MĚŘENÍ GENEROVÁNÍ NÁHODNÝCH BODŮ A ARITMETICKÝCH OPERACÍ S BODY NA ELIPTICKÝCH KŘIVKÁCH

BENCHMARKS OF GENERATING RANDOM POINTS AND ARITHMETIC OPERATIONS WITH POINTS ON ELLIPTIC CURVES

Josef Brychta, Radek Fujdiak
Fakulta elektrotechniky a komunikačních technologií VUT v Brně,
Technická 12, Brno
E-mail: {xbrych07, fujdiak}@vutbr.cz

Abstrakt:
Tento článek se zabývá měřením generování náhodných bodů a aritmetických operací s body na eliptických křivkách. Tyto měření byly implementovány na hardwarové platformy Raspberry Pi Zero/1/2/3. Byl kladen důraz na efektivitu využití systémových prostředků a energetickou náročnost dané hardwarové platformy. Konkrétně, se jednalo o využití paměťových a časových prostředků. Celkem bylo proměřeno 48 eliptických křivek standardů NIST a BRAINPOOL, okrajově také GOST a FRP.

Klíčová slova: Generování náhodných bodů, aritmetické operace, eliptické křivky, Raspberry Pi Zero/1/2/3, NIST, BRAINPOOL, GOST, FRP.

Abstract:
This article deals with benchmarks of generating random points and arithmetic operations with points on elliptic curves. These benchmarks were implemented on hardware platforms Raspberry Pi Zero/1/2/3. Focus was to efficiency of use system resources and energetic consumption of that hardware platform. Specifically, it was memory and time resources. Totally was measured 48 elliptic curves of standards NIST and BRAINPOOL marginally GOST and FRP too.

Keywords: Generating random points, arithmetic operations, elliptic curves, Raspberry Pi Zero/1/2/3, NIST, BRAINPOOL, GOST, FRP.

Úvod

Kryptografie je v dnešní době používána prakticky ve všech oblastech, které jakkoliv souvisí s počítači či sítěmi. Od internetového bankovnictví, platebních karet, šifrování souborů, disků nebo jen zabezpečený přístup na webové stránky. Jednou z možností, jak posunout kryptografii na další úroveň, je použití eliptických křivek, což je spojitá křivka, která je definovaná matematickou rovnicí, které umožňují použití pseudonáhodných či náhodných proměnných do kryptografických protokolů. K tomuto účelu se používá generování náhodných bodů na eliptických křivkách a dále použití aritmetických operací s body, čímž může být sčítání, násobení apod. těchto bodů na eliptických křivkách [1, 2].

Nemůžeme s jistotou tvrdit, že použití eliptických křivek je nejlepším řešením do budoucna, díky stále se rozvíjejícím kvantovým počítačům, a v současnosti se již od tohoto řešení pomalu upouští. Moderní kryptografické knihovny umožňují práci s eliptickými křivkami jako vedlejší funkcionalitu, či jako hotové, sestavené kryptografické protokoly pracující na eliptických křivkách (například Elliptic curve Diffie Hellman a Elliptic Curve Digital Signature Algorithm) [3, 4, 5]. Motivací k napsání tohoto článku bylo vytvoření benchmarků pro měření generování náhodných bodů a aritmetických operací s těmito body na eliptických křivkách a tím zjistit jejich reálnou efektivitu na daných platformách. Použitými hardwarovými platformami zde byla zařízení Raspberry Pi Zero/1/2/3, o kterých je možno se více dozvědět na stránkách výrobce [6].

Článek v první řadě seznamuje čtenáře s použitými kryptografickými knihovnami, jejich stručným popisem a funkcionalitou. Ve druhé řadě seznamuje s principem benchmarkového měření, hlavními měřenými parametry, a nakonec prezentuje dosažené výsledky měření a jejich diskuzi.

KNIHOVNY A ELIPTICKÉ KŘIVKY

Zde budou ve stručnosti popsány zvolené knihovny, a eliptické křivky, na kterých probíhalo měření generování náhodných bodů a aritmetických operací.

