

# **Aplikace eliptických křivek v moderní kryptografii**

Application of elliptic curves in modern cryptography

František Špaček



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2011/2012

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **František ŠPAČEK**  
Osobní číslo: **A09575**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Aplikace eliptických křivek v moderní kryptografii**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Nastudujte a popište příslušný matematický aparát.
3. Popište problematiku využití eliptických křivek v moderních kryptografických systémech.
4. Vytvořte aplikaci, která bude demonstrovat aplikaci eliptických křivek v moderní kryptografii.
5. Porovnejte kryptografické systémy, založené na eliptických křivkách se současně rozšířenými šifrovacími systémy DSA / RSA.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. MENEZES, Alfred J, Paul VAN OORSCHOT a Scott VANSTONE. Handbook of applied cryptography. Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 08-493-8523-7.
2. KATZ, Jonathan a Yehuda LINDELL. Introduction to modern cryptography. Boca Raton: Chapman, 2008, 534 s. ISBN 978-158-4885-511.
3. TILBORG, C. Encyclopedia of cryptography and security. Vyd. 1. New York: Springer, 2005, 684 s. ISBN 03-872-3473-X.
4. HANKERSON, Darrel, Alfred J MENEZES a Scott VANSTONE. Guide to elliptic curve cryptography. Vyd. 1. New York: Springer, 2004, 311 s. ISBN 03-879-5273-X.
5. WASHINGTON, Lawrence C. Elliptic curves: number theory and cryptography. 2nd ed. Boca Raton, FL: Chapman, 2008, 513 s. ISBN 14-200-7146-7.
6. SWENSON, Christopher. Modern cryptanalysis: techniques for advanced code breaking. Indianapolis, IN: Wiley Pub., 2008, 236 s. ISBN 04-701-3593-X.

Vedoucí bakalářské práce:

**Ing. Roman Šenkeřík, Ph.D.**

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

**24. února 2012**

Termín odevzdání bakalářské práce:

**8. června 2012**

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.  
*děkan*



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

## ABSTRAKT

Táto bakalárska práca rozoberá využitie eliptických kriviek v moderných kryptografických systémoch. Práca popisuje potrebné matematické princípy a algoritmy, ktoré sa využívajú pri aritmetike eliptických kriviek. Ako konkrétne možnosti aplikácie sú rozoberané kryptografické systémy ECDH, ECDSA a ECIES. Popísané sú aj možnosti útoku na systémy, ktoré sa spoliehajú na problém diskretného logaritmu. Praktická časť rozoberá softvérové riešenie aplikácie, ktorá umožňuje vyskúšanie základných operácií založených na eliptických krivkách a demonštruje princípy kryptografických systémov spomenutých vyššie.

### Kľúčové slová:

Eliptická krivka, elektronický podpis, asymetrická kryptografia, kryptografické systémy založené na eliptických krivkách

## ABSTRACT

This bachelor thesis discuss about usage of elliptic curves in modern cryptographic systems. Paper describes necessary mathematical principles and algorithms which are used in elliptic curves arithmetic. As specific possibilities of application are mentioned cryptographic systems such as ECDH, ECDSA and ECIES. Some discussion is about cryptanalysis method for systems that rely on discrete logarithm problem. Practice output is about software solution, which is useful for tryout basic operations over elliptic curve field. Application also demonstrates principles of cryptographic systems mentioned above.

### Keywords:

Elliptic curve, asymmetric cryptography, cryptographic systems base on elliptic curves, digital signature

**Pod'akovanie**

Ďakujem svojmu vedúcemu bakalárskej práce Ing. Romanovi Šenkeříkovi Ph.D., za jeho cenné rady a pripomienky. Taktiež by som sa chcel poďakovať svojim rodičom za to, že mi umožnili študovať na vysokej škole a za ich neustálu podporu.

**Motto**

*O eliptických krivkách je možné písať donekonečna. ( Toto nie je hrozba.)*

*It is possible to write endlessly on elliptic curves. (This is not a threat.)*

Serge Lang

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD.....</b>	<b>8</b>
<b>I TEORETICKÁ ČASŤ.....</b>	<b>9</b>
<b>1 ZÁKLADNÉ POJMY KRYPTOGRAFIE .....</b>	<b>10</b>
1.1 ZÁKLADNÉ POJMY .....	10
1.2 ELEKTRONICKÝ PODPIS .....	12
1.3 SYMETRICKÁ KRYPTOGRAFIA.....	13
1.4 ASYMETRICKÁ KRYPTOGRAFIA .....	14
<b>2 ELIPTICKÉ KRIVKY .....</b>	<b>17</b>
2.1 MATEMATICKÝ ZÁKLAD.....	17
2.2 ALGORITMY .....	20
2.2.1 Euklidov algoritmus .....	20
2.2.2 Shanks-Tonelli algoritmus .....	22
2.2.3 Right to Left algoritmus .....	24
2.3 PRINCÍP .....	25
2.3.1 Eliptické krivky nad množinou reálnych čísiel .....	25
2.3.2 Eliptické krivky nad telesom $F_p$ .....	29
2.3.3 Eliptické krivky nad telesom $F(2^m)$ .....	31
2.4 APLIKÁCIA V KRYPTOGRAFII .....	33
2.4.1 Prevod správy na bod eliptickej krivky .....	33
2.4.2 Elliptic Curve Diffie-Hellman.....	33
2.4.3 Elliptic Curve Integrated Encryption Scheme.....	35
2.4.4 Elliptic Curve Digital Signature Algorithm .....	36
2.5 KRYPTOANALÝZA .....	37
2.5.1 Baby-Step Giant-Step.....	37
2.5.2 Pollard $\rho$ metóda .....	38
2.6 ŠTANDARDY .....	40
<b>3 POROVNANIE SO SÚČASNE POUŽÍVANÝMI SYSTÉMAMI .....</b>	<b>42</b>
<b>II PRAKTICKÁ ČASŤ .....</b>	<b>44</b>
<b>4 SOFTVÉROVÉ RIEŠENIE .....</b>	<b>45</b>
4.1 POPIS RIEŠENIA.....	45
4.1.1 Knižnica ECC.....	45
4.1.2 Windows Presentation Foundation .....	48
4.1.3 Návrhový vzor MVVM.....	49
4.1.4 Grafické rozhranie.....	50
4.2 UŽÍVATEĽSKÁ PRÍRUČKA .....	52
<b>ZÁVER .....</b>	<b>56</b>
<b>CONCLUSION .....</b>	<b>57</b>
<b>ZOZNAM POUŽITEJ LITERATÚRY.....</b>	<b>58</b>
<b>ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK.....</b>	<b>61</b>
<b>ZOZNAM OBRÁZKOV .....</b>	<b>62</b>
<b>ZOZNAM TABULIEK .....</b>	<b>63</b>

## ÚVOD

Ľudia majú už od nepamäti potrebu ukrývať osobné a dôležité informácie pred zvedavými spoluobčanmi. Šifrovanie správ naberalo na dôležitosť postupnou aplikáciou vo vojenskej a politickej oblasti. Prvým takto využívaným algoritmom bola tzv. Ceasarova šifra. Išlo o jednoduchý posun každého znaku latinskej abecedy o tri znaky doprava. Dešifrovanie prebiehalo inverzným spôsobom. Od týchto dôb prešla kryptografia značným vývojom. Najviac sa rozvinula v 20. storočí, počas prvej a druhej svetovej vojny. V súčasnosti sa s kryptografiou stretávame prakticky denne a možno ani o tom nevieme.

Táto práca je zameraná na aplikáciu eliptických kriviek v súčasnej (modernej) kryptografii. Kryptografické systémy založené na eliptických krivkách sa v dnešnej dobe ukazujú ako veľmi perspektívne najmä vďaka tomu, že umožňujú použitie kratších kľúčov ako pri využívaní súčasných majoritných systémov (RSA, DSA), pri zachovaní približne rovnakej miery zabezpečenia.

Eliptické krivky sa začali pomaly, ale isto nasadzovať na reálne systémy. Napríklad zabezpečené webové aplikácie od spoločnosti Google, začali nedávno využívať výmenný mechanizmus kľúčov založených na eliptických krivkách.

Hlavnou nevýhodou eliptických kriviek je nutná investícia do počiatočného výskumu, pretože ich aplikovanie v kryptografii nie je tak dobre zdokumentované ako pri známych systémoch RSA/DSA a taktiež veľa efektívnych algoritmov je chránených rôznymi patentmi, čo predstavuje ďalšie náklady na poplatky.

V prvej kapitole sú vysvetlené základne pojmy používané v kryptológii, ktorým je nutné rozumieť pre pochopenie ďalšieho textu.

Druhá kapitola predstavuje nutné minimum pre porozumenie a implementovanie systému založenom na eliptických krivkách pre edukačné účely. Pri načrtnutí niektorej zaujímavej oblasti iba okrajovo je priamo v texte odkázané na rozširujúcu literatúru.

V praktickej časti je popísaná softvérová implementácia. Softvérová aplikácia je zameraná na vizualizáciu princípov popísaných v tejto práci. Ako implementačný jazyk bol použitý programovací jazyk C# a platforma .NET.



## **I. TEORETICKÁ ČASŤ**

# 1 ZÁKLADNÉ POJMY KRYPTOGRRAFIE

## 1.1 Základné pojmy

### Kryptológia

Je to veda, ktorá skúma utajovanie obsahu správ. Dnešná kryptológia je považovaná za časť matematiky, keďže využíva pokročilé matematické princípy. Kryptológia sa delí na kryptografiu a kryptoanalýzu, prípadne sa ešte zaraduje samostatne steganografia.

### Kryptografia

Je to vedná disciplína kryptológie, ktorá skúma a navrhuje šifrovacie systémy. Tieto systémy musia spĺňať podmienky ako autenticitu, dôvernosť, dostupnosť, integritu a pôvod dát. Popisuje mechanizmy, ktoré sú využívané ku ukrytiu obsahu správy (otvorený text) do nečitateľnej podoby (šifrovaní text). Tento postup sa nazýva šifrovanie. Opačný postup ku šifrovaniu sa nazýva dešifrovanie.

### Steganografia

Oblasť kryptológie, ktorá má za cieľ utajiť správu do inej správy. Príkladom môže byť použitie neviditeľného atramentu. V informačných technológiách je možné správy ukryť pomocou redundantných bitov v obrázkových, zvukových prípadne video súboroch.

### Kryptoanalýza

Vedná disciplína kryptológie, ktorá dáva za cieľ hľadať spôsoby ako prelomiť šifrovacie systémy. Je to veľmi dôležitá súčasť kryptológie, pretože z histórie poznáme, že úspešné lúštenie šifier rozhodovalo o ľudských životoch.

### Frekvenčná analýza

Je to najznámejšia metóda kryptoanalýzy. Spočíva v tom, že je známa frekvencia výskytu jednotlivých znakov prakticky každého jazyka. Pri použití substitučnej šifry sa totiž frekvencia výskytu jednotlivých znakov nemení, mení sa len znak pod ktorým sa znak v šifrovanom texte vyskytuje. Preto jednoduchým porovnaním môžeme zistiť, ktorý znak šifrovaného textu predstavuje znak v abecede daného jazyka. [1]

**Kľúčový pár**

Predstavuje množinu, ktorá obsahuje verejný a privátny kľúč. Najčastejšie verejný kľúč slúži na šifrovanie a je voľne dostupný. Privátny kľúč slúži zvyčajne na dešifrovanie obsahu. Privátny a verejný kľúč sú istým matematickým spôsobom spojené.

**Hešovacia funkcia**

Je to jednosmerná matematická funkcia, ktorá prevedie vstupné dáta konečnej dĺžky na krátky výstupný reťazec fixnej dĺžky. Nazýva sa aj odtlačok, heš (hash), kontrolný súčet. Používa sa na overovanie pravosti dát, hľadanie duplicít atď. Veľký význam má pri elektronickom podpise. Najznámejšími hešovacími funkciami sú MD5 a SHA-1. MD5 už nie je doporučená pre používanie, pretože bol dokázaný kolízny útok, kedy dve rôzne správy mali rovnaký heš. [2]

**Bloková šifra**

Je to šifra, ktorá spracováva dáta po blokoch fixnej veľkosti. V prípade, ak nie je možné rozdeliť dáta na rovnaké bloky, sú doplnené nulami. Príkladom sú šifry DES a AES.

## 1.2 Elektronický podpis

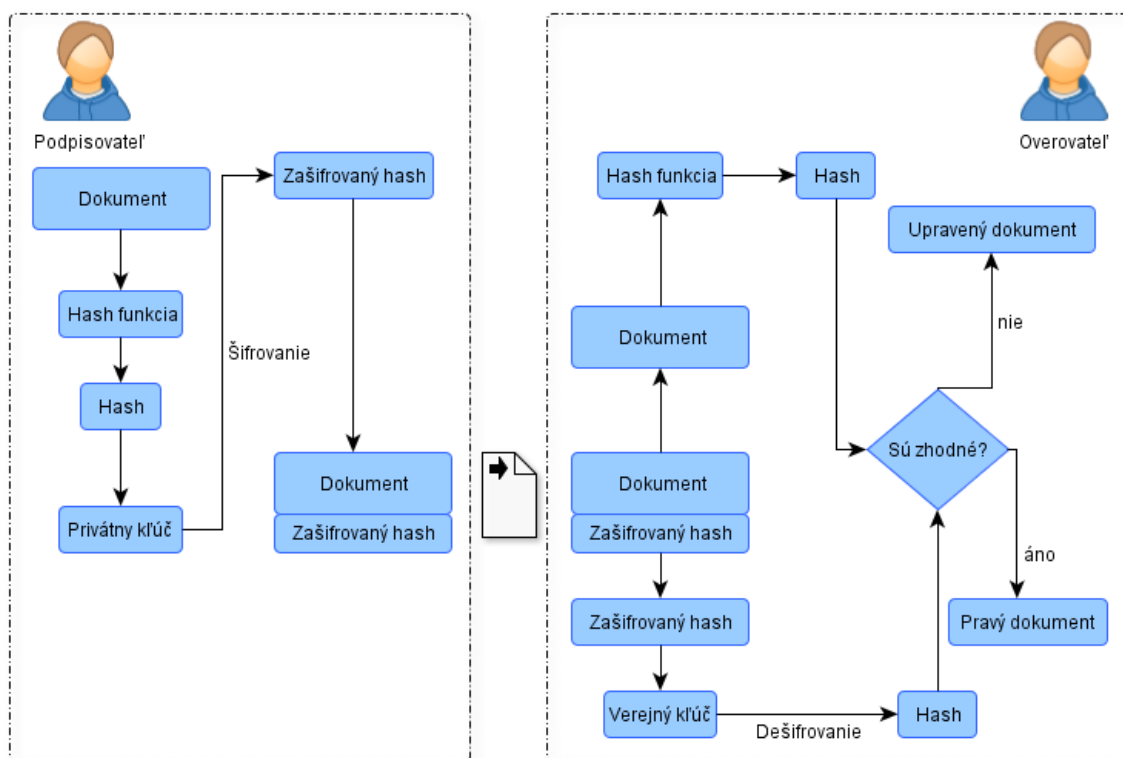
Elektronický podpis je unikátny, jednoznačný identifikátor dokumentu, na základe ktorého sa dá overiť jeho pravosť a autor dokumentu.

Základným princípom elektronického podpisu je vypočítanie hešu (odtlačku) dokumentu, ktorý chceme podpísať. Elektronický podpis potom predstavuje zašifrovanie tohto odtlačku pomocou nejakého asymetrického kryptografického systému.

Osoba A (Podpisovateľ) zašifruje heš pomocou súkromného kľúča. Takto zašifrovaný odtlačok je pripojený k dokumentu a tým sa z neho stáva elektronický podpis. Keďže odtlačok bol zašifrovaný asymetrickým kryptografickým algoritmom, je možného ho odšifrovať iba verejným kľúčom, ktorý osoba A zverejní spolu s dokumentom.

Osoba B (Overovateľ) si vďaka znalosti verejného kľúča, hešovacej funkcie a dokumentu overí či je dokument pravý, alebo či nebol zmenený počas prenosu. Schematický znázornený postup aplikovaný pri elektronickom podpise je zobrazený na obrázku 1. [3]

Pre viac informácií o elektronickom podpise odporúčam publikáciu Jiřího Peterku [4].



Obrázok 1 : Schéma vytvorenia a overovania elektronického podpisu

### 1.3 Symetrická kryptografia

Princíp symetrickej kryptografii spočíva vo využití rovnakého kľúča pre šifrovanie ako aj dešifrovanie správy (dát).

