křivce bod řádu q. Protože máme křivku řádu $2q$, víme, že bude izomorfní grupě $\mathbb{Z}_{2q}$ z tvrzení 9, a tedy obsahuje jen prvky řádu 1, 2, q, $2q$. A protože q je liché, stačí jen spočítat qL, abychom zjistili, zda je L řádu q. Že je číslo z kvadratickým zbytkem modulo p pak znamená, že existuje $c \in \mathbb{Z}_p$ takové, že $c^2 = z \bmod p$. Snadno si lze uvědomit, že jich bude zhruba polovina – existují „dvě druhé odmocniny“.

Algorithm 4 Redukční krok (*REDUKUJ*)

VSTUP: p (dokazované prvočíslo)
1: $((A,B),q) \leftarrow GENERUJ(p)$
$L \leftarrow VYBER(p,q,(A,B))$
2: **return** $((A,B),L,q)$

Tento algoritmus využívá věty 12 k redukci problému.

Celý test (algoritmus 5) pak probíhá tak, že opakuje redukční krok do té doby, než hodnota čísla, jehož prvočíselnost je potřeba dokázat aby byla dokázána prvočíselnost zadaného čísla, klesne natolik, že je možné rychle ověřit prvočíselnost tohoto čísla nějakým deterministickým testem.

4.2 Certifikát prvočíselnosti

Výstup tohoto testu je tvaru $((A_0,B_0), L_0, p_1)$, …, $((A_{k-1},B_{k-1}), L_{k-1}, p_k)$. Ukážeme, že s jeho pomocí můžeme rychle ověřit, že v našem algoritmu nenastala chyba, a že p je skutečně prvočíslo. Z toho důvodu budeme takovému výstupu říkat Goldwasser-Killianův certifikát.

Tvrzení 13. *Pokud ověřovací algoritmus přijme vstup tvaru* $p, ((A_0,B_0), L_0, p_1)$, …, $((A_{k-1},B_{k-1}), L_{k-1}, p_k)$, *pak je* p *prvočíslo.*

Důkaz. Pokud ověřovací algoritmus přijme vstup tvaru $p, ((A_0,B_0), L_0, p_1)$, …, $((A_{k-1},B_{k-1}), L_{k-1}, p_k)$, pak zjevně p_k musí být prvočíslo. Dále v každém kroku využíváme větu 12, abychom ověřili, že pokud p_i je prvočíslo, pak i p_{i-1} je prvočíslo. Tedy pokud náš vstup projde ověřovacím algoritmem, pak dostáváme řetěz implikací: p_i je prvočíslo $\Rightarrow$ p_{i-1} je prvočíslo $\Rightarrow$ … $\Rightarrow$ $p_0 = p$ je prvočíslo.

Algorithm 5 Goldwasser-Killianův test

VSTUP: p (dokazované prvočíslo), $DOLNÍ\_MEZ$ (kam chceme redukovat)

certifikát $\leftarrow p$
$p_0 \leftarrow p$
$i \leftarrow 0$
repeat
    certifikát $\leftarrow$ (certifikát, $REDUKUJ(p_i)$)
    $i \leftarrow i + 1$
until $p_i <$*DOLNÍ_MEZ* **or** cyklus běží příliš dlouho
if cyklus běžel příliš dlouho **then**
    goto 1.
else
    if p_i je prvočíslo **then**
        return certifikát
    else
        goto 1.
    end if
end if

Algorithm 6 Ověřovací algoritmus pro Goldwasser-Killianův certifikát

VSTUP: p, $((A_0,B_0),L_0,p_1)$, …, $((A_{k-1},B_{k-1}),L_{k-1},p_k)$

if $p_i > max(DOLNÍ\_MEZ,37)$ **then**
    return NEPLATNÝ VSTUP
end if
if p_i není prvočíslo **then**
    return NENÍ PRVOČÍSLO
else
    $p_0 \leftarrow p$.
    for all $j \in [0,i-1]$ **do**
        if $p_j \bmod 2 = 0$ **then**
            return NENÍ PRVOČÍSLO
        else if $p_j \bmod 3 = 0$ **then**
            return NENÍ PRVOČÍSLO
        else if $GCD(4A_j^3 + 27B_j^2,p_j) \neq 1$ **then**
            return NENÍ PRVOČÍSLO
        else if $p_{j+1} \leq p_j^{1/2} + 2p_j^{1/4} + 1$ **then**
            return NENÍ PRVOČÍSLO
        else if $p_{j+1}L_j \neq I_{p_j}$ **then**
            return NENÍ PRVOČÍSLO
        else if $L_j = I_{p_j}$ **then**
            return NENÍ PRVOČÍSLO
        end if
    end for
end if
return JE PRVOČÍSLO