Knihovny

Jednalo se o knihovny OpenSSL, LibECC a Python.

- **Knihovna OpenSSL** se zabývá mimo své široké uplatnění na poli bezpečnosti také kryptografií na eliptické křivce. Tato knihovna poskytuje rozsáhlou sadu funkcí pro provádění operací na eliptických křivkách s konečnými poli, konkrétně práci s testovanými kryptografickými primitivy. OpenSSL je základem pro implementaci ECDSA (Elliptic Curve Digital Signature Algorithm a ECDH. Primárním zaměřením je tedy implementace těchto algoritmů. Dovoluje však i primitivní operace s eliptickými křivkami [7].
- **Knihovna LibECC** se naopak oproti OpenSSL zabývá výhradně kryptografií na eliptických křivkách. Tato knihovna poskytuje rozsáhlou sadu funkcí na provádění operací s eliptickými křivkami v konečném poli. Knihovna poskytuje patřičné nástroje k provedení testovacích měření. Mezi další využití této knihovny spadají možnosti implementace protokolů na bázi protokolu ECDH [8].
- **Python** je interpretovací vysokoúrovňový programovací jazyk pro obecné programování. Jednou z možností tohoto programovacího jazyka je možnost přesného definování matematické funkce, tedy eliptické křivky. Dále deklaraci bodu na eliptické křivce a úkony s tím související. Byl zde zařazen pro porovnání výkonnosti se specializovanými knihovnami.

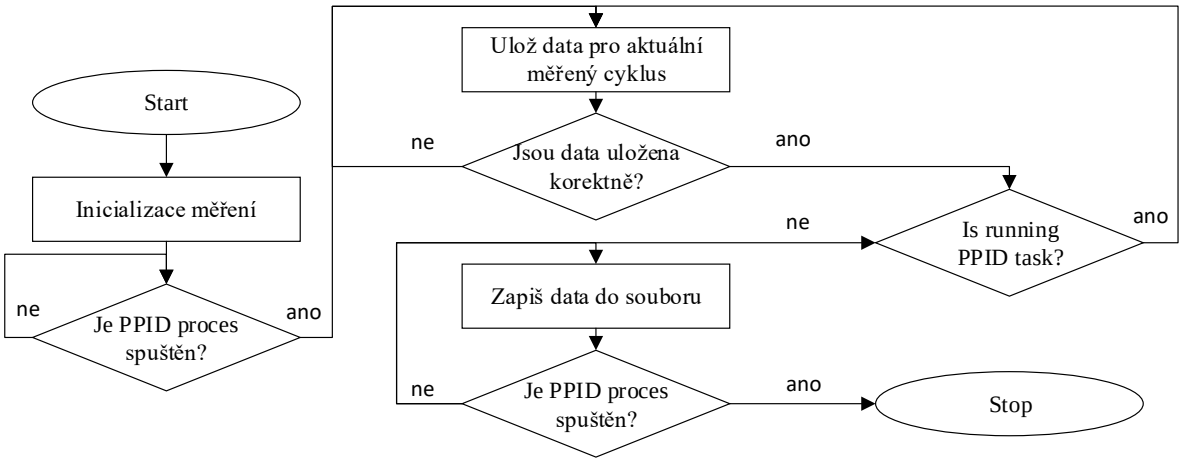
Eliptické křivky

Zvolené eliptické křivky standardů NIST a BRAINPOOL, okrajově také GOST a FRP jsou z intervalu velikosti pole 112 až 512 bitů. Použité parametrické označení eliptických křivek ve výsledcích je R, K, T. Parametrickým označením R se myslí eliptická křivka s náhodně zvolenými parametry v rozsahu bitů stejném jako je velikost pole, nad kterým se eliptická křivka nachází. Parametrickým označením K se myslí Koblitzova eliptická křivka, která se řídí svou vlastní parametrizací. Nakonec parametrickým označením T se myslí Twisted Edwards eliptické křivky, které se řídí také svou vlastní parametrizací. Více o této parametrizaci je možné se dočíst v oficiální dokumentaci daných standardů NIST (SECP, SECT) [9] a BRAINPOOL [10].