Medzi najjednoduchšie symetrické šifry patrí v úvode spomenutá Ceasarova šifra, ktorú je možné jednoducho lúštiť pomocou frekvenčnej analýzy. V dnešnej dobe sa už nevyužíva, avšak v upravenej podobe sa často vyskytuje na rôznych filmových fórach. Úprava sa nazýva ROT13 (posun o 13) a používa sa na skrývanie filmových spoilerov<sup>1</sup>.

Do kategórie symetrických kryptografických systémov patrí aj slávny nemecký systém Enigma. Princíp šifrovania je založený na transpozícii s časovo premenným algoritmom.

Algoritmus je určený nastavením stroja – typom použitých rotorov, ich pozíciou vo zväzku, vzájomnou polohou a nastavením komutačnej dosky. Rotorov je k dispozícii 5 rôznych typov a sú dostupné štyri pozície pre rotory. Každý rotor je možné nastaviť do jednej z 26 polôh. Každým stlačením klávesy sa rotory pootočia. Vďaka tejto rôznorodej konfigurácii bolo veľmi komplikované tento systém prelomiť. [5]

S rozvojom výpočtovej techniky bolo nutné vymyslieť dômyselnejšie systémy pre šifrovanie správ. Preto bol v roku 1970 vyvinutý spoločnosťou IBM kryptografický systém DES, ktorý sa neskôr v roku 1977 stal americkým štandardom FIPS 46. [6]

V súčasnosti sa pôvodná verzia považuje za nespoľahlivú, pretože používa kľúč iba o dĺžke 64bitov, z toho 8 je kontrolných a 56 efektívnych. Algoritmus taktiež obsahuje aj slabiny, ktoré umožňujú prelomiť šifru útokom hrubou silou (brute force) za menej než 24 hodín.

Posledná revízia štandardu pre DES FIPS 46-3, hovorí o variante Triple DES(3TDES), ktorý pracuje s kľúčom o celkovej dĺžke 168 bitov. Tento variant funguje tak, že sa tri krát zašifrujú dáta pomocou DES, po každé s iným 64bitovým kľúčom. Tento variant je bezpečnejší, avšak oproti novším algoritmom ako napríklad AES je oveľa pomalším.

Funkcionálny zápis postupu 3TDES algoritmu je  $DES(k3, DES(k2, DES(k1, M)))$ , kde  $k3, k2, k1$  sú rôzne kľúče, a  $M$  je blok dát.

---

<sup>1</sup> Ukážky, úryvky z filmov prípadne seriálov, ktoré prezrádzajú zásadné fakty dejovej línie

V roku 1997 americký úrad NIST vyhlásil verejnú súťaž na výber nového symetrického blokového algoritmu, ktorý by nahradil starší DES. Víťazom sa stal algoritmus Rijndael od belgických autorov Joana Daemena a Vincenta Rijmena. Tento algoritmus sa stal štandardom FIPS 197 v roku 2002 pod názvom AES. [7]

AES spracováva dáta po 128bitových blokoch s využitím 128, 192 prípadne 256 bitovým kľúčom. Táto šifra využíva teoretické základy šifry DES. Princíp fungovania a implementácie je mimo zameranie tejto práce, preto pre viac informácií odporúčam preštudovať štandard FIPS 197. [8]

V súčasnosti neexistuje rozumný spôsob ako túto šifru prelomiť. [9]

Možné útoky na kryptografický systém AES popisuje známy kryptograf Bruce Schneier na svojom blogu [10].

Pri použití rovnakého kľúča pre obe kryptografické operácie spočíva komplikácia v distribúcii tohto kľúča medzi oboma stranami komunikačného kanálu.

## 1.4 Asymetrická kryptografia

Koncept kryptografie s verejným kľúčom bol prvý krát predstavený kryptografmi Whitom Diffie a Martinom Hellmanom v roku 1976.

Asymetrická kryptografia sa zaoberá systémami, ktoré využívajú kľúčový pár. Zabezpečenie takýchto systémov spočíva v nemožnosti riešiť niektoré matematické problémy v reálnom čase. Keďže využívajú pokročilú matematiku a pracujú s obrovskými číslami, nie sú vhodné na šifrovanie veľkého objemu dát. [11]

### RSA

RSA je asymetrickým kryptografickým systémom, ktorý bol prvý krát verejne predstavený v roku 1978. Názov RSA pochádza z prvých písmen priezvisk jeho vynálezcov, ktorými boli Ron Rivest, Adi Shamir a Leonard Adleman. Bezpečnosť tohto spôsobu asymetrického šifrovania je založený na tom, že nie je možné v rozumnom čase faktorizovať<sup>2</sup> veľké prvočísla. [1]

---

<sup>2</sup> Rozklad veľkého prvočísla na súčin prvočísel

---

**Generovanie kľúčového páru pre šifrovanie pomocou RSA**

---

1. Vygenerujeme si dve veľké a od seba vzdialené prvočísla  $p$  a  $q$ , každé približne rovnakej veľkosti.
2. Vypočítame  $n = pq$  a  $\phi = (p - 1)(q - 1)$ . ( $\phi$  je Eulerova funkcia)
3. Zvolíme si náhodné celé číslo  $1 < e < \phi$ , pričom musí platiť  $\text{nsd}(e, \phi) = 1$ .
4. Použitím rozšíreného Euklidovho algoritmu spočítame unikátne celé číslo  $d$ ,  $1 < d < \phi$ , pričom musí platiť  $ed = 1 \pmod{\phi}$ .
5. Verejný kľúč je  $(n, e)$  a súkromný kľúč je číslo  $d$ .

Celé čísla  $e$  a  $d$  použité pri generovaní kľúčového páru sa nazývajú šifrovací a dešifrovací exponent, číslo  $n$  sa nazýva modulus.

---

**Šifrovanie a dešifrovanie správy pomocou RSA**

---

1. Šifrovanie
  - a. Získame verejný kľúč  $(n, e)$ .
  - b. Otvorený text prevedieme na číslo  $m$  z intervalu  $[0, n - 1]$
  - c. Spočítame  $c = m^e \pmod{n}$ .
2. Dešifrovanie
  - a. Správu dešifrujeme pomocou privátneho kľúča  $d$
  - b. Vypočítame  $m = c^d \pmod{n}$ .

**DSA**

Ten systém šifrovania s verejným kľúčom bol navrhnutý priamo pre potreby digitálneho podpisu. Je americkým štandardom FIPS a je používaný v DSS. Tento algoritmus je variantom ElGamal podpisovej schémy. Využíva malé pologrupy z množiny prirodzených čísiel  $\mathbb{Z}_p^*$ , pričom si dáva za cieľ zmenšiť veľkosť podpisov. [11]

---

**Generovanie kľúčového páru pomocou DSA**

---

1. Vyberieme prvočíslo  $q$ , ktoré spĺňa podmienku  $2^{159} < q < 2^{160}$ .
2. Nájdeme 1024 bitové prvočíslo  $p$ , ktoré spĺňa vlastnosť, že  $q$  je deliteľné  $p - 1$ .
3. Zvolíme si element  $h \in Z_p^*$  a spočítame  $g = h^{\frac{p-1}{q}} \bmod p$ ; opakujeme pokiaľ  $g \neq 1$ . (g je generátor unikátnej cyklickej grupy rádu  $p$  v množine  $Z_p^*$ ).
4. Zvolíme si náhodné číslo  $x$  v intervale  $[1, q - 1]$ .
5. Spočítame  $y = g^x \bmod p$ .
6. Verejný kľúč je štvorica  $(p, q, g, y)$ , súkromným kľúčom je číslo  $x$ .

---

**Podpisovanie pomocou DSA**

---

1. Máme správu  $m$ .
2. Vybereme náhodné celé číslo  $k$  z intervalu  $[1, q - 1]$ .
3. Vypočítame  $r = (g^k \bmod p) \bmod q$ .
4. Vypočítame  $k^{-1} \bmod q$ .
5. Spočítame  $s = k^{-1}\{h(m) + xr\} \bmod q$ , kde  $h$  je Secure Hash Algorithm (SHA-1).
6. Ak  $s = 0$ , tak sa vrátime ku kroku 1. (Ak je  $s = 0$ , potom  $s^{-1} \bmod q$  neexistuje;  $s^{-1}$  je vyžadované pre druhý krok verifikácie podpisu).
7. Podpis správy  $m$  predstavuje pár celých čísiel  $(r, s)$ .

---

**Overenie podpisu vytvoreného pomocou DSA**

---

1. Máme verejný kľúč  $(p, q, g, y)$ .
2. Vypočítame  $w = s^{-1} \bmod q$  a  $h(m)$ .
3. Spočítame si  $u_1 = h(m)w \bmod q$  a  $u_2 = rw \bmod q$ .
4. Vypočítame  $v = (g^{u_1} y^{u_2} \bmod p) \bmod q$ .
5. Podpis je akceptovaný iba ak je splnené, že  $v = r$ .



## 2 ELIPTICKÉ KRIVKY

Zmienka o využitie eliptických kriviek v kryptografii sa objavila v roku 1985, keď nezávisle na sebe predstavili Victor Miller a Neal Koblitz kryptografický systém s verejným kľúčom, kde je modulárna aritmetika nahradená operáciami definovanými nad eliptickou krivkou. [12]

Hlavnou výhodou použitia takéhoto systému je relatívne malá veľkosť kľúčov, 160 bitový kľúč v kryptografickom systéme založenom na eliptických krivkách je považovaný za rovnako bezpečný ako napríklad 1024 bitový kľúč v prípade systému RSA. [12]

### 2.1 Matematický základ

Pre pochopenie ako je možné využiť eliptické krivky pre kryptografické účely je nutné oboznámiť sa s niektorými pojmami z algebry a teórie čísiel. Definície z algebry sú prevzaté z [13] a [14].

#### Binárna operácia

Ak je  $G$  neprázdna množina, potom binárnou operáciou nad  $G$  rozumieme ľubovoľné zobrazenie  $\circ : G \times G \rightarrow G$ .

To znamená, že binárna operácia " $\circ$ " priradí každej usporiadanej dvojici  $(x, y)$  prvkov z  $G$  jednoznačne určený prvok  $\circ(x, y)$ , ktorý budeme značiť  $x \circ y$  a nazývať výsledkom operácie " $\circ$ ". Pri konkrétnych binárnych operáciách budeme miesto symbolu " $\circ$ " používať symboly "+", " $\cdot$ ", "...".

#### Algebraická štruktúra

Pod pojmom algebraická štruktúra rozumieme neprázdnu množinu  $G$ , spolu s neprázdny systémom  $\{f_\alpha ; \alpha \in I\}$   $n$ -árnych algebraických operácií na množine  $G$ . (číslo  $n$  môže byť pre rôzne operácie rôzne.) Značíme  $\mathcal{G} = (G, f_\alpha ; \alpha \in I)$ .

#### Grupoid

- Ak je " $\circ$ " binárna operácia na množine  $G \neq \emptyset$ , potom algebraická štruktúra  $\mathcal{G} = (G; \circ)$  sa nazýva grupoid.
- Ak je " $\circ$ " komutatívna, tj. ak platí  $\forall a, b \in G; a \circ b = b \circ a$ , potom sa  $\mathcal{G}$  nazýva komutatívny grupoid.

### Pologrupa

Ľubovoľný grupoid  $\mathcal{G} = (G; \circ)$ , v ktorom je operácia " $\circ$ " asociatívna, tj. platí

$$\forall a, b, c \in G; a \circ (b \circ c) = (a \circ b) \circ c.$$

### Neutrálny prvok

Povedzme, že grupoid  $\mathcal{G} = (G; \circ)$  má neutrálny prvok, ak je pravdivý výrok

$$\exists n \in G \forall a \in G; a \circ n = a = n \circ a.$$

Potom sa  $n$  nazýva neutrálnym prvkom grupoidu. V grupoide  $G$ , môže existovať najviac jeden neutrálny prvok.

Grupoid  $\mathcal{G} = (G; +)$  sa nazýva aditívny. V prípade aditívneho grupoidu sa neutrálny prvok označuje  $0$  a nazýva sa nulový prvok.

Grupoid  $\mathcal{G} = (G; \cdot)$  sa nazýva multiplikatívny. V prípade multiplikatívneho grupoidu sa neutrálny prvok označuje  $e$  a nazýva sa jednotkový prvok.

### Symetrický prvok

Nech grupoid  $\mathcal{G} = (G; \circ)$  má neutrálny prvok  $n$  a nech  $a \in G$ . Potom  $a^* \in G$ , sa nazýva symetrickým prvkom k prvku  $a$ , ak platí

$$a \circ a^* = n = a^* \circ a.$$

Symetrický prvok v aditívnom grupoide sa označuje  $-a$  a nazýva sa opačný prvok. Symetrický prvok v multiplikatívnom grupoide sa označuje  $a^{-1}$  a nazýva sa inverzný prvok.

### Grupa

Pologrupa  $\mathcal{G} = (G; \circ)$  sa nazýva grupa, ak obsahuje neutrálny prvok a ak v nej existuje ku každému prvku symetrický prvok.

### Okruh

Okruhom nazveme trojicu  $\mathcal{R} = (G; +, \cdot)$ , kde  $G \neq \emptyset$  je množina, " $+$ " a " $\cdot$ " sú binárne operácie na  $G$ , ktoré splňujú

- $(G; +)$  je komutatívna grupa
- $(G; \cdot)$  je pologrupa
- Pre každé tri prvky  $a, b, c \in G$  platí

$$\left. \begin{aligned} a \cdot (b + c) &= a \cdot b + a \cdot c - \text{ľavý} \\ (b + c) \cdot a &= b \cdot a + c \cdot a - \text{pravý} \end{aligned} \right\} \text{distributívny zákon.}$$

Príkladom môže byť okruh celých čísiel  $(\mathbb{Z}; +, \cdot)$ .

### Teleso

- a) Okruh  $\mathcal{R} = (G; +, \cdot)$  sa nazýva telesom, ak obsahuje aspoň dva rôzne prvky a sú splnené podmienky

$$\begin{aligned} \exists 0 \neq e \in G \forall 0 \neq a \in G; ae = a = ea, \\ \forall 0 \neq a \in G \exists 0 \neq a^{-1} \in G; aa^{-1} = a = a^{-1}a. \end{aligned}$$

- b) Ak je pologrupa  $(G; \cdot)$  komutatívna, potom  $\mathcal{R}$  sa nazýva komutatívnym telesom.

### Kvadratické rezíduum

„Číslo  $a \in \mathbb{Z}_n^*$  (teda číslo z množiny  $\{1, \dots, n-1\}$ , ktoré je nesúdeliteľné s  $n$ ) sa nazýva kvadratickým rezíduom modulo  $n$  ( $QR_n$ ), ak existuje  $b \in \mathbb{Z}_n$  také, že  $b^2 \equiv a \pmod{n}$ . Ak také  $b$  neexistuje, nazývame  $a$  kvadratickým nerezíduom modulo  $n$  ( $QNR_n$ )“. [15]

### Modulárna inverzia

Pod pojmom modulárna inverzia sa skrýva netriviálne riešenie operácie, ktorá je definovaná ako  $a^{-1} \equiv x \pmod{p}$ . K riešeniu modulárnej inverzie sa využíva rozšírený Euklidov algoritmus. V celočíselných telesách predstavuje inverzia zápis operácie delenia.

### Bod v nekonečne

Je to nulový bod eliptickej krivky. Označujeme ho ako  $O$ . Pre tento bod platí, že

$$P + O = P \text{ a } P + (-P) = O.$$

### Rád bodu

Za rád bodu  $P$  je považované také číslo  $n$  pre ktoré platí  $nP = O$ .

### Rád eliptickej krivky

Za rád krivky  $\#E(Fp)$  považujeme počet všetkých bodov eliptickej krivky  $E$  nad konečným poľom  $Fp$ . Medzi body eliptickej krivky patrí aj bod v nekonečne.

### Kofaktor

Kofaktor predstavuje číselne vyjadrenie podielu rádu krivky a rádu bodu  $h = \frac{\#E}{n}$ , pričom cieľom je aby bol čo najmenší, ideálne  $h = 1$ .

## Diskrétny logaritmus

Problém diskretného logaritmu predstavuje základný bezpečnostný pilier mnohých moderných kryptografických systémoch. Príkladom môže byť známy systém DSA. V prípade eliptických kriviek sa tento problém označuje ako problém diskretného logaritmu eliptických kriviek (ECDLP).

Body  $P$  a  $Q$  sú bodmi eliptickej krivky pre ktoré platí, že  $kP = Q$ , pričom  $k$  je skalár. Pri znalosti  $P$  a  $Q$  nie je možné v reálnom čase získať číslo  $k$ , ak je  $k$  dostatočne veľké. Číslo  $k$  predstavuje diskretný logaritmus  $Q$  pri základe  $P$ . Metódy, ktoré je možné použiť na výpočet diskretného logaritmu sú popísané v kapitole 2.5.

