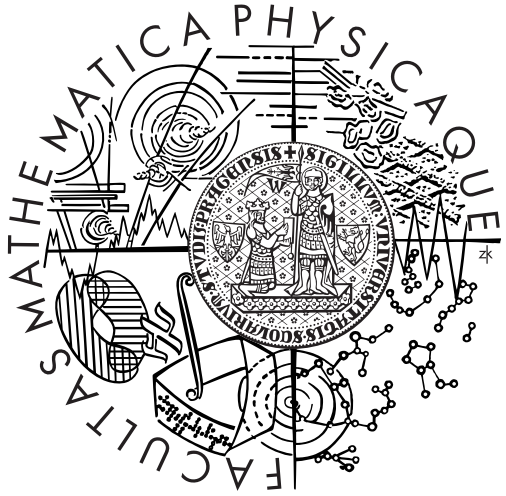
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

Bakalářská práce



Veronika Heglasová

Kryptografie založená na eliptických křivkách

Katedra algebry

Vedoucí: **Mgr. Libor Barto, Ph.D.**

Studijní program: **Obecná matematika**

2010

Poděkováni

Děkuji vedoucímu mé bakalářske práce Mgr. Liborovi Bartovi, Ph.D. za zadání zajímavého tématu bakalářske práce a také za cenné rady, připomínky a návrhy při psaní práce.

Čestné prohlášení

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

V Praze dne 25. května 2010 Veronika Heglasová

Obsah

Název práce: Kryptografia založená na eliptických krivkách

Autor: Veronika Heglasová

Katedra (ústav): Katedra algebry

Vedoucí diplomové práce: Mgr. Libor Barto, Ph.D.

e-mail vedoucího: Libor.Barto@mff.cuni.cz

Abstrakt: Práca sa zaoberá kryptografickými systémami založenými na eliptických krivkách. V prvej kapitole je možné nájsť zhrnutie základných pojmov z teórie eliptických kriviek a ich vlastnosti, ktoré sú potrebné k pochopeniu konkrétnych systémov. V sekcii Kryptografické systémy sú uvedené príklady konkrétnych systémov a opísané výhody kryptografie využívajúcej poznatky z teórie eliptických kriviek. Ku práci je priložený program vytvorený v matematickom softvéri Mathematica, ktorý demonštruje prácu s jedným zo systémov opísaných v druhej kapitole.

Kľúčové slová: Kryptografia na báze eliptických kriviek, eliptická krivka, problém eliptického diskrétneho logaritmu, eliptický Diffie-Helman, kryptografický systém ElGamal, kryptografický systém Massey-Omura

Title: Elliptic curve cryptography

Author: Veronika Heglasová

Department: Department of algebra

Supervisor: Mgr. Libor Barto, Ph.D.

Supervisor's e-mail address: Libor.Barto@karlin.mff.cuni.cz

Abstract: In this bachelor thesis we study elliptic curve cryptography. In the first chapter you can find basic definitions and properties, which are necessary for underestanding elliptic curve cryptography. In section Cryptographic systems are specifically described three systems and advantages of elliptic curve cryptography compare to classical public-key cryptography. A part of this bachelor thesis is a program created in mathematical software Mathematica. The programe uses algorithm from second chapter for encryption and decryption messages.

Keywords: Elliptic curve cryptography, elliptic curve, elliptic discrete logarithm problem, elliptic curve Diffie-Helman, ElGamal encryption system, Massey-Omura encryption system

Úvod

Témou práce je popis kryptografických systémov založených na teórii eliptických kriviek nad konečnými telesami[1] (ECC). V kapitole 1 sú definované základné pojmy z teórie eliptických kriviek, nevyhnutné k pochopeniu ECC a ich všeobecné vlastnosti. Kapitola 2 obsahuje konkrétne príklady kryptografických systémov.

Práca s jedným z týchto systémov, konkrétne so systémom ElGamal, bude demonštrovaná programom vytvoreným v matematickom softvéri Mathematica, ktorým môže užívateľ, s istými obmedzeniami na tvar krivky a prvočíselnú charakteristiku konečného telesa, kódovať alebo dekódovať správy. Špecifikácia programu je obsiahnutá v kapitole 3 a zdrojový kód je uvedený v prílohe.

ECC je moderný a nádejný smer kryptografie s verejným kľúčom. S návrhom použiť grupy založené na eliptických krivkách na účely kryptografie prišli v roku 1985 nezávisle na sebe Neil Koblitz a Victor Miller.

Hlavnou výhodou kryptosystémov na báze eliptických kriviek je ich veľká kryptografická bezpečnosť vzhľadom k veľkosti kľúča. Menšia dĺžka kľúčov vedie ku kratším bezpečnostným certifikátom aj menším parametrom systému, a teda aj k lepšej výpočetnej zložitosti algoritmov. Ďalšou výhodou je, že v podstate všetky doteraz známe systémy pracujúce na princípe problému diskrétneho logaritmu je možné previesť na systém na báze eliptických kriviek.

Aj napriek veľkému množstvu výhod eliptických kriviek, väčšina predajcov a akademikov vidí najväčšiu prekážku implementácie a používania ECC v *nehmotnom vlastníctve* okolo eliptických kriviek. Rôzne aspekty ECC boli totiž patentované rôznymi ľuďmi a firmami po celom svete. Napríklad spoločnosť Certicom Inc. má viac ako 130 patentov týkajúcich sa eliptických kriviek a kryptografie s verejným kľúčom všeobecne.

[1] V celom texte bude používaný pojem teleso ako je zaužívané na katedre algebry MFF UK v Prahe. V slovenskom jazyku sa väčšinou používa pojem pole (field).

1 Úvod do ECC

1.1 Základy z teórie eliptických kriviek

Všetky definície v tejto kapitole je možné nájsť v [4], str.167-169.

Definícia 1.1 (Eliptická krivka)**.** Nech $\mathbf{T}$ je teleso charakteristiky rôznej od 2 a 3 a x^3+ax+b (kde $a, b \in \mathbf{T}$) je kubický polynóm bez viacnásobných koreňov. *Eliptická krivka nad telesom* $\mathbf{T}$ je množina bodov $(x, y) \in \mathbf{T}$, ktoré sú riešením rovnice

$$y^2 = x^3 + ax + b \tag{1.1}$$

a samostatný prvok $\mathcal{O}$ nazývaný "bod v nekonečne"
Ak je $\mathbf{T}$ teleso charakteristiky 3 a $a, b, c \in \mathbf{T}$, potom eliptická krivka nad telesom $\mathbf{T}$ je množina bodov splňujúca rovnicu

$$y^2 = x^3 + ax^2 + b + c \tag{1.2}$$

a "bod v nekonečne" $\mathcal{O}$.
Ak je $\mathbf{T}$ teleso charakteristiky 2 a $a, b, c \in \mathbf{T}$, potom eliptická krivka nad telesom $\mathbf{T}$ je množina bodov splňujúca rovnicu typu

$$y^2 + cy = x^3 + ax + b \tag{1.3}$$

alebo

$$y^2 + xy = x^3 + ax^2 + b \tag{1.4}$$

a "bod v nekonečne" $\mathcal{O}$. V tomto prípade nemusí byť splnená podmienka, že kubický polynóm na pravej strane nemá viacnásobné korene.

Poznámka 1.2. Existuje všeobecný tvar rovnice eliptickej krivky, ktorý sa dá použiť pre ľubovolné teleso: $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$. Nazýva sa Weierstrassova rovnica a dá sa upraviť na rovnice z definície 1.1.

