SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-5384-8739

PODPISOVÉ SCHÉMY V POSTKVANTOVEJ KRYPTOGRAFII

DIPLOMOVÁ PRÁCA

2016 **Pavol Dobročka**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-5384-8739

PODPISOVÉ SCHÉMY V POSTKVANTOVEJ KRYPTOGRAFII

DIPLOMOVÁ PRÁCA

Študijný program:	Aplikovaná informatika
Číslo študijného odboru:	2511
Názov študijného odboru:	9.2.9 Aplikovaná informatika
Školiace pracovisko:	Ústav informatiky a matematiky
Vedúci záverečnej práce:	doc. Ing. Pavol Zajac, PhD.

Bratislava 2016 **Pavol Dobročka**

Slovenská technická univerzita v Bratislave
Ústav informatiky a matematiky

Fakulta elektrotechniky a informatiky
Akademický rok: 2015/2016
Evidenčné číslo: FEI-5384-8739

ZADANIE DIPLOMOVEJ PRÁCE

Študent:	**Bc. Pavol Dobročka**
ID študenta:	8739
Študijný program:	Aplikovaná informatika
Študijný odbor:	9.2.9. aplikovaná informatika
Vedúci práce:	doc. Ing. Pavol Zajac, PhD.
Miesto vypracovania:	Ústav informatiky a matematiky

Názov práce: **Podpisové schémy v postkvantovej kryptografii**

Špecifikácia zadania:

Cieľom práce je implementovať podpisovú schému pomocou prostriedkov postkvantovej kryptografie. Zameriame sa prioritne na schémy využívajúce dekódovací problém.

Úlohy:
1. Naštudujte problematiku podpisových schém pomocou postkvantovej kryptografie založenej na dekódovacom probléme.
2. Analyzujte knižnicu BitPunch a navrhnite potenciálne rozšírenie knižnice o podpisové schémy.
3. Implementujte vybranú podpisovú schému.
4. Otestujte a vyhodnoťte riešenie.

Zoznam odbornej literatúry:

1. F. Uhrecký: Implementácia kryptografickej knižnice s McEliece kryptosystémom. Diplomová práca, 2015.
2. M. Repka, P. Zajac: "Overview of the Mceliece Cryptosystem and its Security." Tatra Mountains Mathematical Publications 60.1 (2014): 57-83.
3. N. Courtois, M. Finiasz, and N. Sendrier. "How to achieve a McEliece-based digital signature scheme." Advances in Cryptology—ASIACRYPT 2001. Springer Berlin Heidelberg, 2001. 157-174.

Riešenie zadania práce od: 21. 09. 2015

Dátum odovzdania práce: 20. 05. 2016

SLOVENSKÁ TECHNICKÁ UNIVERZITA
Fakulta elektrotechniky a informatiky
Ú. S.
informatiky
a matematiky
812 19 BRATISLAVA

Bc. Pavol Dobročka
študent

prof. RNDr. Otokar Grošek, PhD.
vedúci pracoviska

prof. RNDr. Gabriel Juhás, PhD.
garant študijného programu

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Aplikovaná informatika
Autor:	Pavol Dobročka
Diplomová práca:	Podpisové schémy v postkvantovej kryptografii
Vedúci záverečnej práce:	doc. Ing. Pavol Zajac, PhD.
Miesto a rok predloženia práce:	Bratislava 2016

Cieľom práce bolo štúdium a implementácia podpisových schém založených na náročnosti riešenia dekódovacieho problému (v angl. code-based cryptography). Práca v úvodnej časti obsahuje prehľad známych code-based kryptosystémov. Ďalej sa zameriava na prehľad návrhov na použitie code-based kryptografie na elektronický podpis. Bližšie skúma McElieceov kryptosystém a k nemu duálny Niederreiterov kryptosystém a ponúka ich vzájomné porovnanie. Z podpisových schém popisuje najstaršiu CFS schému a novšiu LDGM schému. Pre ďalšiu prácu sme zvolili LDGM schému vzhľadom na jej novosť a menšie kľúče. Fungovanie celej LDGM schémy demonštruje v práci ukážkový príklad. Výsledkom práce je funkčná implementácia LDGM.

Kľúčové slová: Postkvantová kryptografia, Elektronický podpis, Dekódovací problém, Code-based kryptografia, McEliece, Niederreiter, QC matice, LDGM, CFS, BitPunch

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Pavol Dobročka
Diploma Thesis:	Post-Quantum Cryptography Digital Signature
Supervisor:	doc. Ing. Pavol Zajac, PhD.
Place and year of submission:	Bratislava 2016

The goal of the thesis was to study and implement code-based digital signature schemes. The first part of the thesis contains the review of the well known code-based cryptosystems. After that it focuses on the proposed digital signature schemes which rely on the code-based cryptography. It reviews in more detail the McEliece and Niederreiter cryptosystems, which are dual to each other and puts them in comparison. In the field of code-based digital signature schemes the thesis reviews the oldest CFS scheme and the recent LDGM scheme. For the further research we picked the LDGM to benefit from its key size reduction feature. The principle of the LDGM scheme is demonstrated in a simple example. The overall output of the thesis is a working LDGM implementation.

Keywords: Post-Quantum cryptography, Digital signature, Decoding problem, Code-based cryptography, McEliece, Niedderreiter, QC matrices, LDGM, CFS, BitPunch

Vyhlásenie autora

Podpísaný Pavol Dobročka čestne vyhlasujem, že som diplomovú prácu Podpisové schémy v postkvantovej kryptografii vypracoval na základe poznatkov získaných počas štúdia a informácií z dostupnej literatúry uvedenej v práci.

Vedúcim mojej diplomovej práce bol doc. Ing. Pavol Zajac, PhD.

Bratislava, dňa 17.5.2016

....

podpis autora

Poďakovanie

Chcem sa poďakovať vedúcemu záverečnej práce, ktorým bol doc. Ing. Pavol Zajac, PhD., za odborné vedenie, rady a pripomienky, ktoré mi pomohli pri vypracovaní tejto diplomovej práce.

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek a značiek

Ak nie je v práci uvedené inak, platia nasledovné označenia

QC - quasi-cyclic
VK - verejný kľúč
PK - privátny kľúč
$A[n][m]$ - prvok z matice A na n-tom riadku a m-tom stĺpci
$A[n:m]$ - matica tvorená riadkami n až m matice A
$A[k][n:m]$ - vektor tvorený prvkami n až m z k-tého riadku matice A

V algoritmoch prvky vektorov a matíc indexujeme od 1

Zoznam algoritmov

Úvod

Vo februári 2016 NIST zverejnil správu [3], v ktorej hodnotí súčasný stav kryptosystémov a ich bezpečnosť v najbližších rokoch. Bezpečnosť dnešných kryptosystémy považuje z dlhodobého hľadiska za nedostatočnú a vyzýva na návrh budúceho postkvantového kryptografického štandardu. Postkvantové kryptosystémy sú odolné voči známym útokom na kvantovom počítači. Hoci zatiaľ nie je známa praktická realizácia kvantového počítača, na ktorom by bolo možné uskutočniť útoky na dnešné kryptosystémy, vývoj napreduje a do výskumu sa investuje veľa finančných prostriedkov. Tento čas treba využiť na výskum a prípravu.

V súčasnosti je známych niekoľko druhov postkvantových asymetrických kryptosystémov. [2] Jedným z nich sú takzvané code-based kryptosystémy. Tieto kryptosystémy vychádzajú z teórie kódovania a ich bezpečnosť je založená na dekódovacom probléme. Dekódovací problém označuje problém opravenia chýb pre všeobecný lineárny kód. Tento problém je NP-úplný, a teda sa predpokladá, že s veľkou pravdepodobnosťou neexistuje efektívny všeobecný dekódovací algoritmus pre vhodne zvolené parametre. Podstata code-based kryptosystémov je skonštruovať kód, pre ktorý poznáme algoritmus na efektívne dekódovanie a následne ho transformovať na kód, v ktorom nepoznáme efektívny dekódovací algoritmus. Táto transformácia je väčšinou súčasťou súkromného kľúča.