## 2.2 Algoritmy

Táto podkapitola popisuje algoritmy, ktoré sú využívané pri aritmetike založenej na eliptických krivkách.

### 2.2.1 Euklidov algoritmus

Dve čísla sú *nesúdeliteľné*, ak nemajú žiadnych spoločných deliteľov okrem 1, čiže ak ich najväčší spoločný deliteľ je rovný číslu 1. Euklidov algoritmus sa často označuje GCD z anglického Greatest Common Divisor. [16]

---

#### *Euklidov algoritmus pre výpočet GCD dvoch celých čísel*

---

**VSTUP:** dve celé čísla  $a, b$  pričom  $a, b > 0$  a  $a \geq b$

**VYSTUP:** najväčší spoločný deliteľ  $a$  a  $b$

---

1. Pokiaľ  $b \neq 0$ , tak vykonaj

a. Nastav  $r \leftarrow a \bmod b$ ,  $a \leftarrow b$ ,  $b \leftarrow r$

2. Vráť  $(a)$ .

Nasledujúci príklad ukazuje výpočet GCD pomocou nerekurzívnej verzie. Príklad je prevzatý z [16].

Určite hodnotu  $\gcd(31615807, 2763323)$

$$31615807 = 1 \times 2763323 + 852484$$

$$2763323 = 3 \times 852484 + 205871$$

$$852484 = 4 \times 205871 + 29000$$

$$205871 = 7 \times 29000 + 2871$$

$$29000 = 10 \times 2871 + 290$$

$$2871 = 9 \times 290 + 261$$

$$290 = 1 \times 261 + 29$$

$$261 = 9 \times 29$$

$$\gcd(31615807, 2763323) = 29$$

Euklidov algoritmus je možné upraviť pre všeobecnejšie použitie (napr. počítanie s polynómami).

---

### *Rozšírený Euklidov algoritmus*

---

**VSTUP:** dve celé čísla  $a, b$  pričom  $a, b > 0$  a  $a \geq b$

**VYSTUP:**  $d = \gcd(a, b)$ , a celé čísla  $x, y$  splňujúce podmienku  $ax + by = d$

---

1. Nastav  $x \leftarrow 1$  a  $d \leftarrow a$ .
2. Ak  $b = 0$ , tak nastav  $y \leftarrow 0$  a vráť  $(d, x, y)$ .
3. Nastav  $x_1 \leftarrow 0, x_2 \leftarrow 1, y_1 \leftarrow 1, y_2 \leftarrow 0$
4. Pokiaľ  $b > 0$  vykonavaj
  - a.  $q \leftarrow [a/b], r \leftarrow a - q \times b, x \leftarrow x_2 - q \times x_1, y \leftarrow y_2 - q \times y_1$
  - b.  $a \leftarrow b, b \leftarrow r, x_2 \leftarrow x_1, x_1 \leftarrow x, y_2 \leftarrow y_1, y_1 \leftarrow y$
5. Nastav  $d \leftarrow a, x \leftarrow x_2, y \leftarrow y_2$  a vráť  $(d, x, y)$

Rozšírený Euklidov algoritmus sa používa hlavne na výpočet modulárnej inverzie, prípadne na riešenie diofantických rovníc. [1][16]

Obe spomenuté varianty Euklidovho algoritmu sú v nerekurzívnej podobe. V prípade softvérovej implementácie sa používajú častejšie ich rekurzívne podoby.

### 2.2.2 Shanks-Tonelli algoritmus

Tento algoritmus slúži na riešenie modulárnej rovnice typu  $y^2 = f(x) \bmod p$ , preto sa hodí pre počítanie bodov eliptickej krivky, najmä pri reprezentovaní správy pomocou bodov krivky.

Na testovanie toho či má číslo kvadratické rezíduum sa použije tzv. Legendre symbol, ktorý predstavuje modifikáciu Euleroveho kritéria. Algoritmus využíva poznatky teórie čísiel preto sa pre viac detailov odkazujem na [17][18][19].

---

#### *Legendre symbol - LS*

---

**VSTUP:** číslo  $a$ , prvočíslo  $p$

**VÝSTUP:**  $r$  – Legendre symbol

---

1.  $r = a^{\frac{p-1}{2}} \bmod p$ .
2. Ak  $r = p - 1$ , tak vráť  $-1$ , číslo  $a$  nie je kvadratickým rezíduom ku  $p$ .
3. Vráť  $r$ .

---

#### *Shanks-Tonelli algoritmus*

---

**VSTUP:** celé číslo  $a$ , prvočíslo  $p$ , platí že  $p > 2$  a  $p$  je relatívnym prvočísлом ku  $a$

**VÝSTUP:** hodnota  $y$ , ktorá predstavuje riešenie rovnice  $y^2 = f(x) \bmod p$

---

1. Ak  $LS(a, p) = -1$ , číslo  $a$  nemá kvadratické rezíduum ku  $p$ , vráť  $-1$ .
2. Ak  $a = 0$ , vráť  $0$ .
3. Ak  $p = 2$ , vráť  $p$ .
4. Ak  $p \bmod 4 = 3$ , vráť  $a^{\frac{p+1}{4}} \bmod p$ .
5. Ak  $LS(a, p) = 1$ , tak rozlož  $p - 1$  na tvar  $s \cdot 2^e$ , kde  $s$  je nepárne a  $e$  je kladné číslo
  - a.  $s \leftarrow p - 1; e \leftarrow 0$ ;
  - b. Pokiaľ platí  $s \bmod 2 = 0$ , vykonávaj

- i.  $s \leftarrow s \gg 1$ , kde  $\gg$  predstavuje bitový posun doprava („zmenšenie“ čísla).
  - ii.  $e \leftarrow e + 1$ .
- 6.  $n \leftarrow 2$ , hľadaj číslo  $n$  pokiaľ platí  $LS(n, p) = -1$  (nemá kvadratické rezíduum)
  - a.  $n \leftarrow n + 1$ .
- 7. Nastav nasledujúce premenné
  - a.  $x \equiv a^{\frac{s+1}{2}} \bmod p$ , prvé potenciálne kvadratické rezíduum
  - b.  $b \equiv a^s \bmod p$ , ako moc je tip na  $x$  mimo
  - c.  $g \equiv n^s \bmod p$
  - d.  $r = e$
- 8. Pokiaľ je pravda( nekonečný cyklus) vykonávajú
  - a.  $t \leftarrow b; m \leftarrow 0$ ;
  - b. Pokiaľ platí  $m < r$ , tak
    - i. Ak  $t = 1$ , preruš (break).
    - ii.  $t \leftarrow t^2 \bmod p$ .
    - iii.  $m \leftarrow m + 1$ .
  - c. Ak  $m = 0$ , vráť  $x$ .
  - d.  $gs \leftarrow g^{2^{r-m-1}} \bmod p$ ;  $g \leftarrow gs^2 \bmod p$ ;
  - e.  $x \leftarrow (x \times gs) \bmod p$ ;  $b \leftarrow (b \times g) \bmod p$ ;  $r = m$ ;

Ak výsledkom algoritmus nie je  $-1$ , tak riešením rovnice je aj opačná hodnota výsledku.

### Príklad

Majme rovnicu  $y^2 = 8 \bmod 113$ .

Na začiatok si môžeme nastaviť niektoré parametre,

$$a = 8; p = 113; p - 1 = 112 = 7 \times 2^4; s = 7; e = 4; LS(a, p) = 1; \frac{s+1}{2} = 4.$$

Vidíme, že Legendre symbol dáva výsledok 1, to značí, že pre túto rovnicu existuje riešenie. Začneme hľadať číslo  $n$ . Zistili sme, že  $n = 3$ . Nastavíme si, ďalšie parametre.

$$x = a^{\frac{s+1}{2}} \bmod p = 8^4 \bmod 113 = 28; \quad b = a^s \bmod p = 8^7 \bmod 113 = 98;$$

$$g = n^s \bmod p = 3^7 \bmod 113 = 40; \quad r = e = 4.$$

Pokračujeme ďalej tým, že nastavíme pomocné parametre  $t = b = 98$  a  $m = 0$ . Platí, že  $m < r$  a  $t \neq 1$ , takže si spočítame  $t = t^2 \bmod p = 98^2 \bmod 113 = 112$  a zvýšime  $m = 1$ . Stále platia podmienky  $m < r$  a  $t \neq 1$ , spočítame si  $t = 112^2 \bmod 113 = 1$ , čo značí, že ide o poslednú iteráciu. Na koniec zvýšime  $m = 2$ . Ako vidíme v tejto iterácii stále platí, že  $m \neq 0$ , čiže nastavíme aktualizované parametre a pomocný parameter  $gs$ .

$$gs = g^{2^{r-m-1}} \bmod p = 40^{2^{4-2-1}} \bmod 113 = 18.$$

$$x = (x \times gs) \bmod p = 28 \times 18 \bmod 113 = 52.$$

$$g = gs^2 \bmod p = 18^2 \bmod 113 = 98.$$

$$b = (b \times g) \bmod p = 98 \times 98 \bmod 113 = 112.$$

$$r = m = 2.$$

V ďalšej iterácii hlavného (nekonečného) cyklu je  $m = 1$ , podmienka  $m \neq 0$  je splnená, aktualizujeme teda parametre,  $gs = 98; x = 11; g = 112; b = 1; r = m = 1$ . V ďalšej iterácii bude  $m = 0$ , čo znamená, že ako výsledok bude vrátená posledná hodnota  $x$  čiže  $x = 11$ , čo predstavuje prvé riešenie.

Druhé riešenie nájdeme tak, že si vypočítame  $-11 \bmod 113 = 102$ . Či sú riešenia správne si overíme dosadením  $11^2 \bmod 113 \equiv 8 \bmod 113$  a  $102^2 \bmod 113 \equiv 8 \bmod 113$ .

### 2.2.3 Right to Left algoritmus

Right to Left algoritmus slúži na zjednodušenie násobenia bodu skalárnym číslom. Využíva to, že v binárnej reprezentácii čísla každá jednotka predstavuje násobok čísla dva.



---

*Right to Left algoritmus*

---

**VSTUP:**  $k = (k_{t-1}, \dots, k_2, k_1, k_0), P \in E(Fp)$ **VÝSTUP:**  $Q = kP$ 

---

1.  $Q \leftarrow \infty$ .
2. Od  $i = 0$ , po  $t - 1$  rob
  - a. Ak  $k_i = 1$ , potom  $Q \leftarrow Q + P$ .
  - b.  $P \leftarrow 2P$ .
3. Vráť  $Q$ .

Tento algoritmus nepredstavuje najrýchlejšie riešenie pre násobenie bodu skalárom, avšak pre pochopenie je najjednoduchší a pre účely tejto práce postačuje. [20]

## 2.3 Princíp

### 2.3.1 Eliptické krivky nad množinou reálnych čísiel

Eliptická krivka vytvorená nad množinou reálnych čísiel  $\mathbb{R}$  je definovaná ako množina bodov  $P = (x, y)$ , kde  $x$  a  $y$  sú reálne čísla, ktoré spĺňajú rovnicu uvedenú nižšie. K bodom krivky patrí aj bod v nekonečne  $O$ .

$$y^2 = x^3 + ax + b \quad (2.1)$$

Koeficienty  $a, b$  určujú eliptickú krivku a musia spĺňať nasledujúcu podmienku

$$4a^3 + 27b^2 \neq 0 \quad (2.2)$$

Krivky vytvorené nad množinou reálnych čísiel nie sú vhodné na aplikáciu v kryptografii, pretože vznikajú nepresnosti pri zaokrúhľovaní, avšak sú vhodné pre ľahšie pochopenie danej problematiky. [12]

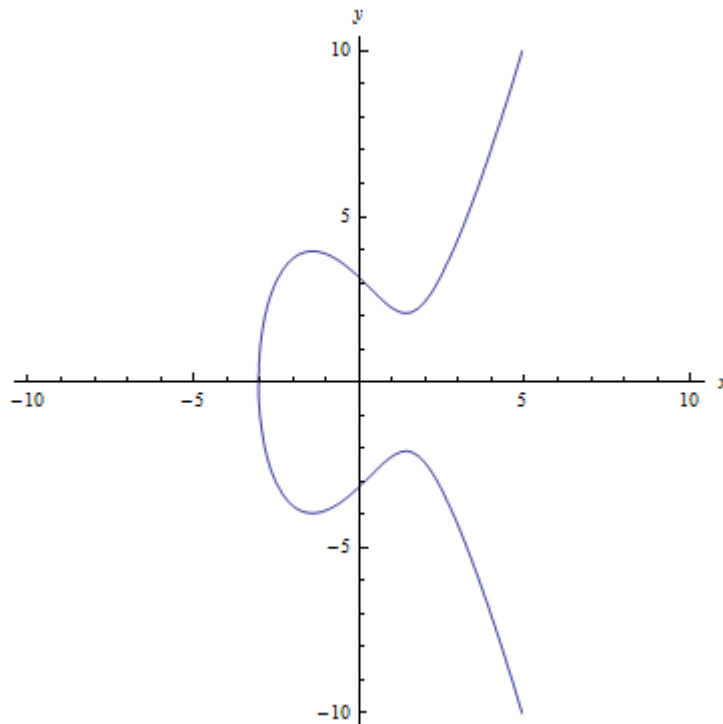
Na obrázku 2 je vidieť ukážku reálnej eliptickej krivky s koeficientmi  $a = -6, b = 10$ , ktoré spĺňajú podmienku  $4 \times (-6)^3 + 27 \times 10^2 \neq 0$ . Rovnica tejto krivky je  $y^2 = x^3 - 6x + 10$ .

### Opačný bod

Ku každému bodu na eliptickej krivke, okrem bodu v nekonečne, existuje opačný bod. Táto operácia sa nazýva aj inverzia. Inverzia bodu je definovaná ako  $Q = -P$ , pričom má zápornú súradnicu  $y$ ,  $-P = (x, -y)$ . [21]

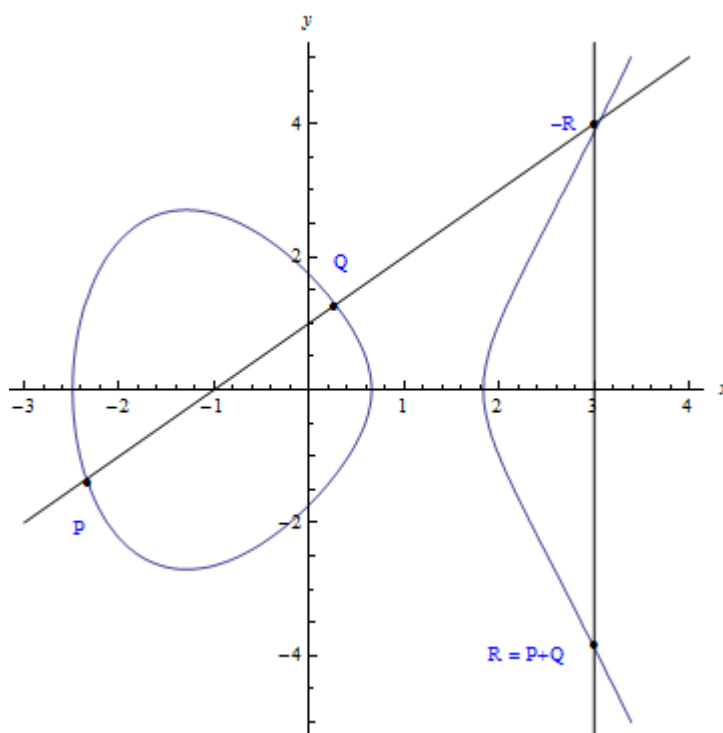
### Sčítanie bodov

Obrázok 3 ukazuje ako vyzerá sčítanie bodov reálnej eliptickej krivky. Sčítanie dosiahneme tak, že spojíme oba body priamkou. Kde sa priamka pretne krivkou vznikne bod  $-R$ , pričom výsledok sčítania je opačný bod ku  $-R$  čiže bod  $R$ . Opačný bod je symetrický podľa osy  $x$ .