Dôkaz. Nech $\mathbf{T}$ je teleso charakteristiky rôznej od 2. Potom substitúcia

$$y = z - \frac{1}{2}(a_1x + a_3)$$

upraví Weierstrassovu rovnicu na:

$$\begin{aligned} z^2 &= x^3 + a_2x^2 + a_4x + a_6 + \frac{1}{4}\left(a_1x + a_3\right)^2 \\ z^2 &= x^3 + \left(a_2 + \frac{1}{4}a_1^2\right)x^2 + \left(a_4 + \frac{1}{2}a_1a_3\right)x + \left(a_6 + \frac{1}{4}a_3^2\right). \end{aligned}$$

Nech $z = y,\ (a_2 + \frac{1}{4}a_1^2) = a,\ a_4 + \frac{1}{2}a_1a_3 = b,\ a_6 + \frac{1}{4}a_3^2 = c$. Potom

$$y^2 = x^3 + ax^2 + bx + c.$$

Ak je charakteristika telesa väčšia ako 3 je možné na predošlý výsledok zaviesť substitúciu $x = z - \frac{1}{3}a$:

$$\begin{aligned} y^2 &= \left(z - \frac{1}{3}a\right)^3 + a\left(z - \frac{1}{3}a\right)^2 + b\left(z - \frac{1}{3}a\right) + c \\ y^2 &= z^3 + \left(b - \frac{1}{3}a^2\right)z + \left(\frac{1}{3}ab - \frac{2}{27}a^3 + c\right), \end{aligned}$$

Substitúcia $z = x,\ b - \frac{1}{3}a^2 = A,\ \ \frac{1}{3}ab - \frac{2}{27}a^3 + c = B$ dáva tvar

$$y^2 = x^3 + Ax + B.$$

Pre teleso charakteristiky 2 sa Weierstrassova rovnica upraví následovne:
Ak $a_1 = 0$ nech $x = X + a_2$. Potom

$$y^2 + a_3y = X^3 + (a_2^2 + a_4)X + (a_2a_4 + a_6)$$

Nech $X = x,\ a_2^2 + a_4 = a,\ a_2a_4 + a_6 = b,\ a_3 = c$. Potom

$$y^2 + cy = x^3 + ax + b.$$

Ak $a_1 \neq 0$ nech $x = a_1^2X + a_1^{-1}a_3,\ y = a_1^3Y + a_1^{-3}(a_1^2a_4 + a_3^2)$. Po substitúciu sa Weierstrassova rovnica upraví na tvar

$$Y^2 + XY = X^3 + \left(\frac{a_2}{a_1^2} + \frac{a_3}{a_1^3}\right)X^2 + \frac{1}{a_1^{12}}(a_1^3a_3^3 - a_3^4 + a_1^5a_3a_4 + a_1^4a_2a_3^2 - a_1^4a_4^2 + a_1^6a_6)$$

Nech $X = x,\ Y = y,\ \frac{a_2}{a_1^2} + \frac{a_3}{a_1^3} = a,\ \frac{1}{a_1^{12}}(a_1^3a_3^3 - a_3^4 + a_1^5a_3a_4 + a_1^4a_2a_3^2 - a_1^4a_4^2 + a_1^6a_6) = b$

$$y^2 + xy = x^3 + ax^2 + b.$$

□

Poznámka 1.3. Podmienka aby koreňe kubického polynómu na pravej strane rovnice eliptickej krivky $y^2 = x^3 + ax + b$ nad $\mathbb{F}_p$ pre $p > 3$ boli jednoduché je ekvivalentná tomu, že všetky body krivky sú nesingulárne. Splnenie nerovnosti $4a^3 + 27b \neq 0 \mod p$ koeficientov kubického polynómu zaručuje požadovanú nesingulárnosť bodov eliptickej krivky. V prípade krivky nad telesom reálnych čísel je podmienka tvaru $4a^3 + 27b \neq 0$.

V nasledujúcich definíciách sa budú uvažovať eliptické krivky nad telesom reálnych čísel. Nespôsobí to žiadne väčšie problémy pri neskoršom prechode ku konečným telesám a geometrická predstava, prečo sa používa pojem krivka, bude názornejšia.

Definícia 1.4 (opačný prvok)**.** Nech E je eliptická krivka nad telesom reálnych čísel a $P \in E$. *Opačný prvok* $-P$ je taký, že:

- Ak P je bod v nekonečne $\mathcal{O}$, potom aj $-P$ je bod v nekonečne $\mathcal{O}$.
- Ak P nie je bod v nekonečne $\mathcal{O}$, potom $-P$ je bod s rovnakou x-ovou súradnicou a opačnou y-ovou súradnicou ako P, tj. $-(x, y) = (x, -y)$.

Definícia 1.5 (súčet bodov)**.** Nech E je eliptická krivka nad telesom reálnych čísel a $P, Q \in E$. Súčet prvkov je definovaný následovne:

- Ak P je bod v nekonečne $\mathcal{O}$ potom $P + Q = Q$. V ostatných častiach definície sa predpokladá, že oba body sú rôzne od bodu v nekonečne $\mathcal{O}$.
- Ak body P a Q majú rôzne x-ové súradnice, označme priamku prechádzajúcu cez tieto 2 body l. Priamka l pretína krivku E buď v ešte práve jednom bode R, alebo je dotyčnicou krivky v bode P resp. Q, v takom prípade položíme $R = P$ resp. $R = Q$. Potom definujeme $P + Q = -R$.
- Ak $Q = -P$, definuje sa $P + Q = \mathcal{O}$.
- Ak $Q = P$ a l nech je dotyčnicou ku krivke v bode P. Potom R je druhý prienik (tj. rôzny od dotykového bodu) l a krivky alebo, v prípade, že l krivku v inom bode nepretne je $R = P$ a teda súčet sa definuje ako $Q + P = 2 \times P = -R$.

V celom texte bude používaný skrátený zápis: $k \times P = P + ... + P$, kde súčet na pravej strane má k členov, $k \in \mathbb{N}$ a P je bod na eliptickej krivke.

Nech $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$, $R = (x_R, y_r)$ a $P \neq Q$. Pre smernicu s priamky l z definície 1.5 platí $s = (y_Q - y_P)/(x_Q - x_p)$. Potom je možné odvodiť nasledujúci vzťah pre súradnice $-R = P + Q$:

$$x_R = s^2 - x_P - x_Q = \left(\frac{y_Q - y_P}{x_Q - x_p}\right)^2 - x_P - x_Q \tag{1.5}$$

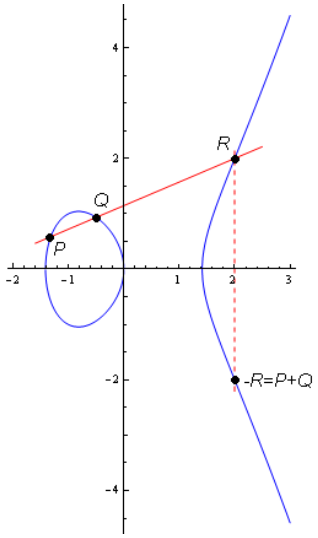
$$y_R = s(x_P - x_R) - y_P = \left(\frac{y_Q - y_P}{x_Q - x_p}\right)(x_P - x_R) - y_P \tag{1.6}$$

V prípade, že $P = Q$ je $s = (3x_P^2 + a)/2y_p$, kde a je koeficient lineárneho člena kubického polynómu v rovnici krivky. Potom platí:

$$x_R = s^2 - 2x_P = \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \tag{1.7}$$

$$y_R = s(x_P - x_R) - y_P = \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_R) - y_P \tag{1.8}$$

Príklad 1.6. Eliptická krivka $y^2 = x^3 - 2x$ nad telesom reálnych čísel je krivka modrej farby na obrázku 1. Zároveň je na obrázku graficky znázornené sčítavanie opísané v definícií 1.5 (súčet bodov $P + Q$ je na obrázku znázornený červenou farbou).



Obrázok 1: Krivka $y^2 = x^3 - 2x$ nad $\mathbb{R}$

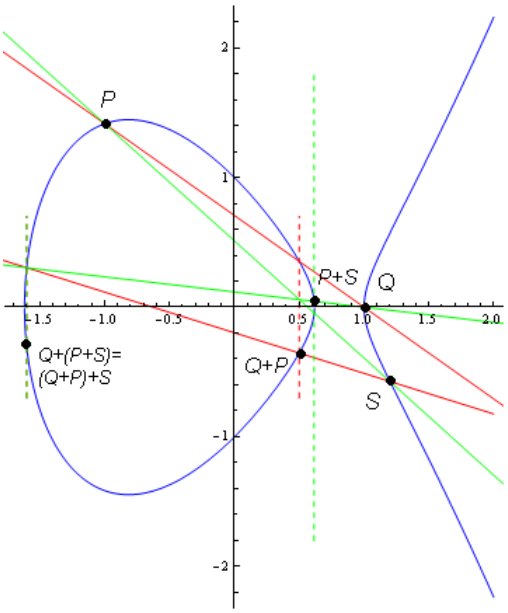
Príklad 1.7. Nech $y^2 = x^3 + x + 2$ je eliptická krivka nad $\mathbb{R}$, body $P = (1, 2)$ a $Q = (-1, 0)$ ležia na tejto krivke. Potom $P + Q = (x, y)$ sa podľa vzorcov (1.5) a (1.6) spočíta nasledovne:

$$\begin{aligned} x &= \left(\frac{0-2}{-1-1}\right)^2 - 1 - (-1) = 1 \\ y &= \left(\frac{0-2}{-1-1}\right)(1-1) - 2 = -2 \end{aligned}$$

Podľa vzorcov (1.7) a (1.8) je jednoduché spočítať napr. $2 \times P$:

$$\begin{aligned} x &= \left(\frac{3.1^2+1}{2.2}\right)^2 - 2.1 = -1 \\ y &= \left(\frac{3.1^2+1}{2.2}(1-(-1)) - 2 = 0 \end{aligned}$$

Poznámka 1.8. Podľa definícií 1.4 a 1.5 je zrejmé, že opačný prvok každého bodu krivky je opäť bod na krivke a zároveň, že množina bodov krivky je invariantná na operáciu sčítavania, ktorá je komutatívna. Ak by operácia sčítavania z definície 1.5 bola asociatívna, tvorili by body eliptickej krivky Abelovu grupu. Asociatívnosť súčtu je možné dokázať pomocou Riemann - Rochovej vety, ktorá je nad rámec tejto práce. Geometrická predstava, že operácia sčítavania z definície 1.5 je asociatívna, je znázornená na obrázku 2.



