Implementácia ECC v .NET framework

Peter Vagánek, Pavol Zajac *
p.vaganek@gmail.com

10. apríla 2008

Abstrakt

Tento článok sa zameriava na výkonnostné testy knižnice pre prácu s eliptickými krivkami. Flexibilitu implementácie krivkovej aritmetiky ukážeme na základných operáciach (adding, doubling, scalar multiplication) v grupe bodov eliptickej krivky nad poľom $\mathbb{F}$, ktoré je možné realizovať v troch rôznych súradnicových reprezentáciach. Hlavným bodom článku sú potom výkonnostne testy troch techník násobenia bodov (addition-subtraction ladder, Montgomery ladder, sliding window multiplication) vo všetkých troch súradnicových systémoch.

1 Úvod

Dnes v ešte stále v relatívne novej oblasti ECC, sa kladie dôraz na skúmanie už existujúcich, a hľadanie inovatívnych riešení ako realizovať *násobenie bodov (scalar multiplaction)* v grupe bodov eliptickej krivky. Táto úloha priamo súvisí s problémom eliptického diskrétneho logaritmu (ECDLP) a bezpečnosťou eliptických kryposystémov (napr. ECDSA). V článku budú teda porovnané časové zložitosti respektíve výkonnosť algoritmov násobenia bodov vzhľadom na danú reprezentáciu a krivku.

*Vedúci práce

2 Eliptické krivky nad $\mathbb{F}_p$ - jemný úvod

Eliptická krivka nad poľom $\mathbb{F}_p$ ($E/\mathbb{F}_p$) je množina bodov (riešení) $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ rovnice

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

kde $a_i \in \mathbb{F}_p$. Ak char $\mathbb{F}_p \neq 2, 3$ je možné upraviť tuto rovnicu tak, že

$$y^2 = x^3 + a_4x + a_6 \quad (2)$$

Všetky takéto body tvoria abelovskú aditívnu grupu $(E(\mathbb{F}, +))$ spolu s adjungovaným bodom $\mathcal{O}$ v nekonečne. Definícia operácie sčítania vychádza z geometrického významu operácie sčítania bodov na eliptickej krivke nad reálnymi číslami. Vzorce je možné odvodiť na základe pravidla že súčet troch bodov na priamke pretínajúcej eliptickú krivku je 0.

Počet bodov eliptickej krivky danej rovnicou (2) môžeme vyjadriť s pomocou Legendreho symbolu ako

$$\sharp E(\mathbb{F}_p) = 1 + p + \sum_{x \in \mathbb{F}_p} \left(\frac{x^3 + a_4x + a_6}{p}\right) \quad (3)$$

Podrobnejšie informácie o štruktúre eliptických kriviek sa môže čitateľ dozvedieť napr. v [1].

3 Reprezentácia bodov eliptickej krivky v knižnici ECC

Knižnica pracuje s krivkami definovanými nad $\mathbb{F}_p$ pre $p > 3$. Fakt, že body eliptickej krivky môžu byť reprezentované v rôznych súradnicových systémoch, poskytuje flexibilitu a „voľnosť" pri realizovaní krivkových operácií. V ECC knižnici som implementoval tri rôzne reprezentácie (súradnicové systémy)

(i) Affine coordinates ($\mathcal{A}$) - *AfinePointEC*

(ii) Projective coordinates ($\mathcal{P}$) - *ProjectivePointEC*

(iii) Jacobian coordinates ($\mathcal{J}$) - *JacobianPointEC*

Za názvami súradnicových systémov a ich označením sa nachádza názov triedy, v ktorej sú implementované aritmeticke operácie charakteristické pre daný súradnicový systém. Medzi charakteristické operácie a črty pre konkrétny súradnicový systém patria:

+ Interná reprezentácia bodu
+ Adding (operácia sčitania dvoch bodov)
+ Doubling (operácia zdvojenia bodu)
+ Reprezentácia opačného bodu
+ Reprezentácia bodu v nekonečne $\mathcal{O}$
+ Rovnosť bodov
+ Tvar rovnice eliptickej krivky

Pre každú jednu reprezentáciu bodu, sú vyššie uvedené operácie unikátne[1]. Rozhranie ako aj implementáciu metód ktoré sú pre všetky tri reprezentácie(triedy) rovnaké im poskytuje materská trieda *PointEC*. Ďalšou dôležitou časťou knižnice je možnosť využiť tri rôzne implementované formy eliptického násobenia (*scalar multiplication)*

+ Adding-subtraction ladder
+ Montgomery ladder
+ Sliding window

Flexibilita knižnice jednak v reprezentácii bodov, jednak v eliptickom násobení teda umožňuje vykonať výkonnostné testy vzhľadom na použitú reprezentáciu a typ násobenia.