Obrázok 2 : Eliptická krivka nad množinou reálnych čísiel

Uvažujme o dvoch odlišných bodoch eliptickej krivky  $P(x_P, y_P)$  a  $Q(x_Q, y_Q)$ , pričom bod  $R(x_R, y_R)$  je výsledok súčtu bodov  $P$  a  $Q$ ,  $R = P + Q$ . [21]



Obrázok 3 : Geometrický pohľad na sčítanie dvoch bodov

Smernica priamky  $s$ , ktorá spája body  $P$  a  $Q$  je určená pomocou nasledujúceho vzťahu:

$$s = \frac{y_Q - y_P}{x_Q - x_P} \quad (2.3)$$

Súradnice výsledného bodu  $R$  sú definované ako:

$$x_R = (s^2 - x_P - x_Q) \quad (2.4)$$

$$y_R = -y_P + s(x_P - x_R) \quad (2.5)$$

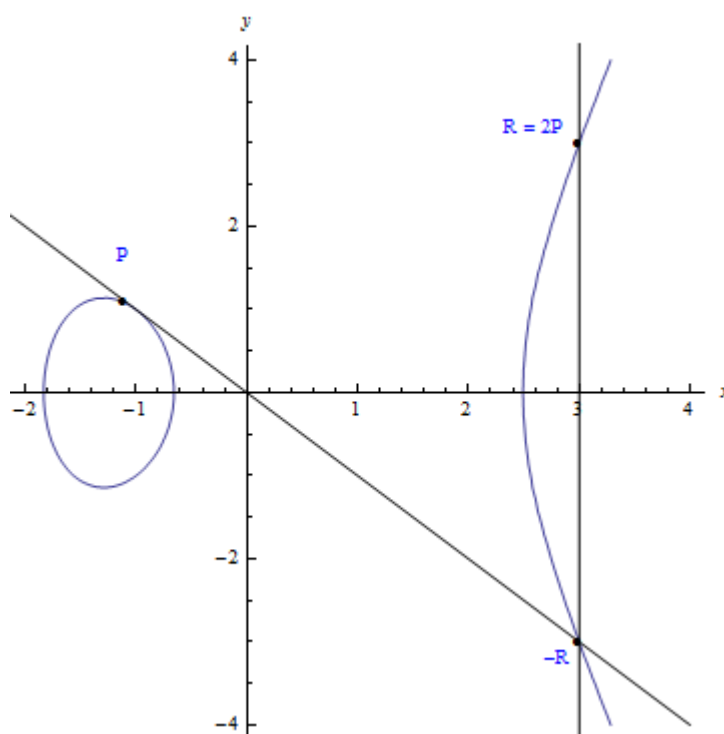
### Zdvojenie bodu

Ide o špeciálny prípad sčítania bodov pre ktorý platí, že  $P = Q$ .

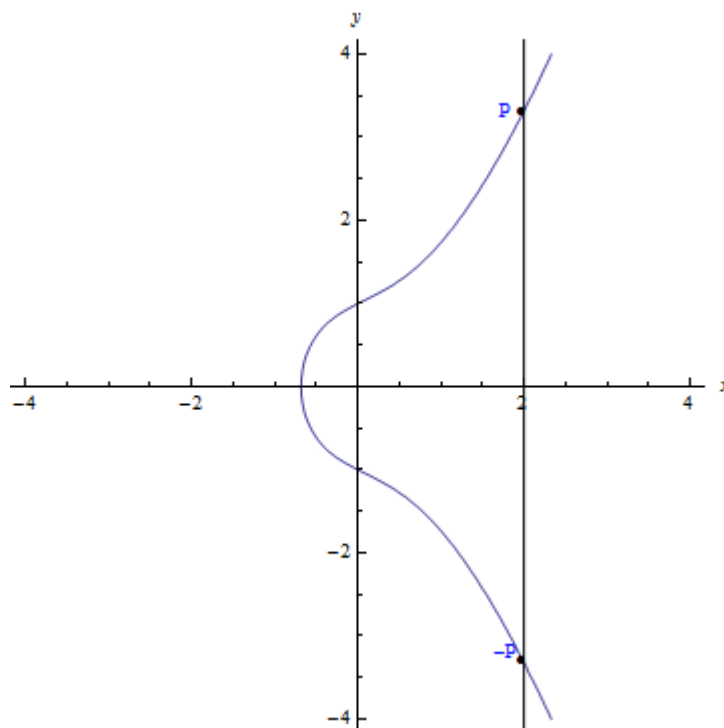
Pre takéto sčítanie je smernica  $s$  definovaná ako:

$$s = \frac{3x_P^2 + a}{2y_P} \quad (2.6)$$

Na výpočet súradníc bodu  $R$  je možné použiť vzťahy uvedené vyššie.



Obrázok 4 : Geometrické zobrazenie zdvojenia bodu P



Obrázok 5 : Sčítanie opačných bodov

### Sčítanie opačných bodov

Obrázok 5 zobrazuje ako vyzerá sčítanie dvoch opačných bodov. Z obrázku je vidieť, že výsledok takéhoto sčítania sa nepreťne s krivkou ako v prípade zobrazenom na obrázku 3.

Výsledok takejto operácie je definovaný ako  $P + (-P) = O$ . Výsledkom je nulový bod, nazývaný bod v nekonečne.

### 2.3.2 Eliptické krivky nad telesom $Fp$

Pre eliptickú krivku vytvorenú nad prvočíselným ( $p$  je prvočíslo) telesom  $Fp$  platí rovnica

$$y^2 \bmod p = (x^3 + ax + b) \bmod p \quad (2.7)$$

Koeficienty  $a, b$  musia spĺňať podmienku

$$4a^3 + 27b^2 \bmod p \neq 0 \quad (2.8)$$

Prvky telesa sú celé čísla z intervalu  $[0, p - 1]$ . Pre všetky operácie ako súčet, rozdiel, podiel a súčin platí, že musia byť definované tak, aby výsledok operácie bol znovu prvkom telesa. [12]

#### Sčítanie bodov

Uvažujme o dvoch odlišných bodoch eliptickej krivky  $P(x_P, y_P)$  a  $Q(x_Q, y_Q)$ , pričom bod  $R(x_R, y_R)$  je výsledok súčtu bodov  $P$  a  $Q$ ,  $R = P + Q$ . Smernica priamky  $s$ , ktorá spája body  $P$  a  $Q$  je určená pomocou nasledujúceho vzťahu:

$$s = \frac{y_Q - y_P}{x_Q - x_P} \bmod p \quad (2.9)$$

Súradnice výsledného bodu  $R$  sú definované ako:

$$x_R = (s^2 - x_P - x_Q) \bmod p \quad (2.10)$$

$$y_R = -y_P + s(x_P - x_R) \bmod p \quad (2.11)$$

V prípade ak  $P = Q$  tak smernica  $s$  je definovaná ako:

$$s = \frac{3x_P^2 + a}{2y_P} \bmod p \quad (2.12)$$

Koeficient  $a$  je jedným z parametrov definujúcich eliptickú krivku.[20]

#### Odčítanie bodov

Uvažujme o dvoch bodoch  $P(x_P, y_P)$  a  $Q(x_Q, y_Q)$ . Rozdiel týchto bodov je potom definovaný ako  $P - Q = P + (-Q)$ , kde  $-Q$  je opačný bod ku  $Q$ .

### Násobenie bodu skalárom

Majme bod  $P(x_P, y_P)$  a celé číslo  $d$ , výsledný bod  $Q(x_Q, y_Q)$  je definovaný ako  $Q = dP$ . Pre lepšiu efektivitu násobenia sa používajú násobky dvojky,  $Q = 2P + 2P \dots$ . Na takéto násobenie môžeme použiť algoritmus Right to Left popísaný v časti 2.2.3. Operácia  $2P$ , čiže zdvojenie bodu je vyriešená klasickým sčítaním, pretože rieši prípad ak sa oba vstupné body sčítania rovnajú.

### Príklad výpočtov pre krivku $E(F_p)$

Majme eliptickú krivku  $y^2 = x^3 + x + 1 \bmod 19$ . Diskriminant tejto krivky je  $4 \times 1^3 + 27 \times 1^2 \bmod 19 = 12$

Je teda splnená podmienka  $12 \neq 0$ , môžeme pokračovať ďalej v počítaní. Body tejto krivky môžeme spočítat' skúšaním všetkých možností (brute force), keďže toto konečné pole nie je také veľké, alebo môžeme k tomu využiť sofistikovaný polynomiálny Shanks Tonelli algoritmus. Body tejto krivky sú zapísané v nasledujúcej tabuľke.

(0, 1)	(7, 3)	(10, 17)	(15, 3)
(2, 7)	(7, 16)	(13, 8)	(15, 16)
(2, 12)	(9, 6)	(13, 11)	(16, 3)
(5, 6)	(9, 13)	(14, 2)	(16, 16)
(5, 13)	(10, 2)	(14, 17)	0

Tabuľka 1 : Body krivky  $y^2 = x^3 + x + 1 \bmod 19$

Podľa tabuľky môžeme určiť, že krivka definovaná rovnicou  $y^2 = x^3 + x + 1 \bmod 19$  má rád 20.

Z Tabuľka 1 si vyberieme napríklad bod  $P = (13, 11)$  a bod  $Q = (16, 3)$ . Súčet týchto bodov je definovaný ako  $R = P + Q$ , pričom  $P \neq Q$  a teda sa nejedná o zdvojenie bodu. Pre súčet bodov si najprv spočítame smernicu

$$s = \frac{y_Q - y_P}{x_Q - x_P} \bmod p = [(3 - 11) \bmod 19] \times [(16 - 13)^{-1} \bmod 19] \bmod 19$$

$$s = [11 \times 13] \bmod 19 = 10.$$

Keď máme smernicu môžeme si spočíta súradnice bodu  $R(x_R, y_R)$ .

$$x_R = (s^2 - x_P - x_Q) \bmod p = (10^2 - 13 - 16) \bmod 19 = 71 \bmod 19 = 14$$

$$y_R = -y_P + s(x_P - x_R) \bmod p = (-11 + 10 \times (13 - 14)) \bmod 19$$

$$= -21 \bmod 19 = 17$$

Výsledný bod  $R = (14, 17)$ .

### 2.3.3 Eliptické krivky nad telesom $F(2^m)$

Eliptické krivky zostavené nad telesom  $F(2^m)$  sú definované rovnicou

$$y^2 + xy = x^3 + ax^2 + b \quad (2.13)$$

Pre koeficient  $b$  musí platiť podmienka  $b \neq 0$ . Toto konečné teleso obsahuje celočíselné prvky s dĺžkou maximálne  $m$  bitov. O týchto číslach môže byť uvažované ako o binárnych polynómoch stupňa  $m - 1$ . V binárnom polynóme môžu byť koeficienty iba 0 a 1.

Binárny reťazec  $[a_{m-1} \dots a_1, a_0]$  môžeme vyjadriť ako polynóm

$$a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \dots + a_2x^2 + a_1x + a_0; a_i \in \{0,1\} \quad (2.14)$$

Štvorbitové číslo  $1011_2$  môže byť reprezentované ako polynóm  $x^3 + x + 1$ .

#### Sčítanie nad $F(2^m)$

Uvažujme o dvoch polynómoch,  $A = x^3 + x + 1$ ,  $B = x^3 + x^2 + 1$ . Súčet týchto dvoch polynómov je  $A + B = 2x^3 + x^2 + x + 2$ . Keďže musí byť splnená podmienka, že  $a_i \in \{0,1\}$ , koeficienty výsledného polynómu podrobíme operácii  $\bmod 2$ . Upravená podoba polynómu teda je  $A + B = x^2 + x$ .

Binárne môžeme body reprezentovať ako  $A = 1011_2$ ,  $B = 1101_2$ . Binárny súčet  $A + B$  je definovaný ako logická operácia XOR, čiže

$A + B = 1011_2 + 1101_2 = 0110_2$ . Sčítanie dvoch binárnych polynómov môže byť dosiahnuté jednoduchou operáciou XOR dvoch čísiel. [22]

#### Sčítanie bodov

Uvažujme o dvoch odlišných bodoch eliptickej krivky  $P(x_P, y_P)$  a  $Q(x_Q, y_Q)$ , pričom bod  $R(x_R, y_R)$  je výsledok súčtu bodov  $P$  a  $Q$ ,  $R = P + Q$ . Smernica priamky  $s$ , ktorá spája body  $P$  a  $Q$  je určená pomocou nasledujúceho vzťahu:

$$s = \frac{y_P - y_Q}{x_P - x_Q} \quad (2.15)$$

Súradnice výsledného bodu  $R$  sú definované ako:

$$x_R = s^2 + s + x_P + x_Q + a \quad (2.16)$$

$$y_R = s(x_P + x_Q) + x_R + y_P \quad (2.17)$$

V prípade ak  $P = Q$  tak smernica  $s$  je definovaná ako:

$$s = \frac{x_P + y_P}{x_P} \quad (2.18)$$

### Odčítanie nad $F(2^m)$

V poli  $F(2^m)$  je operácia odčítanie rovnaká ako operácia sčítanie. Majme dva polynómy  $A = x^3 + x + 1$ ,  $B = x^3 + x^2 + 1$ . Rozdiel týchto polynómov je  $A - B = -x^2 + x$ . Musí platiť rovnako ako pri sčítaní, že  $a_i \in \{0,1\}$ , takže po úprave dostávame polynóm  $A - B = x^2 + x$ . Ako možno vidieť výsledok je rovnaký ako v prípade operácie sčítania. Vďaka tomu môžeme rozdiel bodov dosiahnuť operáciou XOR medzi binárnou reprezentáciou  $A$  a  $B$ ;  $A - B = 1011_2 - 1101_2 = 0110_2$ . [22]

### Odčítanie bodov

Uvažujme o dvoch rozdielnych bodoch eliptickej krivky  $P(x_P, y_P)$  a  $Q(x_Q, y_Q)$ , pričom bod  $R(x_R, y_R)$  je výsledok rozdielu bodov  $P$  a  $Q$ ,  $R = P - Q$ . Operácia rozdiel je definovaná ako  $P - Q = P + (-Q)$ , kde  $-Q = (x_Q, x_Q + y_Q)$ . [22]



## 2.4 Aplikácia v kryptografii

### 2.4.1 Prevod správy na bod eliptickej krivky

V súčasnosti (2012) neexistuje deterministický algoritmus na prevod správy na bod eliptickej krivky. [23]

Aby sme mohli správu zakódovať ako bod eliptickej krivky je nutné nájsť korene rovnice eliptickej krivky (2.7). Túto rovnicu môžeme zjednodušené zapísať ako

$$y^2 = f(x) \bmod p \quad (2.19)$$

---

#### *Prevod správy na body eliptickej krivky podľa N.Koblitz*

---

1. Majme prvočíslo  $p$ , pre ktoré platí  $p \equiv 3 \bmod 4$ .
2. Majme eliptickú krivku  $E: y^2 = x^3 + ax + b \bmod p$ .
3. Zvoľme číslo  $K$ , také, že  $\left(\frac{1}{2}\right)^K$  je pravdepodobnosť chyby prevodu. V praxi sa volí  $K = 30$  alebo  $K = 50$ .
4. Správa je reprezentovaná ako číslo  $m \in F_p$ , pričom  $m < \frac{p-K}{K}$ .
5. Od  $i = 0$ , po  $K - 1$  vykonaj
  - a. Nastav  $x_i \leftarrow m \times K + i$
  - b. Vypočítaj rovnicu  $y_i^2 = x_i^3 + ax_i + b \bmod p$ , pomocou Shanks-Tonelli algoritmu (2.2.2).
  - c. Ak má rovnica riešenie, vráť bod  $R(x_i, y_i)$ , ktorý reprezentuje správu  $m$ . Inak nastav  $i \leftarrow i + 1$ .

Ak sa v cykle v 5. kroku nepodarí nájsť riešenie rovnice, bod reprezentujúci správu sa nenašiel. Pravdepodobnosť zlyhania prevodu je  $P, P \leq \left(\frac{1}{2}\right)^K$ .

Správu z bodu  $R(x, y)$  dekodujeme vzťahom  $m = \lfloor x/K \rfloor$ . ( $\lfloor n \rfloor$  predstavuje operáciu zaokrúhlenia nadol). Viac informácií v [24].