Predmetom tejto práce je preskúmať známe code-based kryptosystémy a návrhy, ktoré ich využívajú na elektronický podpis a následne implementovať vybranú podpisovú schému ako súčasť knižnice BitPunch [10]. V prvej kapitole skúmame niektoré code-based kryptosystémy. Konkrétne McElieceov kryptosystém a k nemu duálny Niederreiterov kryptosystém. V závere kapitoly je porovnanie týchto kryptosystémov. Druhá kapitola sa venuje code-based podpisovým schémam. Obsahuje najstaršiu CFS schému a LDGM schému, ktorá patrí k novším návrhom. Schémy, ktoré boli považované za zlomené pred začiatkom písania tejto práce neboli zahrnuté. Na konci kapitoly je uvedený ukážkový príklad, ktorý demonštruje fungovanie LDGM schémy. Tretia kapitola obsahuje konkrétny návrh LDGM a implementačné detaily. Predstavuje algoritmy na generovanie kľúčov, podpisovanie, overovanie a organizačnú štruktúru v knižnici BitPunch. Posledná kapitola je venovaná meraniu výkonnosti implementácie. Obsahuje výsledky testov invertovania kvázicyklických matíc a výkonnosti celej podpisovej schémy.

1 Code-based kryptografia

Táto kapitola sa venuje McElieceovmu a Niederreiterovmu kryptosystému. Predstavuje algoritmy na šifrovanie a dešifrovanie a vzájomné porovnanie kryptosystémov.

1.1 McEliece

Najstarším a pravdepodobne najznámejším code-based kryptosystémom je McElieceov kryptosystém [7]. Jadro systému tvorí kód C dĺžky n s dimenziou k a minimálnou vzdialenosťou $d \geq 2t + 1$, kde t je počet chýb, ktorý vie kód opraviť. Podľa pôvodného návrhu sa používajú Goppove kódy, ku ktorým existuje efektívny dekódovací algoritmus.
Verejný a súkromný kľúč zostrojíme nasledovne. Určíme generujúcu maticu G s rozmermi $k \times n$ pre kód C. Ďalej zvolíme náhodnú binárnu regulárnu maticu S s rozmermi $k \times k$ a permutačnú maticu P s rozmermi $n \times n$. Verejný kľúč tvorí matica $G' = SGP$ a parameter t. Súkromný kľúč tvoria matice S, G, P.

Algoritmus 1 McEliece - Algoritmus šifrovania

Vstup: Správa m dĺžky k

Výstup: Zašifrovaná správa c

$c' \leftarrow mG'$

K zakódovanej správe pripočítame náhodný chybový vektor s váhou t.

$c \leftarrow c' + e$, $wt(e) = t$

return c

Algoritmus 2 McEliece - Algoritmus dešifrovania

Vstup: Zašifrovaná správa c

Výstup: Otvorená správa m

Správu c vynásobíme s P^{-1}

$c' \leftarrow cP^{-1} = mSG + eP^{-1}$

$m' \leftarrow Decode(c') = mS;$

$m \leftarrow m'S^{-1}$

return m

V súčasnosti sa pre praktickú bezpečnosť odporúčajú hodnoty parametrov kódu
$n = 1833, k = 1356, t = 44$ [8].

1.2 Niederreiter

K McElieceovmu kryptosystému existuje variant, ktorý namiesto generujúcej matice G využíva kontrolnú maticu H. Táto duálna forma je známa ako Niederreiterov kryptosystém [2]. V tomto kryptosystéme sa správa m najskôr transformuje na vektor m' dĺžky n s Hammingovou váhou t. Funkciu, ktorá vykonáva túto transformáciu označujeme $\phi_{n,t}(m)$. Verejný kľúč tvorí matica $H' = SHP$ a parameter t. Matica S je náhodná regulárna binárna matica s rozmermi $(n-k) \times (n-k)$ a P je permutačná matica s rozmermi $n \times n$ a súkromný kľúč tvoria matice S,H,P. Šifrovaný text sa vypočíta ako syndróm slova m', $c = H'm'^T$ Na dešifrovanie slova c vlastník súkromného kľúča najskôr vynásobí slovo c maticou S^{-1} zľava, následne aplikuje dekódovací algoritmus a výsledok vynásobí maticou P^{-1} zľava. $m = P^{-1}decode(S^{-1}SHPm)$

Algoritmus 3 Niederreiter - Algoritmus šifrovania

Vstup: Správa m

Výstup: Zašifrovaná správa c

$m' \leftarrow \phi(m)$, dostaneme chybové slovo dĺžky n s váhou t

$c \leftarrow H'm^T$

return c

Algoritmus 4 Niederreiter - Algoritmus dešifrovania

Vstup: Vektor c, ktorý predstavuje šifrovanú správu m

Výstup: Dešifrovaná správa, pôvodné m

$c' \leftarrow S^{-1}c$

$e' \leftarrow Decode(c')$

$e \leftarrow P^{-1}e'$

$m \leftarrow \phi^{-1}(e)$

1.3 Porovnanie McEliece a Niederreiter

Zhrňme si a porovnajme parametre oboch kryptosystémov a ako sa zvolené parametre kódu prejavia na veľkosti správ a kľúčov.

Tabuľka 1: Porovnanie parametrov McEliece a Niederreiter

	McEliece	**Niederreiter**
Verejný kľúč	G', t	H', t
Privátny kľúč	S, G, P	S, H, P
Veľkosť VK	nk	$n(n-k)$
Veľkosť PK	$n^2 + nk + k^2$	$3(n^2 - nk) + k^2$
Veľkosť otvorenej správy	k	$\log_2 \sum_{i=0}^{t} \binom{n}{i}$
Veľkosť šifrovanej správy	n	$n - k$
Počet možných správ	2^k	$\sum_{i=0}^{t} \binom{n}{i}$

Pri vhodne zvolených parametroch vieme dosiahnut s Niederreiterovým systémom v porovnaní s McElieceom zmenšenie kľúčov a šifrovaného textu. McEliece však ponúka jednoduchšiu implementáciu a nepotrebuje funkciu na mapovanie otvoreného textu na slová s pevnou váhou.

2 Code-based podpisové schémy

2.1 Úvod

Prechod na postkvantovú kryptografiu so sebou prináša aj potrebu implementovať podpisové schémy pomocou postkvantového kryptosystému. Vo všeobecnosti sa na realizáciu digitálneho podpisu využívajú asymetrické kryptosystémy, respektíve kryptosystémy s verejným kľúčom. Kryptosystémy, ktoré sme si predstavili v predchádzajúcej časti spĺňajú toto kritérium. Všeobecná schéma na vytvorenie digitálneho podpisu správy má podľa definície [6] tieto časti

- Algoritmus na generovanie páru privátnych a verejných kľúčov
- Podpisový algoritmus závislý od privátneho kľúča, ktorý vytvorí podpis pre danú správu
- Overovací algoritmus závislý od verejného kľúča, ktorý príjme alebo zamietne podpis pre zodpovedajúcu správu

Niektoré kryptosystémy túto schému implementujú tak, že ako podpisovú funkciu použijú dešifrovací algoritmus so súkromným kľúčom a ako overovaciu zvolia šifrovací algoritmus s verejným kľúčom. Podpis a overenie v tejto implementácii môže vyzerať takto

Algoritmus 5 Schéma digitálneho podpisu

Vstup: Správa m, odtlačková funkcia H

Výstup: Podpis správy m, ozn. sig

$h \leftarrow H(m)$

$sig \leftarrow Decode(h, PrivateKey)$

return sig

Možnosť tejto implmentácie je silne závisla od toho, ako sa prekrývajú množiny šifrovaných textov a odtlačkov v konkrétnom kryptosystéme. Ako si ukážeme v ďalších častiach práce, nie všetky odtlačky musia byť dešifrovateľné správy.