3.1 Reprezentácia $\mathcal{A}$

Eliptická krivka $E(\mathbb{F}_p)$ je daná rovnicou

$$y^2 = x^3 + ax + b \tag{4}$$

Ak bod $P = (x_1, y_1)$ leží na E, potom k nemu opačný bod je reprezentovaný ako $P = (x_1, -y_1)$. Pri takejto reprezentácii bodov si *adding* a *doubling* operácie vyžadujú $I+2M+S$ resp. $I+2M+2S$ (kde I je operácia multiplikatívnej inverzie, S squaring a M násobenie súradníc).

3.2 Reprezentácia $\mathcal{P}$

Eliptická krivka $E(\mathbb{F}_p)$ je daná rovnicou

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{5}$$

Bod $(X_1 : Y_1 : Z_1)$ na krivke E zodpovedá $\mathcal{A}$-bodu $(X_1/Z_1, Y_1/Z_1)$ kde $Z_1 \neq 0$. Bod v nekonečne je reprezentovaný ako $(0 : 1 : 0)$ a opačný bod k bodu $(X_1 : Y_1 : Z_1)$ je daný ako $(X_1 : -Y_1 : Z_1)$.

Pri takejto reprezentácii bodov si *adding* a *doubling* operácie vyžadujú $12M + 2S$ resp. $7M + 5S$[2].

[1] Jednotlivé súradnicové systémy je možné pohodlne meniť pomocou triedy *ConvertPoint*

[2] žiadna multiplikatívna inverzia nie je potrebná, čo je veľmi dôležité pri neefektívnej implementázii výpočtu inverzného prvku modulo p

3.3 Reprezentácia $\mathcal{J}$

Eliptická krivka $E(\mathbb{F}_p)$ je daná rovnicou

$$Y^2 = X^3 + aXZ^4 + bZ^6 \tag{6}$$

Bod $(X_1 : Y_1 : Z_1)$ na krivke E zodpovedá $\mathcal{A}$-bodu $(X_1/Z_1^2, Y_1/Z_1^3)$ kde $Z_1 \neq 0$. Bod v nekonečne je reprezentovaný ako $(1 : 1 : 0)$ a opačný bod k bodu $(X_1 : Y_1 : Z_1)$ je daný ako $(X_1 : -Y_1 : Z_1)$.

Pri takejto reprezentácii bodov si *adding* a *doubling* operácie vyžadujú $12M + 4S$ resp. $4M + 6S$[3].

Samozrejme uvedené tri možné reprezentácie nie sú jediné. V 13. kapitole sekcii 13.1. a 13.2 v [1] čitateľ nájde definíiu operácii v jednotlivých systémoch ako aj ďaľšie možné reprezentácie napr. pomocou Chudovski Jacobian systému, Montgomeryho tvar eliptických kriviek atď..

4 Implementované algoritmy pre eliptické násobenie

Základom podpisových algoritmov založených na eliptických krivkách je operácia násobenia bodu eliptickej krivky celým číslom Q = nP.

4.1 Addition-subtraction ladder

Vstup *bod* $P \in E(\mathbb{F}_p)$, *nezáporné celé číslo* n
Výstup $[n]P$

Implementácia využíva podobný princíp ako squaring algoritmus využívaný v modulárnom umocňovaní. Oproti klasickému squaring algoritmu sa však využíva fakt, že inverzia bodu eliptickej krivky je jednoduchá operácia, iba negujeme y-ovú súradnicu bodu. Odčítanie bodu má teda rovnakú zložitost' ako pripočítanie. V závislosti na zvolenej reprezentácii bodu eliptickej krivky je možné realizovať operáciu zdvojnásobenia bodu jednoduchšie ako operácia sčítania dvoch rôznych bodov. Vtedy sa s výhodou dá využiť tzv. sčítaco-odčítací binárny rebrík *(„addition-subtraction ladder")[3].*

4.2 Montgomery ladder

Vstup *bod* $P \in E(\mathbb{F}_p)$, *nezáporné celé číslo* n
Výstup $[n]P$

Tento algoritmus sa využíva najmä ak je eliptická krivka prevedená do Montgomeryho tvaru, ktorý je detailne vysvetlený v [1], avšak je aplikovateľný na všetky tri implementované reprezentácie bodov.