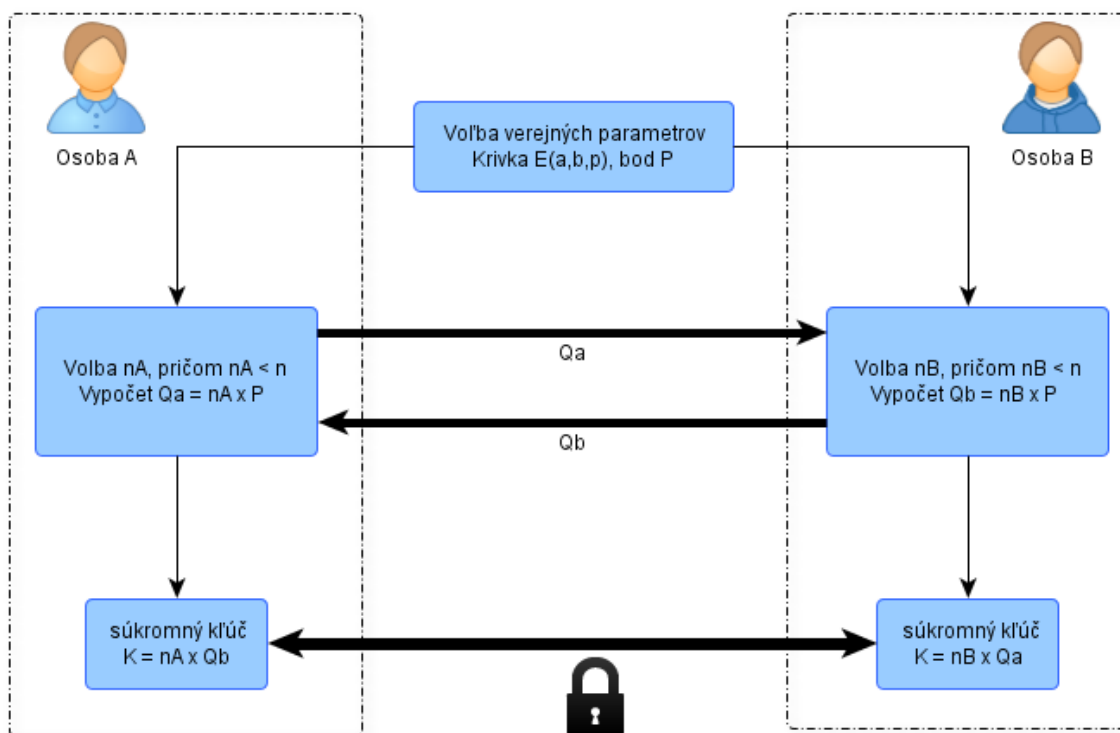
### 2.4.2 Elliptic Curve Diffie-Hellman

ECDH – Diffie-Hellman algoritmus prispôbený pre použitie s eliptickými krivkami. Algoritmus slúži na zabezpečenie výmeny privátneho kľúča pomocou verejného

komunikačného kanálu. Predpokladom pre úspešnú výmenu kľúčov je získanie verejného kľúča dôveryhodnou cestou.

### *Výmena kľúčov podľa Diffie-Hellman algoritmu, úprava pre eliptické krivky*

1. Obe strany komunikácie sa dohodnú voľbe verejných parametrov; na krivke  $E(a, b, p)$  a bode  $P$  (generátor), kde  $p$  je veľké prvočíslo.
2. Osoba A si zvolí celé číslo  $n_A$ , pričom toto číslo musí byť menšie ako rád bodu  $P$ .
3. Osoba B si zvolí celé číslo  $n_B$ , pričom toto číslo musí byť menšie ako rád bodu  $P$ .
4. Osoba A si spočíta bod  $Qa = n_A \times P$ ; Osoba B si spočíta bod  $Qb = n_B \times P$ .
5. Osoba A pošle Osobe B svoj bod  $Qa$  a Osoba B pošle osobe A svoj bod  $Qb$ .
6. Osoba A vynásobí získaný bod zvoleným číslom  $n_A$ ; Osoba B vynásobí získaný bod zvoleným číslom  $n_B$ .
7. Obe osoby si vymenili súkromný kľúč, a môžu komunikovať zabezpečeným kanálom.



Obrázok 6 : Výmena kľúčov pomocou EC Diffie-Hellman algoritmu

Obe strany vychádzajú z rovnakého bodu  $P$ , preto nakoniec vypočítajú zhodný kľúč  $K$ .

### 2.4.3 Elliptic Curve Integrated Encryption Scheme

ECIES – tento algoritmus bol vymyslený kryptografmi Bellare a Rogaway v roku 1998. Tento protokol bol štandardizovaný v ANSI X9.63, ISO/IEC 15946-3 a v IEEE P1363.

Osoba A chce poslať správu  $m$  osobe B.[25]

---

#### *Vytvorenie verejného kľúča osoby B*

---

1. Zvolí si eliptickú krivku  $E(Fp)$ .
2. Vyberie si bod  $P$ , ktorý patrí krivke  $E(Fp)$ , bod  $A$  by mal byť veľkého rádu  $n$ .
3. Zvolí si celé číslo  $b$  a vypočíta  $B = bA$ .
4. Verejným kľúčom je päťica  $(p, E, n, A, B)$ .

Algoritmus ďalej potrebuje dve hešovacie funkcie  $H_1, H_2$  a funkciu symetrickej kryptografie  $E_k$ , kde  $k$  je verejne dohodnutý kľúč.

---

#### *Poslanie správy osobou A*

---

1. Získanie verejného kľúča osoby B.
2. Zvolí si náhodné celé číslo  $k$ ,  $1 \leq k \leq n - 1$  a spočíta si  $P = kA$ ,  $Q = kB$ .
3. Výstup funkcie  $H_1(P, Q)$  rozpíše ako spojenie reťazcov  $k_1 || k_2$ , kde  $k_1$  a  $k_2$  špecifickú dĺžku.
4. Spočíta si  $C = E_{k_1}(m)$  a  $t = H_2(C, k_2)$ .
5. Osoba A pošle osobe B  $(P, C, t)$ .

---

#### *Prečítanie správy osobou B*

---

1. Vypočíta si  $Q = bP$ , pomocou jeho súkromného kľúča  $b$ .
2. Vypočíta si  $H_1(P, Q)$  a výsledok rozpíše ako spojený reťazec  $k_1 || k_2$ .
3. Vypočíta si  $H_2(C, k_2)$ , ak výsledok nie je rovný  $t$ , tak šifrovaný text je odmietnutý, inak pokračuje ďalej.
4. Vypočítanie správy  $m = D_{k_1}(C)$ , kde  $D_{k_1}$  je dešifrovacia funkcia ku  $E_{k_1}$ .

Najdôležitejšou vlastnosťou ECIES algoritmu je autentifikačný proces v treťom kroku.

#### 2.4.4 Elliptic Curve Digital Signature Algorithm

ECDSA - variant DSA algoritmu modifikovaný pre použitie nad celočíselným telesom, ktoré generuje eliptickú krivku. Na začiatok porovnanie DSA s ECDSA.[26]

Grupa	$\mathbb{Z}_p^*$	$E(\mathbb{Z}_p)$
Prvky grupy	Celé čísla $\{1, 2, \dots, p-1\}$	Body $(x, y)$ krivky $E$ a bod v nekonečne $O$
Grupová operácia	Násobenie modulo $p$	Sčítanie bodov
Zápis	Prvky: $g, h$ Násobenie: $g \cdot h$ Inverzia: $g^{-1}$ Delenie: $g/h$ Umocnenie: $g^a$	Prvky: $P, Q$ Sčítanie: $P + Q$ Inverzia: $-P$ Odčítanie: $P - Q$ Násobok: $aP$
Problém diskrétného logaritmu	$g \in \mathbb{Z}_p^*$ $h = g^a \bmod p$ , nájdenie $a$	$P \in E(\mathbb{Z}_p)$ $Q = aP$ , nájdenie $a$

Tabuľka 2 : Porovnanie zápisov v  $\mathbb{Z}_p^*$  a  $E(\mathbb{Z}_p)$ , prevzaté z [26]

Tabuľka 1 porovnáva ako sa líši označovanie ktoré sa používa v prípade celočíselných grup a označenie, ktoré sa používa pri grupách ktoré sú generované pomocou eliptickej krivky.

Zápis DSA	Zápis ECDSA
$q$	$n$
$g$	$P$
$x$	$d$
$y$	$Q$

Tabuľka 3 : Paralela medzi zápisom DSA a ECDSA, prevzaté z [26].

Prvým krokom pri generovaní kľúčového páru je voľba eliptickej krivky, ktorú je najlepšie zvoliť podľa nejakého štandardu (FIPS 182-3, SEC 2, IEEE P1363...).

---

#### Generovanie kľúčového páru pomocou ECDSA

---

1. Vyberieme eliptickú krivku definovanú nad celočíselným poľom  $\mathbb{Z}_p$ . Počet bodov  $E(\mathbb{Z}_p)$  by mal byť deliteľný veľkým prvočíslom  $n$ .
2. Vyberieme bod  $P \in E(\mathbb{Z}_p)$ , ktorého rád je  $n$ .
3. Zvolíme si náhodné unikátne celé číslo  $d$  z intervalu  $[1, n - 1]$ .
4. Spočítame bod  $Q = dP$ .
5. Verejným kľúčom je štvorica  $(E, P, n, Q)$ . Súkromný kľúč predstavuje číslo  $d$ .

---

**Podpisovanie pomocou ECDSA**

---

1. Zvolíme si náhodné unikátne celé číslo  $k$  z interval  $[1, n - 1]$ .
2. Vypočítame bod  $(x_1, y_1) = kP$  a spočítame  $r = x_1 \bmod n$ . (Súradnica  $x_1$  je v tomto prípade celé číslo)
3. Ak je  $r = 0$  tak je potrebné sa vrátiť a pokračovať od kroku 1. (Ide o bezpečnostnú podmienku: v tomto prípade by podpisová rovnica  $s = k^{-1}\{h(m) + dr\} \bmod n$  nezahrňovala súkromný kľúč.
4. Spočítame  $z = k^{-1} \bmod n$ .
5. Spočítame  $s = z \times \{h(m) + dr\} \bmod n$ , kde  $h$  je Secure Hash Algorithm (SHA-1).
6. Ak je  $s = 0$  vrátime sa ku kroku 1, pretože neexistuje modulárna inverzia  $s^{-1} \bmod n$ , ktorá je nutná pri overovaní podpisu.
7. Podpis správy  $m$  je reprezentovaný ako dvojica celých čísiel  $(r, s)$ .

---

**Overovanie podpisu vytvoreného pomocou ECDSA**

---

1. Máme verejný kľúč  $(E, P, n, Q)$ , a podpis správy  $(r, s)$ .
2. Overíme si, či čísla  $r$  a  $s$  sú z intervalu  $[1, n - 1]$ .
3. Vypočítame  $w = s^{-1} \bmod n$  a vytvoríme heš správy  $h(m)$ .
4. Spočítame  $u_1 = h(m)w \bmod n$  a  $u_2 = rw \bmod n$ .
5. Spočítame bod  $(x_0, y_0) = u_1P + u_2Q$  a  $v = x_0 \bmod n$ .
6. Podpis akceptujeme iba ak platí  $v = r$ .

## 2.5 Kryptoanalýza

Bezpečnosť kryptografických systémov založených na eliptických krivkách spočíva v nemožnosti riešiť problém diskretného logaritmu v rozumnom čase. To, že tento problém nemožno riešiť rýchlo neznamená, že neexistujú algoritmy pre riešenie úlohy diskretného logaritmu, ktoré sú efektívnejšie ako metóda brute force útoku.

### 2.5.1 Baby-Step Giant-Step

Jeden z najjednoduchších algoritmov pre riešenie diskretného logaritmu. Využíva vopred vypočítane mocniny, ktoré sú uložené vo vyhľadávacej tabuľke, najčastejšie v hash

tabuľke ( zložitosť vyhľadávania je  $O(1)$ ). Tento algoritmus uviedol matematik Daniel Shanks.

Algoritmus rieši rovnicu  $a^x = b \bmod p$ , v cyklickom konečnom poli  $\mathbb{Z}_p$ , kde  $x$  je neznáma a  $n$  je rád prvku. [1][27]

---

### ***Baby-Step Giant-Step algoritmus***

---

**VSTUP:**  $a$  (generátor cyklickej grupy  $G$  rádu  $n$ ), prvok  $b \in G$ .

**VÝSTUP:** diskretný logaritmus  $x = \log_a b$ .

---

1. Vypočítame si  $m \leftarrow \sqrt{n}$ , zaokrúhlené nahor.
2. Vytvoríme hash tabuľku  $(j, a^j \bmod p)$  pre exponenty  $\{0, \dots, m-1\}$ .
3. Vypočítame  $h \leftarrow (a^{-1} \bmod p)^m \bmod p$  a nastavíme  $t \leftarrow b$ .
4. Pre  $i$  od 0 po  $m-1$ , urob
  - a. Skontroluj či sa  $t$  nenachádza vo vyhľadávacej tabuľke.
  - b. Ak  $t = a^j$ , potom vráť  $(x = i \times m + j)$ .
  - c. Nastav  $t \leftarrow t \times h \bmod p$ .

Algoritmus má zložitosť  $O(\sqrt{n})$  a potrebuje dostatok pamäť pre uloženie vypočítaných mocnín  $\sqrt{n}$  prvkov grupy. Pri veľkých konečných poliach ako sa používajú v kryptografií sa môže stať, že nám jednoducho nebude stačiť pamäť.

### **2.5.2 Pollard $\rho$ metóda**

Pollard  $\rho$  metóda umožňuje riešiť problém diskretného logaritmu podobne ako Baby-Step Giant-Step, avšak s takmer zanedbateľnou pamäťovou náročnosťou (nemusí si držať vypočítané hodnoty vo vyhľadávacej tabuľke).

Predpokladajme, že grupa s ktorou pracujeme je množina celých čísel z rozsahu 0 až  $p-1$ , spolu s operáciami sčítanie a násobenie. Pre použitie Pollard  $\rho$  algoritmu rozdelíme túto grupu na tri časti približne rovnakej veľkosti.

Tieto časti budeme označovať ako  $P_0, P_1$  a  $P_3$ . Tieto podmnožiny môžeme vytvoriť jednoducho tak, že rozdelíme čísla grupy podľa toho či sú menšie ako tretiny čísla  $p$ , alebo podľa pravidla

$x \in P_0$ , ak  $x \equiv 1 \pmod{3}$ ,  $x \in P_1$ , ak  $x \equiv 0 \pmod{3}$  a  $x \in P_2$ , ak  $x \equiv 2 \pmod{3}$ . [20]

Algoritmus spoľieha na tri funkcie. Tieto funkcie sú definované podľa časti (podmnožiny), z ktorej je ich vstupný argument.

$$\text{Ak } x \in G_0: \begin{cases} f(x) = (bx) \pmod{p} \\ g(x, n) = (n + 1) \pmod{(p - 1)} \\ h(x, n) = n \pmod{(p - 1)} \end{cases}$$

$$\text{Ak } x \in G_1: \begin{cases} f(x) = (x^2) \pmod{p} \\ g(x, n) = (2n) \pmod{(p - 1)} \\ h(x, n) = (2n) \pmod{(p - 1)} \end{cases}$$

$$\text{Ak } x \in G_2: \begin{cases} f(x) = (ax) \pmod{p} \\ g(x, n) = n \pmod{(p - 1)} \\ h(x, n) = (n + 1) \pmod{(p - 1)} \end{cases}$$

---

### **Pollard $\rho$ algoritmus**

---

**VSTUP:**  $a$  (generátor cyklickej grupy  $G$  rádu  $n$ ), prvok  $b \in G$ .

**VÝSTUP:** diskretný logaritmus  $x = \log_a b$ .

---

1. Nastavíme  $a_0 \leftarrow 0, b_0 \leftarrow 0$ , ktoré predstavujú pomocné štartovacie hodnoty
2. Nastavíme  $x_0 \leftarrow 1$ , náš štartovací bod v grupe  $G$ .
3. Nech  $i \leftarrow 0$ .
4. Opakujeme nasledujúce kroky až pokiaľ  $x_i = x_{2i}$ , v tom prípade sa naše „cesty“ spojili.
  - a. Nech  $i \leftarrow i + 1$ .
  - b. Spočítame ďalšie  $x: x_i = f(x_{i-1})$ .
  - c. Spočítame ďalšie  $a: a_i = g(x_{i-1}, a_{i-1})$ .
  - d. Spočítame ďalšie  $b: b_i = h(x_{i-1}, b_{i-1})$ .
  - e. Spočítame ďalšie  $x: x_{2i} = f(f(x_{2i-2}))$ .
  - f. Spočítame ďalšie  $a: a_{2i} = g(f(x_{2i-2}), g(x_{2i-2}, a_{2i-2}))$ .
  - g. Spočítame ďalšie  $b: b_{2i} = h(f(x_{2i-2}), h(x_{2i-2}, b_{2i-2}))$ .
5. Ak  $b_i = b_{2i}$ , tak algoritmus zlyhal.

6. Nastavíme  $m \leftarrow (a_i - a_{2i}) \bmod (p - 1)$ .
7. Nastavíme  $n \leftarrow (b_{2i} - b_i) \bmod (p - 1)$ .
8. Vyriešime rovnicu  $mx \equiv n \bmod (p - 1)$ , kde  $x$  je neznáma. Rovnica má zopár možných riešení, avšak ku nájdeniu toho správneho ich budeme musieť skúsiť všetky.

Algoritmická zložitosť je rovnaká ako v predchádzajúcom prípade  $O(\sqrt{p})$ . Tento algoritmus má aj modifikáciu pre využitie paralelných výpočtov, viac informácií je možné nájsť v [1][27].