2.2 Prehľad code-based podpisových schém

V nasledujúcich častiach práce sa už budeme zaoberať iba code-based podpisovými schémami, teda schémami, ktoré využívajú code-based kryptosystémy. Veľkou prekážkou týchto kryptosystémov je v súčasnosti veľkosť kľúča, ktorá je v porovnaní s dnešnými kryptosystémami rádovo tisícnásobne väčšia. Pri implementácii a následne v praxi je

dôležité nájsť vhodný kompromis medzi požadovanou bezpečnosťou a výpočtovou a dátovou náročnosťou, ktorá zavisí od voľby veľkosti kľúča.
Existuje niekoľko potenciálnych návrhov code-based kryptosystémov, z ktorých si bližšie predstavíme CFS (Courtois-Finiasz-Sendrier) [4] a LDGM (Low-density generator matrix) [1].

2.3 CFS schéma

Jedným z nádejných návrhov code-based podpisových schém je CFS schéma (pomenovaná podľa autorov), ktorá používa na podpisovanie Niederreiterov kryptosystém . Základný problém, ktorý treba vyriešiť pri podpisovaní založenom na kódovaní, je ako získať taký odtlačok správy, ktorý je dekódovateľné slovo. Ak máme lineárny kód $C(n, k, 2t + 1)$, syndróm slova je vektor dĺžky $n - k$. Počet všetkých syndrómov je 2^{n-k} a počet dekódovateľných syndrómov je $\sum_{i=0}^{t} \binom{n}{i}$. To znamená, že $\frac{\sum_{i=0}^{t} \binom{n}{i}}{2^{n-k}}$ všetkých syndrómov je dekódovateľných. Pre Goppove kódy je to približne $\frac{1}{t!}$ [4]. Pravdepodobnosť, že odtlačok správy bude zároveň dekódovateľný, je teda približne $p = \frac{1}{t!}$. Nato, aby sme vedeli podpísať každú správu, budeme musieť ku správe pridať bity navyše a pokúsiť sa podpísať túto upravenú správu. Priemerný počet pokusov na podpísanie jednej správy je približne $t!$.

Algoritmus 6 Algoritmus podpisovania v CFS

Vstup: Správa m, odtlačková funkcia H ktorá vracia odtlačky dĺžky $n - k$
Výstup: Podpis správy m, ozn. sig

```
i ← 0
repeat
    h ← H(m||i)
    if h nie je dekódovateľné slovo then
        i ← i + 1
    end if
until h je dekódovateľné slovo
e ← Decode(h)
sig ← (e, i)

return sig
```

Implementácia uvedeného algoritmu môže byť vylepšená po viacerých stránkach. Prvé vylepšenie sa dá realizovať pri hľadaní dekódovateľného syndrómu. Na začiatku podpisovania si vypočítame hash samotnej správy $h' = H(m)$ a v ďalších krokoch počítame $h = H(h'||i)$. Ďalší priestor na vylepšenie, tentokrát dĺžka výsledného podpisu, sa ponúka v spôsobe uloženia časti e z podpisu. Autori tejto podpisovej schémy navrhli ukladať e ako index I z množiny všetkých n bitových vektorov s váhou t. To predstavuje číslo z rozsahu $[1, \binom{n}{t}]$ [4]

Algoritmus 7 Algoritmus overovania v CFS

Vstup: Podpis (e, i), správa m, verejný kľúč H_{pub}, odtlačková funkcia H ktorá vracia odtlačky dĺžky $n - k$

Výstup: True/False - podpis prijímame/zamietame

$s_1 \leftarrow H_{pub}e^T$

$s_2 \leftarrow H(m||i)$

return $s_1 = s_2$

Kedže podpis správy tvorí chybové slovo s pevnou váhou, pri parametroch $n = 2^{16}, t = 9$, ktoré navrhli autori, môže byť podpis výrazne komprimovaný, čo je značná výhoda oproti schéme, ktorá využíva klasický McElieceov kryptosystém.

2.4 LDGM schéma

Ďalší z možných návrhov pre code-based kryptografiu sa pokúša zmenšiť potrebnú veľkosť kľúča pomocou vhodne zvoleného kódu, respektíve pomocou vhodne zvolenej generujúcej matice. LDGM (Low-density generating matrix) kódy, čiže kódy s generujúcou maticou s nízkou váhou, sa v niektorých prípadoch dajú zapísať kompaktne pomocou cirkulantných matíc. Generujúca matica G kódu dĺžky n s dimenziou k sa skladá z $k_0 n_0$ blokov s rozmermi $p \times p$, kde $n_0 = n/p$ a $k_0 = k/p$.

$$G = \begin{bmatrix} C_{0,0} & C_{0,1} & \cdots & C_{0,n_0-1} \\ C_{1,0} & C_{1,1} & \cdots & C_{1,n_0-1} \\ \vdots & \vdots & \ddots & \cdots \\ C_{k_0-1,0} & C_{k_0-1,1} & \cdots & C_{k_0-1,n_0-1} \end{bmatrix}$$

Každé $C_{i,j}$ je $p \times p$ cirkulatná matica. Vďaka tomu nám stačí uložiť z každého bloku iba jeden riadok. Tým zmenšíme veľkosť klúča p-násobne. Matica v takomto tvare sa nazýva kvázicyklická (QC). K matici G vypočítame kontrolnú maticu H v systematickom tvare, t.j. $H = [X|I]$. Kontrolná matica H je súčasťou súkromného kľúča. Ďalšiu časť kľúča tvoria matice Q a S. Postup ako určiť maticu Q je zjednodušene zhrnutý v algoritme 8. [1]

Algoritmus 8 Výpočet matice Q

Určíme náhodne matice a, b s rozmermi $z \times r_0$, $z \leq r_0$

Vypočítame maticu $R \leftarrow a^T b \otimes 1_{p,p}$

Určíme maticu T poskladanú z $r_0 \times r_0$ cirkulantných matic tak, aby váha každého riadku aj stĺpca bola w_t a aby $rank(R+T) = r$

$Q \leftarrow R + T$

Pozn. $r_0 = n_0 - k_0$, $1_{p,p}$ *- matica* $p \times p$ *samé jednotky,* $\otimes$ *- Kroneckerov súčin*

Maticu S určíme ako náhodnú maticu poskladanú z $n_0 \times n_0$ cirkulantných blokov veľkosti $p \times p$ tak, aby váha každého riadku aj stĺpca bola w_s a aby mala plnú hodnosť.

Verejný kľúč tvorí upravená kontrolná matica $H_{pub} = Q^{-1} H S^{-1}$. Spôsob, ktorým počítame matice Q a S zachovávajú QC vlastnosti pôvodnej matice H, čo nám umožnuje zmenšenie veľkosti verejného kľúča.

Pri generovaní musíme riešiť podobný problém ako pri CFS schéme. Potrebujeme výstup z hashovacej funkcie transformovať na vektor, ktorý spĺňa podmienky určené štruktúrou kódu. Prepokladajme, že máme funkciu ϕ, ktorá jednoznačne priradí vektoru dĺžky l

vektor dĺžky r s váhou w a funkciu ψ, ktorá pre každú správu m vyberie kódové slovo s nízkou váhou w_c z kódu generovaného maticou G. Generovanie podpisu je formálne zapísané ako Algoritmus 9.

Algoritmus 9 Podpis v LDGM

Vstup: Správa m

Výstup: Podpis správy m, ozn. sig

Vypočítame odtlačok správy m

$h \leftarrow H(m)$

$i \leftarrow 0$

repeat

$s \leftarrow \phi(h||i)$

$b_1 \leftarrow b \otimes 1_{1,p}$

if $b_1 s \neq 0$ **then**

$i \leftarrow i + 1$

end if

until $b_1 s = 0$

$s' \leftarrow Qs$

$e \leftarrow [0_{1,k}||s'^{\mathrm{T}}]$

$c \leftarrow \psi(m)$

$e' \leftarrow (e + c)S^T$

$sig \leftarrow (e', i)$

return sig

Pozn. $0_{1,k}$ *- nulový vektor dĺžky k*

Na overenie podpisu najskôr skontrolujeme, či váha slova, ktoré je výstupom podpisu, spĺňa parametre schémy. Ďalej potrebujeme zrekonštruovať syndróm, ktorý sa použil na vytvorenie podpisu. Ten tvorí výstup funkcie ϕ, ktorej vstup bol hash správy rozšírenej o hodnotu počítadla, ktoré je súčasťou podpisu. Potom pomocou verejného kľúča získame syndróm pre slovo, ktoré tvorí podpis a porovnáme ho so zrekonštruovaným syndrómom. Ak sa zhodujú, popis akceptujeme. Postup je zhrnutý v algoritme 10.