TEORIE MĚŘENÍ

Měření bylo prováděno pomocí měřicího algoritmu naprogramovaného v Pythonu s použitím knihovny PSutil. Dále bylo třeba naprogramovat výpočetní operace, tím je myšleno generování bodů na eliptických křivkách a aritmetické operace s těmito body. Generování náhodných bodů na eliptické křivce bylo naprogramováno pomocí interních algoritmů daných knihoven, které umožňují přizpůsobit generování bodů požadovaným kritériím. Algoritmus pracoval tak, že generoval body v dané grupě, ve které se nachází eliptická křivka a ověřoval, zda se na této křivce nachází. Tudíž generování trvá tak dlouho, dokud se nevygeneruje bod, který odpovídá. Konkrétní měřící algoritmus měří průměr tisíce vygenerovaných bodů na eliptické křivce. Dále byly naprogramovány aritmetické operace s již vygenerovanými body na eliptické křivce. Tímto je myšleno sčítání, násobení apod. Ve článku bylo měřeno násobení, jelikož se používá u kryptografických protokolů nejčastěji a je zároveň nejnáročnější. Realizováno bylo tak, že se vzala jedna souřadnice, například x, náhodného bodu na eliptické křivce c o velikosti n. Tudíž, bylo získáno n-bitové číslo (n je velikost grupy) a také naše konstanta pro danou křivku, např. A ($A=x$ o velikosti n). Tato konstanta byla stejná pro všechna měření dané křivky napříč platformami. Konstanta A se generovala pro každou velikost křivky jiná. Matematické násobení bodů na eliptické křivce probíhalo základní skalární metodou Double-and-add, dále skalární metodou wNAF a skalární Montgomeryho žebřík. Rozdíly mezi těmito metodami jsou ryze matematického charakteru a používají se v současnosti spolu s ostatními [11, 12].

Samotný měřící algoritmus funguje následovně. Každé aritmetické operaci (dále výpočetní operace) je po zahájení konání dané operace přidělen PPID identifikátor. Tato operace běží v určitém cyklu. Zde konkrétně 1000 opakování. Po načtení kryptografických knihoven a spuštění výpočetních operací těchto knihoven se spustí jednotlivá měření. Po skončení běhu výpočetních operací se ukončí také měření. Výsledky jednotlivých měření se zapíší do souboru. Hodnoty v tomto článku jsou měřeny v milisekundách a byly průměrovány. Vývojový diagram tohoto měřícího algoritmu je znázorněn na obr. 1.



Obr. 1: Vývojový diagram měřícího algoritmu.

VÝSLEDKY MĚŘENÍ OPENSSL

V následující kapitole jsou výsledky měření generování náhodných bodů na eliptických křivkách a aritmetická operace s těmito body. Z důvodu kapacity článku bylo vybráno pouze generování náhodných bodů a aritmetická operace násobení náhodných bodů na eliptické křivce.

Generování náhodného bodu

Výsledky měření časové náročnosti generování náhodného bodu na eliptických křivkách standardů SECP, SECT a BRAINPOOL na zařízeních Raspberry Pi Zero/1/2/3 viz tab. 1.

Tab. 1: Generování náhodného bodu v OpenSSL.

