VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Patrik Burda

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

MOŽNOSTI POSTKVANTOVÉ KRYPTOGRAFIE
POSSIBILITIES OF POST-QUANTUM CRYPTOGRAPHY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Patrik Burda**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Petr Dzurenda**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Student: Patrik Burda ***ID:*** 195833
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Možnosti postkvantové kryptografie

POKYNY PRO VYPRACOVÁNÍ:

Student se seznámí s možnostmi a principy post-kvantové kryptografie. Student provede srovnání existujících kryptografických schémat v závislosti na výpočetní a paměťové složitosti. Výstupem bakalářské práce bude kryptografická knihovna pro benchmarkové testy post-kvantových kryptografických schémat. Student dále implementuje vybrané schéma na výkonově omezeném zařízení např. na čipové kartě.

DOPORUČENÁ LITERATURA:

[1] BUCHMANN, Johannes, et al. Post-quantum cryptography: lattice signatures. Computing, 2009, 85.1: 105-125.

[2] CHEN, Lily, et al. Report on post-quantum cryptography. US Department of Commerce, National Institute of Standards and Technology, 2016.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: Ing. Petr Dzurenda
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

S rychlostí jakou se vyvíjejí technologie je potenciální hrozbou vývoj kvantových počítačů, které mají ohrozit veškerou dnešní bezpečnou komunikaci a je potřeba hledat způsoby jak této hrozbě čelit. Tato práce se zabývá rozborem kryptografických schémat odolných vůči dnešním známým kvantovým útokům jako je například Shorův algoritmus. Dále práce obsahuje měření a vyhodnocení rychlosti a paměťové náročnosti. Na základě měření pak vybírá vhodné postkvantové schéma pro implementaci na čipovou kartu, na které poté tuto implementaci realizuje.

KLÍČOVÁ SLOVA

postkvantová kryptografie, Raspberry Pi 3 Model B, šifrovací schéma, podpisové schéma, New Hope, NTRU, NIST, knihovna měření postkvantových schémat, postkvantová kryptografie na IoT

ABSTRACT

With rapidly evolving technologies and a potential threat of quantum computers that could break all of today's secure communication, there is a need for ways to deal with this threat. This work deals with the analysis of cryptographic schemes resistant to today's known quantum attacks such as Shore's algorithm. The work also includes measurement and evaluation of speed and memory usage. Based on the measurement a suitable postquantum scheme is then implemented on smart card.

KEYWORDS

postquantum cryptography, Raspberry Pi 3 Model B, encryption algorithms, signature algorithms, New Hope, NTRU, NIST, library for postquantum algorithms, postquantum cryptography on IoT

BURDA, Patrik. *Možnosti postkvantové kryptografie*. Brno, 2019, 38 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Petr Dzurenda,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Možnosti postkvantové kryptografie“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Petru Dzurendovi za odborné vedení, trpělivost, za pomoc a rady při zpracování této práce.

Brno ....

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

S neustále se zrychlujícím vývojem technologií je hrozba funkčních kvantových počítačů, které by dokázaly využít například Shorova algoritmu velice aktuální. Proto je dnes zájem o postkvantovou kryptografii jak od vládních agentur, tak od soukromých firem veliký. Stále více odborníků se věnuje právě tomuto tématu a snaží se neustále přicházet s novějšími algoritmy, které by byly odolné proti kvantovým počítačům. Dnes již existuje celá řada postkvantových schémat, které by měly útokům kvantových počítačů odolat. Využívají různé druhy kryptografie a matematických problémů. Bohužel tyto algoritmy nejsou optimalizované a použitelné na omezených zařízeních což s rostoucím trhem *Internet of Things* (IoT) může být velké bezpečnostní riziko.

Cílem této práce je vyhodnotit aktuální stav a možnosti využití postkvantové kryptografie na omezených zařízeních a to jak šifrovacích tak podpisových schémat. Kritérii použitelnosti budou především rychlost a paměťová náročnost, která budou měřena na Raspberry Pi 3 Model B. Na základě těchto dat potom bude vybrán kandidát pro implementaci na omezené zařízení a to konkrétně na čipovou kartu Multos.

V první části práce vysvětlí pojmy jako kvantový počítač, postkvantová kryptografie a rozebere jednotlivé druhy kryptografie odolné vůči kvantovým počítačům. Představí *National Institute of Standards and Technology* (NIST) a jeho „soutěž“ postkvantových schémat a ve druhé části popíše vytvoření knihovny a její následné použití pro měření rychlosti a paměťové náročnosti schémat. Na závěr pak na základě naměřených hodnot vybere jedno postkvantové schéma, které implementuje na čipovou kartu.

1 Kvantový počítač

První zmínky o možném využití kvantových jevů k bezpečnému přenosu informace jsou již z šedesátých let minulého století. Ovšem až o dvacet let později přišli Richard Feynman a Yuri Manin s myšlenkou kvantového počítače, který by byl schopen dokonale simulovat kvantové systémy. Kvantový počítač využívá jako základní jednotku qubit. Qubit je narozdíl od bitu, který je buď 0 nebo 1, superpozice pravděpodobností mezi 0 a 1, to znamená, že jeho hodnota leží na spektru a až měřením hodnota zkolabuje do hodnoty 0 nebo 1. K popsání stavu jednoho qubitu potřebujeme dva bity, což znamená, že množství obsažené informace v qubitech roste exponenciálně. Například 10 qubitů již obsahuje 2^{10} bitů informace, avšak z Holevova teorému vyplývá, že množství informace, kterou po změření můžeme dostat z n qubitů nepřesáhne n bitů [1]. Proto aby mohlo být využito výhod kvantového počítače je potřeba použít specifických algoritmů, které řeší daný problém bez nutnosti měření v jeho průběhu. Dva nejznámnější jsou Groverův algoritmus [2] a Shorův algoritmus [3], který způsobil rozruch v oblasti kryptografie. Právě nutnost speciálně navržených algoritmů je důvod proč kvantové počítače nejspíše nikdy nenahradí ty klasické.

Historicky první demonstrace kvantového algoritmu byla v roce 1998, kdy byl použit *Nuclear Magnetic Resonance* (NMR) kvantový počítač [4]. Prvním komerčním kvantovým počítačem se stal v roce 2011 stroj od firmy D-Wave [5], který disponoval 14 qubitovým registrem. Dnes již tato firma dodává kvantové počítače pro *National Aeronautics and Space Administration* (NASA) a Google a používá 2000 qubitů. Naproti tomu společnosti IBM a Google oznámily příchod kvantových počítačů, které mají používat 50 až 72 qubitů. Rozdíl v počtu qubitů používaných v kvantových počítačích těchto firem je způsoben používáním rozdílných technologií. Firma D-Wave využívá tzv. kvantové žíhání, což je způsob jak využít kvantové fyziky k řešení určitých typů úloh, kterým se říká optimalizační problémy. Na tomto počítači však nemůže být použito například Shorova algoritmu. K využití všech výhod kvantového počítače potřebujeme tzv. kvantová hradla. Google oznámil kvantový procesor Bristlecone využívající právě kvantová hradla v roce 2018 [6]. Tento procesor využívá 72 qubitů a pokud se podaří snížit chybovost tohoto systému dostatečně nízko mohlo by se jednat o první kvantový procesor, který v určité úloze překoná dnešní superpočítače.

Chybovost je jeden z hlavních problémů dnešních kvantových počítačů. Kvůli potřebě extrémního chlazení až na tisíciny Kelvinů, nutnosti stínění před magnetickým polem a dalšími vnějšími faktory jsou dnešní kvantové počítače mimo vědecké experimenty nepoužitelné. Spousta vědců se také domnívá, že se o dnešních strojích nedá jako o kvantových počítačích hovořit a že kvantové počítače nikdy sestrojeny nebudou.

2 Postkvantová kryptografie

Postkvantová kryptografie se zabývá kryptografickými schématy, které jsou odolné vůči kvantovým počítačům. Konkrétněji vůči dnes již známým kvantovým algoritmům jako jsou Groverův algoritmus nebo Shorův algoritmus. To ovšem neznamená, že se nemohou objevit další algoritmy pro kvantové počítače, které budou ještě účinnější. Groverův algoritmus nabízí kvadratické zrychlení v případě symetrických šifer jako AES a DES. Řešením by tedy bylo zdvojnásobení velikosti klíčů symetrických šifer. Velkým problémem je však Shorův algoritmus, díky kterému lze jak faktorizovat velká čísla, tak řešit problém diskrétního logaritmu v polynomiálním čase. To znamená, že dnešní nejpoužívanější asymetrické šifry jako RSA, *Digital Signature Algorithm* (DSA), *Elliptic Curve Diffie Hellman* (ECDH) a *Elliptic Curve Digital Signature Algorithm* (ECDSA) jsou v ohrožení [7].