Obrázok 2: Krivka $y^2 = x^3 - 2x + 1$ nad $\mathbb{R}$.

Odteraz sa bude predpokladať, že $\mathbf{T}$ je konečné teleso $\mathbb{F}_q$, kde $q = p^r$ pre nejaké prvočíslo p.

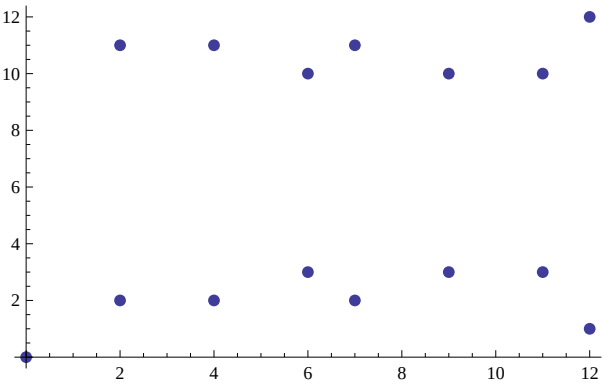
Definícia 1.9 (Rád prvku)**.** Nech E je eliptická krivka nad $\mathbb{F}_q$ a $P \in E$. Potom *rád prvku* P je najmenšie $n \in \mathbb{N}$, pre ktoré platí $n \times P = \mathcal{O}$

.

Príklad 1.10. Krivka z príkladu 1.6, tentokrát definovaná nad $\mathbb{Z}_{13}$, demonštruje, prečo je pri prvotnom zoznamovaní sa s eliptickými krivkami názornejšie uvažovať krivky definované nad telesom reálnych čísel. Množina bodov, ktoré tvoria krivku nad $\mathbb{Z}_{13}$ je znázornená v tabuľke 1 a na obrázku 3.

$(0,0)$	$(2,2)$	$(4,2)$	$(6,3)$
$(2,11)$	$(7,2)$	$(9,3)$	$(11,3)$
$(4,11)$	$(9,10)$	$(7,11)$	$(12,1)$
$(6,10)$	$(11,10)$	$(12,12)$	$\mathcal{O}$

Tabuľka 1: Body krivky $y^2 = x^3 - 2x$ nad $\mathbb{Z}_{13}$.



Obrázok 3: Body krivky $y^2 = x^3 - 2x$ nad $\mathbb{Z}_{13}$.

1.2 Výpočetné postupy

V tejto kapitole budú predstavené niektoré výpočetné postupy, prevažne z teórie čísel. Ich použitie je potrebné pri šifrovaní opísanom v nasledujúcich kapitolách.

Je potrebné vedieť, či existuje riešenie rovnice $y^2 = x \mod p$. Bude sa to totiž využívať pri hľadaní bodov krivky nad $x \in \mathbb{Z}_p$, $p > 2$. Konkrétne či pre dané $x \in \mathbb{Z}_p$ existuje y také, že $y^2 = f(x)$, kde $f(x)$ je hodnota kubického polynómu na pravej strane rovnice eliptickej krivky v bode x.

Definícia 1.11 (Kvadratické reziduum)**.** Nech p je prvočíslo, $x \in \mathbb{Z}_p$. Potom x je *kvadratické reziduum (kvadratický zvyšok)* práve vtedy, ak existuje $y \in \mathbb{Z}_p$ také, že $y^2 = x \mod p$.

Definícia 1.12 (Legendreov symbol)**.** Nech a je prirodzené číslo a $p > 2$ prvočíslo. Potom

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{ak } a \text{ je kvadratické reziduum;} \\ -1 & \text{ak } a \text{ nie je kvadratické reziduum;} \\ 0 & \text{ak } p \mid a \end{cases} \tag{1.9}$$

je *Legendreov symbol.*

Poznámka 1.13. Legendreov symbol má mnoho vlastností, ktoré uľahčujú jeho výpočet. Dôkaz vlastností a výpočetných postupov uvedených v tejto kapitole je možné nájsť v skriptách [2], 3. kapitola.

Nech $p, q > 2$ je prvočíslo, $a, b \in \mathbb{N}$

- $\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \mod p$
- $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$

- $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$
- $\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)(q-1)}{4}}$ (zákon reciprocity)

Poznámka 1.14. V prípade, že $x \in \mathbb{Z}_p$ je kvadratické reziduum a $y^2 = x \mod p$ je ešte potrebné nájsť y. Nie je však potrebné overovať všetky čísla $(0, ..., p-1)$; existuje efektivnejší algoritmus, ktorý je opísaný v [4], str. 182. V prípade, že $p = 3 \mod 4$ je $x = 1.x = \left(\frac{x}{p}\right).x = x^{(p-1)/2}.x = x^{(p+1)/2} = (x^{(p+1)/4})^2$ a hľadaná odmocnina $y = x^{(p+1)/4}$.

Pre účely ECC je dôležité vedieť vypočítať počet bodov N eliptickej krivky E nad $\mathbb{F}_q$. Je zrejmé, že $N \leq 2q + 1$: krivka obsahuje bod v nekonečne $\mathcal{O}$ a body (x, y), pričom pre každú z q možností x existujú maximálne 2 odpovedajúce y. Omnoho presnejší odhad na počet bodov eliptickej krivky definovej nad konečným telesom poskytuje nasledujúca veta:

Veta 1.15 (Hasseho veta)**.** *Nech N je počet bodov eliptickej krivky definovanej nad $\mathbb{F}_q$. Potom $|N - (q + 1)| \leq 2\sqrt{q}$.*

1.3 Problém diskrétneho logaritmu

Teória čísel ponúka viacero *jednosmerných (one-way)* úloh, tzn. úloh, ktoré nie je komplikované vypočítať, ale z výsledku už nie je také jednoduché zistiť vstupné parametre úlohy. Asi najznámejším príkladom je faktorizácia veľkých čísel, kde z dvoch veľkých prvočísiel p, q je jednoduché vypočítať $n = p \times q$, ale neexistuje algoritmus, ktorý by zo znalosti n v reálnom čase zistil prvočísla p, q. Problém faktorizácie veľkých čísel používa kryprografický systém RSA.

Príkladom jednosmernej úlohy je aj tzv. *problém diskrétneho logaritmu*. V prípade počítania v telese reálnych čísel $\mathbb{R}$ je výpočet $a^x = b$ približne rovnako komplikovaný ako výpočet $\log_a b = x$. Avšak, ak bude výpočet prebiehať v konečnej multiplikatívnej grupe ako $\mathbb{Z}_q^*$ existujú efektívne algoritmy[2] na výpočet $a^x = b$. Zároveň neexistuje efektívny algoritmus, ktorý by z b pri fixnom základe a vypočítal x. Tomuto výpočtu sa hovorí problém diskrétneho logaritmu.

Definícia 1.16. Nech G je konečná multiplikatívna grupa, $a \in G$ a $y \in G$ je mocnina prvku a. Potom *diskrétny logaritmus* b pri základe a je ľubovolné x také, že $a^x = b$.

Príklad 1.17. Nech $G = \mathbb{Z}_{13}$, $a = 3$ a $b = 9$. Keďže $3^5 = 9 \mod 13$, je 5 diskrétny logaritmus prvku 9 pri základe 3.

[2]Presný opis algoritmov je možné nájsť v skriptách [7], 3. kapitola

2 Kryptografické systémy

2.1 Príprava textu na šifrovanie

Skôr, ako je možné začat správu šifrovať konkrétnym kryptografických systémom využívajúcim teóriu eliptických kriviek, je potrebné správu *previesť* na body danej eliptickej krivky. Tento prevod je potrebné urobiť jednoduchým a systematickým spôsobom, aby pri spätnom dešifrovaní správy bolo jednoduché priradiť k bodom opäť znaky tvoriace prijatú správu.