□

Nyní si uvědomíme, že ověřovací algoritmus vždy přijme Goldwasser-Killianův certifikát. Z definice Goldwasser-Killianova testu je p_i prvočíslo. Z definice algoritmu *GENERUJ* vidíme, že $\mathrm{GCD}(4A_j^3 + 27B_j^2, p_j) = 1$. Z definice algoritmu *GENERUJ* a z Hasseho odhadu vidíme, že $p_{j+1} \geq (p_j + 1 - 2\sqrt{p_j})/2$. O tom ale můžeme snadno zjistit, že pro $p_j > 37$ je větší než $p_j^{1/2} + 2p_j^{1/4} + 1$. Z definice Goldwasser-Killianova testu je ale $p_j > 37$. Konečně z kroku *VYBER* vidíme, že $L_j \neq I_{p_j}$ a $p_{j+1}L_j = I_{p_j}$. A tedy ověřovací algoritmus přijme Goldwasser-Killianův certifikát.

Ukážeme, kolik operací je tedy potřeba pro ověření l-bitového čísla p. Všimneme si, že $p_{j+1} = p_j/2 + o(p_j)$, a tedy $k = O(\log(p)) = O(l)$. V každém z k kroků algoritmu je třeba provést konstantní počet jednoduchých aritmetických operací, jeden výpočet GCD a vynásobení bodu L_j číslem q_j. To vše lze stihnout v čase $O(l^3)$ [Sta11], a tedy lze ověřovacím algortimem ověřit Goldwasser-Killianův certifikát l-bitového čísla v čase $O(l^4)$.

4.3 Rychlost algoritmu

Již víme, že Goldwasser-Killianův algoritmus funguje, ale nevíme, jaká je jeho asymptotická složitost.

4.3.1 Analýza redukčního kroku

Nyní provedeme analýzu běhu redukčního kroku algoritmu v závislosti na tom, kolik prvočísel pro něj připadá v úvahu. Tedy definujme $S(p)$ jako

$$S(p) = \{q : q \in \left[\frac{p+1-\sqrt{p}}{2}, \frac{p+1+\sqrt{p}}{2}\right], q \text{ je prvočíslo}\}$$

Nejprve ukážeme, jak dlouho trvá generování křivky pomocí algorimtu *GENERUJ*. Zjevně je potřeba vynásobit čas nutný pro vygenerování a otestování jedné křivky, zda je vhodného řádu, s očekávaným počtem křivek, které je potřeba vyzkoušet, než narazíme na vyhovující.

Nejnákladnější krok z kroků nutných pro vygenerování a otestování křivky je zjištění počtu jejích bodů pomocí Schoofova algoritmu, jehož asymptotická složitost je $O(\log^8(p))$ ([IKY00] sekce 2.1). Všechny ostatní kroky (počítání GCD pomocí Euklidova algoritmu a pravděpodobnostní prvočíselné testy), mají menší složitost [Sta11].

Zjistíme tedy, kolik křivek (A,B) takto musíme otestovat. Snadno si uvědomíme, že pro prvočíslo p libovolné A existují pouze dvě hodnoty B (a sice $\pm\sqrt{4A^3/27}$) takové, že $4A^3 + 27B^2 = 0$. Tedy pokud náhodně vybereme A a B, pak téměř jistě bude (A,B) eliptická křivka.

Zbývá tedy jen určit, kolik takových křivek je řádu $2q$, kde q je prvočíslo. Snadnou aplikací lemmatu 11 si uvědomíme, že platí:

$$P[|E_{A,B}| \text{ je tvaru } 2q] > \frac{c}{\log(p)} * \frac{|S(p)| - 2}{2\lfloor\sqrt{p}\rfloor + 1}.$$

Existuje totiž bijekce mezi prvočísly z intervalu $\left[\frac{p+1-\sqrt{p}}{2}, \frac{p+1+\sqrt{p}}{2}\right]$ a dvojnásobky prvočísel z intervalu $[p+1-\lfloor\sqrt{p}\rfloor\, p+1+\lfloor\sqrt{p}\rfloor]$.