Tak ako predchádzajňuci algoritmus, aj Mongomeryho rebrík využíva binárnu reprezentáciu čísla n, ale rozdiel je v tom že v tomto algoritme sa pri každej iterácii vykonávajú až dve krivkové operácie (dobling, addition),čo pri reprezentácii $\mathcal{A}$ znamená zvýšenie počtu multiplikatívnych inverzií (v každom kroku 2, v a/s je to v najhoršom prípade 2). Pre $\mathcal{P}$ a $\mathcal{J}$ sa rapídne zvyšuje počet základných aritmetických operácii v poli $\mathbb{F}_p$, a to v každom prípade.

4.3 Sliding window

Vstup*bod* $P \in E(\mathbb{F}_p)$, *nezáporné celé číslo* n, *parameter* $k \geq 1$, *a predvypočítané hodnoty* $[3]P, [5]P, [7]P, ..., [(2^k - 1)]P$
Výstup $[n]P$

Algoritmus je použiteľný aj pre n v binárnej reprezentácií aj v NAF expanzii. V knižnici je implementovaná prvá možnosť.

[3]pozn. žiadna multiplikatívna inverzia nie je potrebná, dôležité pri neefektívnej implementázii výpočtu inverzného prvku modulo p

So vstupných prametrov je možné vyčítať ze tento algoritmus potrebuje pre svoju činnosť sadu (pole) prevypočítaných hodnôt. Ich počet závisí od vstupného parametra k - šírka okna (*„sliding window width"*). Presne ide o $2^{k-1} - 1$ predvypočítaných bodov[4].

Keďže algoritmus potrebuje mať už niektoré hodnoty násobku bodu známe, jeho časová náročnosť pri bežných riešeniach je jednoznačne vyššia ako pri ostatných dvoch algritmoch. Ale v prípade vhodného oddelenia fázy predvýpočtu bodov[5] a samotného násobenia, sa zo zvyšujúcou šírkou okna znižuje časová „cena" eliptického násobenia.

5 Výsledky

Výsledné časy[6] uvedené v tabuľkách vznikli ako aritmetický priemer 20 hodnôt. Pred začatím testovania bolo napevno vygenerované pole 20 pseudonáhodných cca. 128-bitových celočíselných činiteľov.

Pre eliptické násobenie „sliding window" je použitá šírka okna od 2 do 10 pre lepšiu ilustráciu.

V nasledujňucich tabuľkách pre krivky P192, P224, P256, P384 a P521 je zvýraznené pole(čas) kde metóda „sliding window"[7] bola rýchlejšia ako násobenie s využitím metódy „addition-subtraction ladder" (**bold**) resp. „Montgomery ladder" (*kurzíva*) .

SW	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{J}$
2	*90*	*109*	*109*
3	**81**	**102**	**100**
4	77	96	93
5	74	92	93
6	73	91	88
7	68	90	85
8	69	87	84
9	68	87	80
10	66	85	80
A/S L	**90**	**108**	**106**
ML	*139*	*161*	*181*

Tab.1: Test reríkov pre krivku P192

SW	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{J}$
2	**107**	*134*	*138*
3	98	**126**	**127**
4	90	122	120
5	88	117	116
6	86	113	111
7	82	112	109
8	81	109	106
9	79	108	102
10	78	106	102
A/S L	**109**	**134**	**135**
ML	*166*	*202*	*230*

Tab. 2: Test reríkov pre krivku P224

SW	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{J}$
2	**127**	*166*	*172*
3	116	**155**	**161**
4	109	149	148
5	105	144	142
6	100	139	135
7	98	137	134
8	98	134	131
9	96	130	127
10	92	131	125
A/S L	**132**	**165**	**169**
ML	*201*	*249*	*185*

[4] V knižnici som implementoval funkciu PrecomputePoints(int), kde vstupný parameter určuje šírku okna nie predvypočítaný počet bodov