Na začiatok je potrebné poznamenať dva fakty:

- Neexistuje deterministický algoritmus pracujúci v polynomiálnom čase, ktorý by dokázal vypočítať veľké množstvo bodov na ľubovolnej krivke E nad $\mathbb{F}_q$. Avšak, existuje algoritmus, ktorého *pravdepodobnosť zlyhania* je dostatočne malá a ten bude predvedený v tejto kapitole. Dôkaz, že táto pravdepodobnosť je dostatočne malá je možné nájsť v [4], str.179-180.
- Je potrebné uvedomiť si, že nestačí generovať náhodné body na E. Je nutné nájsť spôsob ako vygenerovať veľké množstvo bodov, ktoré budú istým spôsobom vo vzťahu s posielanou správou.

Príklad vhodného pravdepodobnostného algoritmu pre prevod znaku s na bod P_s na krivke E nad $\mathbb{F}_p$ pre veľké prvočíslo p:

1. Priradenie prirodzeného čísla ku každému znaku správy.
   Jedna z možností ako priraďovať prirodzené čísla k znakom je napríklad postupne abecedne pre A-Z čísla 10-35 a čísla 0-9 priraďovať k číslam v správe (zhodne).
2. Voľba κ.
   Pri predpoklade, že znakom odpovedajú prirodzené čísla s, $0 < s < S$, musí voľba κ spĺňať $q > S\kappa$.
3. Hľadanie vhodného x.
   Nech $f(x)$ je kubický polynóm z rovnice krivky E, $y^2 = f(x)$. Potom pre každé s je $x = s\kappa + j$ (kde $j = 1, 2, ...\kappa$) prvok $\mathbb{F}_p$.
4. $j := 1$
5. Pre j sa spočíta hodnota $f(x)$.
   Ak $f(x)$ je kvadratický zvyšok $\bmod p$, je možné vypočítať hodnotu y. V takom prípade $P_s = (x, y)$.
   Ak $f(x)$ nie je kvadratický zvyšok $\bmod p$, $j := j + 1$, opakovať krok 5.

Vzťah medzi súradnicou x a znakom s po použití algoritmu je potom nasledovný: $\lfloor \frac{x-1}{\kappa} \rfloor = n$, kde $n \in \mathbb{N}$ odpovedá znaku s.

V prípade eliptickej krivky definovanej nad $\mathbb{F}_q$, $q = p^r$ kde $r > 1$ je možné analogicky nájsť vhodný algoritmus.

Príklad 2.1. Podobný príklad sa nachádza aj v [4], str.185.
Nech E je eliptická krivka nad $\mathbb{F}_{751}$ splňujúca rovnicu $y^2 = x^3 - x + 188$. Predpokladá sa, že znakom v správe odpovedajú prirodzené čísla spôsobom opísaným v 1. bode pravdepodobnostného algoritmu na prevod znaku, $\kappa = 20$. Prevod správy AHOJ3:

- Na výpočty sa používaju spôsoby opísané v kapitole 1.2. Kedže $751 = 3 \mod 4$ je možné pre nájdenie odmocniny použiť špeciálny prípad z poznámky 1.14, $f(x) = x^3 - x + 188$.
- Prevod znaku A: A ~ 10, $\kappa.10 = 200$. Minimálne $j \in 1, 2, ..., 20$ také, že $\left(\frac{f(200+j)}{751}\right) = 1$ je $j = 1$.
  Teda, pre $x = 201$ sa dá vypočítať $y^2 = f(201) = 25$ a teda je zrejmé, že $y = 5$ aj bez použitia algoritmu z kapitoly 1.2. $P_A = (201, 5)$.
- Prevod znaku H: H ~ 17, $\kappa.17 = 340$. Minimálne $j \in 1, 2, ..., 20$ také, že $\left(\frac{f(340+j)}{751}\right) = 1$ je $j = 1$.
  Teda, pre $x = 341$ sa dá vypočítať $y^2 = f(y) = 370$. Podľa poznámky 1.14 je $y = 370^{188} \mod 751 = 362$. $P_H = (341, 362)$.
- Prevod znaku O: O ~ 24, $\kappa.24 = 480$. Minimálne $j \in 1, 2, ..., 20$ také, že $\left(\frac{f(480+j)}{751}\right) = 1$ je $j = 4$.
  $y^2 = f(484) = 387$. $y = 387^{188} \mod 751 = 590$. $P_O = (484, 590)$.
- Prevod znaku J: J ~ 19, $\kappa.19 = 380$. Minimálne $j \in 1, 2, ..., 20$ také, že $\left(\frac{f(380+j)}{751}\right) = 1$ je $j = 1$.
  $y^2 = f(381) = 255$. $y = 255^{188} \mod 751 = 682$. $P_J = (381, 682)$.
- Prevod znaku 3: 3 ~ 3, $\kappa.3 = 60$. Minimálne $j \in 1, 2, ..., 20$ také, že $\left(\frac{f(60+j)}{751}\right) = 1$ je $j = 2$.
  $y^2 = f(62) = 387$. $y = 387^{188} \mod 751 = 590$. $P_J = (62, 590)$.
- Záver: odpovedajúca postupnosť bodov krivky E pre správu AHOJ3 je $(201, 5), (341, 362), (484, 590), (381, 682), (62, 590)$.

V pripáde dešifrovania prijatej správy na postupnosť bodov zo záveru sa odpovedajúca postupnosť znakov nájde následovne:
$\lfloor \frac{201-1}{20} \rfloor = 10 \sim$ A, $\lfloor \frac{341-1}{20} \rfloor = 17 \sim$ H, $\lfloor \frac{484-1}{20} \rfloor = 24 \sim$ O, $\lfloor \frac{381-1}{20} \rfloor = 19 \sim$ J, $\lfloor \frac{62-1}{20} \rfloor = 3 \sim 3$.

2.2 Bezpečnosť

Množstvo súčasných kryptografických systémov s verejným kľúčom je založených na operácií umocňovania v konečných grupách. Kryptografická sila týchto systémov je závislá na zložitosti problému diskrétneho logaritmu, respektíve na zložitosti nájdenia prvočíselného rozkladu veľkých čísel. Obvykle sú používané konečné multiplikatívne grupy $\mathbb{F}_q^*$, kde $q = p^r$ pre prvočíslo p.

Keď v roku 1985 N. Koblitz a V. Miller nezávisle na sebe navrhli využitie eliptických kriviek v kryptografii s verejným kľúčom, vyvolala táto myšlienka veľkú pozornosť. Ukázalo sa, že ECC má dve výhody oproti konvenčným systémom: veľká rôznorodosť vhodných eliptických kriviek a absencia algoritmu pracujúceho v subexponenciálnom čase, ktorý by vyriešil problém diskrétneho logaritmu v grupe bodov eliptickej krivky.

Poznámka 2.2. Nech E je vhodná eliptická krivka, tj. s veľkým počtom bodov. Body eliptickej krivky E tvoria podľa 1.8 konečnú Abelovu grupu $\mathbb{E}$. Pre účely kryptografie sa uvažuje veľká cyklická podgrupa grupy $\mathbb{E}$. Problém diskrétneho logaritmu je v prípade grupy prvkov eliptickej krivky práve nalezenie $k \in \mathbb{N}$, že $k \times P = Q$ pri znalosti $P, Q \in \mathbb{E}$.[3]

Zatiaľ najúčinejšia metóda na riešenie problému diskrétneho logaritmu v grupe bodov eliptickej krivky je tzv. *Pollardova ρ-metóda* so zložitosťou rádovo $(\pi\frac{r}{2})^{\frac{1}{2}}$ krokov. Pre $r = 2^{256}$, kde r je rád generátoru cyklickej podgrupy $\mathbb{E}$, je to približne 2^{128} krokov, čo je z výpočtového hľadiska nereálne. Príslušná šifra je teda bezpečná.[3]

Pôvodný algoritmus, ktorý sa používal na riešenie diskrétneho logaritmu v grupe bodov eliptickej krivky bolo možné použiť pre ľubovolnú grupu. V roku 1991 A. Menezes, T. Okamota a S. Vanston vymysleli algoritmus, ktorý využíva bohatú štruktúru grupy prvkov eliptickej krivky. Tento algoritmus prevádza problém diskrétneho logaritmu na eliptickej krivke E na problém diskrétneho logaritmu v $\mathbb{F}_{q^k}^*$. Tento prevod sa nazýva *MOV (Menezes,Okamota,Vanston) redukcia* alebo niekedy aj *MOV útok*. MOV redukcia využíva Weilovo párovanie. Hlbšie informácie o Weilovom párovaní (tzv. *Wiel pairing*) sú nad rámec tejto práce. Podrobnú teóriu je možné nájsť v [6]. Mov redukcia funguje iba na tzv. *supersingulárne eliptické krivky*. Najznámejším príkladom sú krivky tvaru $y^2 = x^3 + ax$, kde charakteristika $\mathbb{F}_q$ je $p = 3 \mod 4$ a $y^2 = x^3 + b$, kde charakteristika $\mathbb{F}_q$ je $p = 2 \mod 3$. Problém diskrétneho logaritmu na eliptických krivkách je v prípade supersingulárnych kriviek možné vyriešiť v subexponenciálnom čase. V prípade potreby bezpečného kryptografického systému je teda potrebné vyhnúť sa množine supersingulárnych kriviek.