Algoritmus 10 Overenie v LDGM

Vstup: Správa m, podpis (e', i)

Výstup: True/False - podpis prijímame/zamietame

```
if wt(e') > (w_t w + w_c) w_s then
    return False
else
    s ← φ(H(m)||i)
    if wt(s) ≠ w then
        return False
    else
        s_1 ← H_pub e'^T
        return s = s_1
    end if
end if
```

Pre lepšiu predstavu ako LDGM podpisová schéma funguje si predvedieme generovanie, podpisovanie a overenie na príklade.

2.5 Ukážkový príklad LDGM

Hodnoty parametrov použité v príklade sú odlišné od hodnôt vhodných pre praktické využitie.

Generovanie kľúčov Majme kód $C(15,9)$ s kontrolnou maticou H v systematickom tvare

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Rozmer cirkulantných matíc je $p \times p$, $p = 3$, $n_0 = 5$, $r_0 = 2$.
Ďalej vypočítame maticu Q. Matica Q je tvorená ako súčet matíc R a T. Na výpočet matice R zvolíme matice a, b s rozmermi $z \times r_0$, kde $z \leq r_0$.

$$a = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Matica $R = (a' \times b) \otimes 1_{p,p}$

$$R = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Maticu T vyberáme tak, aby sa skladala z $r_0 \times r_0$ cirkulatných blokov, váha každého stĺpca a riadku bola w_t a hodnosť $R + T$ bola $r = r_0 p$. Zvolme $w_t = 1$.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Matica S je regulárna matica tvorená z $n_0 \times n_0$ cirkulatných blokov, každý stĺpec a riadok má hodnosť w_s. Nech $w_s = 1$, potom matica S môže vyzerať nasledovne

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Keď máme matice Q a S, môžeme vypočítať maticu $H_{pub} = Q^{-1}HS^{-1}$, ktorá tvorí verejný kľúč

$$H_{pub} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Vytvorenie podpisu Predpokladajme, že máme správu m, pre ktorú je výstup z funkcie ϕ vektor s váhou w

$$s = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}^{\mathrm{T}} \qquad w = 2$$

Ďalej vypočítame $s' = Q \cdot s$

$$s' = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

Vektoru s' zodpovedá chybové slovo v tvare $e = [0_{1\times k}, s'^{\mathrm{T}}]$

$$e = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Predstavme si, že funkcia $\psi(m)$ vyberie kódové slovo c s váhou $w_c = 4$ pre správu m

$$c = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Posledný krok podpisu je výpočet $e' = (e + c) \cdot S^{\mathrm{T}}$

$$e' = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Podpis tvorí dvojica (e', i). V tomto príklade však i neuvádzame.

Overenie podpisu Ako prvé overíme, či $wt(e') \leq (w_t \cdot w + w_c) \cdot w_s$. V našom príklade $w_t = 1$, $w = 2$, $w_c = 4$ a $w_s = 1$. Potom $6 \leq (1 \cdot 2 + 4) \cdot 1$, $6 \leq 6$ Prvá podmienka je splnená. Ďalej pomocou verejného kľúča zrekonštruujeme odtlačok správy $s_0 = H_{pub}e'^{\mathrm{T}}$

$$s_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$$

Vidíme, že $s = s_0$ a podpis teda akceptujeme.

2.6 Porovnanie

Pozrime sa na porovnanie CFS a LDGM schémy.

Tabuľka 2: Porovnanie CFS a LDGM schémy

	CFS	**LDGM**
Velkosť VK	$(n-k)n$	$(n-k)n/p$
Veľkosť PK	$3(n^2-nk)+k^2$	$[3(n^2-nk)+k^2]/p$
Veľkosť podpisu	n	n

Vidíme, že pri rovnakých parametroch kódu LDGM schéma ponúka p-násobnú úsporu pamäte. Treba však podotknúť, že kódy použité v každej schéme sa líšia a z praktického hľadiska sa volia iné hodnoty parametrov.

3 Návrh a implementácia LDGM schémy

V tejto časti uvedieme návrh a implementáciu LDGM podpisovej schémy spolu s konkrétnymi algoritmami na generovanie kľúčov, podpisov a overovanie.

3.1 Parametre

Nasledujúca tabuľka obsahuje všetky voliteľné parametre a funkcie pre celú podpisovú schému LDGM spolu s odporúčanými hodnotami na dosiahnutie 80 bitovej a 120 bitovej bezpečnosti. [1]

Tabuľka 3: Parametre a funkcie LDGM

Par.	**Popis**	**SL80**	**SL120**
n	dĺžka kódu (n_0p)	9800	24960
k	dimenzia kódu (k_0p)	4900	10000
p	veľkosť cirkulantných matíc	50	80
z	počet riadkov matice a,b	2	2
w_g	váha riadku generujúcej matice G	20	25
w_t	váha riadku/stĺpca matice T	1	1
w_s	váha riadku/stĺpca matice S	9	14
w_c	váha slova c, ktoré určuje funkcia ψ	160	325
H	hashovacia funkcia	-	-
ψ	funkcia, na priradenie slova c k správe m	-	-
ϕ	funkcia, zobrazujúca slová na vektory s pevnou váhou	-	-

3.2 Funkcia ϕ

Dôležitou súčasťou pre celú implementáciu je zvoliť funkciu ϕ, ktorá jednoznačne mapuje vektor dĺžky n na vektor dĺžky l s váhou t. Pre tento účel zvolíme algoritmus, ktorý navrhol Sendrier v rámci CFS schémy v [9]. Tento algoritmus má lineárnu zložitosť a oproti algoritmu, ktorý použili autori LDGM schémy ponúka výrazné zrýchlenie celého výpočtu.

Implementácia ako ju navrhol Sendrier je uvedená v algoritme 11. [9]

Algoritmus 11 Funkcia ϕ

Vstup: Dĺžka slova - n, váha slova - t, prúd bitov - B

Výstup: Vektor dĺžky n s váhou t

```
t_tuple ← BtoCW(n, t, 0, B)
return convertTupleToVector(t_tuple)

function BtoCW(n, t, δ, B)
    if t = 0 then
        return
    else if n ≤ t then
        return δ, BtoCW(n − 1, t − 1, 0, B)
    else
        d ← (n − (t−1)/2)(1 − 1/2^(1/t))
        if read(B, 1) = 1 then
            return BtoCW(n − d, t, δ + d, B)
        else
            i ← DecodeFD(d, B)
            return δ + i, BtoCW(n − i − 1, t − 1, 0, B)
        end if
    end if
end function

function DecodeFD(d, B)
    u ← ⌈log_2(d)⌉
    δ ← read(B, u − 1)
    if δ ≥ 2^u − d then
        δ ← 2δ + read(B, 1) − 2^u + d
    end if
end function
```

Pozn. $read(n, B)$ je funkcia, ktorá prečíta n bitov z prúdu v desiatkovej reprezentácii a posunie prúd o n bitov.

3.3 Funkcia ψ

Na zamaskovanie chybového slova, ktoré tvorí podstatu podpisu a má špecifický tvar $[0k|s]$ (viď algoritmus 9), k nemu pripočítame kódové slovo, ktoré sa odvodí v závislosti od správy. Toto slovo musí mať váhu nižšiu alebo rovnú ako predpísaný parameter schémy w_c. Existuje veľa spôsobov, ako takéto slovo určiť a v algoritme 12 je uvedený postup, ktorý používame v našej implementácii. Generujúca matica G je špeciálne vytvorená, aby každý riadok mal vopred určenú váhu w_g a kedže je to náhodná matica s nízkou váhou, je malá pravdepodobnosť, že jednotky v riadkoch sú na rovnakých pozíciách. Preto na určenie slova s váhou menšou alebo rovnou w_c stačí spočítať w_c/w_g riadkov z matice G.