	RPi3	RPi2	RPiZero	RPi1
	čas[ms]	**čas[ms]**	**čas[ms]**	**čas[ms]**
SECP				
112R1	1,010	1,559	3,405	4,629
112R2	1,024	1,576	3,587	4,715
128R1	1,112	1,715	3,578	5,109
128R2	1,133	1,777	3,690	5,632
160K1	1,769	2,797	5,987	8,934
160R1	1,614	2,514	5,227	7,437
160R2	1,616	2,518	5,078	7,350
192K1	2,389	3,951	8,169	11,816
224K1	3,180	5,321	10,946	15,593
224R1	2,850	4,720	9,742	13,751
256K1	4,199	5,999	13,666	19,746
384R1	8,874	14,865	27,434	39,592
521R1	19,349	33,420	60,699	86,438
SECT				
113R1	1,402	1,600	3,428	5,082
113R2	1,406	1,607	3,417	5,398
131R1	2,561	2,795	6,180	9,365
131R2	2,638	2,857	6,344	9,430
163K1	3,354	3,680	8,064	11,755
163R1	3,607	3,894	8,615	12,631
163R2	3,626	3,916	8,791	13,150
193R1	4,683	4,729	11,190	16,134
193R2	4,537	4,626	10,629	15,598
233K1	6,286	6,277	14,593	21,863
233R1	6,948	6,813	16,470	23,877
239K1	6,443	6,428	15,118	22,092
283K1	11,575	11,371	27,020	39,883
283R1	12,955	12,420	30,496	44,364
409K1	26,247	23,248	61,165	82,758
409R1	30,014	26,219	69,676	93,140
571K1	61,400	54,230	133,040	189,659
571R1	70,901	61,326	151,954	230,124
BRAINPOOL				
P160T1	1,620	2,448	5,026	7,196
P160R1	1,698	2,626	5,426	7,752
P192T1	2,179	3,470	7,053	10,051
P192R1	2,310	3,747	7,396	10,427
P224T1	2,870	4,616	9,154	13,031
P224R1	3,081	5,034	10,176	14,430
P256T1	3,779	5,346	11,602	17,328
P256R1	4,126	5,747	12,883	18,274
P320T1	6,007	10,034	18,695	27,125
P320R1	6,633	11,360	20,952	29,248
P384T1	8,919	14,593	27,097	39,348
P384R1	9,996	16,512	31,223	43,727
P512T1	17,499	25,864	54,354	77,879
P512R1	19,963	29,478	62,178	87,791

Z měření je patrná určitá závislost úměrně se zvyšujícím časovým nárokům generování bodu na velikosti grupy dané křivky. Od nejméně časově náročných křivek standardu SECP, přes křivky standardu BRAINPOOL a nejvíce náročné křivky SECT.

Násobení náhodných bodů

Výsledky měření časové náročnosti aritmetické operace násobení náhodných bodů na eliptických křivkách standardů SECP, SECT, BRAINPOOL na zařízeních Raspberry Pi Zero/1/2/3 viz tab. 2.

Tab. 2: Násobení náhodných bodů v OpenSSL.

	RPi 3	RPi2	RPiZero	RPi1
	čas[ms]	**čas[ms]**	**čas[ms]**	**čas[ms]**
SECP				
112R1	0,912	1,392	2,857	4,130
112R2	1,867	2,859	6,065	8,660
128R1	2,896	4,436	9,309	13,341
128R2	3,964	6,099	12,758	18,262
160K1	5,611	8,672	18,236	26,427
160R1	7,103	10,968	23,023	33,161
160R2	8,597	13,290	27,671	39,873
192K1	10,855	17,045	35,298	51,137
224K1	13,877	22,117	45,595	66,120
224R1	16,574	26,580	54,339	78,552
256K1	20,600	32,349	67,363	97,732
384R1	29,175	46,836	94,474	136,282
521R1	48,083	79,955	154,408	219,871
SECT				
113R1	1,365	1,543	3,291	4,871
113R2	2,722	3,072	6,529	9,642
131R1	5,297	5,815	12,599	18,803
131R2	7,871	8,555	18,829	27,825
163K1	11,168	12,108	26,553	39,404
163R1	14,732	15,902	35,079	51,568
163R2	18,301	19,693	43,451	64,069
193R1	22,855	24,213	54,037	79,647
193R2	27,403	28,733	64,516	95,221
233K1	33,679	34,845	79,013	116,606
233R1	40,609	41,455	94,971	140,103
239K1	47,058	47,722	109,801	162,036
283K1	58,714	58,857	136,721	198,530
283R1	71,726	71,103	166,221	239,284
409K1	98,187	94,002	226,094	320,861
409R1	128,309	119,805	293,841	420,041
571K1	189,930	173,672	431,807	618,178
571R1	260,024	234,549	585,000	850,972
BRAINPOOL				
P160T1	3,052	2,475	9,574	13,907
P160R1	1,574	4,790	4,875	7,185
P192T1	7,195	11,692	22,638	33,130
P192R1	5,189	8,380	16,374	23,757
P224T1	12,743	20,931	40,513	59,014
P224R1	10,073	16,539	31,970	46,720
P256T1	20,156	31,468	63,582	92,537
P256R1	16,614	26,426	52,557	76,548
P320T1	32,129	52,100	101,096	146,643
P320R1	26,455	42,392	83,279	121,098
P384T1	50,189	82,573	156,017	226,702
P384R1	41,694	68,291	130,928	188,768
P512T1	86,289	137,637	258,467	387,540
P512R1	69,440	111,767	210,662	312,774