V praxi sa v súčastnosti doporučuje používať krivky s náhodne generovanými parametrami. Pravdepodobnosť, že sa tak vygeneruje *nevhodná* krivka je veľmi malá. Pre stanovenie optimálnych parametrov je potrebné mať možnosť vypočítať

počet bodov eliptickej krivky. To umožňuje tzv. *Shoofov algoritmus*, ktorý je však pomerne dosť náročný. Preto sú k dispozícií doporučené množiny parametrov[3].

Príklad 2.3. Porovnanie dĺžky kľúča potrebnej pri blokových šifrách, kryptografických systémoch na báze eliptických kriviek a RSA/systémoch na báze diskrétneho logaritmu (DL) pre rovnakú bezpečnosť podľa národnej bezpečnostnej služby Spojených štátov amerických (NSA):

Blokové šifry	Eliptické krivky	RSA/DL
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Tabuľka 2: Konkrétne dĺžky kľúčov, pre rôzne systémy, odpovedajúce porovnateľnej bezpečnosti.

2.3 Príklady konkrétnych kryptografických systémov

V tejto kapitole budú predstavené analógie systémov s verejným kľúčom na báze diskrétneho logaritmu so systémami na báze diskrétneho logaritmu v grupe bodov eliptickej krivky. Konrétne protokol výmeny kľúča Diffie-Helman v stati 2.3.1, systém Massey-Omura v stati 2.3.2 a systém ElGamal v stati 2.3.3.

2.3.1 Eliptický Diffie-Helman (ECDH): výmena kľúča

Alica a Bob sa chcú dohodnúť na spoločnom kľúči, ktorý neskôr budú používať pri šifrovaní klasickým kryptografickým systémom. To znamená, že potrebujú ľubovoľný bod P na eliptickej krivke E definovanej nad $\mathbb{F}_q$, $q = p^r$, p je prvočíslo. Spoločný kľúč potom môže byť odvodený napríklad zo súradnice x bodu P, ktorá je nedeterministicky zvolený prvok $\mathbb{F}_q$.

Voľba prvku P musí prebehnúť verejnou výmenou parametrov takým spôsobom, aby nakoniec boli iba Alica a Bob schopní určiť hodnotu P. Ako prvú si verejne zvolia eliptickú krivku E a bod $B \in E$ slúžiaci ako *základ*. Pri používaní pojmu *základ* sa využíva zrejmá analógia medzi logaritmom a diskrétnym logaritmom. Pre zachovanie obtiažnosti riešenia problému DL je potrebné aby B generoval $\mathbb{E}$ tj. grupu prvkov eliptickej krivy E, alebo aspoň jej dostatočne veľkú podgrupu.

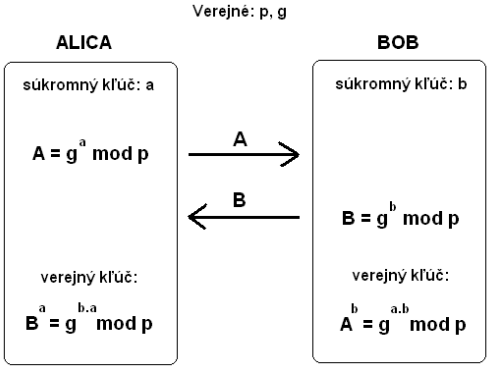
[3]Množiny parametrov vytvára napríklad NIST (National Institute of Standards and Technology) a SECG (Standards for Efficient Cryptography Group)

V prípade, že N je počet prvkov $\mathbb{E}$, potom rád B musí byť rovný N alebo aspoň veľkému deliteľu N.

Alica aj Bob si vyberú prirodzené číslo $k \in \mathbb{N}$, ktorého veľkosť v bitoch je rovná veľkosti q. Toto číslo bude slúžiť ako súkromný kľúč. Nech k_A je Alicin a k_B je Bobov súkromný kľúč. Alica vypočíta bod $K_A = k_A \times B \in E$ a to bude jej verejný kľúč. Podobne Bob vypočíta svôj verejný kľúč $K_B = k_B \times B \in E$.

Potom zdieľaný verejný kľúč je:
$P = k_A \times K_B = k_A \times k_B \times B = k_B \times k_A \times B = k_B \times K_A$.

Poznámka 2.4. Na obrázku 4 je graficky znázornení klasický protokol Diffie-Helman (s využitím konečného telesa $\mathbb{F}_p$) na výmenu kľúča s verejnými parametrami prvočíslom p a generátorom g grupy $\mathbb{F}_p$. Analógia s ECDH je zrejmá.



Obrázok 4: Schéma výmeny kľúča Diffie-Helman

2.3.2 Massey-Omura kryptografický systém

Pred samotným posielaním správ Alica a Bob verejne zvolia eliptickú krivku E nad $\mathbb{F}_q$, kde q je veľké. Potom vypočítajú N = počet bodov na krivke E. Číslo N je tiež verejný parameter.

Obidvaja užívatelia si zvolia súkromný kľúč ako prirodzené číslo $1 < k < M$ nesúdeľné s N. Z nesúdeľnosti vyplýva existencia $a, b \in \mathbb{N}$ takých, že $ak + bN = 1$ a teda $ak = 1 \mod N$. Nech Alicin súkromý kľúč sú čísla k_A, a_A a Bobov čísla k_B, a_B

Alica chce poslať Bobovi správu S, ktorú už má prevedenú na postupnosť bodov na krivke E (napríklad algoritmom opísaným v kapitole 2.1). Nech P_s je jedným z bodov postupnosti. Ako prvý pošle Bobovi bod $k_A \times P_s$. Bob, ani nikto iný, bez znalosti Alicinho súkromného kľuča nevie z prijatého bodu vypočítať P_s. V tomto momente pošle Bob Alici bod $k_B \times k_A \times P_s$. Ako tretí krok Alica vynásobí

prijatý bod a_A a bod:

$$\begin{aligned} a_A \times k_B \times k_A \times P_s &= a_A \times k_A \times k_B \times P_s = (1 - bN) \times k_B \times P_s \\ &= k_B \times P_s - bN \times k_B \times P_s = k_B \times P_s - \mathcal{O} \\ &= k_B \times P_s. \end{aligned}$$

Predposledná rovnosť vyplýva z toho, že N je rád grupy a rád každého prvku delí rád grupy. V poslednom kroku Bob prenásobí prijatý bod a_B, $a_B \times k_B \times P_s = P_s$ a dostane tak bod odpovedajúci znaku v poslanej správe.

Poznámka 2.5. Všetky kryptografické systémy s verejným kľúčom kladú na verejný kanál, po ktorom sa posielajú správy, podmienku autentizácie. V prípade, že by kanál na prenos správ nespĺňal podmienku autentizácie, mohlo by dojsť k nasledovnému útoku: Alica by poslala Bobovi bod $k_A \times P_s$. Nepriateľka Eva, ktorá sa napojila na ich kanál správ tento bod zachytí a pošle Alici $k_E \times k_A \times P_s$, kde k_E a a_E sú Evine súkromné kľúče zvolené rovnakým spôsobom ako Alicine a Bobove. Alica po prijatí bodu pošle naspäť bod $a_A \times k_E \times k_A \times P_s = k_B \times P_s$, z ktorého už Eva dokáže vypočítať vynásobením s a_E bod P_s. Rovnakým postupom teraz Eva pošle správu Bobovi. V takomto prípade si teda Alica aj Bob myslia, že komunikujú spolu a nevedia, že všetky ich správy sú odchytávané a dešifrované protivníkom. Tento typ útoku sa nazýva *man in the middle attack*.

2.3.3 ElGamal kryptografický systém

Autorom tohoto systému je Taher ElGamal, ktorý ho popísal vo svojej dizertačnej práci v roku 1984. Môže byť definovaný na ľubovolnej cyklickej grupe čiže aj na cyklickej podgrupe bodov eliptickej krivky.