## 2.6 Štandardy

Pre použitie eliptických kriviek pre reálnu aplikáciu je doporučené voliť koeficienty podľa schválených štandardov. Tieto štandardizované krivky sú dlhodobo testované a spĺňajú dané bezpečnostné nároky. Hlavný element na ktorom spočíva bezpečnosť kryptografického systému založenom na eliptických krivkách je rád bodu, s ktorým vykonáva jednotlivé operácie.

Ak je rád bodu nízky, tak algoritmy spomenuté v tejto podkapitole majú menej práce a „rýchlejšie“ môžu nájsť riešenie diskrétného logaritmu. Preto sa v jednotlivých štandardoch uvádza aj bod generátor, ktorý má veľmi vysoký rád. Ďalším dôvodom je aj to, že výpočet rádu tohto bodu je časovo náročná úloha, ktorá by spomalila celý proces, a eliptické krivky by stratili svoju hlavnú výhodu.

Ak už je nutné generovať náhodné krivky, jednotlivé štandardy popisujú spôsoby ako vybrať tú vhodnú z jednotlivých náhodne vygenerovaných.

Implementácia podľa štandardu zabezpečí kompatibilitu medzi jednotlivými stranami (aplikáciami), ktoré sa môžu zúčastňovať na kryptografickom procese.

### FIPS

Štandardy označené ako FIPS vydáva americká federálna vláda. Verejnou kryptografiou a eliptickými krivkami sa zaoberá štandard FIPS 182-3.



**SECG**

Komerčné zoskupenie popredných IT firiem. ECC sa zaoberajú štandardy SEC1 a SEC2. SEC1 popisuje prácu s eliptickými krivkami a jednotlivými protokolmi. SEC2 predstavuje súhrn doporučených parametrov eliptických kriviek.

**ISO**

Medzinárodná štandardizačná organizácia. Použitie eliptických kriviek je definované v štandardoch ISO 15946-2 (ECDSA) a ISO 15946-3 (ECDH).

**IEEE**

Medzinárodná štandardizačná organizácia, ktorá vydáva štandard napríklad pre Ethernet (IEEE 802.3). Verejnou kryptografiou a problematikou s ňou spojenou sa zaoberajú štandardy skupiny IEEE P1363.

**ANSI**

Americký národný štandardizačný inštitút. Použitie eliptických kriviek v kryptografii sú definované v štandardoch ANSI X9.62(ECDSA), ANSI X9.63.

### 3 POROVNANIE SO SÚČASNE POUŽÍVANÝMI SYSTÉMAMI

Najrozšírenejším súčasne používaným verejným kryptografickým systémom je RSA. Je to spôsobené tým, že je dobre známy a preskúmaný. Avšak počítačový svet stále napreduje a pre zabezpečenie pomocou RSA sú v súčasnosti už doporučené kľúče o veľkosti 2048 bitov. Zvyšovanie dĺžky používaných kľúčov spôsobuje zvýšenie potrebného výpočtového výkonu. A práve výpočtová náročnosť predstavuje problém pre napríklad mobilné telefóny, ktoré nedosahujú výkonu stolných počítačov, notebookov. Preto sa dostávajú do popredia moderné kryptografické systémy založené na problematike eliptických kriviek.

Prečo sú lepšie? Je to spôsobené viacerými faktormi. Použitie eliptických kriviek nie je tak preskúmaná oblasť, čo predstavuje aj výhodu aj nevýhodu. Výhodou je, že ešte neboli objavené potenciálne zraniteľnosti ako v prípade systémov RSA/DSA, no naopak nevýhodou je fakt, že práve pre nedostatok poznatkov sa priemysel nehrnie do ich nasadzovania. Druhým faktorom je, že pre riešenie diskretného logaritmu eliptických kriviek neexistuje subexponenciálny algoritmus.

To umožňuje použiť kratšie kľúče pri použití kryptografie eliptických kriviek. Vďaka kratším kľúčom je potrebný výpočtový výkon oveľa nižší ako v prípade RSA/DSA. Od toho sa odvíjajú ďalšie veci ako menšie nároky na dátový tok, menšia spotreba elektrickej energie, ... [28]

	<b>RSA</b>	<b>DSA</b>	<b>ECC</b>
<b>Matematický problém</b>	Faktorizácia celých čísiel	Diskretný logaritmus	Diskretný logaritmus eliptických kriviek
<b>Riešenie matematického problému</b>	Číselné sieťové pole (eng. GNFS)	Číselné sieťové pole (eng. GNFS)	Pollard rho algoritmus pre diskretný logaritmus
<b>Časová náročnosť riešenia matematického problému</b>	Subexponenciálna	Subexponenciálna	Exponenciálna

Tabuľka 4 : Porovnanie matematických problémov jednotlivých systémov

Eliptické krivky sú postavené buď nad celočíselnom telese  $Fp$ , alebo binárnym telese  $F^{2m}$ . Krivky postavené na binárnym telese sú vhodnejšie pre hardvérovú realizáciu, prípadne pre použitie v mikropočítačových systémoch.

Najdôležitejšou vlastnosťou kryptografických systémov je bezpodmienečne ich bezpečnosť. Tabuľka 5 porovnáva bezpečnosť jednotlivých kryptografických systémov pri daných parametroch. Tabuľka je prevzatá z [28].

Bezpečnosť (bity)	Minimálna dĺžka kľúčov v bitoch		
	RSA	DSA	ECC
80	1024	1024	160
112	2048	2048	224
128	3072	3072	256
192	7680	7680	384
256	15360	15360	521

Tabuľka 5 : Porovnanie bezpečnosti jednotlivých systémov

Ďalšie bezpečnostné riziko predstavuje implementácia jednotlivých systémov. Aby sa zamedzilo prípadným chýbam v implementácii, kryptografický systém by mal implementovať skutočný odborník v kryptografii. Ďalšou možnosťou je použiť hotové riešenia, ktoré sú dobre otestované a overené časom. Takýmto príkladom môže byť knižnica Bouncy Castle, ktorá implementuje veľa kryptografických systémov, vrátane tých, ktoré sú založené na eliptických krivkách.

Životnosť parametrov	RSA	DSA	ECC
Do roku 2030 (minimálne 112 bitové zabezpečenie)	2048	2048	224
Po roku 2030 (minimálne 128 bitové zabezpečenie)	3072	3072	256

Tabuľka 6 : Doporučené dĺžky kľúčov podľa NIST

Tabuľka 6 predstavuje prehľad ako dlho budú jednotlivé dĺžky kľúčov dostatočne silné aby bola dodržaná daná bezpečnosť. Tabuľka pochádza z [29] , avšak tento draft pochádza z roku 2007 a novší nie je ešte kompletný preto považujte tento prehľad ako ilustráciu budúceho vývoja.

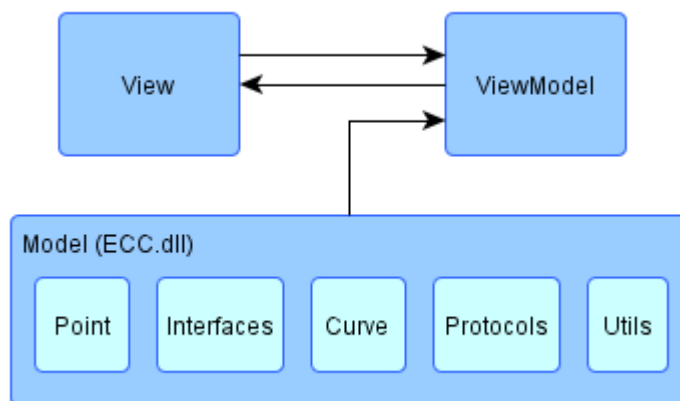
## **II. PRAKTICKÁ ČASŤ**

## 4 SOFTVÉROVÉ RIEŠENIE

Výstupom tejto bakalárskej práce je aplikácia, ktorá umožňuje vyskúšať si aritmetiku eliptických kriviek a v zjednodušenej miere demonštruje jednotlivé možnosti aplikácie eliptických kriviek v kryptografii. Implementácia tohto softvérového riešenia bola prevedená v jazyku C# a na platforme .NET od spoločnosti Microsoft. Ako vývojové prostredie bolo použité Visual Studio 2010 Premium a pre návrh rozhrania grafickej časti aplikácie bol použitý nástroj Microsoft Expression Blend 4. K obom týmto komerčným nástrojom som mal prístup zadarmo, vďaka programu MSDN AA v ktorom je zapojená naša fakulta.

### 4.1 Popis riešenia

Aplikáciu som sa snažil navrhnuť tak aby bolo možné jednotlivé komponenty znovu použiť. Celá aritmetika a jednotlivé protokoly sú implementované ako binárna knižnica ECC.dll. Výsledky jednotlivých operácií sú prenášané do grafického rozhrania, ktoré je postavené na technológii WPF. Pri implementácii grafickej časti som sa snažil dodržiavať návrhový vzor MVVM (Model – View – ViewModel). Obrázok 7 zobrazuje model tejto aplikácie, ktorý je reprezentovaný ako knižnica ECC.dll.



Obrázok 7 : Zjednodušená schéma aplikácie

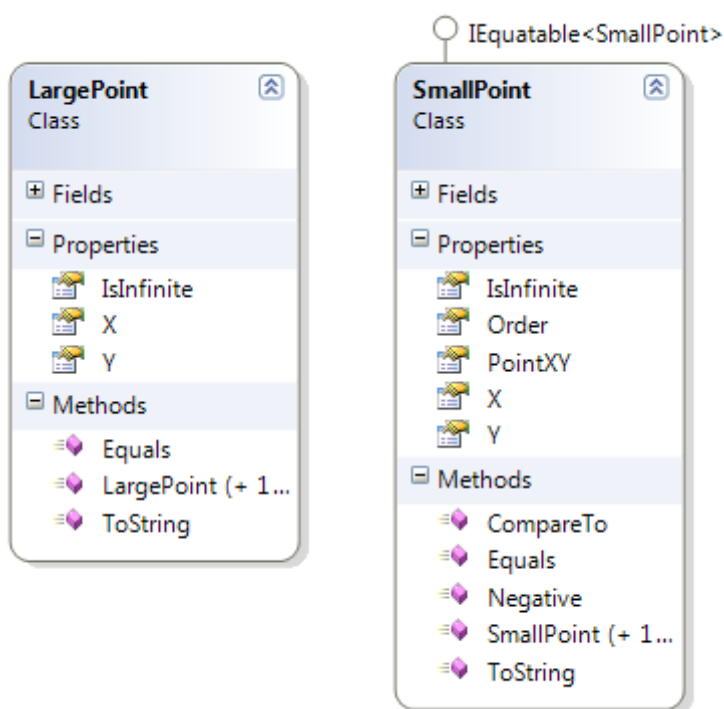
#### 4.1.1 Knižnica ECC

Knižnica ECC.dll predstavuje samotné jadro aplikácie. ECC.dll je rozdelená do šiestich namespace vrátane hlavného ECC.

## Namespace Point

Tento namespace obsahuje implementáciu dátových typov `SmallPoint` a `LargePoint`. Separátna implementácia bola zvolená pretože, pre veľké čísla, ktoré sa používajú pri štandardizovaných krivkách bolo nutné použiť dátový typ `BigInteger`, ktorý je súčasťou namespace `System.Numerics`. Popritom niektoré veci sa pri veľkých krivkách vzhľadom na použitie typu `BigInteger` museli riešiť inak. Určite by to šlo zjednotiť pomocou generickej triedy, avšak rozhodol som sa pre separátne riešenie.

Obe triedy majú dva konštruktory. Jeden s parametrami, súradnice `x` a `y` a druhý bez parametrov, ktorý vytvorí bod v nekonečne nasetovaním property `IsInfinite`. Preto aby bolo možné použiť metódu `Contains` na generický `List<SmallPoint>`, `SmallPoint` implementuje interface `IEquatable`. Samozrejmosťou je override metód `Equals` a `ToString`.



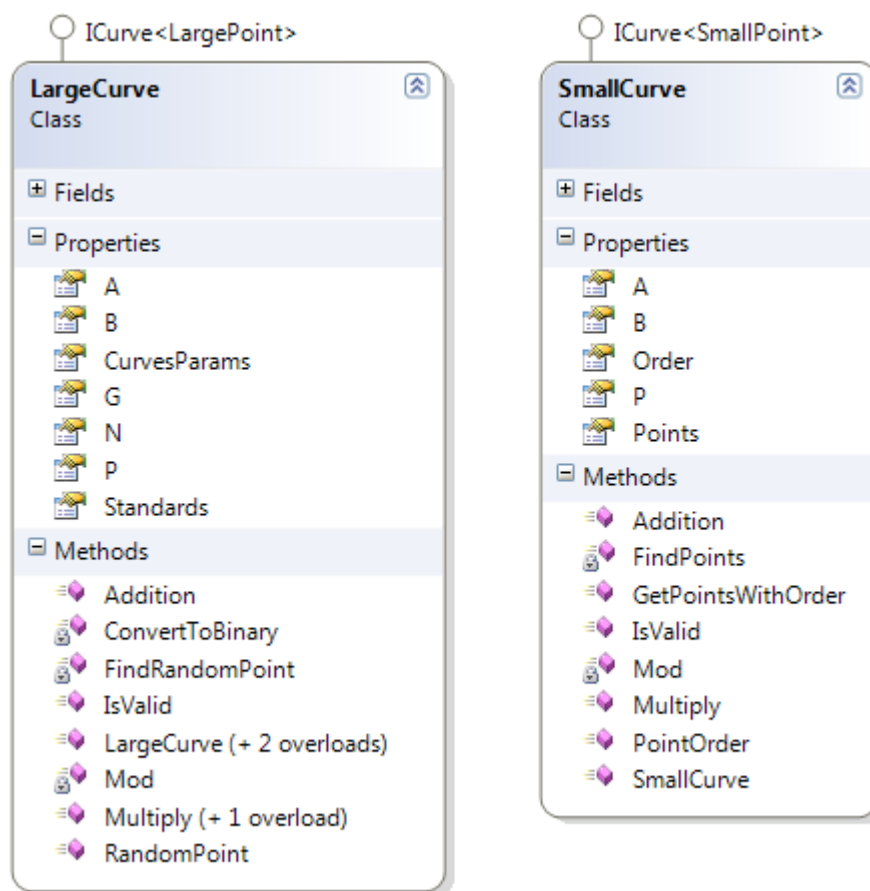
Obrázok 8 : Diagram tried `LargePoint` a `SmallPoint`

## Namespace Interfaces

Tento namespace obsahuje interface `ICurve<T>`, ktorý definuje základné operácie krivky. `ICurve<T>` slúži na zjednotenie implementácií kriviek `SmallCurve` a `LargeCurve`.

## Namespace Curve

V tomto namespace sú definované triedy (dátové typy) `SmallCurve` a `LargeCurve`. Ako už bolo spomenuté obe implementujú interface `ICurve`.



Obrázok 9 : Diagram tried LargeCurve a SmallCurve

Trieda `LargeCurve` má tri konštruktory. Jeden je bez parametru a slúži na získanie implementovaných štandardizovaných parametrov, ktoré si drží property `CurvesParams`. Druhý preberá ako parameter názov štandardu. Implementovanými štandardným krivkami sú krivky z SEC2. [30] Tretí konštruktor umožňuje nastaviť priamo parametre krivky,  $a, b, p$ .

Štandardy sú reprezentované ako štruktúra `Dictionary<string, StandardParams>`, kde `StandardParams` je pomocná trieda, ktorá si drží všetky potrebné parametre jednotlivých kriviek.

`SmallCurve` je podobná ako `LargeCurve`, avšak vzhľadom na použité rôznych číselných dátových typov sa jednotlivé implementácie značne líšia. Metoda `PointOrder` umožňuje výpočet rádu bodu. Tento výpočet je riešený brute force metódou ( násobí body dovtedy, pokiaľ nenarazí na bod v nekonečne), preto pri väčších konečných poliach môže výpočet dlho trvať.

## Namespace Protocols

Namespace Protocols implementujú prevod správy na body eliptickej krivky podľa Koblitz, mechanizmus výmeny kľúčov ECDH a podpisový algoritmus ECDSA.

Prevod správy pomocou Koblitzovho algoritmu je riešený pomocou veľkých kriviek zo štandardu SEC2. [30]

Snažil som sa zakódovať správu aj pomocou malých kriviek (parameter  $p < 229$ ), ale to sa mi veľmi nedarilo. Pretože takéto krivky nemajú toľko bodov, tak znaky boli prevádzané po päťiciach. Problém vznikol pri dekódovaní, kde výsledkom bolo číslo ktoré predstavovalo súčet ASCII hodnôt znakov tejto päťice, z ktorého nie je možné určiť o aké znaky šlo. Aj preto som od tohto riešenia upustil.