Z měření je patrná určitá závislost, stejně jako u předchozího měření ale se značně vyššími časovými nároky výpočetní operace.

VÝSLEDKY MĚŘENÍ LIBECC

V následující kapitole jsou výsledky provedených měření generování náhodných bodů na eliptických křivkách a aritmetická operace s těmito body. Z důvodu kapacity článku bylo vybráno pouze generování náhodných bodů a aritmetická operace násobení náhodných bodů na eliptické křivce. Oproti knihovně OpenSSL knihovna LibECC obsahuje méně křivek ale více standardů. Byly tedy proměřeny všechny dostupné křivky ze všech standardů knihovny LibECC.

Generování náhodného bodu

Výsledky měření časové náročnosti generování náhodného bodu na eliptických křivkách standardů SECP, BRAINPOOL, GOST a FRP na zařízeních Raspberry Pi Zero/1/2/3 viz tab. 3.

Tab. 3: Generování náhodných bodů v LibECC.

	RPi3	RPi2	RPiZero	RPi1
	čas[ms]	**čas[ms]**	**čas[ms]**	**čas[ms]**
SECP				
192R1	2,185	3,636	6,933	4,927
224R1	35,57	60,251	106,047	80,239
256R1	4,179	7,030	12,620	10,069
384R1	12,083	19,958	37,309	26,306
521R1	30,995	53,005	96,303	70,905
BRAINPOOL				
P224R1	3,675	6,097	11,734	9,248
P256R1	4,211	7,176	12,997	9,626
P384R1	11,414	19,474	36,639	26,471
P512R1	24,574	42,312	76,518	57,872
GOST				
256	17,695	29,903	53,358	40,521
512	25,742	43,622	76,583	56,027
FRP				
256V1	4,153	7,105	12,881	11,380

Z měření je patrná určitá závislost úměrně se zvyšujícím časovým nárokům na generování bodu na velikosti grupy dané křivky. Od nejméně časově náročných křivek standardu SECP, přes křivky standardu FRP, BRAINPOOL a nejvíce náročné křivky SECT.

Násobení náhodných bodů

Výsledky měření časové náročnosti aritmetické operace násobení náhodných bodů na eliptických křivkách standardů SECP, SECT, BRAINPOOL na zařízeních Raspberry Pi Zero/1/2/3 bylo provedeno základní metodu skalární Double-and-add viz tab. 4 a metodou Montgomeryho žebřík.

Tab. 4: Násobení náhodných bodů v LibECC základní metodou skalární Double-and-add.

	RPi3	RPi2	RPiZero	RPi1
	čas[ms]	**čas[ms]**	**čas[ms]**	**čas[ms]**
SECP				
192R1	26,557	44,908	80,884	59,276
224R1	44,395	74,212	134,592	99,910
256R1	46,037	76,872	136,913	102,985
384R1	111,587	183,921	324,403	242,413
521R1	289,114	475,409	853,945	629,021
BRAINPOOL				
P224R1	44,343	73,834	134,703	97,159
P256R1	45,962	76,646	135,978	99,817
P384R1	111,297	184,984	333,615	247,983
P512R1	222,814	368,641	662,321	483,166
GOST				
256	45,870	76,246	137,81	102,343
512	237,269	392,419	683,528	522,168
FRP				
256V1	45,969	77,364	138,597	103,410

Výsledky měření primitiva násobení náhodných bodů na eliptických křivkách standardů SECP, SECT, BRAINPOOL na zařízeních Raspberry Pi Zero/1/2/3 metodou Montgomeryho žebřík viz tab. 5.