Tedy můžeme formulovat následující tvrzení:

Tvrzení 14. *Nechť p je prvočíslo a $k = \log(p)$. Potom pokud platí, že $|S(p)| = O(\sqrt{p}/\log^c(p))$, pak algoritmus GENERUJ skončí v očekávaném čase $O(k^{9+c})$*

Důkaz. Víme, že testování křivky má složitost $O(k^8)$ (Schoofův algoritmus). A budeme ho muset provést asi $(\frac{c_1}{\log(p)} * \frac{\sqrt{p}-2}{\log^c(p)(2\lfloor\sqrt{p}\rfloor+1)})^{-1} = O(\log^{c+1}(p)) = O(k^{c+1})$-krát.

□

Proč jsme zvolili zrovna tuto podmínku ($|S(p)| = O(\sqrt{p}/\log^c(p))$), se ukáže, až budeme dokazovat složitost celého algoritmu. Nyní ověříme, že složitost algoritmu *VYBER* není větší než složitost algortimu *GENERUJ*.

Předpokládejme, že máme křivku $E_{A,B}(p)$ řádu $2q$. Z lemmatu 9 víme, že grupa jejích bodů je izomorfní $\mathbb{Z}_{2q}$. O té víme, že má $q-1$ bodů řádu q.

Všimneme si, že je můžeme spárovat, protože platí $(x,y) + (x,-y) = I$, pak pokud je (x,y) řádu q, musí být i $(x,-y)$ řádu q. Tedy je asi $(q-1)/2$ možností jak zvolit x a $y = \sqrt{x^3 + Ax + B}$ tak, aby bod (x,y) byl řádu q. Očekávaná složitost tohoto kroku je tedy $O(2p/(q-1)) = O(1)$.

Dále je pravda, že můžeme zjistit, že $z = x^3 + Ax + B$ je kvadratický zbytek, a spočítat $\sqrt{z}$ se složitostí $O(k^4)$. Postup je možné najít například v [CSF12]. K spočítání qL je třeba $O(k^3)$ kroků pomocí binárního algoritmu, a tedy složitost algoritmu *VYBER* je $O(k^4)$.

Tvrzení 15. *Nechť platí, že $|S(p)| = O(\sqrt{p}/\log^c(p))$. Pak očekávaná složitost algoritmu REDUKUJ je $O(k^{c+9})$.*

Důkaz. V redukčním kroku jednou provádíme dva algoritmy. Právě jsme dokázali (za daných předpokladů), že jsou oba $O(k^{c+9})$.

□

4.3.2 Analýza celého algoritmu

Ukážeme, že pokud platí jistá doměnka, pak tento test vždy skončí v očekávaném polynomiálním čase.

Věta 16. *Nechť $\pi(x)$ značí počet prvočísel menších než x. Nechť dále platí, že $\exists c_1,c_2 > 0 : \pi(x+\sqrt{x}) - \pi(x) \geq \frac{c_2\sqrt{x}}{\log^{c_1}(x)}$. Pak Goldwasser-Killianův test skončí v očekávaném čase $O(\log^{c_1+10}(p))$.*

Poznámka. Platnosti výše uvedené domněnky nasvědčuje (s $c_1 = 1$) například prvočíselná věta, která říká, že $\pi(x) \sim x/\ln(x)$.

Nejprve budeme pro jednoduchost předpokládat, že se algorotmus nikdy nesplete, tedy že všechna čísla, která projdou pravděpodobnostními testy, jsou opravdu prvočísla, a že algoritmus nikdy nespustíme znovu, protože by běžel příliš dlouho.