Rovnako ako pri väčšine systémov sa začína s fixným, verejne známym konečným telesom $\mathbb{F}_q$, eliptickou krivkou E definovanou nad týmto telesom a bodom $B \in E$. Pre bezpečnosť systému nie je nevyhnutné aby B generoval celú eliptickú krivku, stačí aby cyklická grupa generovaná bodom B bola dostatočné veľká. Narozdiel od systému opísaného v stati 2.3.2 nie je potrebné vedieť počet bodov E.

Obidvaja užívatelia si zvolia súkromný kľúč $k \in \mathbb{N}$. Nech Alicin súkromný kľúč je k_A a verejný $K_A = k_A \times B$. Bobov súkromný kľúč nech je k_B a verejný $K_B = k_B \times B$.

Alica chce poslať Bobovi správu S, ktorú už má prevedenú na postupnosť bodov na krivke E (napríklad algoritmom opísaným v kapitole 2.1). Nech P_s je jedným z bodov postupnosti. Alica zvolí ľubovolné $n \in \mathbb{N}$, vypočíta body $Q_1 = n \times B$, $Q_2 = P_s + n \times K_B$ a Bobovi pošle dvojicu bodov (Q_1, Q_2).

Na dešifrovanie Bob vynásobí prvý bod svojím súkromným kľúčom a výsledok odčíta od druhého bodu:
$Q_2 - k_B \times Q_1 = P_s + n \times \ k_B \times B - k_B \times \ n \times B = P_s$.

Poznámka 2.6. Podľa [1] môže byť pri šifrovaní použitá ľubovolná *invertibilná* operácia (v prípade eliptickej krivky je to aditívna operácia). Klasický systém ElGamal s multiplikatívnou operáciou vyzerá následovne:
Nech p je veľké prvočíslo a g generátor $\mathbb{F}_p$ sú verejne parametre. Alica chce Bobovi poslať nezáporné $s \leq (p-1)$, ktoré odpovedá znaku v správe S. Ako prvý si zvolí súkromný kľúč k_A, $0 < k_A < (p-1)$ a vypočíta verejný kľúč $K_A = q^{k_A}$. Bob si analogicky zvolí súkromný kľúč k_B a vypočíta verejný kľúč $K_B = q^{k_B}$. Potom Alica pošle Bobovi dvojicu bodov $(K_A \mod p, K_B^{k_A} \times s \mod p)$. Bob umocní prvý bod svojím súkromným kľúčom a výsledkom vydelí druhý bod:
$(K_B^{k_A} \times s)/(K_A^{k_B}) = (q^{k_B.k_A} \times s)/(q^{k_A.k_B}) = s$.

Príklad 2.7. Podobný príklad sa nachádza aj v [4], str.186.
V príklade 2.1 bolo ukázané ako je možné správu AHOJ3 previesť na postupnosť bodov $(201, 5), (341, 362), (484, 590), (381, 682), (62, 590)$. Šifrovanie tejto postupnosti kryptografickým systémom ElGamal:

- Pripomenutie zadania z príkladu 2.1: krivka E $y^2 = x^3 - x + 188$, prvočíslo $p = 751$
- Voľba zvyšných parametrov: generátor $B = (0, 0)$, súkromný kľúč $k = 196$, postupnosť prirodzených čísel, ktoré budú použité pri kódovaní jednotlivých bodov: $312, 194, 502, 423, 103$
- Podľa vzorcov 1.5, 1.6, 1.7 a 1.8 je verejný kľúč:
  $K = 196 \times B = \left[2\left(2\left(1 + 2\left(2\big(2(2(1+2))\big)\right)\right)\right)\right] \times B = (485, 738).$
- Kódovanie P_A: $312 \times B = (551, 231)$, $P_A + 312 \times kB = (169, 619)$
- Kódovanie P_H: $194 \times B = (607, 733)$, $P_H + 194 \times kB = (291, 767)$
- Kódovanie P_O: $502 \times B = (77, 6)$, $P_O + 502 \times kB = (333, 316)$
- Kódovanie P_J: $423 \times B = (172, 255)$, $P_J + 423 \times kB = (233, 256)$
- Kódovanie P_3: $103 \times B = (41, 477)$, $P_3 + 103 \times kB = (157, 55)$
- Záver: Správa AHOJ3 je zašifrovaná na postupnosť bodov:
  $\big((551,231),(169,619)\big)$, $\big((607,733),(291,767)\big)$, $\big((77,6),(333,316)\big)$, $\big((172,255),(233,256)\big)$, $\big((41,477),(157,55)\big)$.

Správnosť výsledku je možné overiť programom priloženým k tejto práci. Pozornosť treba venovať rozdielnemu tvaru krivky v programe, konkrétne je potrebné zaviesť substitúciu $y = z + 375$.
Prvočíslo 751 je pre prax nevyhovujúce (príliš malé) a jeho voľba slúžila len na ukážku práce systému ElGamal.

3 Implementácia v Mathematica 7.0

Obmedzenia kladené na užívateľa:

- tvar rovnice eliptickej krivky E definovanej nad $\mathbb{F}_p$ je $y^2 + y = x^3 - x$. Na dosiahnutie tvaru rovnice z definície 1.1 je potrebné zaviesť substitúciu $y = z + \frac{p-1}{2}$ a pri používaní vzorcov (1.5), (1.6), (1.7), (1.8) pracovať s krivkou $z^2 = x^3 - x - \left(\frac{p-1}{2} + \left(\frac{p-1}{2}\right)^2\right)$

- prvočíselná charakteristika $p > 3$ konečného telesa $\mathbb{F}_p$ spĺňa $p = 3 \mod 4$. Toto obmedzenie je z dôvodu zjednodušenia výpočtu druhej odmocniny v telese s takouto charakteristikou.

V celom popise procedúr sa pod pojmom *prvočíslo* myslí prvočíselná charakteristika $\mathbb{F}_p$.

Popis funkcií a procedúr:

- `xko, yko` : funkcie používajúce vzorce (1.5) a (1.6) slúžiace na výpočet súradnic x a y pri súčte dvoch rovnakých bodov, tj. pri dvojnásobku bodu. Podiel zo vzorcov je nahradený výpočtom inverzného prvku menovateľa. Vstupnými parametrami sú súradnice bodu krivky a prvočíslo.

- `xko2, yko2`: funkcie používajúce vzorce (1.7) a (1.8) slúžiace na výpočet súradnic x a y pri súčte dvoch rôznych bodov. Podiel zo vzorcov je nahradený výpočtom inverzného prvku menovateľa.
  Vstupnými parametrami sú súradnice bodov krivky a prvočíslo.

- `sucetA`: súčet dvoch rovnakých bodov použitím funkcií `xko` a `yko`.
  Vstupnými parametrami sú súradnice bodu krivky, názvy súradnic výsledku a prvočíslo.

- `sucetB`: súčet dvoch rôznych bodov použitím funkcií `xko2` a `yko2`.
  Vstupnými parametrami sú súradnice bodov, ktoré sa majú sčítať, názvy súradnic výsledku a prvočíslo.

- `nasobenie`: výpočet $k \times P$, $P \in E$. Číslo k sa prevedie do binárnej sústavy a postupom analogickým s metódou binárneho mocnenia (algoritmus opísaný napr. v [7]) sa súčin prevedie na $\sum 2^i \times P$, kde i prebieha *polohy jednotiek* v binárnom zápise čísla k, tj. plati $\sum 2^i = k$. Na výpočty sú použité procedúry `sucetA` a `sucetB`.
  Vstupnými parametrami sú bod krivky, súradnice výsledku, násobok k a prvočíslo.

- `over`: overenie vhodnosti vstupu zadaného užívateľom. Procedúra overuje, či sú splnené podmienky:
  - prvočíselná charakteristika je naozaj prvočíslo,
  - $p = 3 \mod 4$,
  - násobok κ spĺňa nerovnosť $\kappa S < p$ z kapitoly 2.1.

  Vstupné parametre κ, prvočíslo a správa určená na šifrovanie.

- `nacitaj`: načítanie vstupnej správy a jej prevod na postupnosť prirodzených čísel podľa algoritmu z kapitoly 2.1

- `prevodX`: podľa algoritmu opísanom v kapitole 2.1 sa najskôr hľadá j, pre ktoré je $f(s\kappa + j)$ kvadratické reziduum.
  Vstupné parametré sú správa, κ a prvočíslo.

- `prevodY`: výpočet súradnice y odpovedajúcej súradnici x.
  Vstupné parametre sú prvočíslo a množina súradnic x.