Tab. 5: Násobení náhodných bodů v LibECC skalární metodou Montgomeryho žebřík.

	RPi3	RPi2	RPiZero	RPi1
	čas[ms]	**čas[ms]**	**čas[ms]**	**čas[ms]**
SECP				
192R1	13,104	21,601	40,917	31,084
224R1	20,606	34,153	63,977	48,011
256R1	23,648	39,316	73,307	55,527
384R1	61,090	101,986	187,708	141,964
521R1	158,933	267,278	475,414	359,115
BRAINPOOL				
P224R1	20,652	34,350	60,841	48,179
P256R1	23,601	39,261	73,112	55,528
P384R1	60,974	102,157	187,647	140,892
P512R1	128,204	215,234	390,125	292,574
GOST				
256	23,526	39,187	69,152	54,737
512	127,869	214,677	388,960	293,821
FRP				
256V1	23,592	39,208	73,225	55,523

VÝSLEDKY MĚŘENÍ PYTHON

V této kapitole jsou měření násobení dvou náhodně vygenerovaných bodů na eliptických křivkách daných standardů. Konkrétně se jednalo o standardy SECP a BRAINPOOL. Časové zatížení generování náhodných bodů na eliptických křivkách je zobrazeno viz tab. 6.

Tab. 6: Generování náhodných bodů v Pythonu.

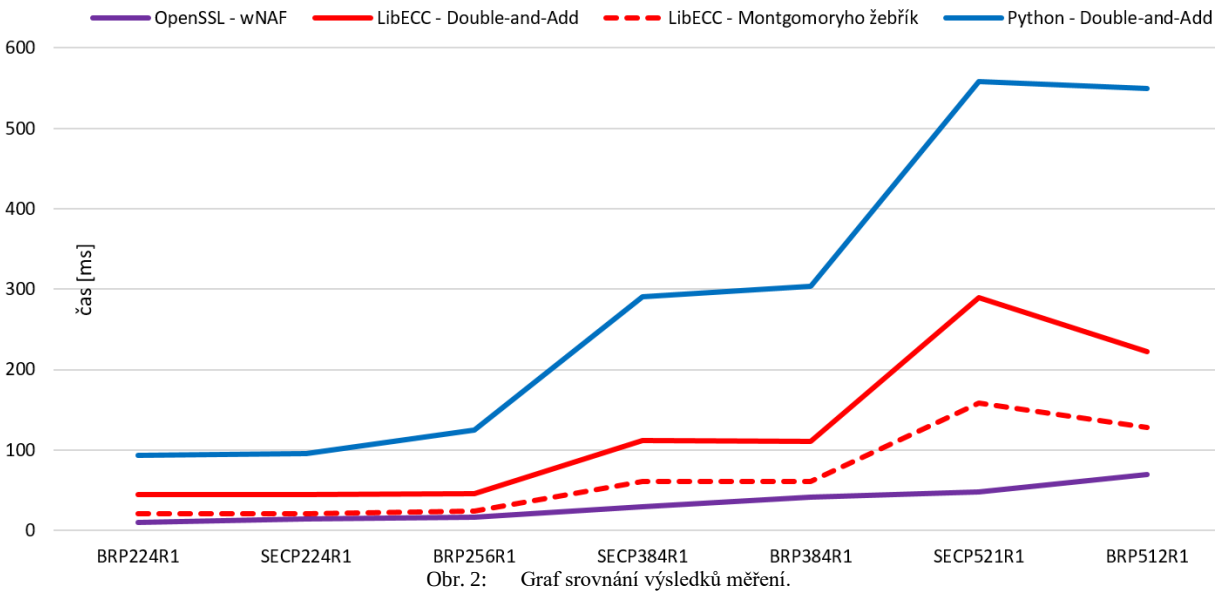
	RPi3	RPi2	RPiZero	RPi1
	čas[ms]	**čas[ms]**	**čas[ms]**	**čas[ms]**
SECP				
192K1	69,059	113,531	298,502	349,552
192R1	70,463	115,833	290,253	351,543
224K1	93,349	154,472	365,240	480,848
224R1	95,120	157,446	356,378	461,448
256K1	125,032	208,003	464,327	613,763
256R1	127,318	217,121	463,424	610,232
384R1	290,350	488,917	1210,727	1360,019
521R1	558,108	958,343	2184,007	2712,240
BRAINPOOL				
P160R1	47,907	82,888	221,820	294,222
P192R1	71,021	122,191	305,478	395,037
P224R1	93,448	157,934	359,223	541,504
P256R1	125,364	205,897	467,335	720,090
P320R1	194,608	335,035	733,409	1049,705
P384R1	303,914	491,724	1022,278	1452,071
P512R1	549,072	903,787	2131,401	2794,490