Pro přehlednost označíme $x = p + 1 - \lfloor\sqrt{p}\rfloor$ a $y = p + 1 + \lfloor\sqrt{p}\rfloor$. Potom $S(p) = \{q \in [x,y]\,, q \text{ prvočíslo}\}$. Pokud $p > 37$, pak $y > x + \sqrt{x}$, a tedy $S(p)$ obsahuje $O(\sqrt{x}/\log^{c_1}(x))$ prvočísel. Stejně jako v důkazu složitosti algoritmu *GENERUJ* vidíme, že algoritmus *GENERUJ* bude běžet v čase $O(\log^{c_1+9}(p))$. Protože se s každým krokem rekurze dokazované prvočíslo zmenší zhruba na polovinu, budeme potřebovat dokázat $O(\log(p))$ prvočísel. Celkem tedy algoritmus poběží v čase $O(\log^{c_1+10}(p))$.

Nyní do našeho odhadu započítáme chyby. Předpokládejme, že algoritmus spustíme znovu vždy po K krocích. Dále označme ρ pravděpodobnost chyby (nějaké q nebude prvočíslo) a jako D označíme očekávaný počet kroků, které algoritmus provede, pokud nenastane chyba. Zjevně pak počet kroků algoritmu bude

$$(\rho + \rho^2 + \rho^3 + ...)K + D = O(K\rho + D),$$

je-li ρ menší než 1 (ρ - pravděpodobnost, že jsme jednou museli algoritmus spustit znovu, ρ^2 - pravděpodobnost, že jsme jej museli spustit znovu dvakrát...). K můžeme zvolit tak, aby $K = k^{\log(k)}$, a o D víme, že $D = O(k^{c_1+10})$. Zbývá ještě omezit ρ. V průběhu algoritmu se provede maximálně K ověření prvočíselnosti pravděpodobnostním testem. Každý může selhat s pravděpodobností $1/$*DOLNÍ_MEZ*. *DOLNÍ_MEZ* pak můžeme zvolit tak malou, aby byla $O(1/K^2)$, pro velká p. Dále z Čebyševovy nerovnosti víme, že pokud nenastane chyba v prvočíselném testu, pak algoritmus provede víc než K kroků s pravděpodobností menší než D/K. Celkem tedy můžeme očekávat, že algoritmus poběží v $O(((D/K) + 1/K^2)K + D) = O(D)$ kroků.

Platnost doměnky ovšem neumíme dokázat pro všechna prvočísla. Ovšem je možné omezit počet těch prvočísel, pro které doměnka neplatí, což použili Goldwasser a Killian v [GK99] k tomu, aby ukázali, že algoritmus běží v očekávaném polynomiálním čase pro všechna až na zanebatelně málo prvočísel.

4.3.3 Úpravy Goldwasser-Killianova algoritmu

Goldwasser-Killianův algoritmus dává dobré teoretické výsledky, ale přesto není dokonalý. Některými problémy jsou, že se nepodařilo prokázat, že běží rychle pro úplně všechna prvočísla (což je spíše teoretický problém) a to, že Schoofův algoritmus, který je v něm použit, je i přes svou asymptotickou rychlost velmi pomalý (což byl spíše historický problém - nyní lze použít Schoof-Elkies-Atkinův algoritmus, který v praxi běží o mnoho rychleji).

Adlemanovi a Huangovi se ale v [AH14] povedlo upravit tento postup tak, aby běžel v očekávaném polynomiálním čase pro všechna prvočísla. Jejich postupem (používají jiné křivky) se „redukuje“ otázka prvočíselnosti čísla p na prvočíselnosti čísla p', kde $p' \approx p^2$. Což je sice větší, ale vybráno náhodně, a proto pro něj nejspíše bude platit tato doměnka. A pokud ne, může se vygenerovat jiné p'.

Atkinovi a Morrainovi se pak v [AM93] povedlo obejít zdlouhavé počítání bodů Schoofovým algoritmem tím způsobem, že místo náhodného zkoušení křivek a následného počítání bodů generují křivku takovým způsobem (pomocí komplexního násobení), že je poté možné rychle určit její řád. Tato verze algoritmu je jednou z nejrychlejších pro praktické dokazování prvočíselnosti čísel, která nejsou speciálního tvaru.

Závěr

Cílem práce bylo seznámit čtenáře s algoritmy, které je možné použít k důkazu prvočíselnosti zadaného čísla. Největší důraz byl kladen na Goldwasser-Killianův algoritmus. Dále bylo cílem práce zároveň čtenáři poskytnout určity vhled do teorie za tímto algoritmem. Proto byla zařazena kapitola o projektivní geometrii, která, ač je z teoretického hlediska velmi důležitá, ve většině článků na toto téma chybí, neboť se předpokládá, že je s ní čtenář již seznámen, a nebo jej příliš nezajímá. Mám ale dojem, že právě základní znalosti z projektivní geometrie mohou čtenáři poskytnout jistou intuici, díky které se mu bude v tématu lépe orientovat.