Jako u většiny technologií není otázka jestli, ale kdy. V případě kvantových počítačů se nejčastěji uvádí rozmezí 10 až 20 let. I když se to může zdát jako dlouhá doba, je třeba myslet na to, že již dnes je potřeba určitá citlivá data zabezpečit a uschovat na 5 i více let. Proto je třeba již dnes řešit postkvantovou kryptografii a hledat schémata, která těmto kvantovým útokům odolají. Nabízí se několik typů kryptografie, které se používají nejen v postkvantové kryptografii.

2.1 Kryptografie založená na teorii kódování

Teorie kódování se zabývá konstrukcemi kódu a studiem jejich vlastností. Hlavní z těchto vlastností je bezpečný přenos zpráv. Pokud by se během přenosu stala chyba měl by algoritmus chybu nalézt a opravit. Takto funguje například Hammingův kód [8]. Schémata založená na teorii kódování, která jsou odolná proti útokům kvantových počítačů využívají takzvané Goppa kódy. Goppa kód je lineární, samoopravný kód, který může být použit k zašifrování a dešifrování zprávy. Možná vyšší rychlost kryptosystému využívajících tohoto způsobu je vykoupená velikostí klíčů, což je činí nepraktickými nebo dokonce nepoužitelnými na zařízeních s omezenou pamětí. Nejznámější kryptosystémy jsou McEliece [9] a Niederreiter [10], který z McEliece vychází jen zvyšuje rychlost šifrování.

2.2 Kryptografie založená na mřížkách

Kryptografie založená na mřížkách je relativě novou oblastí výzkumu, která začala prací M. Ajtaiho a Cynthie Dwork [11], jenž jako první využil problémů spojených s

mřížkami k sestrojení kryptografického systému. Jak již název napovídá tato kryptografie využívá mřížky což si můžeme představit jako vektorový prostor s omezením násobení vektorů v mřížce celými čísly.

Mřížky využívají dva základní problémy. *Shortest Vector Problem* (SVP) a *Closest Vector Problem* (CVP).

SVP je hlavním problémem, který se týká mřížek. Při řešení tohoto problému máme mřížku L a bázi a cílem je najít nejkratší nenulový vektor, který patří do mřížky. Složitost tohoto problému spočívá v tom, že jedna mřížka má spoustu různých bází a že dostáváme mřížku zadanou pomocí báze, která obsahuje vektory mnohem delší než je nejkratší nenulový vektor.

CVP obecně využívanější problém spojený s mřížkami je hledání nejbližšího bodu mřížky. Pokud máme terč (terčový vektor) a mřížku ve stejném prostoru R^n, pak máme najít takový bod mřížky, který je nejblíž k zadanému terči [12]. Mezi známější kryptosystémy založené na mřížkách patří například New Hope [13] nebo NTRU [14].

Na SVP a CVP jsou postaveny další dva matematické problémy a to *Learning With Errors* (LWE) a *Ring Learning With Errors* (RLWE).

LWE je robustní kryptografická metoda. Na začátku vytvoříme matici s a e. Poté si vezmeme matici náhodných čísel A a spočítáme

$$B = As + e$$

Matice s bude soukromý klíč a matice A a B budou veřejným klíčem. Regev ukázal, že LWE problém je složitý jako nejsložitější problémy mřížek [15].

RLWE pracuje s polynomiálními okruhy namísto reálných čísel. Tímto se snižuje výpočetní náročnost. I když jsou klíče RLWE značně větší než například u RSA jsou velikostně odmocninou velikosti klíčů LWE [16].

2.3 Kryptografie založená na hashovacích funkcích

Hashovací funkce je algoritmus pro převod vstupních dat libovolné délky na řetězec konstantní délky. Tento proces je jednosměrný to znamená, že je jednoduché ze vstupních dat dostat výsledný řetězec, ale prakticky nemožné z hashe dostat zpět původní data.

Hashovací funkce se využívají například k rychlému vyhledávání v tabulkách nebo v kryptografii na vytváření a ověřování elektronického podpisu. Při používání hashovacích funkcí je třeba myslet na to, že se mohou vyskytnout kolize. To znamená, že pro různá vstupní data budeme mít stejný výsledný řetězec. Těmto kolizím se nelze vyhnout, protože možností vstupních dat je teoreticky nekonečno, ale možnosti výstupního řetězce jsou omezeny. Co se týče bezpečnosti proti kvantovým

počítačům tak třeba při použití SHA-2/3 nám bude stačit použití větších výstupních řetězců stejně jako u symetrické kryptografie by nám zatím stačilo zvětšit klíče. Kryptosystémy využívající hashovacích funkcí odolné vůči kvantovým počítačům jsou například SPHINCS+ [17] nebo Picnic [18].

2.4 Kryptografie založená na polynomiálních rovnicích

Tato kryptografie využívá algebraickou geometrii. Obtížnost problému spočívá v řešení soustavy rovnic o více neznámých nad konečným polem [19].

Základem bezpečnosti kryptografie založené na polynomech je problém výpočtu vícerozměrných kvadratických rovnic. Tento problém, někdy nazýván *Multivariate Quadratic* (MQ) je NP-úplný [20]. Kryptosystém využívající problémy spojené s polynomy je například MQDSS [21].

2.5 Kryptografie založená na supersingulárních eliptických křivkách

Potřeba využívat supersingulární eliptické křivky vychází z premisy, že příchod kvantových počítačů s dostatečným výkonem bude znamenat konec algoritmům založených na eliptických křivkách.

Hlavním rozdílem od ECDH je to, že postkvantové protokoly využívající supersingulární křivky mají izogenní tajné klíče. To znamená, že body jedné křivky zobrazuje do druhé se zachováním vrcholů. Veřejným klíčem je pak samotná supersingulární křivka [22]. Na stránkách NIST najdeme pouze jeden postkvantový kryptosystém využívající tuto metodu a to kryptosystém SIKE [23].

3 NIST

NIST byl založen v roce 1901 a je jednou z nejstarších fyzikálních laboratoří v zemi. Cílem instituce je podpora inovací a zlepšování konkurenceschopnosti USA.

NIST inicioval proces vyhodnocení a standardizace jednoho nebo více kvantově odolných asymetrických algoritmů. Záměrem je, aby nové standardy asymetrického šifrování specifikovaly algoritmy, které budou dostupné na celém světě, a které budou schopny chránit citlivé vládní informace v dohledné budoucnosti i po příchodu kvantových počítačů. Jako první krok NIST požádal o veřejný názor na návrh minimálních požadavků a hodnotících kritérií pro kandidátské algoritmy [24].

Základním kritériem je bezpečnost, pro kterou NIST ustanovil pět stupňů. Kryptosystém splňuje bezpečnost určité úrovně pokud útok na tento kryptosystém je časově alespoň tak náročný jako na kryptosystém uvedený v následující tabulce a to jak na klasickém, tak na kvantovém počítači. NIST požádal přispěvatele, aby se soustředili především na úrovně 1, 3 a 5.

Tab. 3.1: Stupně bezpečnosti podle NIST

Úroveň	**Algoritmus**
1	AES128
2	SHA256/SHA3-256
3	AES192
4	SHA384/SHA3-384
5	AES256

3.1 První kolo NIST soutěže

První kolo probíhalo od prosince 2017 do ledna 2019. V tomto kole se přihlásilo 82 kandidátů z toho 69 jich bylo přijato jako „úplných a správných“ (5 odstoupilo) [25].

Do druhého kola prošlo 26 algoritmů z toho 17 šifrovacích a 9 podpisových. V srpnu 2019 se koná druhá konference o postkvantové standardizaci. Ukončení druhého kola je plánováno na rok 2020/2021 [26].