Z měření je patrná určitá závislost úměrně se zvyšujícím časovým nárokům na generování bodu na velikosti grupy dané křivky. Od nejméně časově náročných křivek standardu SECP, přes křivky standardu FRP, BRAINPOOL a nejvíce náročné křivky SECT.

SROVNÁNÍ VÝSLEDKŮ MĚŘENÍ

Tato kapitola se zabývá srovnáním naměřených hodnot násobení náhodně vygenerovaných bodů na eliptických křivkách standardů SECP a BRAINPOOL z důvodu kapacity článku a také z toho důvodu, že tyto dva standardy se podařilo implementovat ve všech knihovnách na všech měřených zařízeních. Zbývající zmíněné a naměřené standardy GHOST a FRP byly implementovány pouze v knihovně LibECC.

Data pro srovnání byla vybrána pouze z nejrychlejšího zařízení Raspberry Pi 3. V závěrečném srovnání výkonnosti byly použity metody skalárního násobení wNAF, což je primární a také nejrychlejší metoda použitá v knihovně OpenSSL. Metoda základního skalárního násobení Double-and-add a skalárního Montgomeryho násobení v knihovně LibECC. V knihovně Python byla použita metoda základního skalárního násobení Double-and-add. Výsledky tohoto srovnání jsou k dispozici na obr. 2.



Obr. 2: Graf srovnání výsledků měření.

Z grafu lze vyčíst, že násobení základní skalární metodou Double-and-add v knihovně Python je značně pomalejší oproti ostatním měřením. Jen o něco málo pomalejší výkon byl naměřen u knihovny LibECC s metodami skalárního násobení Double-and-add a Montgomeryho žebřík oproti knihovně OpenSSL. V knihovně LibECC je navíc patrno rychlejší násobení užitím skalárního násobení metodou Montgomeryho žebříku. Nejrychlejší je metoda skalárního násobení metodou wNAF, která je implementovány pouze v knihovně OpenSSL. Na základě provedeného rozsáhlejšího měření nad rámec publikovaných hodnot v tomto článku bylo naměřeno sčítání náhodně vygenerovaných bodů na eliptické křivce rychlejší v knihovně LibECC než tento úkon v knihovně OpenSSL [13].