Algoritmus 12 Funkcia ψ

Vstup: Správa m, váha výsledného slova w_c
Výstup: Kódové slovo s váhou $\leq w_c$

$b \leftarrow \log_2(k)$
$count \leftarrow w_c/w_g$
$h \leftarrow H(m)$
$c \leftarrow 0_n$
for $i \leftarrow 0$ **to** $count$ **do**
  $index_{row} \leftarrow binToDec(h[(ib+1) : (i+1)b]) + 1$
  $c \leftarrow c \oplus G[index_{row}]$
end for
return c
Pozn. G - generujúca matica, k - počet riadkov G, H - hash. funkcia
$binToDec$ je funkcia, ktorá pre reťazec bitov vráti reprezentáciu v desiatkovej sústave

---

V pseudokóde algoritmu 12 predpokladáme, že $\log_2(k)(w_c/w_g) \geq length(H(m))$.

3.4 Generovanie kľúčov

Generovanie páru kľúčov sa skladá z viacerých krokov

- Vygenerovať generujúcu maticu G
- Vygenerovať maticu Q
- Vygenerovať maticu S
- Vypočítať maticu H_{pub}

Algoritmus 13 Generovanie kľúčov

Vstup: Parametre kódu a schémy - *params*

Výstup: Pár kľúčov

$G \leftarrow generateGenMatrix(params)$
repeat
  $Q \leftarrow generateMatrixQ(params)$
  $Q_{inv} \leftarrow tryToInvert(Q)$
until $exists(Q_{inv})$
repeat
  $S \leftarrow generateMatrixS(params)$
  $S_{inv} \leftarrow tryToInvert(S)$
until $exists(S_{inv})$

$H_{pub} \leftarrow buildMatrixH(G, Q_{inv}, S_{inv})$
$PublicKey \leftarrow H_{pub}$
$PrivateKey \leftarrow G, Q_{inv}, S_{inv}$
return $PrivateKey$, $PublicKey$

Pseudokód generovania kľúčov je zhrnutý v algoritme 13

Každý krok generovania rozoberieme v osobitnej časti. Podľa návrhu schémy potrebujeme inverzné matice Q^{-1} a S^{-1}, takže matice Q a S musia byť regulárne. Pseudokód v algoritme 13 ukazuje, že sme na generovanie matíc Q a S zvolili stratégiu generuj a testuj a teda, ak sa nám nepodarí vygenerovnú maticu invertovať, generovanie opakujeme.

Invertovanie kvazicyklických matíc zohráva pri generovaní kľúčov podstatnú úlohu. Preto predtým, ako si priblížime algoritmy na generovanie matíc Q a S, sa budeme venovať invertovaniu kvázicyklických matíc.

3.5 Invertovanie QC matice

Binárne cirkulantné matice veľkosti n môžu byť reprezentované polynómom $p(x) \in GF(2)[x]/(x^n - 1)$. Polynóm $p(x)$ je stupňa maximálne $n - 1$ a reprezentuje prvý riadok cirkulantnej matice, i-ty riadok matice je reprezentovaný polynómom $x^{i-1}p(x)$ pre $i \in 1, ..., n$.

Kvázicyklická matica je matica, ktorej všetky prvky sú polynómy z $GF(2)[x]/(x^n - 1)$. Každý polynóm predstavuje jeden cirkulantný blok. Pre kvázicyklickú maticu môžeme riadkovo ekvivalentné operácie rozšíriť o operácie, ktoré pracujú nad blokmi. To znaľmená, že na počítanie inverznej matice môžeme použiť Gaussovu eliminačnú metódu nad okruhom polynómov. Kvázicyklickú maticu M veľkosti n rozšírime zprava o jednotkovú maticu (tiež v QC tvare). Rožšírenú maticu upravíme do redukovaného stupňovitého tvaru. Maticu M^{-1} potom tvoria stĺpce $[n+1, 2n]$.

$$\left[\begin{array}{ccc|cccc} p_{1,1}(x) & \cdots & p_{1,n}(x) & 1 & 0 & \cdots & 0 \\ p_{2,1}(x) & \cdots & p_{2,n}(x) & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n,1}(x) & \cdots & p_{n,n}(x) & 0 & 0 & \cdots & 1 \end{array}\right] \longrightarrow \left[\begin{array}{cccc|ccc} 1 & 0 & \cdots & 0 & p'_{1,1}(x) & \cdots & p'_{1,n}(x) \\ 0 & 0 & \cdots & 0 & p'_{2,1}(x) & \cdots & p'_{2,n}(x) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & p'_{n,1}(x) & \cdots & p'_{n,n}(x) \end{array}\right]$$

Algoritmus 14 Invertovanie QC matice

Vstup: QC matica M

Výstup: QC matica M^{-1} ak existuje, inak 0

```
M_eye ← [M|I]
for i ← 1 to n do
    for j ← i to n do
        if existsPolynomialInverse(M_eye[j][i]) then
            inverse ← polynomialInverse(M_eye[j][i])
            swap(M_eye[i], M_eye[j])
            M_eye[i] ← M_eye[i] * inverse
            break
        end if
    end for
    if exists(inverse) then
        for j ← 1 to n do
            if j = i then
                continue
            end if
            M_eye[j] ← M_eye[j] + M_eye[i] * M_eye[j][i]
        end for
    else
        return 0
    end if
end for
return M_eye[1 : n][n + 1 : 2n]
```

Na to, aby sme pomocou jedného riadku mohli eliminovať ostatné, potrebujeme nájsť vedúci prvok (pivot). V polynomickej reprezentácii to znamená nájsť v stĺpci prvok, ktorý sa dá invertovať modulo $x^n - 1$. Inverziu hľadáme pomocou Euklidovho rozšíreného algoritmu. Ak nájdeme pivot, pripočítame k ostatným riadkom taký násobok riadku, ktorý obsahuje pivot, aby sme v stĺpci nad aj pod ním dostali nuly. Ak v stĺpci pivot nenájdeme, buď matica nie je regulárna a teda inverzia neexistuje, alebo je štruktúra matice taká, že jednoduchou Gaussovou elimináciou nie sme schopní inverziu nájsť. Je dôležité zdôrazniť, že nie sme schopní z výsledku algoritmu určiť či inverzia existuje alebo nie. Ak inverzia neexistuje, algoritmus je neúspešný, ale môže nastať prípad, keď inverzia existuje, ale napriek tomu algoritmus inverziu nevypočíta. Úspešnosť algoritmu do veľkej miery zavisí od veľkosti cirkulatných blokov. Čím viac faktorov má polynóm $x^n - 1$, tým je väčšia šanca, že náhodný polynóm s ním bude súdeliteľný a teda k nemu neexistuje inverzný prvok. Ak je malá šanca nájsť inverzný prvok, znižuje to celkovú šancu na úspech Gaussovej eliminácie, pretože nevieme nájsť pivot. Predchádzajúce tvrdenie sa potvrdzuje aj vo výsledkoch meraní, ktoré sú uvedené v samostatnej kapitole. Napríklad pri veľkosti bloku 21 počítame s polynómom $x^{21}-1$. Tento polynóm má rozklad nad $GF(2)$ $(x+1)(x^2+x+1)(x^3+x+1)(x^3+x^2+1)(x^6+x^4+x^2+x+1)(x^6+x^5+x^4+x^2+1)$. Polynóm $x^{19} - 1$ má rozklad $(x+1)(x^{18}+x^{17}+x^{16}+x^{15}+x^{14}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1)$. Invertovanie by teda malo byť výrazne úspešnejšie pri velkosti bloku 19 ako pri veľkosti 21, čo aj potvrdzujú merania.