Tab. 3.2: Rozdělení kandidátů prvního kola

	Podpisy	**Ustanovení klíčů/Šifrování**	**Celkem**
Mřížky	5	21	26
Kódy	2	17	19
Polynomy	7	2	9
Hashe	3	0	3
Ostatní	2	5	7
Celkem	19	45	64

3.2 Stručné zhodnocení kandidátů druhého kola

3.2.1 Šifrování [mřížky]

Crystals-Kyber Jednoduché rozšíření, nenáročný na výkon
FrodoKEM Větší klíče, náročnější na výkon
LAC Nenáročný na výkon, parametry úrovně 5 nefungují
NewHope Nenáročný na výkon
NTRU Prověřen časem, lepší struktura mřížek
NTRU Prime Nenáročný na výkon, pouze parametry úrovně 5
Round5 Nenáročný na výkon a paměť, problém s chybovostí
Saber Nenáročný na výkon a paměť
ThreeBears Rychlá aritmetika, novější bezpečnostní předpoklad

3.2.2 Šifrování [kódy]

Classic McEliece Velké množství analýz, velmi velké veřejné klíče
NTS-KEM Velice podobný McEliece jen s designovými úpravami
BIKE Náročnost na výkon a paměť srovnatelná s mřížkovými algoritmy
HQC Nízká chybovost, větší klíče a šifrované zprávy

3.2.3 Šifrování [kódy a křivky]

LEDAcrypt Potřeba hlubší analýzy
Rollo Spojení 3 algoritmů, novější bezpečnostní předpoklad
RQC Žádná chybovost, náročnější na výkon
SIKE Velmi malé klíče, velmi náročné na výkon

3.2.4 Podpis [mřížky]

Crystals-Dilithium	Nenáročný na výkon
Falcon	Nenáročný na výkon, komplikovaný na implementaci
qTesla	Nenáročný na výkon, potřeba hlubší analýza

3.2.5 Podpis [hashe]

Sphincs+	Malé veřejné klíče, ale velké podpisy, pomalejší podepisování
Picnic	Malé veřejné klíče, ale velké podpisy, náročnější na výkon

3.2.6 Podpis [polynomy]

GeMSS	Velmi malé podpisy, náročnější na výkon
LUOV	Malé velikosti klíčů, potřeba hlubší analýza
MQDSS	Malé veřejné klíče, ale velké podpisy, potřeba lepší optimalizace
Rainbow	Dobře prověřeno, potřeba lepší implementace

4 Měření postkvantových schémat

V této kapitole se práce bude zabývat měřením jednotlivých schémat z druhého kola „soutěže“ NIST [27] jak z hlediska výkonové náročnosti (rychlosti), tak z hlediska paměťové náročnosti.

Testování proběhlo na notebooku s 8GB RAM a Intel procesorem i5-6200U 2.30GHz a na Raspberry Pi 3 Model B s 1GB RAM a 1.2GHz ARMv8 procesorem. Na notebooku byl používán operační systém Ubuntu verze 18.10 a na Raspberry byly vyzkoušeny dva systémy. První byl oficiální 32bitový systém Raspbian stretch with desktop [28] a druhý 64bitový byl neoficiální a založený na Gentoo [29].

4.1 Knihovna

Pro testování velkého množství postkvantových schémat na více zařízeních, bylo potřeba vytvořit knihovnu, která byla schopná měření zautomatizovat.

První bylo potřeba si na stránce NIST [30] stáhnout archivy všech postkvantových schémat. Každý z těchto archivů musí obsahovat referenční implementaci a dokumentaci. Některé z nich také obsahují různé varianty optimalizovaných implementací popřípadě vlastní testy výpočetní náročnosti. Jelikož ve většině případu se referenční implementace shoduje s optimalizovanou bylo výhodnější použít právě referenční implementaci, protože u některých schémat nebylo možné optimalizované verze zprovoznit a to jak na notebooku tak na Raspberry. Bohužel z důvodu nedostatečných nebo někdy i neexistujících návodů na zprovoznění, nebylo možné testovat některá schémata. Seznam testovaných schémat spolu s funkční verzí knihovny, stejně tak další potřebné knihovny, se nachází jako veřejný repositář na bitbucketu [31].

4.1.1 Potřebný software

Ke zprovoznění každého postkvantového schématu je potřeba jedna nebo více knihoven, které slouží buď ke zkompilování jednotlivých schémat, měření rychlosti, paměťové náročnosti nebo samotným matematickým operacím používaných ve schématu.

Nejznámější používanou knihovnou v kryptografii je OpenSSL, ze které se ve výše zmíněných schématech využívá prakticky jen šifrování pomocí AES. Tato funkce se v tomto případě používá pro rozšíření bloků seedu [32], který se dále používá v generátoru pseudonáhodných čísel. V praxi však tato funkčnost musí být nahrazena ideálně reálným fyzickým generátorem náhodných čísel. Dále se používá například knihovna NTL, která obsahuje datové struktury a algoritmy pro počítání s čísly, vektory, maticemi a polynomiálními rovnicemi libovolné délky [33]. Další ze známějších jsou

GMP [34], Keccak [35] nebo Valgrind [36]. Na zkompilování samotného programu potřebujeme nainstalovat balíčky jako make, gcc a g++.

4.1.2 Funkce knihovny

Knihovnu lze po stažení [31] zprovoznit pouhým spuštěním souboru install32 či install64 podle verze systému. Následně se stáhnou a nainstalují potřebné balíčky. Na závěr se zkompiluje i samotná knihovna do výsledného souboru run. Po spuštění v konzoli knihovna nabízí možnost vypsání všech schémat dostupných pro měření, měření rychlosti průběhu, měření paměťové náročnosti nebo otevření dokumentace k příslušnému schématu. Menu knihovny se prochází pomocí zadávání čísel před jednotlivými možnostmi.

Knihovna nabízí možnost měření všech schémat najednou či jednoho konkrétního schématu a jeho verzí s libovolným počtem průběhů přičemž výsledné hodnoty jsou průměr jednoho průběhu ve vteřinách. Dále je možnost měřit paměťovou náročnost pomocí již zmíněné knihovny Valgrind. V tomto případě lze však měřit pouze jednu verzi vybraného schématu, jelikož výpis z knihovny Valgrind do konzole je při větším počtu testů nepřehledný.

V následujícím výpisu lze vidět zjednodušený kód hlavního souboru schémat. Tato ukázka se liší pouze v tom, že pro přehlednost neobsahuje podmínky, které ověřují zda se funkce provedly správně. Názvy funkcí pro generování klíčů, šifrování a dešifrování stanovil NIST v souboru api.h, kterou obsahuje každé schéma. Stejně tak byl přiložen hlavní soubor pro *Known Answer Test* (KAT) s počtem průběhu nastaveným na 100. Některé implementace schémat přesto nedodržely jednodnou strukturu složek a obsah hlavního souboru a proto bylo potřeba specifických úprav kódu.