V první kapitole byly čtenáři představeny některé algoritmy, které je možné použít k dokázání prvočíselnosti zadaného čísla. U některých bylo navíc ukázáno proč fungují tak, jak fungují.

Ve druhé kapitole byla vybudována teorie potřebná k zavedení pojmu eliptické křivky, a dále je ukázáno, jak lze na eliptické křivce definovat sčítání. Navíc je dokázáno (pomocí několika nedokazovaných tvrzení), že sčítání na eliptické křivce tvoří grupu.

Ve třetí kapitole se čtenář seznámil s početní stránkou věci, a bylo ukázáno, jak odvodit vzorce pro sčítání dvou bodů na afinní eliptické křivce. Také bylo ukázáno ukázáno, jakou strukturu musí mít grupa o velikosti $\mathbb{Z}_{2q}$, kde q je liché prvočíslo.

Ve čtvrté kapitole byl pak čtenáři konečně představen a vysvětlen Goldwasser-Killianův algoritmus. Dále byla zanalyzována časová složitost algoritmu za předpokladu jisté (pravděpodobné) hypotézy. V závěru kapitoly (a celé práce) pak byly zmíněny jisté možné úpravy tohoto postupu, které mají nějaký praktický, nebo teoretický význam.

Celkem by čtenář po přečtení této práce měl nejen vědět, jak Goldwasser-Killianův algoritmus (a některé další) funguje, ale měl by mít i určitou, byť matnou představu o tom proč.

Seznam použité literatury

[AH14] L.M. Adleman and M.D.A. Huang. *Primality Testing and Abelian Varieties Over Finite Fields*. Springer, 2014.

[AKS04] M. Agraval, S. Kayal, and N. Saxena. Primes is in p. *Annals of Mathematics*, 160:781–793, 2004.

[AM93] A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61(203):29–68, 1993.

[CSF12] Zhengjun Cao, Qian Sha, and Xiao Fan. *Information Security and Cryptology: 7th International Conference, Inscrypt 2011, Beijing, China, November 30 – December 3, 2011. Revised Selected Papers*, chapter Adleman-Manders-Miller Root Extraction Method Revisited, pages 77–85. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Ful] William Fulton. Algebraic curves (an introduction to algebraic geometry).

[GK86] S Goldwasser and J Kilian. Almost all primes can be quickly certified. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, pages 316–329, New York, NY, USA, 1986. ACM.

[GK99] Shafi Goldwasser and Joe Kilian. Primality testing using elliptic curves. *J. ACM*, 46(4):450–472, 1999.

[IKY00] Tetsuya Izu, Jun Kogure, and Kazuhiro Yokoyama. *Public Key Cryptography: Third International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2000, Melbourne, Victoria, Australia, January 18-20, 2000. Proceedings*, chapter Efficient Implementation of Schoof's Algorithm in Case of Characteristic 2, pages 210–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[Leh30] D. H. Lehmer. An extended theory of Lucas' functions. *Annals of Mathematics*, 31:419–448, 1930.

[Len87] Hendrik W. Lenstra. Factoring integers with elliptic curves. *The Annals of Mathematics*, 126(3):649–673, 1987.

[Poc16] H. C. Pocklington. The determination of the prime or composite nature of large numbers by Fermat's theorem. *Proceedings of the Cambridge Philosophical Society*, 18:29–30, 1916.

[Pro78] F. Proth. Théorèmes sur les nombres premiers. *Comptes Rendus des Séances de l'Académie des Sciences*, 87:926, 1878.

[RÖ94] Öystein J. Rödseth. A note on primality tests for $N = h \cdot 2^n - 1$. *BIT Numerical Mathematics*, 34:451–454, 1994.

[Sch85] Rene Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of Computation*, 44(170):483–494, 1985.

[Sta11] Libor Stanovský, David a Bárto. *Počítačová algebra*. MATFYZPRESS, 1 edition, 2011.

[Was08] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Chapman & Hall/CRC, 2 edition, 2008.

Seznam obrázků