ZÁVĚR

Tento článek se zabýval měřením generování náhodných bodů a aritmetických operací s body na eliptických křivkách. Tato měření byla implementována na hardwarové platformy Raspberry Pi Zero/1/2/3. Byla proměřena časová náročnost těchto operací na Raspberry Pi Zero/1/2/3. Celkem bylo proměřeno 48 eliptických křivek standardů NIST a BRAINPOOL, okrajově také GOST a FRP. Měřené metody násobení byly ty, které bylo možno do daných knihoven implementovat. V knihovně OpenSSL to byla metoda wNAF, v LibECC to byly metody Double-and-Add a nakonec v knihovně Python metoda Double-andAdd. Metoda srovnání knihoven byla zvolena pro jednoduchost následovně. Na nejvýkonnějším zařízení, tedy Raspberry Pi 3 a na nejnáročnější operaci – násobení náhodně vygenerovaných bodů na eliptických křivkách. Naměřeno bylo, že nejvýkonnější knihovnou je OpenSSL s nativní metodou násobení wNAF. Druhou nejvýkonnější knihovnou se, v našem měření, stala LibECC s nativními metodami násobení Double-and-Add a Montgomeryho žebřík. V knihovně LibECC je z těchto dvou metod rychlejší Double-and-Add. Nakonec byla proměřena knihovna Python s metodou násobení Double-and-Add, která se, mezi těmito knihovnami, umístila na posledním místě. Měřením nad rámec článku bylo dále zjištěno, že knihovna OpenSSL není nejrychlejší knihovnou ve všech měřených aritmetických operacích. Například při stejných podmínkách a měření sčítání v těchto knihovnách byla nejrychlejší knihovna LibECC. Konkrétně metoda sčítání Montgomeryho žebřík. Tato metoda byla rychlejší než Double-and-Add ve stejné knihovně.

LITERATURA

[1] MAO, W. Modern Cryptography: Theory and Practice.. 2004, ISBN 1-306-6943-1. [cit.2.10.2017].

[2] HANKERSON, D; MENEZES, A; VANSTONE, S.Guide to Elliptic Curve Cryptography. 2004, ISBN 978-0387952734 [cit.2.102017].

[3] ANOOP, M. ECC [online]. 2015, poslední aktualizace 2015 [cit.2.10.2017]. Dostupné z URL: <https://www.johannes-bauer.com/compsci/ecc>.

[4] JOHNSTON, O. Elliptic Curve Cryptosystems [online]. 2010, poslední aktualizace 2010 [cit.2.10.2017]. Dostupné z URL: <https://eprint.iacr.org/ 2010/575.pdf>.

[5] BOS, W; HALDERMAN, A; HENINGER, N. Elliptic Curve Cryptography in Practice [online]. 2013, poslední aktualizace 2013 [cit.2.10.2017]. Dostupné z URL: <https://eprint.iacr.org/2013/ 734.pdf>.

[6] Raspberry Pi 3 [online]. 2016, poslední aktualizace 2016 [cit. 23. 10. 2017]. Dostupné z URL: <http://rpishop.cz/kategorie/283-raspberry-pi-3-model-b-64-bit.html>.

[7] OpenSSL library [online]. 2017, poslední aktualizace 2017 [cit.9.10.2017]. Dostupné z URL: <https://github.com/openssl/openssl>.

[8] Libecc library [online]. 2017, poslední aktualizace 2017 [cit.14.3.2018]. Dostupné z URL: <https://github.com/ANSSI-FR/libecc>.

[9] SEC 2: Recommended Elliptic Curve Domain Parameters [online]. 2010, poslední aktualizace 2010 [cit.3.3.2018]. Dostupné z URL: <http://www.secg.org/sec2-v2.pdf>.

[10] Brainpool Standard Curves and Curve Generation [online]. 2010, poslední aktualizace 2017 [cit.3.3.2018]. Dostupné z URL: <https://tools.ietf.org/html/ rfc5639>.

[11] GNU. Elliptic curve arithmetic [online]. 2017, poslední aktualizace 2017 [cit.2.10.2017]. Dostupné z URL: <https://rosettacode.org/wiki/Elliptic_curve_arithmetic>.

[12] MENEZES, A; OORSCHOT P, VANSTONE S. Handbook of Applied Cryptography. 1997, CRC Press [cit.2.10.2017]. Dostupné z URL: <https://zygoteiot.wordpress.com/2017/03/16/ a primer-to-elliptic-curve-diffie-hellman>.

[13] VUT, BRNO. Proceedings of the 24th Conference STUDENT EEICT 2018 [online]. 2018, poslední aktualizace 2018, ISBN 978-80-214-5614-3.[cit.19.4.2018].

[14] FUJDIAK, R. Analýza a optimalizace datové komunikace pro telemetrické systémy v energetice .2017 [cit.8.12.2017]. Dostupné z URL: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=157492>.