Výsledné riešenie funguje tak, že každý znak vstupného textu je reprezentovaný ako bod krivky. V prípade ak už bol znak predtým zakódovaný, tak by novým kódovaním iný bod nevznikol. V takomto prípade sa kódovanie neprevádza znovu. Mapovanie znakov je riešené pomocou hash tabuľky, ktorá sa v C# deklaruje ako štruktúra *Dictionary*.

Výmenný mechanizmus ECDH bolo asi najjednoduchšie implementovať. Trieda ECDH podporuje aj veľké aj malé krivky.

Najproblematickejšie implementovanie predstavoval algoritmus ECDSA. Knižnica ECC podporuje ECDSA pomocou štandardizovaných kriviek SEC2, ale aj malé krivky. Implementácia však nie je úplne deterministická. Pri veľkých krivkách overenie podpisu zlyhá vždy. Nefunkčnosť overovania je pravdepodobne spôsobená generovaním náhodných čísel, pretože sa používa iná metóda generovania ako v prípade malých kriviek. Pre malé krivky funguje aj podpisovanie aj overovanie. Avšak pre správnu funkčnosť je nutné vybrať bod, ktorý má prvočíselný rád. Pre ostatné body by ECDSA nefungovalo, preto pri výbere takého bodu aplikácia zobrazí chybovú správu.

## Namespace Utils

Tento namespace obsahuje pomocné triedy a nástroje. Najdôležitejšou časťou je trieda NTMath, ktorá obsluhuje všetky nutné matematické operácie z teórie čísel.

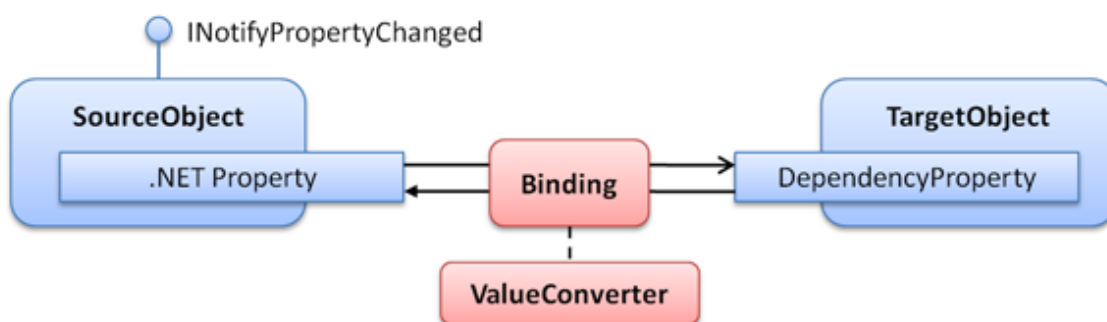
### 4.1.2 Windows Presentation Foundation

WPF je moderný subsystém pre tvorbu grafického rozhrania aplikácií postavených na technológii .NET. Pre lepšiu odozvu aplikácií využíva hardvérovú akceleráciu pomocou



technológie DirectX. WPF dáva možnosť oddeliť definíciu rozhrania aplikácia od aplikačnej logiky. Rozhranie aplikácie sa deklaruje pomocou značkovacieho jazyka XAML, ktorý vychádza z XML. Takéto oddelenie umožňuje designerom vytvárať rozhranie bez toho aby zasahovali do práce vývojárom. WPF je priamo prispôsobený pre použitie s návrhovým vzorom MVVM. Umožňuje to mechanizmus Data Binding.

Tento mechanizmus zabezpečuje jednoduchý spôsob aktualizácie grafického rozhrania vzhľadom na zmeny na aplikačnej logike. Notifikácia zmien medzi zdrojovým a cieľovým objektom je riešená pomocou implementácie interface *INotifyPropertyChanged*. Obrázok 10 schematicky znázorňuje Data Binding. Samotný Binding môže byť jednosmerný (OneWay), alebo obojsmerný (TwoWay). Štandardne sa používa obojsmerný Data Binding. ValueConverter slúži na prípadne konvertovanie objektov, ktoré prichádzajú z aplikačnej logiky, aby sa dali „rozumne“ zobrazit' v grafickom rozhraní.



Obrázok 10 : Data Binding diagram prevzatý z [31]

V značkovacom jazyku XAML je Binding riešený pomocou deklarácia {Binding}. Nasledujúca ukážka zobrazuje Binding premennej typu string pre parameter text elementu TextBox.

```
<StackPanel Orientation="Horizontal">
    <Label Content="Solution" />
    <TextBlock Text="{Binding ShanksResult}" />
</StackPanel>
```

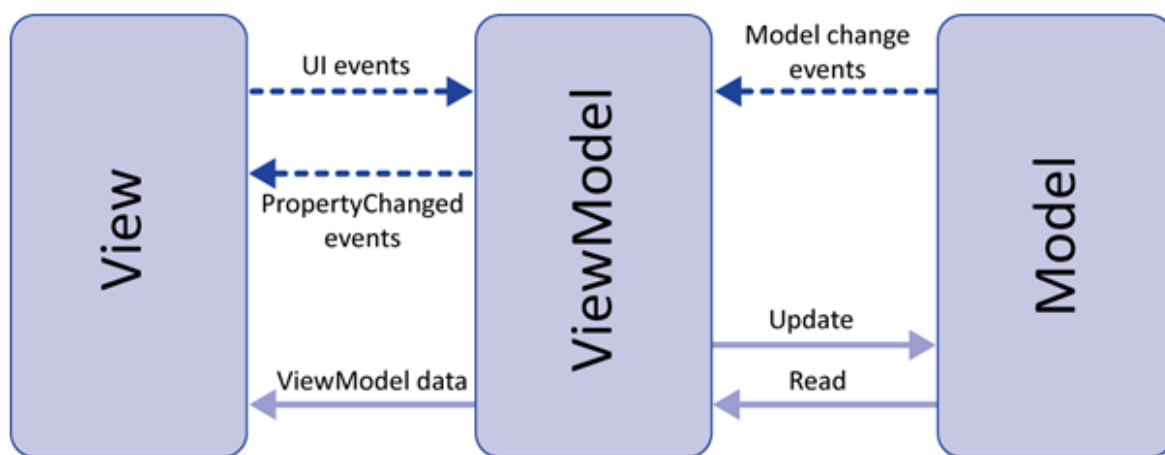
#### 4.1.3 Návrhový vzor MVVM

Ako bolo načrtnuté v úvode kapitoly, moderné aplikácie dodržiavajú nejakú schému, vzor. Dodržiavanie určité vzoru, umožňuje lepšie možnosti pre tímovú spoluprácu a v budúcnosti jednoduchšiu správu zdrojového kódu. Vzor MVVM je moderný spôsob tvorby grafických aplikácií.

Model predstavuje triedy, ktoré deklarujú jednotlivé dátové kontajnery, napríklad pri databázovej aplikácii, triedy modelu reprezentujú jednotlivé tabuľky databázy. View sa stará o grafické rozhranie a nemal by obsahovať aplikačnú logiku.

ViewModel slúži na riešenie „prenosu“ aktualizácií z View do Modelu a naopak. Stará sa aj o konvertovanie modelu do podoby vhodnej pre View. Okrem jednoduchšej správy kódu, je takéto rozloženie vhodné pre automatické testovanie pomocou Unit testov.

V aplikácii, ktorú bola vytvorená ako súčasť tejto práce, si ViewModel drží všetky výpočty a štruktúry, ktoré mu „dodáva“ ECC.dll knižnica. ECC.dll reprezentuje Model aplikácie. Z ViewModelu sú potom jednotlivé property bindované do View. Každá záložka aplikácie je riešená ako jeden ViewModel a jeden View. Unit testy boli použité na testovanie funkčnosti jednotlivých metód z knižnice ECC.dll



Obrázok 11 : MVVM diagram, prevzatý z [32]

Vzor MVVM sa dá dosiahnuť aj za použitia „čistého“ WPF a .NET, avšak pre komplexnejšie aplikácie sú dostupné rôzne MVVM frameworky. Medzi známejšia patria MVVM Light, Apex ...

Každá záložka aplikácie je riešená ako jeden ViewModel a jeden View.

#### 4.1.4 Grafické rozhranie

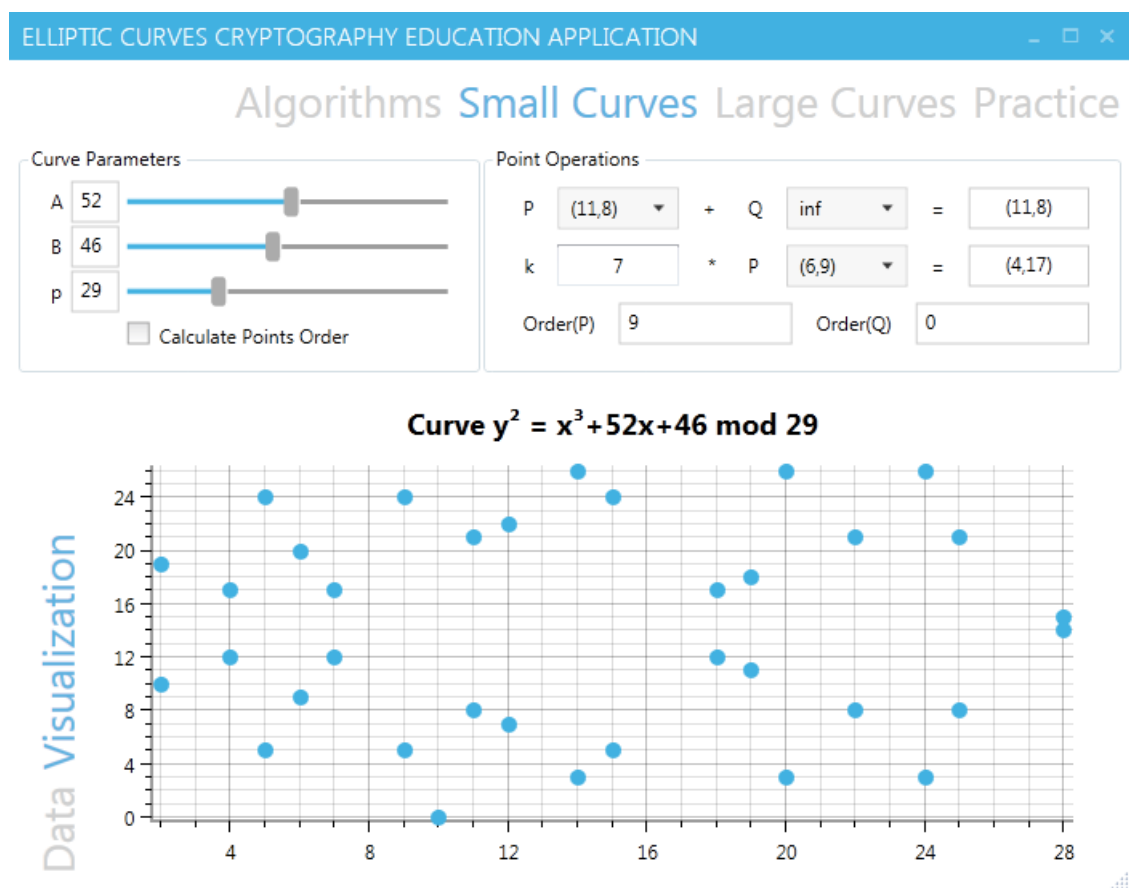
Grafické rozhranie bolo deklarované pomocou jazyka XAML. Tento jazyk je podobne štyľovateľný ako HTML s pomocou CSS. Globálne nastavenie vlastností jednotlivých elementov sa pri štandardne vytvorenom projekte WPF deklaruje v App.xaml. MainWindow.xaml v takomto prípade predstavuje hlavné okno aplikácie.

Pre vzhľad aplikácie som použil knižnicu MahApps.Metro, ktorá mení vzhľad aplikácie na štýl rozhrania Metro, ktoré bude preferovaným štýlom v blížiacom sa Windows 8.

Keďže štandardný WPF Toolkit nemá Spinner element, využil som knižnicu WPFToolkit.Extended, ktorá definuje dodatočné chýbajúce elementy, vrátane DecimalUpDown, ktorý má rovnakú funkciu ako Spinner.

Pre vykresľovanie grafu malých kriviek som použil knižnicu OxyPlot. Všetky spomenuté a použité knižnice sú uvoľnené pod nejakou formou Open Source licencie.

Výsledkom spojenia všetkých spomenutých elementov som dosiahol prehľadnú, jednoducho vyzerajúcu aplikáciu.



Obrázok 12 : Grafické rozhranie demonštračnej aplikácie

## 4.2 Uživatelská příručka

Grafické rozhranie (Obrázok 12) bolo rozdelené do viacerých tabov (záložiek), medzi ktorými je možno sa jednoducho prepínať.

### Záložka Algorithms

ELLIPTIC CURVES CRYPTOGRAPHY EDUCATION APPLICATION

Algorithms Small Curves Large Curves Practice

Calc Shanks Tonelli

$y^2 = 5 \pmod{40961}$  **SOLVE**

Step	m	x	b	g	r
1	11	8145	37802	20237	11
2	9	35907	21227	8603	9
3	7	33206	26432	19734	7
4	6	34087	21153	14529	6
5	5	31533	8555	19808	5
6	4	32336	9282	32406	4
7	3	16114	26420	31679	3
8	2	19424	1	14541	2

Solution 19424 21537

Obrázok 13 : Záložka Algorithms

Záložka *Algorithms* obsahuje ešte dve podzáložky, *Shanks Tonelli* a *Calc*. *Shanks Tonelli* umožňuje výpočet rovnice  $y^2 \equiv x \pmod{p}$  pomocou Shanks Tonelli algoritmu, aj s príslušným výpisom jednotlivých krokov. Pod výpisom jednotlivých krokov je výsledné riešenie, ak existuje.

Podzáložka *Calc* umožňuje výpočet najväčšieho spoločného deliteľa a výpočet multiplikatívnej modulárnej inverzie  $k^{-1} \pmod{p}$ .

### Záložka Small Curves

ELLIPTIC CURVES CRYPTOGRAPHY EDUCATION APPLICATION

Algorithms Small Curves Large Curves Practice

Curve Parameters

A 92 B 40 p 41

☐ Calculate Points Order

Point Operations

P (4,12) + Q (5,16) = (7,17)

k 5 \* P (7,17) = (11,28)

Order(P) 8 Order(Q) 24

Obrázok 14 : Ovládacie prvky záložky Small Curves

Celý obsah záložky *Small Curves* je vidieť na obrázku 12. Nastavovanie parametrov eliptických kriviek na nastavované pomocou sliderov. Po zmene niektorého zo sliderov sa pregeneruje zoznam bodov v comboboxoch. Ak je zaškrnuté *Calculate Points Order* v podzáložke *Data* k jednotlivými bodom priradí ich rád.

Pre takéto riešenie som sa rozhodol, pretože prepočet rádu bodov spomaľoval aplikáciu. Pre danú krivku je možné vykonať operáciu sčítanie medzi vybranými bodmi a násobenie bodu celým číslom. Pri každom výbere bodu sa prepočíta rád bodu ( $\text{Order}(P)$ ,  $\text{Order}(Q)$ ). Nastavenie čísla pre násobenie bodu je možný aj stlačením dolnej alebo hornej šípky.

Pravým kliknutím na graf bodov sa dá graf uložiť ako obrázok formátu PNG.

## Záložka Large Curves

[illegible]

Obrázok 15 : Záložka Large Curves

Záložka *Large Curves* zobrazuje parametre kriviek definovaných v štandarde SEC2. Umožňuje previesť číselný formát z hexadecimálneho na dekadický. Po rozkliknutí Groupboxu *Point Operations* je možné vykonať operáciu sčítanie medzi dvoma bodmi. Ako bod je možné použiť generátor definovaný v štandarde alebo náhodný bod.

### Založka Practice

Záložka *Practice*, demonštruje aplikovanie eliptických kriviek. *Practice* záložka zahrňuje prevod správy na body pomocou Koblitzovho algoritmu, výmenný mechanizmus ECDH, pomocou malý kriviek a podpisový algoritmus ECDSA taktiež pomocou malých kriviek.

The screenshot shows the 'Practice' application window. At the top, there's a title bar 'ELLIPTIC CURVES CRYPTOGRAPHY EDUCATION APPLICATION'. Below it, a navigation bar has 'Algorithms', 'Small Curves', 'Large Curves', and 'Practice' (highlighted in blue). On the left, a vertical sidebar lists 'ECDSA', 'ECDH', and 'Koblitz'. The main area has a 'Standard' dropdown set to 'secp224r1'. Below it, an 'Open Text' input field contains 'testovací text prevodu znakov na body', followed by an 'ENCODE' button. A table with columns 'CHAR', 'POINT', and 'SELECT' displays a list of points. The points are grouped by character: 't', 'e', 's', 'o', 'v', 'a', 'c', 'i', 'x', 'p', 'r', 'd'. The points for 'e', 's', 'o', 'v', 'a', and 'c' are selected, indicated by checked checkboxes. At the bottom, a 'Decoded Points' input field shows 'e s o v a c'.