- `sifruj`: výpočet bodov výstupnej správy $(n \times B, P_s + n \times\ k \times B)$ využitím procedúr `násobenie`,`sucetA` a `sucetB`.
  Vstupné parametre sú množiny súradnic x a y a prvočíslo.

- `desifruj`: výpočet bodov vstupnej správy a ich prevod na znaky správy využitím procedúr `násobenie`,`sucetA` a `sucetB`.
  Vstupné parametre sú prvočíslo a κ.

V hlavnej časti programu užívateľ zadáva parameter κ, prvočíselnú charakteristiku p, hodnotu súkromného kľúča k a rovnica krivky je substitúciou prevedená na rovnicu $y^2 = x^3 - x - a$. Táto substitúcia je dôležitá pre využitie vzorcov pre súčet bodov na krivke uvedených v kapitole 1.1. Potom užívateľ zadá bod B, ktorý generuje všetky body krivky E alebo pre ktorý platí (rád B) $> k.n$ kde k je súkromný kľúč a n je maximálny prvok postupnost $n_1, ..., n_d$ prirodzených čísel slúžiacich na kódovanie. Menovateľ vo funkciách `xko, yko, xko2, yko2` sa nahrádza inverzným prvkom, ktorý sa nájde výpočtom Bézoutových koeficientov. V prípade, že B nespĺňa danú podmienku, nastane problém vo chvíli keď B je násobený svojím rádom. Konkrétne v tomto kroku inverzný prvok neexistuje a teda použitý bezoutov koeficient je nekorektný. Program neoveruje správnosť užívateľom zadaného bodu B.

Následne si užívateľ zvolí či chce svoju správu kódovať alebo dekódovať. V prípade

1. voľby kódovania: zadá text svojej správy veľkými písmenami, bez medzier a postupnosť prirodzených čísel, ktoré budú pri kódovaní použité (viď stať 2.3.3). V prípade, že sa počet prvkov tejto postupnosti nebude zhodovať s počtom znakov správy, procedúra `sifruj` vypíše chybové hlásenie.
   Na výstupe dostane užívateľ informácie, či voľba κ a prvočísla bola vhodná a postupnosť bodov odpovedajúcich správe.

2. voľby dekódovania: zadá postupnosť bodov správy. Postupnosť musí byť zadaná v tvare x1,y1,x2,y2,.... V prípade chybného zadania tvaru správy procedúra `desifruj` buď nedá žiaden výstup alebo dá nesprávny výstup, ale nevypíše chybové hlásenie.
   Na výstupe dostane užívateľ dekódovanú správu.

Poznámka 3.1. Voľba bodu B je najjednoduchšia v prípade kriviek, ktorých počet bodov N je prvočíslo. Pre tieto krivky je možné použiť ich ľubovolný bod rôzny od $\mathcal{O}$. Príkladom prvočíselnej charakteristiky, pre ktorú má krivka $y^2 + y = x^3 - x$ prvočíselný počet bodov je 751. Vtedy je $N = 727$ podľa 3. príkladu v [4], str. 185. V praxi sa používajú telesá podstatne väčších charakteristík.

Zdrojový kód programu je uvedený v prílohe.

Záver

Náplňou bakalárskej práce bol opis kryptografických systémov založených na eliptických krivkách. Hlavným cieľom bolo zhrnúť viaceré známe fakty o ECC a predstaviť konkrétne systémy.

Eliptické krivky prinášajú do sveta kryptografie nástroj na skrátenie dĺžky kľúča pri zachovaní požadovanej bezpečnosti. Tento pokrok je veľmi podstatný vzhľadom k rýchlemu technologickému vývoju. Do roku 2010 odporúčal NIST pre RSA 1024 bitový kľúč čo je v prípade ECC ekvivalentné 160 bitovému kľúču. V súčasnosti je už 1024 bitový kľúč pre RSA považovaný za málo bezpečný.

Zachovanie bezpečnosti pri použití rádovo kratšieho kľúča nie je jediná výhoda systémov ECC. V porovnaní s klasickými systémami sú aj výpočetne oveľa efektivnejšie.

V roku 1997 firma Certicom vypísala výzvu na výpočet súkromného kľúča ECC. Výzva sa nazýva *Certicom Elliptic Curve Cryptosystem Challenge*[4]. Poďla dĺžok kľúčov je rozdelená do dvoch úrovní. Úlohou je spočítať súkromný kľúč z príslušného zoznamu verejných kľúčov a súvisiacich systémových parametrov. Jedná sa o typ problému, ktorému čelí protivník, ktorý chce prelomiť kryptografický systém. Výzva sa týka eliptických kriviek definovaných nad konečnými telesami $\mathbb{F}_{2^m}$ a $\mathbb{F}_p$. Najväčšia dĺžka kľúča ECC, ktorá bola doteraz verejne prelomená, v prípade prvočíselnej charakteristiky telesa, je 112 bitov. K prelomeniu došlo v Apríly 2009 a výpočet trval približne 3 mesiace a 2 týždne.

V dôsledku výhod, ktoré systémy ECC ponúkajú je ich použitie stále častejšie v súvislosti s čipovými kartami, pri komplexných schémach riešenia prístupu k elektronickým čipovým dokladom a štandardne sú podporované v moderných operačných systémoch.

[4]Viac informácií je možné nájsť na webových stránkach firmy Certicom

Použitá literatúra

[1] ElGamal,T. (1985): *A public cryptosystem and a signature scheme based on discrete logarithms.* IEEE Trans. on Information Theory, 469-472.

[2] Drápal, A. (2009): *Skriptá k predmetu NMIB001-Teória čísel a RSA.* `http://www.karlin.mff.cuni.cz/~holub/soubory/skriptaRSA.pdf`

[3] Klíma, V.: *Eliptické krivky a šifrováni (1).* Chip, september 2002.

[4] Koblitz, N. (1994): *A Course in Number Theory and Cryptography.* Springer-Verlag New York, Inc., 2nd ed.

[5] Pinkava, J.: *Jak je to s bezpečností eliptických kryptosystému?* Crypto-World, november 1999. `http://crypto-world.info/casop1/crypto11_99.pdf`

[6] Somberg, P. (2010): *Elliptic curves.* `http://www.karlin.mff.cuni.cz/~somberg/lectures.htm`