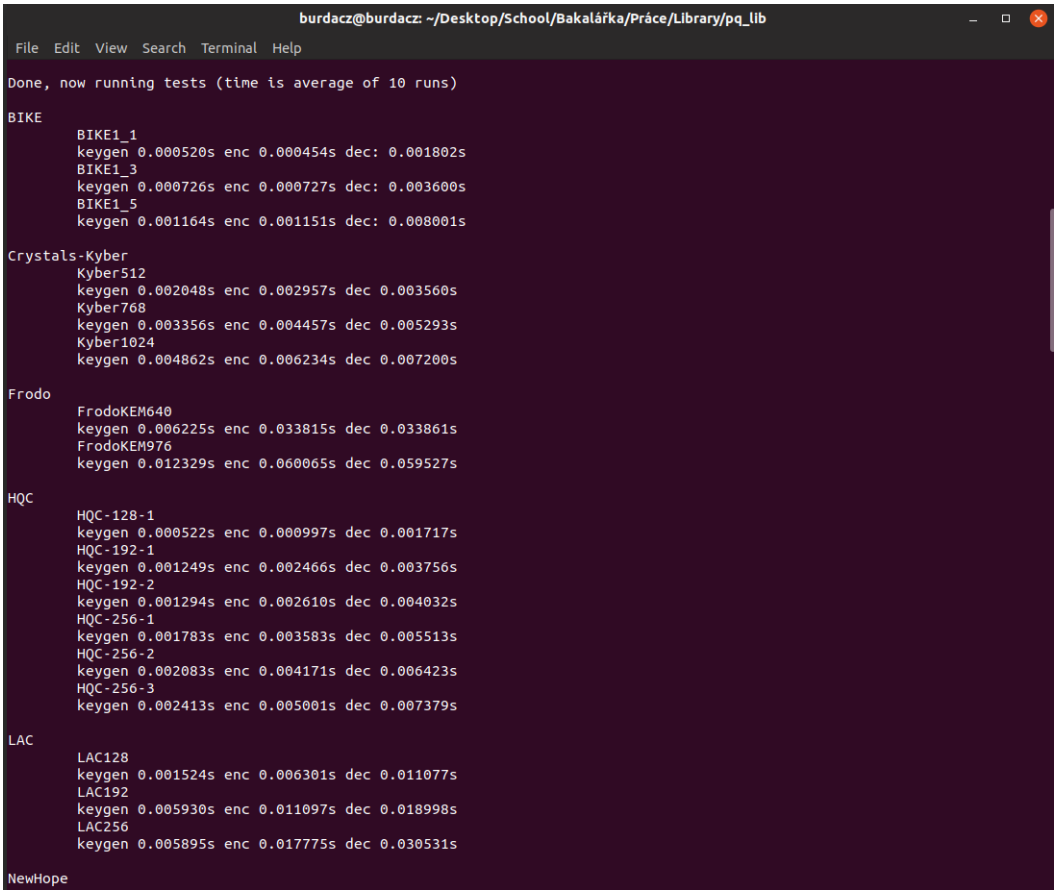
Výpis 4.1: Zjednodušený kód hlavního souboru pro měření rychlosti průběhu jednoho ze schémat v jazyce C.

```
do {
// Uložení počtu tiků hodin od spuštění programu
start = clock();
// Generování veřejného (pk) a soukromého klíče (sk)
crypto_kem_keypair(pk, sk);
// Uložení aktuálního počtu tiků
end = clock();  total_keygen += (end-start); start = clock();

// Šifrování zprávy (ss) veřejným klíčem (pk)
// a uložení výsledku (ct)
crypto_kem_enc(ct, ss, pk);
end = clock();  total_enc += (end-start); start = clock();

// Dešifrování (ct) soukromým klíčem (sk)
// a uložení výsledku (ss1)
crypto_kem_dec(ss1, ct, sk);
end = clock();  total_dec += (end-start);

// Porovnání původní zprávy (ss) a dešifrované zprávy (ss1)
memcmp(ss, ss1, CRYPTO_BYTES);
// Konec do-while cyklu (počet průběhů)
} while(!done);
// Výpis výsledků
printf("\tkeygen %fs enc %fs dec %fs",
  (double)total_keygen/CLOCKS_PER_SEC/runs,
  (double)total_enc/CLOCKS_PER_SEC/runs,
  (double)total_dec/CLOCKS_PER_SEC/runs);
```

Jak je vidět ve výpisu výše, základní funkce byly „obaleny“ funkcí clock() z knihovny time. Tato funkce umožňuje získat procesorový čas, který daný program potřeboval. Rozdílem hodnot funkce ve dvou částech kódu získáme čas, který skonzumovala daná funkce. V kompletním kódu, který lze nalézt v příloze je také zapisování a načítání ze souboru, kdy se první do jednoho souboru zapíší hodnoty seedů a poté se ze souboru tyto seedy načítají a používají se do pseudonáhodného generátoru čísel.

4.1.3 Měření výkonové náročnosti

Pomocí vytvořené knihovny mohou být měřena všechna schémata najednou. Při počtu průběhů více než 100 může měření trvat i na běžném počítači v řádu desítek či dokonce stovek minut. Proto při měření na výkonově omezených zařízeních volíme maximálně 100 průběhů a u náročnějších schémat, kdy i 1 průběh může trvat vyšší desítky vteřin volíme maximálně 10 průběhů.



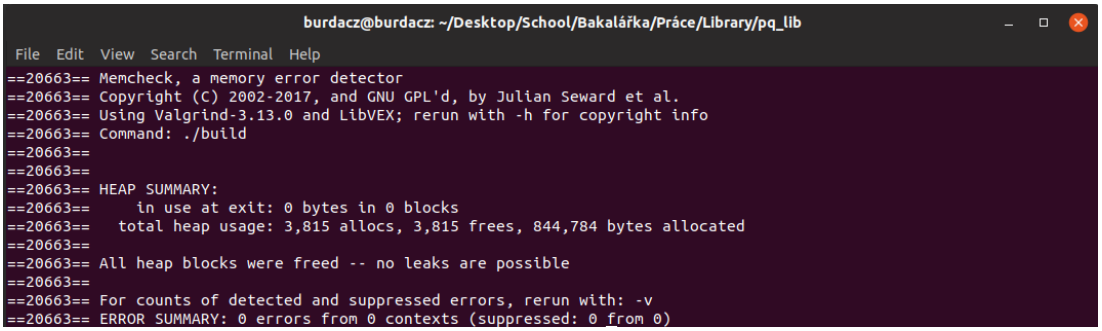
Obr. 4.1: Měření rychlosti průběhu jednotlivých schémat.

Na uvedeném obrázku můžeme vidět postupný výpis názvu schématu, konkrétní implementace a jednotlivé časy průběhů. První hodnota udává dobu generování dvojice klíčů, druhá a třetí hodnota na řádku pak ukazuje dobu potřebnou pro šifrování a dešifrování zprávy těmito klíči.

4.1.4 Měření paměťové náročnosti

Nejen pro měření paměťové náročnosti programů obecně, ale i pro kontrolu neinicializované paměti, čtení či zápis do paměti po jejím uvolnění nebo i špatné použití

příkazu malloc je Valgrind snad jediným spolehlivým nástrojem pro Linux. Jeho použití je velice jednoduché, pro základní výpis stačí pouze zavolat valgrind s cestou k programu.

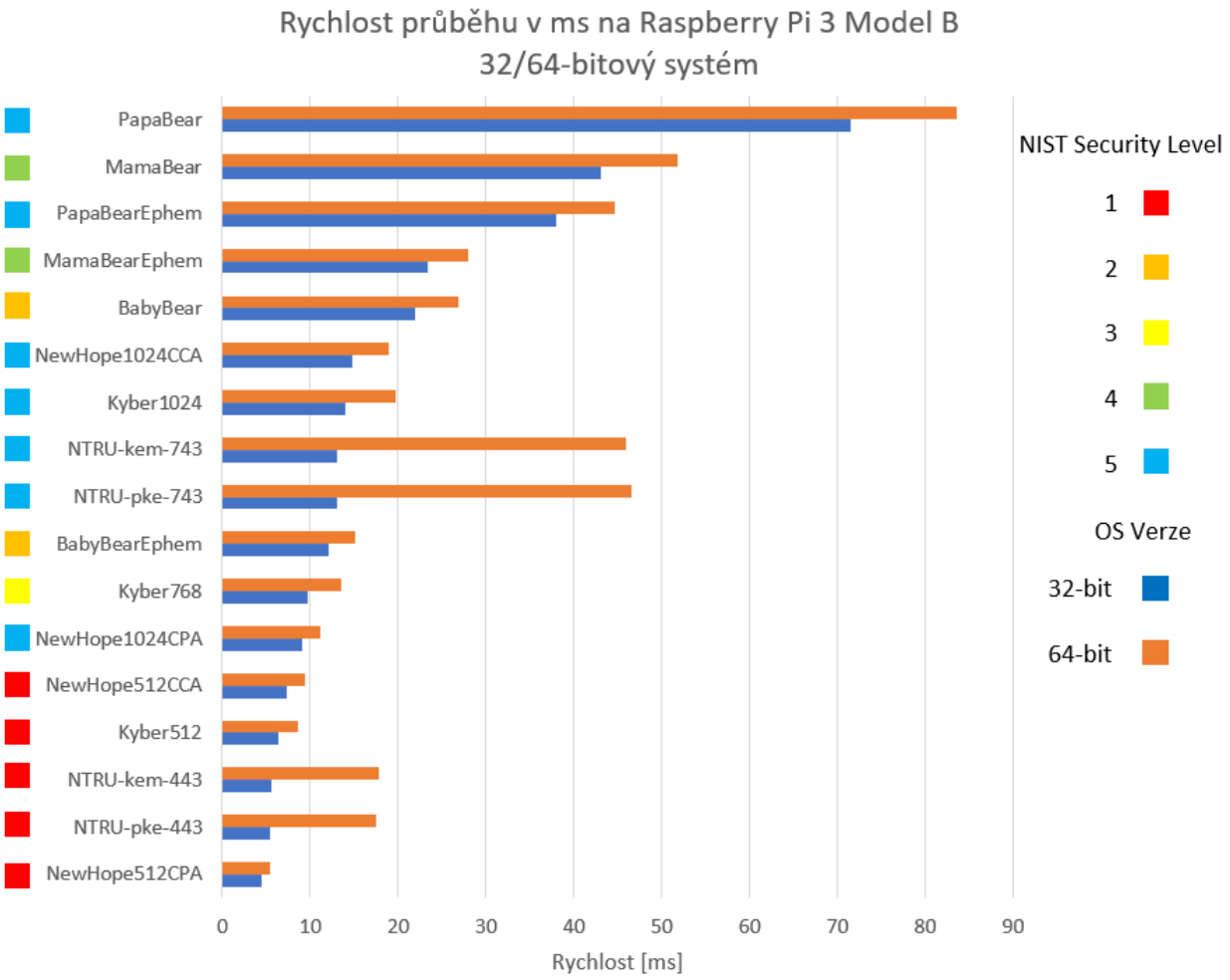


Obr. 4.2: Měření paměťové náročnosti.