CHAR	POINT	SELECT
t	(0D9D,0F39B8221B71F68EDB523AA2562C714B692B8C4FE95B6F7E078E7D7EE)	<input type="checkbox"/>
e	(0BD7,0E9EE22212C1227E028BFB440DB9FFB7D2EF2B44E637228C60D2084AF)	<input checked="" type="checkbox"/>
s	(0D7A,0B50B6C61B9DEAE3979E023497E2166493A59ACF155F983980F013619)	<input checked="" type="checkbox"/>
o	(0D02,082F408DB431DE3E6328AF2862CE5587D76B70D0F8FEEDA987C2AEC93)	<input checked="" type="checkbox"/>
v	(0DD4,0D0B068C675A763342AF764844B475DA660B1A2E0073F67A483C3C5DA)	<input checked="" type="checkbox"/>
a	(0B5E,33941B06E589262B1DDA25B03550C2B3893F78229AED5F78757DD124)	<input checked="" type="checkbox"/>
c	(0B9A,77494C7DB2ED1C7400483B51A50B3034C867C5082340BB41EF00DC52)	<input checked="" type="checkbox"/>
i	(1BC6,08F19964E1E119D2305CFAE6DC5EB1911035B09F9AD25D7D899597032)	<input type="checkbox"/>
x	(0E10,09B52779EB34769291DF1E78B46C87F15F4C6AA15B15972BADD82336A)	<input type="checkbox"/>
p	(0D20,0A267E60E2C0B9687F0DD323DAC22E408E51883063A2D773C14215560)	<input type="checkbox"/>
r	(0D62,220FE5AAAC365E3341BB28E286903BE99F63C1FCF0F1582DA5C3F15)	<input type="checkbox"/>
d	(0BBC,0E793F90B41E2374E993933AE47414778AC53F85A9F23BBEE6ED23669)	<input type="checkbox"/>

Decoded Points: e s o v a c

Obrázok 16 : Prevod správy v grafickom rozhraní na záložke Practice

Prevod správy začína výberom štandardu, podľa ktorého sa nastaví parametre krivky. Výsledkom prevodu je zoznam (mapa) bodov. Po výbere bodu sa bod dekoduje späť na znak.

Podzáložka *ECDH*, demonštruje výmenný mechanizmus kľúčov, cez nezabezpečený kanál. Užívateľ si vyberie krivku a bod pomocou ktorého chce výmenu vykonať. Nastaví si súkromný kľúč osoby A a súkromný kľúč osoby B. Vynásobený bod súkromným kľúčom sa hneď prepočítava a zobrazuje sa v okienku *Point Qa* a *Point Qb*. Po kliknutí na tlačidlo *Exchange* sa vykoná výmena, a zobrazí sa spoločný kľúč. Ovládanie výmenného mechanizmu je zobrazené na obrázku 17.

Poslednou podzáložkou je *ECDSA* (Obrázok 18). Táto podzáložka demonštruje podpisovanie textovej správy a overovanie podpisu. Algoritmus je implementovaný nad

malými krivkami  $p < 541$ . Parameter  $p$  musí byť prvočíslo, v opačnom prípade sa zobrazí chybová správa. Pre obe vstupné správy sa zobrazuje hash, aby bolo vidieť prípadnú zmenu.

ELLIPTIC CURVES CRYPTOGRAPHY EDUCATION APPLICATION

Algorithms Small Curves Large Curves Practice

ECDSA ECDH Koblitz

Curve Parameters

A: 61

B: 22

p: 73

Point: (20,65)

Entity A

Private Key: 4

Point Qa: (27,72)

Entity B

Private Key: 10

Point Qb: (20,65)

EXCHANGE

Common Key: (27,72)

Obrázok 17 : Grafické rozhranie výmenného mechanizmu ECDH

ELLIPTIC CURVES CRYPTOGRAPHY EDUCATION APPLICATION

Algorithms Small Curves Large Curves Practice

ECDSA ECDH Koblitz

Public Parameters

Curve E:  $y^2 = x^3 + 35x + 68 \pmod{229}$

Point P: (33,73) Order(P): 59

Signing

Message: testovacia sprava pre podpisovanie

Message Hash: 019CF2244826ACD7C39B1275C562427EB5AD1CBF

Signature: (5,13)

Verify

Message: testovacia sprava pre podpisovanie

Signature (r,s): 5 13

Message Hash: 019CF2244826ACD7C39B1275C562427EB5AD1CBF

Verification result: Message was not changed

Obrázok 18 : Grafické rozhranie podpisového algoritmu ECDSA

## ZÁVER

Táto bakalárska práca zhrnula základné poznatky o využití eliptických kriviek v kryptografii. Popísané boli jednotlivé operácie s krivkami, algoritmy, ktoré sa používajú pri práci s krivkami, ale aj kryptografické systémy, ktoré sa používajú na zabezpečenie reálnych aplikácií.

Praktickým prínosom tejto práce bolo vytvorenie aplikácie, ktorá umožňuje pomocou jednoduchého grafického rozhrania si vyskúšať elementárne operácie nad eliptickými krivkami a pochopiť jednotlivé algoritmy a princípy využívané pri aplikovaní aritmetiky kriviek v kryptografických systémoch.

Aplikovanie eliptických kriviek v kryptografií ešte len čaká na svoje výsledky. Tento nádejný smer si určite nájde uplatnenie predovšetkým na mobilných zariadeniach a iných prístrojoch, ktoré nedisponujú vysokým výpočtovým výkonom, ale sú súčasťou komunikácie, ktorú je nutné zabezpečiť.

Ako už bolo spomenuté, táto práca zhrnula elementárne poznatky, preto sa pre viac informácií odkazujem na Washingtona [25], ktorý popisuje viac matematické pozadie a na knihu od Menezes a Vanstone [20], ktorá rozoberá eliptické krivky a príslušné algoritmy do detailu. Ku kryptológii patrí aj kryptoanalýza.

Dobre čitateľná publikácia o kryptoanalýze, bez zbytočne veľa matematických forém, je kniha Christophera Swensona [1].

Hodnotným zdrojom informácií a novinek z oblasti kryptografie je blog známeho autora Bruce Schneiera [10].

Na priloženom CD sú umiestené všetky zdrojové kódy aplikácie, vrátane elektronickej verzie tejto práce.



## CONCLUSION

This bachelor thesis summarized basic knowledge about application of elliptic curves in cryptography. Basic operation with curves and points was described in first part. As important part of paper was discussed cryptographic systems and algorithms which are used in real world applications.

Output of this paper is application, which could be used to tryout elementary operations over elliptic curves placed in simple intuitive graphical interface. Application has also capabilities for demonstration corresponding algorithms and elliptic curves cryptographic systems.

Application of elliptic curve in cryptography is part of next generation cryptographic systems. This promising area is definitely usable mainly on mobile devices and other equipment, which do not have high rate computing performance, but they are need to be part of secure communication.

As was mentioned, this thesis summarized only basic information, so I refer to Washington's book [25], which discuss more about mathematical background and to book from Menezes and Vanstone [20], which discuss elliptic curves and corresponding algorithms in detail. Part of cryptology is also cryptanalysis.

Readable publication about cryptanalysis, without unnecessary mathematical formulations, is Swenson's book [1].

Valuable source of information and news from cryptography is blog from good known author Bruce Schneier [10].

The attached CD contains full sources of application and digital form of this thesis.

## ZOZNAM POUŽITEJ LITERATURY

- [1] SWENSON, Christopher. Modern cryptanalysis: techniques for advanced code breaking. Indianapolis, IN: Wiley Pub., 2008, 236 s. ISBN 04-701-3593-X.
- [2] KLÍMA, Vlastimil. Finding MD5 Collisions. [online]. Praha, 2005. Dostupné z: [http://cryptography.hyperlink.cz/md5/MD5\\_collisions.pdf](http://cryptography.hyperlink.cz/md5/MD5_collisions.pdf)
- [3] Teoretický základ elektronického podpisu. NBÚ SR. *Národný bezpečnostný úrad* [online]. 2011. Dostupné z: <http://www.nbusr.sk/sk/elektronicky-podpis/elektronicky-podpis/teoreticky-zaklad-elektronickeho-podpisu/index.html>
- [4] PETERKA, Jiří. Báječný svět elektronického podpisu. Praha: CZ.NIC, c2011, 430 s. CZ.NIC. ISBN 978-80-904248-3-8.
- [5] Jak pracuje Enigma. *Enigma.eleferno.cz* [online]. 2007. Dostupné z: <http://enigma.eleferno.cz/index.php?text=13-jak-pracuje-enigma>
- [6] FIPS 46-3.Data Encryption Standard (DES). 5.vyd. USA: National Institute of Standards and Technology, 1999. Dostupné z: <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- [7] KLÍMA, V.: Nová šifra nastupuje, Chip 5/2002, 142-144.
- [8] FIPS 197.Advanced Encryption Standard (AES). USA: National Institute of Standards and Technology, 2001. Dostupné z: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- [9] How secure is AES against brute force attacks?. In: EETimes [online]. 2012. Dostupné z: <http://www.eetimes.com/design/embedded-internet-design/4372428/How-secure-is-AES-against-brute-force-attacks>
- [10] SCHNEIER, Bruce. Another New AES Attack. Schneier on Security [online]. 2009. Dostupné z: [http://www.schneier.com/blog/archives/2009/07/another\\_new\\_aes.html](http://www.schneier.com/blog/archives/2009/07/another_new_aes.html)
- [11] JURISIC, Aleksandar a Alfred J. MENEZES. Elliptic Curves and Cryptography. [online]. 1999. Dostupné z: <http://www.cs.nthu.edu.tw/~cchen/CS4351/jurisic.pdf>
- [12] KLÍMA, V.: Eliptické křivky a šifrování 1. Chip 9/2002, 134–136.
- [13] HORT, Daniel a Jiří RACHŮNEK. Algebra I. 1. vyd. Olomouc: Univerzita Palackého, 2003, 172 s. Skripta (Univerzita Palackého). ISBN 80-244-0631-4.
- [14] CHAJDA, Ivan. Okruhy a moduly. 1. vyd. Olomouc: Univerzita Palackého, 2003, 65 s. Skripta (Univerzita Palackého). ISBN 80-244-0687-X.

- [15] STANEK, Martin. Základy kryptologie. In: Katedra informatiky [online]. 2004. Dostupné z: <http://www.dcs.fmph.uniba.sk/~staneck/crypto/main2.pdf>
- [16] Základy modulárnej aritmetiky. [online]. 2003. Dostupné z: [http://www.kemt.fei.tuke.sk/predmety/KEMT414\\_AK/\\_materialy/Cvicenia/kryp\\_1\\_2.pdf](http://www.kemt.fei.tuke.sk/predmety/KEMT414_AK/_materialy/Cvicenia/kryp_1_2.pdf)
- [17] BROWN, Ezra. Square Roots from 1;24,51,10 to Dan Shanks. The Department of Mathematics at Virginia Tech [online]. 1999. Dostupné z: <http://www.math.vt.edu/people/brown/doc/sqrts.pdf>
- [18] Computing modular square roots in Python. Eli Bendersky's website [online]. 2009. Dostupné z: <http://eli.thegreenplace.net/2009/03/07/computing-modular-square-roots-in-python/>
- [19] STEIN, William A. Elementary number theory: primes, congruences, and secrets : a computational approach. New York, NY: Springer, c2009, x, 166 p. ISBN 03-878-5524-6.
- [20] HANKERSON, Darrel, Alfred J MENEZES a Scott VANSTONE. Guide to elliptic curve cryptography. Vyd. 1. New York: Springer, 2004, 311 s. ISBN 03-879-5273-X.
- [21] TILBORG, Henk C. Fundamentals of cryptology: a professional reference and interactive tutorial. Boston: Kluwer Academic Publishers, c2000, xiv, 490 p. ISBN 07-923-8675-2.
- [22] MS, Anoop. Elliptic Curve Cryptography. [online]. 2006. Dostupné z: [http://www.tataelxsi.com/whitepapers/ECC\\_Tut\\_v1\\_0.pdf](http://www.tataelxsi.com/whitepapers/ECC_Tut_v1_0.pdf)
- [23] BURŠÍK, F. Problematika převodu zprávy na body eliptické křivky. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2007. 7 s.
- [24] KOBLITZ, N. Elliptic Curve Cryptosystems. Mathematics of Computation. 1987.
- [25] WASHINGTON, Lawrence C. Elliptic curves: number theory and cryptography. 2nd ed. Boca Raton, FL: Chapman, 2008, 513 s. ISBN 14-200-7146-7.
- [26] DON B. JOHNSON, Don a Alfred J. MENEZES. Elliptic Curve DSA. Department of Computer Science [online]. 1998. Dostupné z: <http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa.pdf>

- [27] MENEZES, Alfred J, Paul VAN OORSCHOT a Scott VANSTONE. Handbook of applied cryptography. Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 08-493-8523-7.
- [28] ECC Holds Key to Next-Gen Cryptography. VANSTONE, Scott. EETimes [online]. 2004. Dostupné z: <http://eetimes.com/electronics-news/4144307/ECC-Holds-Key-to-Next-Gen-Cryptography>
- [29] NIST 800-57. Recommendation for Key Management. USA: NIST, 2007. Dostupné z: [http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2\\_Mar08-2007.pdf](http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf)
- [30] SEC2. Recommended Elliptic Curve Domain Parameters. 2. vyd. Canada: Certicom, 2010. Dostupné z: <http://www.secg.org/download/aid-780/sec1-v2.pdf>
- [31] DataBinding in WPF. WPF Tutorial [online]. 2011. Dostupné z: <http://www.wpfutorial.net/DataBindingOverview.html>
- [32] Implementing the Model-View-ViewModel Pattern. MSDN Library [online]. 2012. Dostupné z: <http://msdn.microsoft.com/en-us/library/ff798384.aspx>

**ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK**

AES	Advanced Encryption Standard.
ANSI	American National Standards Institute
CSS	Cascading Style Sheets
DES	Data Encryption Standard.
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme/Standard
FIPS	Federal Information Processing Standard
GCD	Greatest Common Divisor
HTML	HyperText Markup Language
IBM	International Business Machines Corporation.
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
NIST	National Institute of Standards and Technology
RSA	Rivest Shamir Adleman
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

## ZOZNAM OBRÁZKOV

Obrázok 1 : Schéma vytvorenia a overovania elektronického podpisu.....	12
Obrázok 2 : Eliptická krivka nad množinou reálnych čísiel .....	26
Obrázok 3 : Geometrický pohľad na sčítanie dvoch bodov.....	27
Obrázok 4 : Geometrické zobrazenie zdvojenia bodu P.....	28
Obrázok 5 : Sčítanie opačných bodov .....	28
Obrázok 6 : Výmena kľúčov pomocou EC Diffie-Hellman algoritmu .....	34
Obrázok 7 : Zjednodušená schéma aplikácie.....	45
Obrázok 8 : Diagram tried LargePoint a SmallPoint.....	46
Obrázok 9 : Diagram tried LargeCurve a SmallCurve .....	47
Obrázok 10 : Data Binding diagram prevzatý z [31].....	49
Obrázok 11 : MVVM diagram, prevzatý z [32] .....	50
Obrázok 12 : Grafické rozhranie demonštračnej aplikácie.....	51
Obrázok 13 : Záložka Algorithms .....	52
Obrázok 14 : Ovládacie prvky záložky Small Curves.....	52
Obrázok 15 : Záložka Large Curves .....	53
Obrázok 16 : Prevod správy v grafickom rozhraní na záložke Practice .....	54
Obrázok 17 : Grafické rozhranie výmenného mechanizmu ECDH .....	55
Obrázok 18 : Grafické rozhranie podpisového algoritmu ECDSA .....	55

**ZOZNAM TABULIEK**

Tabuľka 1 : Body krivky $y^2 = x^3 + x + 1 \bmod 19$ .....	30
Tabuľka 2 : Porovnanie zápisov v $Z_p^*$ a $E(Zp)$ , prevzaté z [26].....	36
Tabuľka 3 : Paralela medzi zápisom DSA a ECDSA, prevzaté z [26].....	36
Tabuľka 4 : Porovnanie matematických problémov jednotlivých systémov.....	42
Tabuľka 5 : Porovnanie bezpečnosti jednotlivých systémov .....	43
Tabuľka 6 : Doporučené dĺžky kľúčov podľa NIST.....	43