[7] Stanovský, D. (2010): *Skriptá k predmetu NMIB003-Počítačová algebra.*

Príloha

```
(*VZORCE NA SUCET*)

(*dvojnasobok, suradnica x*)
xko[x_, y_, prvocislo_]:=((3 x^2 - 1).
                          ExtendedGCD[2 y, prvocislo][[2,1]])^2
                          - 2 x;

(*dvojnasobok, suradnica y*)
yko[x_, y_, prvocislo_]:=((3 x^2 - 1).
                          ExtendedGCD[2 y, prvocislo])[[2,1]].
                          (x - xko[x, y]) - y;

(*sucet, suradnica x*)
xko2[x1_, y1_, x2_, y2_, prvocislo_]:=((y2 - y1).
                                       ExtendedGCD[x2 - x1, prvocislo][[2,1]])^2
                                         - x1 - x2;

(*sucet suradnica y*)
yko2[x1_, y1_, x2_, y2_, prvocislo_]:= -y1 + ((y2 - y1).
                                        ExtendedGCD[x2 - x1, prvocislo][[2,1]])
                                        (x1 - xko2[x1, y1, x2, y2]);

(*sucet 2 rovnakych bodov na krivke*)
sucetA = Function[{vstupX, vstupY, vystupA, vystupB, prvocislo},
   Module[{pomx, pomy, pomx2},
    pomx = vstupX; pomy = vstupY;
    pomx2 = Mod[xko[pomx, pomy], prvocislo];
    pomy = Mod[yko[pomx, pomy], prvocislo];
    vystupA := pomx2; vystupB := pomy]];

(*sucet 2 roznych bodov na krivke*)
sucetB = Function[{vstupX1, vstupY1, vstupX2, vstupY2, vystupX,
 vystupY, prvocislo},
   Module[{pomx2, pomy},
    pomx2 = Mod[xko2[vstupX1, vstupY1, vstupX2, vstupY2], prvocislo];
    pomy = Mod[yko2[vstupX1, vstupY1, vstupX2, vstupY2], prvocislo];
    vystupX := pomx2; vystupY := pomy]];

(*NASOBENIE BODU NA KRIVKE*)
nasobenie = Function[{vsX, vsY, vysX, vysY, krat, prvocislo},
   Module[{pomocx, pomocy, pomocx2, pomocy2, i, a, mocnina, j},
    pomocx := vsX; pomocy := vsY;
    mocnina = IntegerDigits[krat, 2];
    a = Length[mocnina];
    i = 0;
    While[mocnina[[a - i]] == 0, i++];
    For[j = 0, j < i, j++,
     sucetA[pomocx, pomocy, pomocx2, pomocy2, prvocislo];
     pomocx = pomocx2; pomocy = pomocy2;
```

```
     Clear[pomocx2, pomocy2]];
    vysX = pomocx; vysY = pomocy;
    If[i + 1 <= a,
     For[j = i + 1, j < a, j++,
      sucetA[pomocx, pomocy, pomocx2, pomocy2, prvocislo];
      pomocx = pomocx2; pomocy = pomocy2;
      Clear[pomocx2, pomocy2];
      If[mocnina[[a - j]] == 1,
       sucetB[vysX, vysY, pomocx, pomocy, pomocx2, pomocy2,
        prvocislo]; vysX = pomocx2; vysY = pomocy2;
       Clear[pomocx2, pomocy2], a = a]], a = a]];

(*overenie*)
over[k_, prvocislo_, vzor_] := Module[{max, i, d},
   If[PrimeQ[prvocislo],
    If[Mod[prvocislo, 4] == 3, Print["Prvocislo je zvolene spravne"],
     Print["Nespravne zvolene prvocislo"]],
    Print["Nezadali ste prvocislo"]];
   d = Length[vzor];
   max = vzor[[1]];
   For[i = 2, i <= d, i++,
    If[vzor[[i]] > max, max = vzor[[i]], max = max]];
   If[prvocislo > (k (max + 1)),
    Print["Nasobok na prevod znaku urceny spravne"],
    Print["Nasobok na prevod znaku je prilis velky, kodovanie ne
           prebehne spravne"]]];

(*sprava*)
nacitaj := Module[{vzor, d, nasobky, i},
   vzor = ToCharacterCode[InputString["sprava na sifrovanie"]];
   d = Length[vzor];
   For[i = 1, i <= d, i++,
    If[vzor[[i]] > 64, vzor[[i]] = (vzor[[i]] - 55),
     vzor[[i]] = (vzor[[i]] - 48)]]; vzor];

(*vypocet bodov na krivke co odpovedaju sprave*)
prevodX[vzor_, k_, prvocislo_] := Module[{i, z, j, Px = {},
  d = Length[vzor], x},
   For[i = 1, i <= d, i++,
    z = vzor[[i]] k;
    x = (prvocislo - 1); j = 1;
    While[x > 1, x = Mod[f[z + j]^((prvocislo - 1)/2), prvocislo]; j++];
    x = Mod[z + j - 1, 751];
    Px = Append[Px, x];]; Px];

prevodY[prvocislo_, x_] := Module[{i, y, Py = {}, d = Length[x]},
   For[i = 1 , i <= d, i++,
    y = Mod[f[x[[i]]]^((prvocislo + 1)/4), prvocislo];
    Py = Append[Py, y]]; Py];

(*sifrovanie*)
```

```
sifruj[Px_, Py_, prvocislo_] := Module[{l, kklucX, kklucY, kBx,
  kBy, vyslX, vyslY, d},
   d = Length[Px];
   If[Length[nasobky] == d,
    Print["Sprava sa zasifruje do nasledujucich ", d,
          " dvojic bodov na krivke"];
    For[l = 1, l <= d, l++,
     Clear[kBx, kBy, kklucX, kklucY];
     nasobenie[Bx, By, kBx, kBy, nasobky[[l]], prvocislo];
     nasobenie[klucX, klucY, kklucX, kklucY, nasobky[[l]], prvocislo];
     If[kklucX == Px[[l]],
      If[kklucY == -Py[[l]], Print["point at infinity"],
       If[kklucY == Py[[l]], Clear[vyslX, vyslY];
        sucetA[kklucX, kklucY, vyslX, vyslY, prvocislo],
        Clear[vyslX, vyslY];
        sucetB[kklucX, kklucY, Px[[l]], Py[[l]], vyslX, vyslY,
         prvocislo];]],
      Clear[vyslX, vyslY];
      sucetB[kklucX, kklucY, Px[[l]], Py[[l]], vyslX, vyslY,
       prvocislo];];
     Print["{", kBx, ",", Mod[kBy + (prvocislo - 1)/2, prvocislo],
           "},{", vyslX, ",", Mod[vyslY + (prvocislo - 1)/2, prvocislo],
           "}"];
     ]
    , Print["pocet prirodzenych cisel sluziacich na sifrovanie sa
             nezhoduje s dlzkou spravy"]]];

(*desifrovanie*)
desifruj[prvocislo_, k_] := Module[{sprava, d, v, x1, y1, vyslX, vyslY},
   sprava = Input["zadaj spravu v tvare
                  {{x1,y1},{x2,y2},{x3,y3},{x4,y4},...}"];
   d = Length[sprava]/2;
   For[v = 1, v <= d, v++,
    sprava[[2 v - 1, 2]] =
     Mod[sprava[[2 v - 1, 2]] - (prvocislo - 1)/2, prvocislo];
    sprava[[2 v, 2]] =
     Mod[sprava[[2 v, 2]] - (prvocislo - 1)/2, prvocislo];
    Clear[x1, y1];
    nasobenie[sprava[[2 v - 1, 1]], sprava[[2 v - 1, 2]], x1, y1,
     kluc, prvocislo];
    y1 = -y1;
    Clear[vyslX, vyslY];
    If[sprava[[2 v, 1]] == x1,
     If[sprava[[2 v, 2]] == y1, Clear[vyslX, vyslY];
      sucetA[x1, y1, vyslX, vyslY, prvocislo], Clear[vyslX, vyslY];
      sucetB[sprava[[2 v, 1]], sprava[[2 v, 2]], x1, y1, vyslX, vyslY,
        prvocislo]], Clear[vyslX, vyslY];
     sucetB[sprava[[2 v, 1]], sprava[[2 v, 2]], x1, y1, vyslX, vyslY,
      prvocislo]];
    vyslX = Mod[vyslX, prvocislo];
    If[IntegerPart[(vyslX - 1)/k] > 9,
```

```
     Print[FromCharacterCode[IntegerPart[(vyslX - 1)/k] + 55]],
     Print[FromCharacterCode[IntegerPart[(vyslX - 1)/k] + 48]]]]];

(*HLAVNY PROGRAM*)

(*Vhodne testovacie parametre su napriklad: kappa=20, prvocislo=751,
sukromny kluc 31, prirodzene cisla na kodovanie {51,304,109,518,402}.
Pri takychto parametroch a spravu MFFUK zakoduje nasledovne:
{{639,718},{36,711},{172,120},{251,319},{392,34},{232,425},{595,296},
{253,483},{403,593},{359,79}} *)

(*Upozornenie: pri desifrovani zadavajte suradnice kazdeho bodu do
mnozinovych zatvoriek {} a taktiez celu postupnost bodov do
mnozinovych zatvoriek, vid. testovaci priklad. *)

(*VSTUPNE PARAMETRE*)
(*kappa na prevod na bod na krivke*)
kappa = Input["Zvolte nasobok na prevod znaku na bod na krivke (kappa)"];


(*prvocislo urcujuce konecne teleso*)
p = Input["Zvolte prvocislo (charakteristika telesa F_p), pre ktore
          plati p=3 mod 4"];

(*sukromny kluc*)
kluc = Input[
   "Zvolte sukromny kluc k"];

(* elipticka krivka v tvare y^2+y=x^3-x upravena do tvaru y^2=x^3-x-b *)
f[m_] := m^3 - m - Mod[(p - 1)^2/4 + (p - 1)/2, p];

(*nacitanie generatora a uprava vzhladom k substitucii*)
b = Input["Zadajte suradnice generatora B v tvare {x,y}"];
Bx = b[[1]];
By = Mod[b[[2]] - (p - 1)/2, p];

a = Input["Ak chcete sifrovat stlacte 1, ak desifrovat 2"];
If[a == 1,

  (*KODOVANIE*)
  sprava = nacitaj;

  (*prirodzene cisla na sifrovanie*)
  nasobky =
   Input["Zvolte d prirodzenych cisel (d-dlzka spravy) sluziacich na
sifrovanie v tvare {n_1,...,n_d}, 1<n_i<p "];
  over[kappa, p, sprava];

  (*generovanie kluca*)
  Clear[klucX, klucY];
  nasobenie[Bx, By, klucX, klucY, kluc, p];
```

```
Print["Verejny kluc je (", klucX, ",", klucY, ")."];
X = prevodX[sprava, kappa, p];
Y = prevodY[p, X];
sifruj[X, Y, p],


(*DEKODOVANIE*)
desifruj[p, kappa]];
```