3.6 Generovanie matice G

Generovanie náhodnej matice G je pomerne priamočiare. G je kvázicyklcká matica tvaru $[I|X]$, kde matica X je náhodná matica, v ktorej každý riadok má rovnakú váhu. Parametre, ktoré vstupujú do generovania sú počet blokov v riadku n, počet blokov v stĺpci k, veľkosť bloku p a váha riadku w_g. Pseudokód generovania je uvedený v algoritme 15

Algoritmus 15 Generovanie matice G

Vstup: Počet blokov v riadku - n, počet blokov v stĺpci - k, veľkosť bloku - p, váha riadku w_g

Výstup: Generujúca matica G

```
X ← 0_{k,n−k}
for i ← 1 to k do
    w ← w_g − 1
    while w > 0 do
        p(x) ← X[i][rand(n)]
        if wt(p(x)) < p then
            addRandBit(p(x))
            w ← w − 1
        end if
    end while
end for
G ← [I|X]
return G
```

Pozn. $rand(n)$ vráti celé číslo z rozsahu $[1, n]$, $addRandBit(p(x))$ pridá jednotku na náhodnú pozíciu v polynóme $p(x)$

3.7 Generovanie matice Q

Matica Q sa skladá z matíc R a T, ktoré sú blokovo cirkulantné. Matica R vznikne ako Kroneckerov súčin $a^T b \otimes 1_{p,p}$. Matice a, b náhodne vygenerujeme v závislosti od parametra z. Matica T sa skladá z cirkulantných matíc s rozmerom $p \times p$ a váha každého riadku a stĺpca je w_t. Generovanie kvazicyklických matíc s pevnou váhou stĺpcov a riadkov je popísane v algoritme 16.

Algoritmus 16 Generovanie QC matice s pevnou váhou

Vstup: Počet cirkulantných blokov - n, veľkosť bloku - p, predpísaná váha w_t

Výstup: Kvázicyklická matica s predpísanou váhou riadkov a stĺpcov

```
blocks ← 0_n
i ← w_t
if odd(w_t) then
    blocks[1] ← 1
    i ← i − 1
end if
while i > 0 do
    r ← rand(n)
    if blocks[r] ≤ n − 2 then
        blocks[r] ← blocks[r] + 2
        i ← i − 2
    end if
end while
T ← 0_{n,n}
for i ← 1 to n do
    for j ← 1 to n do
        T[i][j] ← randPolyOfWeight(blocks[(i + j)  mod n])
    end for
end for
X ← permuteRowBlocks(X)
X ← permuteColumnBlocks(X)
return X
```

3.8 Generovanie matice S

S je náhodná kvázicyklická matica s pevnou váhou riadkov a stĺpcov. Na jej generovanie môžeme použiť algoritmus 16, ktorý sme použili pri generovaní matice Q

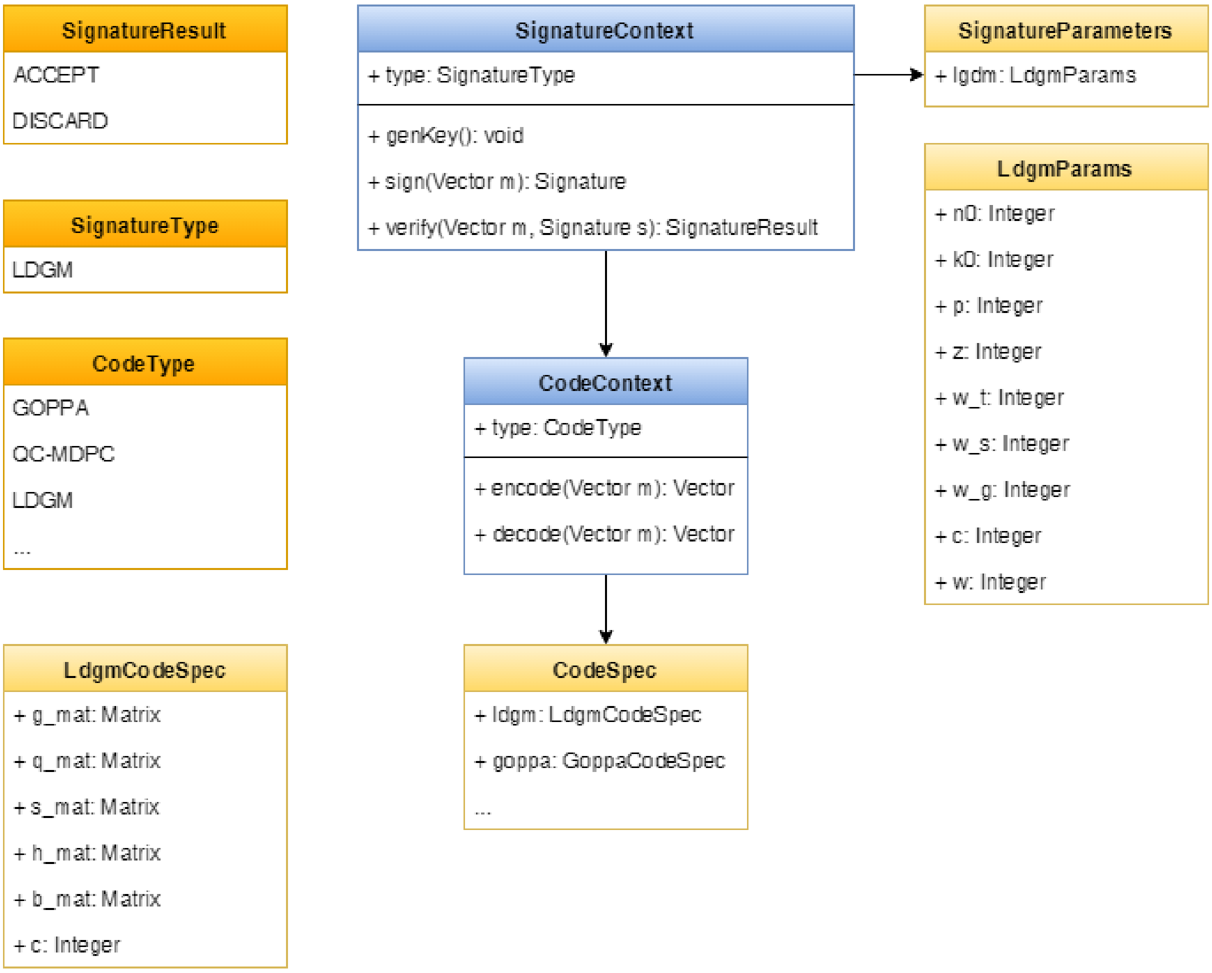
3.9 Výpočet matice H_{pub}

Predtým, ako sa pustíme do výpočtu matice H_{pub}, potrebujeme inverzné matice ku Q a S. Ak nevieme k niektorej z matíc nájsť inverziu, opakujeme generovanie a skúsime nájsť inverziu znova. Ak sa nám úspešne podarí nájsť inverziu k obom maticiam, maticu H_{pub} vypočítame ako súčin matíc $Q^{-1}HS^{-1}$.

3.10 Implementácia v BitPunch

BitPunch je opensource kryptografická knižnica napísaná v jazyku C, ktorú postupne vyvíjajú študenti FEI STU. Primárne zameranie knižnice je na postkvantovú kryptografiu založenej na dekódovacom probléme [10]. Táto práca rozširuje funkčnosť o digitálne podpisy, konkrétne o LDGM schému. BitPunch obsahoval jednoduchú implementáciu kvázicyklických matíc, ktoré sa používali pri QC-MDPC implementácii McElieceovho kryptosystému. Táto implementácia však bola prispôsobená špeciálne pre potreby QC-MDPC schémy a musela byť rozšírená pre všeobecné použitie. Do knižnice bolo pridaných veľa funkcií, ktoré pracujú s kvázicyklickými maticami, vektormi a polynómami nad $GF(2)$. Dokumentácia je uvedená v prílohe. Spôsob, akým sú digitálne podpisy integrované do BitPunch knižnice je načrtnutý na diagrame.