Na obrázku výše je výpis měření z knihovny Valgrind jedné verze postkvantového schématu. Jelikož Valgrind neobjevil žádné paměťové chyby a všechno alokované místo bylo po ukončení programu uvolněno je ve výpisu pouze počet alokací a počet alokovaných bytů, tady konkrétně 3,815 alokací a 844,784 bytů.

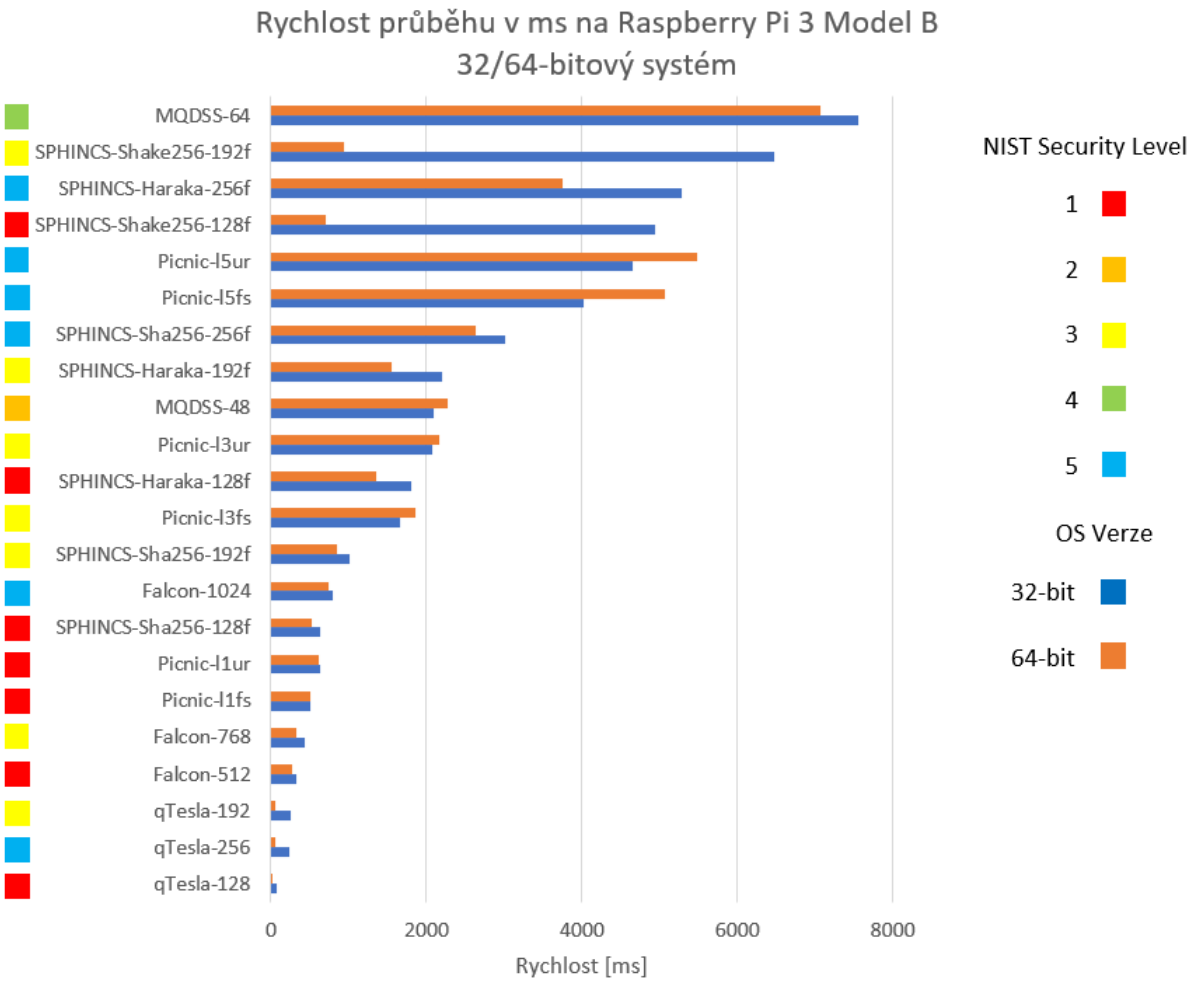
4.2 Výsledky měření

Pro zobrazení výsledků je použito čtyř grafů. Dva obsahující rychlosti průběhů a dva s množstvím alokovaných bytů šifrovacích a podpisových postkvantových schémat. Kvůli měření na dvou velice výkonově odlišných zařízeních jsou v následujících grafech z praktických důvodů uvedeny hodnoty pouze z měření na Raspberry s 32bitovým a 64bitovým systémem. V grafech obsahujících počty alokovaných bytů se jedná pouze o měření na dříve specifikovaném notebooku, jelikož tato data budou napříč zařízeními shodná. Přesná data z měření na obou zařízeních naleznete v příloze.



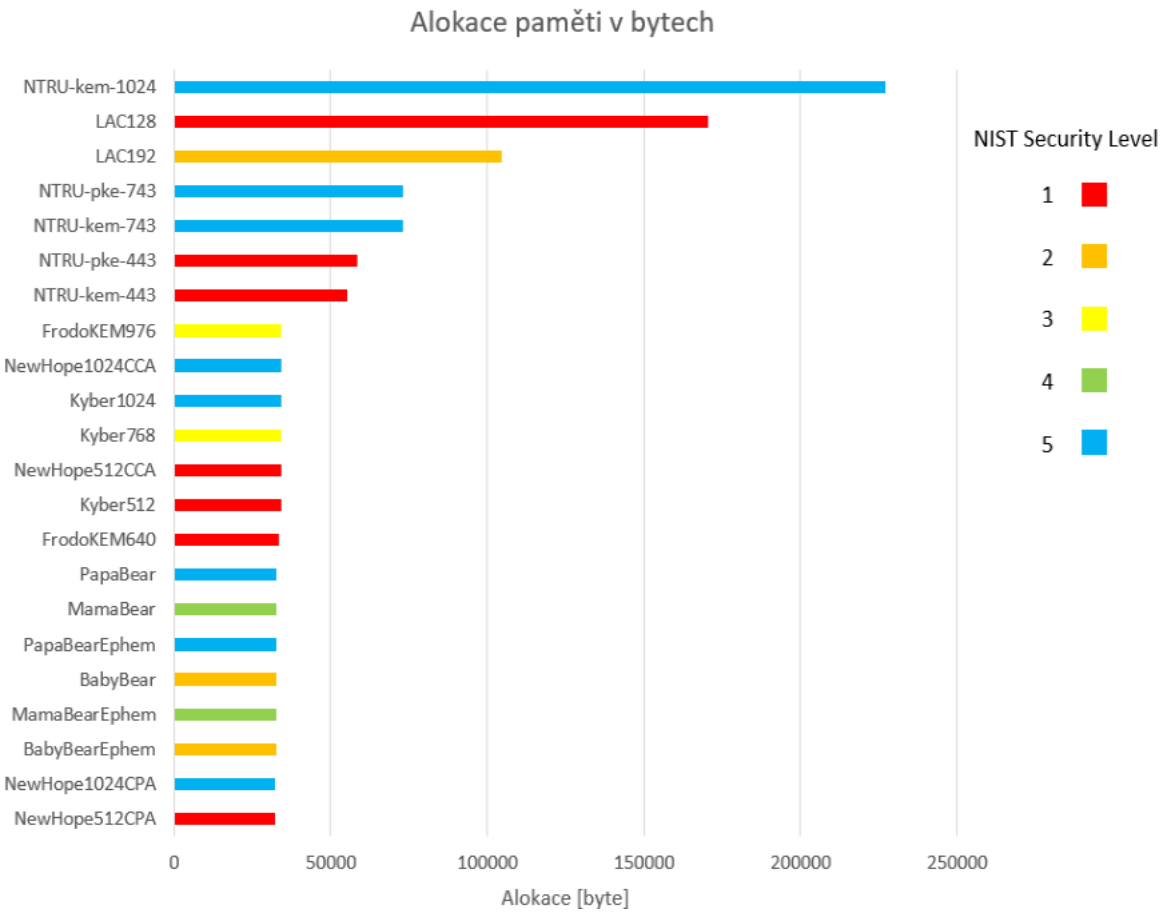
Obr. 4.3: Rychlosti průběhů šifrovacích schémat