[5] napríklad vhodné úložisko LDAP

[6] Údaje sú v milisekundách

[7] fáza predvýpočtu bodov nie je zahrnutá v nameraných hodnotách

Tab. 3:Test reríkov pre krivku P256

SW	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{J}$
2	**228**	*326*	*336*
3	209	**304**	**314**
4	191	293	293
5	183	282	281
6	177	271	272
7	172	268	264
8	168	259	257
9	167	259	253
10	161	253	248
A/S L	**234**	**324**	**333**
ML	*365*	*494*	*565*

Tab. 4: Test reríkov pre krivku P384

SW	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{J}$
2	**351**	**558**	*583*
3	322	524	**541**
4	300	499	508
5	286	480	486
6	278	469	468
7	273	465	455
8	266	453	444
9	259	444	435
10	250	438	431
A/S L	**389**	**562**	**573**
ML	*570*	*856*	*984*

Tab. 5: Test reríkov pre krivku P521

Pri experimentoch boli použité krivky a body podľa štandardu [5].

6 Záver

Testy ukázali že pre túto implmentáciu krivkovej aritmetiky je najvhodnejšia kombinácia reprezentácie bodov v afinných súradniciach a použitia skalárneho násobenia pomocou algoritmu „addition-subtraction ladder".

Reprezentácia bodov v afinných súradniciach si pri operáciach adding, doubling vyžaduje výpočet multiplikatívne inverzného prvku v poli $\mathbb{F}_p$, čo je vzhľadom na ostatné operácie v poli „drahá" operácia.

Ak však nie je použitá reprezentácia pomocou afinných súradníc relatívne prudko narastá počet elementárnych operácií v poli (tabuľka 7.). V tabuľke je tiež vidno že operácia doubling-u je „lacnejšia" ako operácia adding-u, okrem systému $\mathcal{A}$.

ADDING	**I**	**M**	**S**
$\mathcal{A}$	1	2	1
$\mathcal{P}$	0	12	2
$\mathcal{J}$	0	12	4
DOUBLING	**I**	**M**	**S**
$\mathcal{A}$	1	2	2
$\mathcal{P}$	0	7	5
$\mathcal{J}$	0	4	6

Tab. 7: Počet operácií v jednotlivých systémoch podľa [1]

Na realizáciu operácií v konečnom poli $\mathbb{F}_p$ je v knižnici použitá voľne dostupná trieda „*BigInteger*", ktorá zvláda efektívne výpočet multiplikatívnej inverzie v pomere k niekoľkým základným operáciam v konečnom poli.

V prípade, že nie je možné operáciu I efektívne realizovať stávajú sa zaujímavejšie aj ostatné alternatívy. Spolu s vhodným úložišťom dát na uchovanie predvypočítaných bodov a s využitím metódy eliptického násobenia pomocou algoritmu „sliding window" je možné dosiahnuť zaujímavé výsledky, zvlášť keď si všimneme že rozdiel časov zo stúpajúcou šírkou okna medzi afinnou a jacobiho reprezentáciou sa zmensuje a tiež rozdiely (zisky) medzi časmi pri zväčšení šírky okna sú väčsie ako pri ostatných reprezentáciach. Tiež absolútne čísla sú nižšie ako pri ďalších dvoch sledovaných algoritmoch.

Naopak pri implementovaných reprezentáciach bodov a krivky je Montgomerycho rebrík

jednoznačne neefektívna a najpomalšia metóda. Je to spôsobené tým že tento algoritmus je efektívne využívaný pri krivkách ktoré sú zadané v Montgomeryho tvare.

7 Použitá literatúra

[1] H.Cohen, G.Frey et.al Handbook of Elliptic and Hyperelliptic Curve Cryptography. Chapman & Hall/CRC, 2005

[2] P.Zajac Efektívne metódy generovania eliptických kriviek. Diplomová práca, 2004

[3] A.Menezes, P.van Oorschot, S.Vanstone Handbook of Applied Cryptography. CRC press, 1996

[4] M.Ciet Elliptic Curves and Smart Cards. UCL CryptoGroup, July 2003

[5] Digital Signature Standar (DSS). U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology, 2000

[6] American National Standard X9.62-1998. Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital SIgnature Algorithm(ECDSA). American Bankers Association,1998