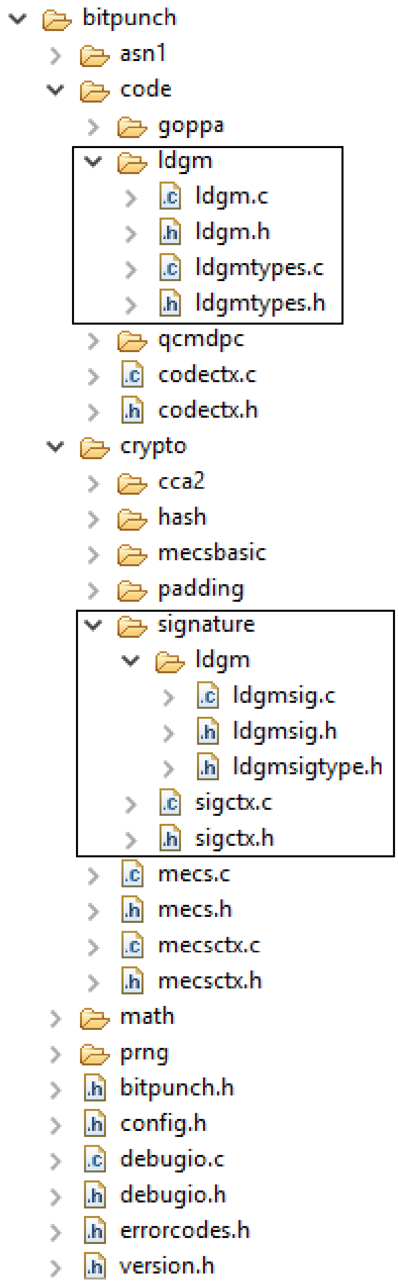
Obrázok 1: Diagram štruktúr v BitPunch



SignatureContext slúži ako premenná prostredia, ktorá obsahuje špecifikáciu kódu a parametre pre podpisovú schému. Štruktúra *CodeSpec* je implementovaná ako union.

Tento spôsob používania union dátového typu má simulovať návrhový vzor *Strategy* [11], ktorý však nie je možné kvalitne implementovať v jazyku C, pretože C nie je objektovo orientovaný jazyk.

Obrázok 2: Organizačná štruktúra v BitPunch



Na obrázku s organizačnou štruktúrou projektu BitPunch sú vyznačené časti, ktoré pribudli ako súčasť tejto práce.

Kompletné zdrojové kódy k implementácii sa nachádzajú v prílohe.

4 Výsledky meraní

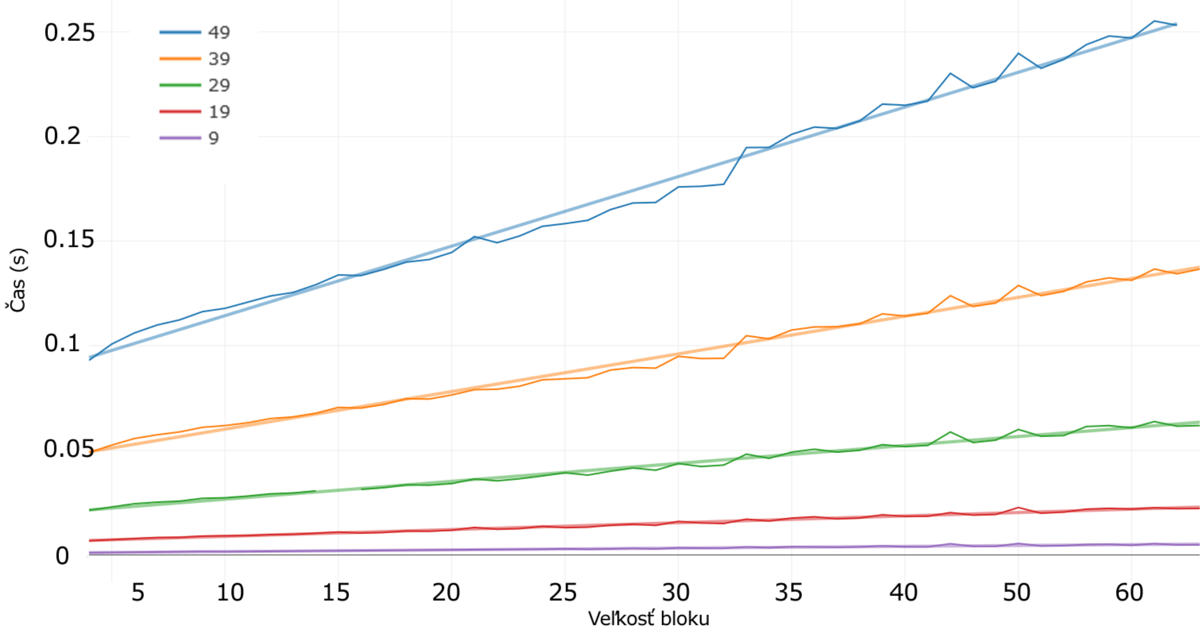
BitPunch s LDGM podpismi sme testovali na zostave Intel Core2 Quad Q8200 @ 2.33GHz, 4GB RAM s operačným systémom Windows 10. Knižnica bola pri testovaní kompilovaná s GCC kompilátorom s O3 optimalizáciou. V rámci testovania sme testovali úspešnosť a časovú zložitosť invertovania kvázicyklických matíc, výkonnosť životného cyklu LDGM podpisovej schémy, čo znamená generovanie kľúčov, podpisovanie náhodnej správy a overenie podpisu. Výsledky sme porovnali s výkonnosťou digitálneho podpisu implementovaného v OpenSSL.

4.1 Invertovanie QC matíc

Vstupné parametre testovania invertovania QC matíc boli počet cirkulantných blokov a veľkosť bloku. Výstupom testu bola úspešnosť invertovania zo 100 pokusov a priemerný čas výpočtu.

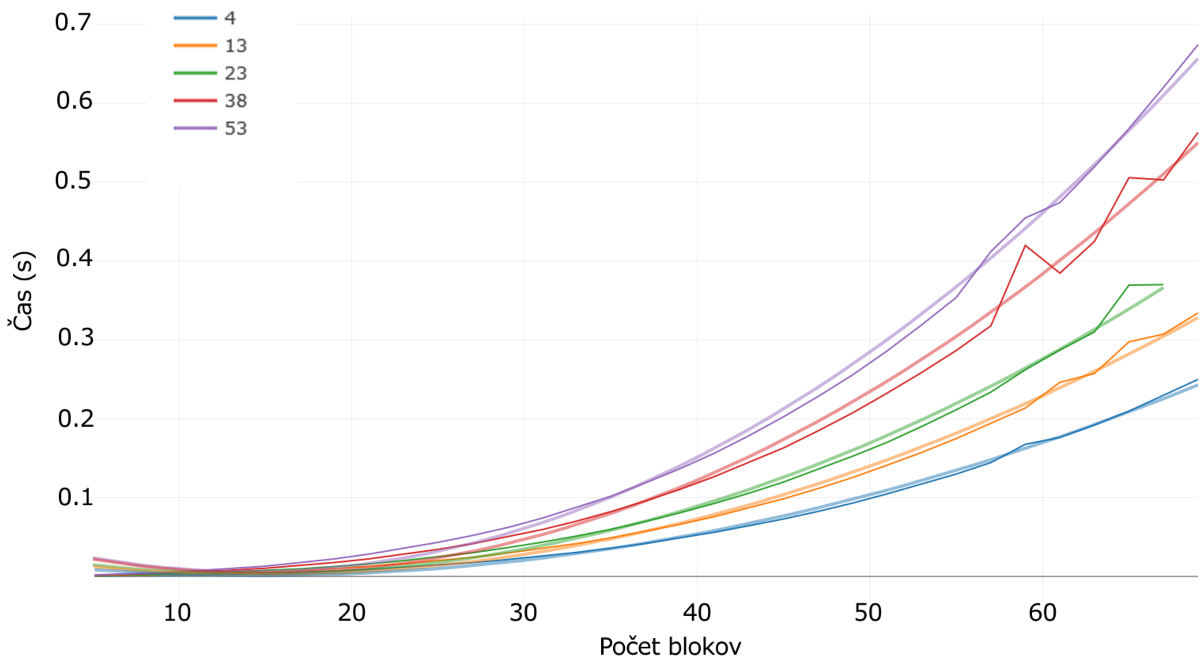
Parameter počtu blokov sme testovali s hodnotami 5, 7, 9, ..., 71 a veľkosť bloku od 4 po 53. Čas výpočtu inverznej matice je kvadraticky závislý od počtu blokov a lineárne od veľkosti bloku. Toto tvrdenie vyplýva z návrhu algoritmu a potvrdzujú ho aj výsledky testovania.