Na tomto grafu 17 vybraných nejrychlejších šifrovacích schémat lze vidět, že 64bitová verze systému na Raspberry je v případě šifrovacích schémat pomalejší. Všechna tato schémata jsou založena na matematických problémech mřížek, které by měly poskytovat nejlepší rychlost šifrování navzdory někdy poněkud velkým klíčům. I když se dalo očekávat, že nejrychlejšími schématy budou ty s nejnižšími úrovněmi bezpečnosti je na grafu vidět, že třeba v případě NewHope1024CPA tomu tak není.



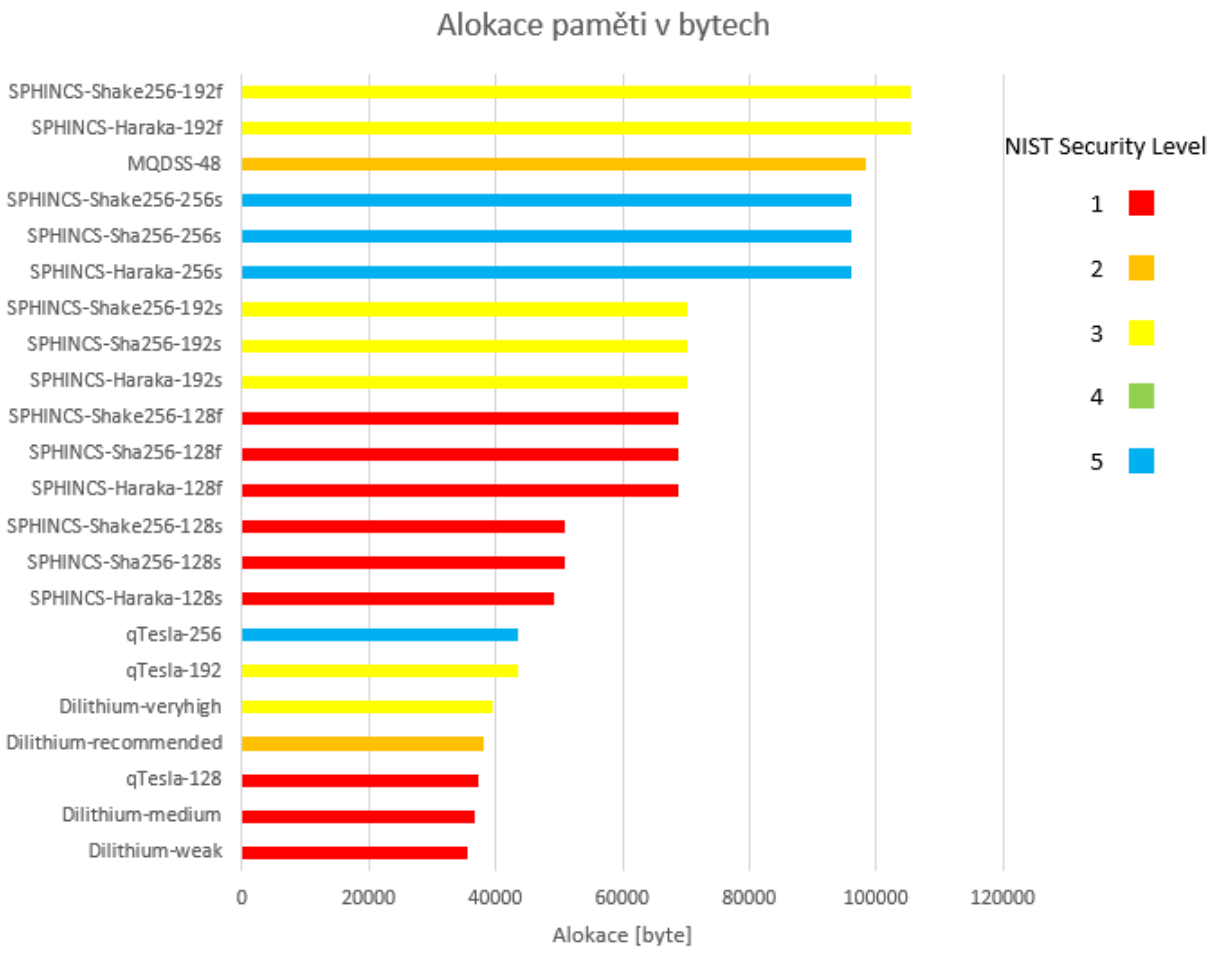
Obr. 4.4: Rychlosti průběhu podpisových schémat

V této kategorii by jednoznačně vyhrálo schéma Crystals-Dilithium, ale na 32bitové verzi systému na Raspberry nebylo možné toto schéma zprovoznit a proto není v grafu uvedeno. V tomto grafu naopak lze pozorovat, že narozdíl od šifrovacích schémat jsou podpisová schémata ve většině rychlejší na 64bitové verzi systému. Všechna schémata v tomto grafu využívají kryptografie založené na hashovacích funkcích kromě schématu qTesla, které je založené na mřížkách a MQDSS, které jako jediné využívá kryptografii založenou na polynomech.Zvláštní je, že qTesla úrovně bezpečnosti 5 je rychlejší než qTesla úrovně 3, ale pouze na Raspberry. Měření tohoto schématu bylo několikrát zopakováno.



Obr. 4.5: Alokované byty šifrovacích schémat

Stejně jako podle grafu s rychlostmi průběhu je zde NewHope nejlépe vycházejícím schématem. Druhý je ThreeBears, který byl naopak v měření vybraných schémat jedním z nejpomalejších.



Obr. 4.6: Alokované byty podpisových schémat

Tento graf obsahuje převážně varianty SPHINCS+, kterých bylo nejvíce. Jsou však jedny z nejpomalejších a paměťově nejnáročnějších.

Na základě dat z těchto grafů bylo vybráno šifrovací schéma NewHope, které využívá kryptografii založenou na mřížkách a je z měřených postkvantových schémat nejrychlejší i paměťově nejméně náročné. Vhodnost použití těchto schémat na omezených zařízeních je však diskutabilní. Výzkumu v této oblasti ještě není zdaleka tolik kolik je potřeba, ale to by se mohlo s příchodem soukromých firem závislých na vysoké bezpečnosti změnit.

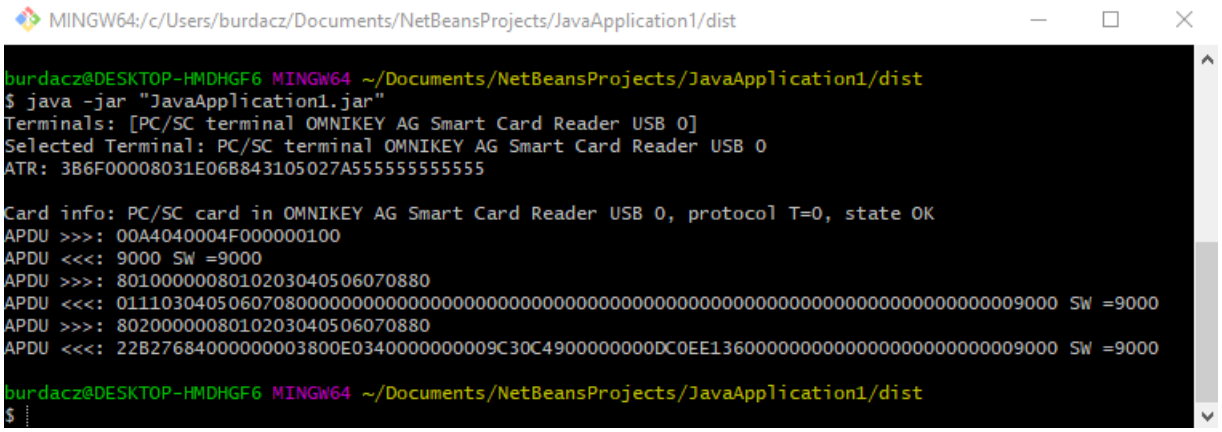
5 Implementace NewHope512CCA

Čipová karta je výpočetně a paměťově nejvíce omezené zařízení, na které můžeme nějaké postkvantové schéma implementovat. Proto bylo nutné změřit a vyhodnotit postkvantové schéma, které bude ve všech ohledech nejméně náročné a přitom jednoduché na implementaci. Tím schématem je NewHope. Implementace tohoto postkvantového schématu se povedla jako prvním na komerčně dostupných bezkontaktních kartách firmě Infineon [37]. NewHope byl také experimentálně používán Googlem [39].

K následující implementaci bude využita vývojářská karta Multos ML3-80K-R1 2013, která umožňuje nahrávání více aplikací najednou, které jsou od sebe izolované systémem firewallů tak, aby data nebyla přístupná bez požadované autorizace. Multos aplikace jsou typicky psané v programovacím jazyce C což je vzhledem k tomu, že je NewHope také v programovacím jazyce C výhodné. K dispozici tudíž je 80KB EEPROM paměti a pouze 2KB RAM paměti. Aby bylo možné na kartu implementovat vybrané schéma je první potřeba projít kód a upravit jej. Na kartu bude nahrán pouze kód potřebný k dešifrování zprávy, jelikož veřejný a soukromý klíč budou vygenerovány na počítači a poté bude na kartu nahrán pouze soukromý klíč. Ověření totožnosti by pak mohlo probíhat tak, že z terminálu bude na kartu poslána náhodně vygenerovaná zpráva o maximální délce 32 bytů, která bude zašifrována veřejným klíčem. To znamená, že pouze majitel karty s odpovídajícím soukromým klíčem bude schopen získanou zprávu dešifrovat a odeslat zpět do terminálu, ten pak porovná odeslanou a přijatou zprávu a vyhodnotí ověření. Karta komunikuje s terminálem pomocí zpráv APDU, která se skládá z povinné 4bytové hlavičky (CLA, INS, P1, P1), kterou můžeme volit aplikace či funkce, které chceme na kartě spouštět a z nepovinných parametrů (LC, DATA, LE). Karta pak odpovídá pouze daty a dvěma statusy SW1 a SW2. Pro komukaci byla vytvořena java aplikace, která bude po spuštění posílat 2 APDU zprávy, které budou volat ty samé bloky funkce na kartě. Takto mohou být rychle otestovány úpravy aplikace karty. Velkou nevýhodou při vývoji aplikací je prakticky nemožnost svůj kód správně debugovat. Eclipse Studio, které je na vývoj této aplikace použito sice debugování poskytuje, ale v mnoha případech se nechová stejně jako karta a proto je lepší si chování aplikace vyzkoušet přímo na kartě.

5.1 Dosažené výsledky

Pro zkompilování kódu musely být vloženy hlavičkové soubory, které jsou ve složce SmartDeckFiles v příloze. Byly potřebné pro využívání datových typů jako například uint64_t což je v podstatě datový typ unsigned long long. Jelikož velikost šifrované zprávy je pro konkrétní schéma NewHope512CCA 1104 bytů a bylo by nutné ji přes terminál posílat ve více APDU zprávách, byla šifrovaná zpráva z praktických důvodů pro testování uložena spolu se soukromým klíčem. Poté již stačilo sloučit .c a .h soubory z implementace NewHope512CCA do hlavního souboru a z lokálních proměnných, které by se nevlezly do RAM, udělat globální proměnné uložené do statické paměti. Dešifrování samotné trvá přibližně 4 minuty, ale bohužel ani po několika úpravách kódu se nepovedlo dostat od karty původní zašifrovanou zprávu. Na následujícím obrázku je vidět poslední přijatá APDU zpráva obsahující dešifrovanou zprávu, která se z většiny skládá z hodnot 0. Testováním jednotlivých funkcí kódu se nepodařilo přijít na žádnou konkrétní chybu, ale nejpravděpodobnější problém bude v bitových posunech právě na velkých číslech a nebo v samotném hlavičkovém souboru stdint, který ve spojení s kartou nefunguje správně.



Obr. 5.1: Dešifrovaná zpráva vrácená kartou na terminál.

6 Závěr

Tato práce pojednává o možnostech postkvantové kryptografie. V rámci teoretické části popisuje jednotlivé typy kryptografie, které se používájí v kryptosystémech odolných proti kvantovým počítačům. V další části již analyzuje konkrétní implementace kryptosystémů. V rámci praktické části se práce zaměřila na měření výpočetní náročnosti verzí kryptosystémů. Měření probíhalo na omezeném zařízení Raspberry Pi 3 Model B s 32bitovou a 64bitovou verzí operačního systému. V první části měření se práce zaměřila na postkvantové šifrovací a podpisové kryptosystémy. Z hlediska naměřených dat byl jako vhodný kandidát na implementaci na čipovou kartu vybrán NewHope což potvrzuje i [37] a [39]. Dalšími potencionálními kandidáty by mohli být Crystals-Dilithium nebo qTesla což jsou podpisová schémata, která byla jak výpočetně tak paměťově nenáročná v porovnání s ostatními.

Práce byla zaměřená hlavně na sestavení veřejně dostupné knihovny pro měření postkvantových schémat, která se bude moci dále rozšiřovat a na využití této knihovny k měření dostupných postkvantových schémat. Dále měla práce na základě těchto dat vybrat vhodného kandidáta pro výpočetně a paměťově omezená zařízení, zhodnotit vhodnost dostupných postkvantových schémat pro omezená zařízení a následnou implementaci na čipovou kartu. Tyto cíle byly z větší části splněny.

Literatura

[1] Quantiki *The Holevo bound* University of Cambridge's Centre for Quantum Computation [online], 2015 [cit. 20.05.2019] Dostupné z URL: <https://quantiki.org/wiki/holevo-bound>.

[2] Lov K. Grover *A fast quantum mechanical algorithm for database search* Bell Labs 600 Mountain Avenue Murray Hill NJ 07974 [online], 1996 [cit. 25.05.2019] Dostupné z URL: <https://arxiv.org/pdf/quant-ph/9605043v3.pdf>.

[3] Peter W. Shor *Algorithms for Quantum Computation: Discrete Log and Factoring* AT&T Bell Labs Room 2D-149 600 Mountain Ave. Murray Hill NJ 07974 USA [online], 1994 [cit. 25.05.2019] Dostupné z URL: <https://ieeexplore.ieee.org/document/365700>.

[4] Wikipedia *Timeline of quantum computing* [online], 2016 [cit. 20.05.2019] Dostupné z URL: <https://en.wikipedia.org/wiki/Timeline_of_quantum_computing>.

[5] D-Wave Systems *The Quantum Computing Company* [online], 2018 [cit. 20.05.2019] Dostupné z URL: <https://www.dwavesys.com>.

[6] Julian Kelly *A Preview of Bristlecone, Google's New Quantum Processorg* Research Scientist, Quantum AI Lab [online], 2018 [cit. 20.05.2019] Dostupné z URL: <https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html>.

[7] John Proos and Christof Zalka *Shor's discrete logarithm quantum algorithm for elliptic curves* Department of Combinatorics and Optimization University of Waterloo, Waterloo, Ontario Canada N2L 3G1 [online], 2008 [cit. 25.05.2019] Dostupné z URL: <https://arxiv.org/pdf/quant-ph/0301141.pdf>.

[8] R. W. Hamming *Error detecting and error correcting codes* Nokia Bell Labs [online], 1950 [cit. 25.05.2019] Dostupné z URL: <https://signallake.com/innovation/hamming.pdf>.

[9] R. J. McEliece *A Public-Key Cryptosystem Based On Algebraic Coding Theory* [online], 1978 [cit. 25.05.2019] Dostupné z URL: <https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF>.

[10] Niederreiter, H *Knapsack Type Cryptosystems and Algebraic Coding Theory* [online], 1986 [cit. 25.05.2019] Dostupné z URL: <https://www.researchgate.net/publication243776483_Knapsack_Type_Cryptosystems_and_Algebraic_Coding_Theory>.

[11] Michael Hartmann *The Ajtai-Dwork Cryptosystem and Other Cryptosystems Based on Lattices* University of Zurich, Institute of Mathematics [online], 2015 [cit. 25.05.2019] Dostupné z URL: <http://user.math.uzh.ch/rosenthal/masterthesis/07711484/Hartmann2015.pdf>.

[12] MICCIANCIO, D. *Lattice-based Cryptography* Courant Institute of Mathematical Sciences [online], 2008 [cit. 14.12.2018] Dostupné z URL: <https://www.cims.nyu.edu/~regev/papers/pqc.pdf>.

[13] Erdem Alkim, Roberto Avanzi, Joppe Bos, Léo Ducas, Antonio de la Piedra, Thomas Pöppelmann, Peter Schwabe, Douglas Stebila *NewHope Algorithm Specifications and Supporting Documentation* [online], 2018 [cit. 26.05.2019] Dostupné z URL: <https://newhopecrypto.org/data/NewHope_2018_12_02.pdf>.

[14] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman *NTRU: A Ring-Based Public Key Cryptosystem* [online], 2006 [cit. 26.05.2019] Dostupné z URL: <https://www.onboardsecurity.com/products/ntru-crypto>.

[15] Oded Regev *The Learning with Errors Problem* [online], 2005 [cit. 20.05.2019] Dostupné z URL: <https://cims.nyu.edu/~regev/papers/lwesurvey.pdf>.

[16] Vikram Singh *A Practical Key Exchange for the Internet using Lattice Cryptography* [online], 2015 [cit. 20.05.2019] Dostupné z URL: <https://eprint.iacr.org/2015/138.pdf>.

[17] Andreas Hülsing *SPHINCS+ – The smaller SPHINCS* [online], 2017 [cit. 26.05.2019] Dostupné z URL: <https://huelsing.net/wordpress/?p=558>.

[18] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Greg Zaverucha *Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives* [online], 2017 [cit. 26.05.2019] Dostupné z URL: <https://eprint.iacr.org/2017/279>.

[19] BERNSTEIN, J. D. *Introduction to post-quantum cryptography* University of Illinois at Chicago: Department of Computer Science [online], 2009 [cit. 14.12.2018] Dostupné z URL: <https://link.springer.com/chapter/10.1007/978-3-540-88702-7_1>.

[20] Enrico Thomae, Christopher Wolf *Solving Systems of Multivariate Quadratic Equations over Finite Fields or: From Relinearization to MutantXL* Faculty of Mathematics Ruhr-University of Bochum [online], 2010 [cit. 26.05.2019] Dostupné z URL: <https://www.researchgate.net/publication30816377_MutantXL_Solving_Multivariate_Polynomial_Equations_for_Cryptanalysis>.

[21] Ming-Shing Chen, Andreas Hulsing, Joost Rijneveld, Simona Samardjiska, Peter Schwabe *MQDSS specifications* [online], 2018 [cit. 26.05.2019] Dostupné z URL: <http://mqdss.org/files/MQDSS_Ver1point1.pdf>.

[22] HUYNH *A. An Introduction to Supersingular Elliptic Curves and Supersingular Primes* University of Washington [online] [cit. 14.12.2018] Dostupné z URL: <https://wstein.org/edu/2011/581g/final/anh-supersingular_elliptic_curves.pdf>.

[23] David Jao *Supersingular Isogeny Key Encapsulation* [online], 2019 [cit. 26.05.2019] Dostupné z URL: <https://sike.org/files/SIDH-spec.pdf>.

[24] NIST *Request for Comments on Submission Requirements and Evaluation Criteria* [online], 2016 [cit. 20.05.2019] Dostupné z URL: <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/rfc-on-submmission-requirements-and-evaluation-cri>.

[25] NIST, Dustin Moody *Round 2 of the NIST PQC "Competition"* [online], 2019 [cit. 20.05.2019] Dostupné z URL: <https://csrc.nist.gov/CSRC/media/Presentations/Round-2-of-the-NIST-PQC-Competition-What-was-NIST/images-media/pqcrypto-may2019-moody.pdf>.

[26] NIST *Post-Quantum Cryptography Workshops and Timeline* [online], 2019 [cit. 20.05.2019] Dostupné z URL: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Workshops-and-Timeline>.

[27] National Institute of Standards and Technology *Post-Quantum Cryptography Round 1 Submissions* University of Illinois at Chicago: Department of Computer Science [online], 2009 [cit. 14.12.2018] Dostupné z URL: <https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions>.

[28] Raspberry Pi Foundation *Raspbian* [online], 2019 [cit. 26.05.2019] Dostupné z URL: <https://www.raspberrypi.org/downloads/raspbian/>.

[29] Sakaki *Bootable 64-bit Gentoo image for the Raspberry Pi 3 B B+* [online], 2019 [cit. 26.05.2019] Dostupné z URL: <https://github.com/sakaki-/gentoo-on-rpi3-64bit>.

[30] Sakaki *Post-Quantum Cryptography Round 2 Submissions* [online], 2019 [cit. 26.05.2019] Dostupné z URL: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.

[31] Patrik Burda *Library for testing postquantum algorithms* [online], 2019 [cit. 26.05.2019] Dostupné z URL: <https://bitbucket.org/burdacz/postquantum_lib/src/master/>.

[32] GeeksforGeeks *Pseudo Random Number Generator (PRNG)* [online], 2018 [cit. 26.05.2019] Dostupné z URL: <https://www.geeksforgeeks.org/pseudo-random-number-generator-prng/>.

[33] Victor Shoup *NTL: A Library for doing Number Theory* New York University Courant Institute [online], 2015 [cit. 26.05.2019] Dostupné z URL: <https://shoup.net/ntl/>.

[34] Torbjorn Granlund *The GNU Multiple Precision Arithmetic Library* [online], 2016 [cit. 26.05.2019] Dostupné z URL: <https://gmplib.org/>.

[35] Gilles Van Assche *eXtended Keccak Code Package* [online], 2015 [cit. 26.05.2019] Dostupné z URL: <https://github.com/XKCP/XKCP>.

[36] Brian Murray [online], 2013 [cit. 26.05.2019] Dostupné z URL: <https://wiki.ubuntu.com/Valgrind>.

[37] Business & Financial Press *Ready for tomorrow: Infineon demonstrates first post-quantum cryptography on a contactless security chip* [online], 2017 [cit. 27.05.2019] Dostupné z URL: <https://www.infineon.com/cms/en/about-infineon/press/press-releases/2017/INFCCS201705-056.html>.

[38] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, Peter Schwabe *Post-quantum key exchange – a new hope* [online], 2016 [cit. 27.05.2019] Dostupné z URL: <https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_alkim.pdf>.

[39] Matt Braithwaite, Google *Experimenting with Post-Quantum Cryptography* [online], 2016 [cit. 20.05.2019] Dostupné z URL: <https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html>.

[40] SafeCrypto *Post-Quantum Cryptography Lounge* Safecrypto [online] [cit. 14.12.2018] Dostupné z URL: <https://www.safecrypto.eu/pqclounge/>.

[41] Lucie Popelová *Metody post-kvantové kryptografie* Brno VUT [online], 2018 [cit. 14.12.2018] Dostupné z URL: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=175435>.

[42] Bc. Jana Divišová *Kryptografie založená na mřížkách* Brno VUT [online], 2010 [cit. 14.12.2018] Dostupné z URL: <https://www.soom.cz/data/DPTX_2007_1_11320_NSZZ016_236020_0_50807.pdf>.

Seznam symbolů, veličin a zkratek

CVP	Closest Vector Problem
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie Hellman
IoT	Internet of Things
KAT	Known Answer Test
KEM	Key Encapsulation Mechanism
LWE	Learning With Errors
MQ	Multivariate Quadratic
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
NMR	Nuclear Magnetic Resonance
RLWE	Ring Learning With Errors
SVP	Shortest Vector Problem

Seznam příloh

A Obsah přiloženého CD

Bakalářská práce - PatrikBurda.pdf (elektronická verze)

Knihovna pro měření postkvantových schémat - PostQuantumLib (archiv z bitbucketu)

Kompletní data měření - Data.xlsx

Projekt s implementací na čipovou kartu - NewHopeCard

Složka obsahující .h soubory pro SmartDeck - SmartDeckFiles

Projekt s aplikací pro komunikaci s čipovou kartou - JavaTerminal