Obrázok 3: Graf závislosti výpočtového času od veľkosti cirkulatných blokov



Na grafe sú zobrazené časové závislosti od veľkosti blokov pre rôzne veľké matice. Pod krivkami sú naznačené regresné priamky.

Obrázok 4: Graf závislosti výpočtového času od počtu blokov v matici

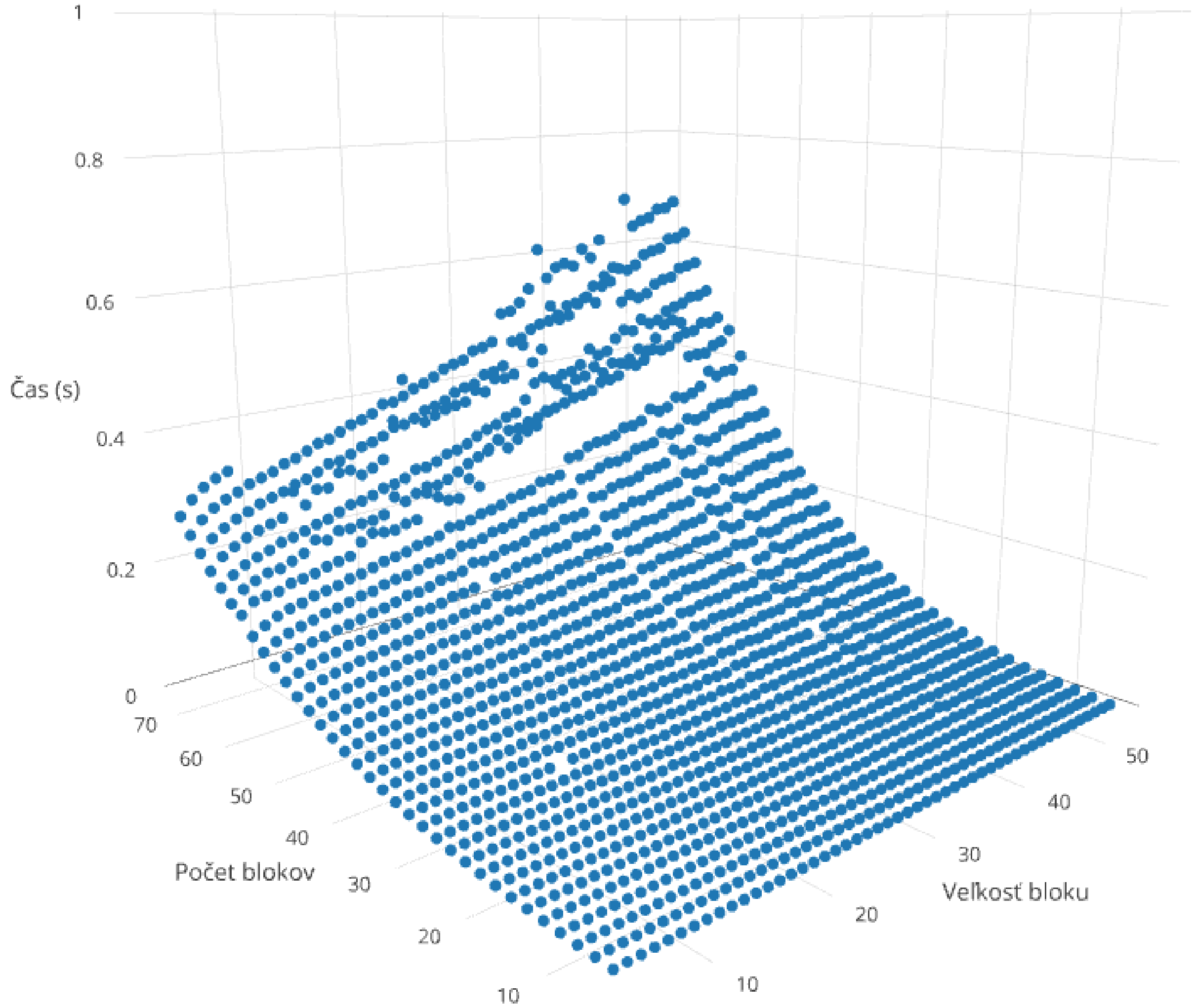


Predchádzajúci graf zobrazuje časové závislosti od počtu blokov pre rôzne veľkosti cirkulantných blokov. Pod krivkami sú naznačené kvadratické regresné krivky.
V tabuľke 4 sú uvedené hodnoty pre vybrané parametre testovania, kompletné výsledky sú uvedené v prílohe.

Tabuľka 4: Výsledky časovej zložitosti invertovania QC matíc

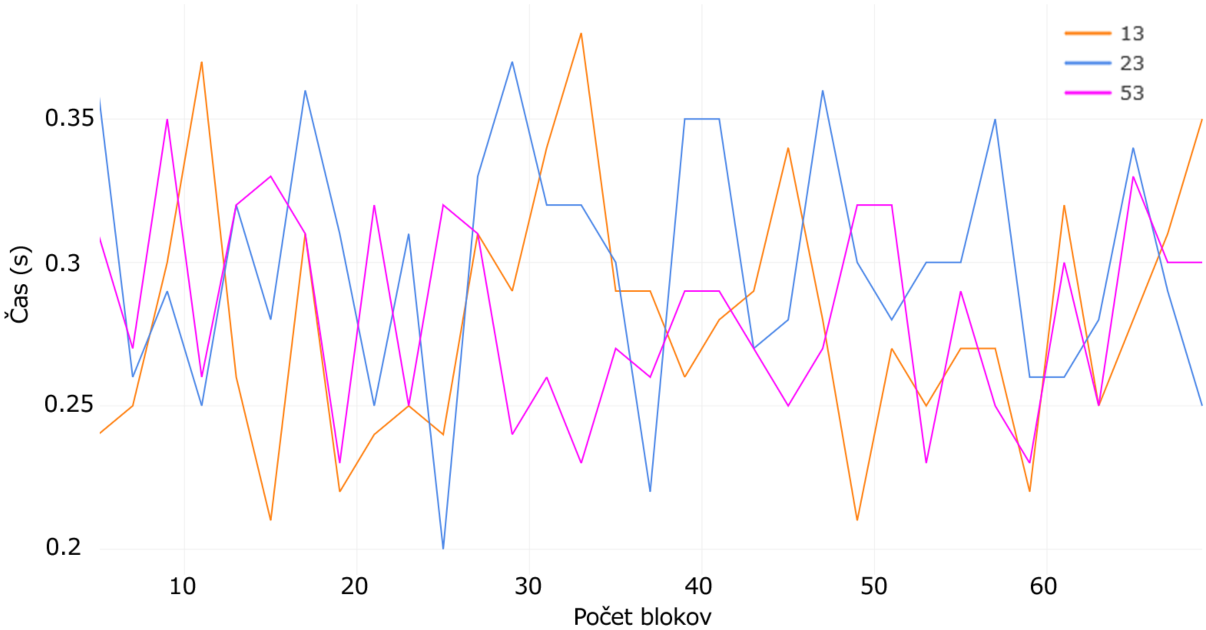
Počet blokov	**Veľkosť bloku**	**Priemerný čas (s)**
5	4	0.0003
9	19	0.0022
13	31	0.0065
21	41	0.0234
31	49	0.0721
41	52	0.1543
51	53	0.2860
63	53	0.5195

Obrázok 5: Graf závislosti výpočtového času od počtu blokov a veľkosti blokov



Z výsledkov testovania nevyplýva, že úspešnosť invertovania závisí od počtu blokov v matici.

Obrázok 6: Graf závislosti úspešnosti od počtu blokov



Ako bolo uvedené v časti, ktorá sa venovala implementácii, úspešnosť nezavisí priamo od veľkosti a počtu blokov ale od počtu faktorov polynómu $x^n - 1$.

Tabuľka 5: Výsledky pre najúspešnejšie veľkosti blokov

Veľkosť bloku	**Úspešnosť (%)**
16	33.7
32	33.6
43	33.6
37	33.5
52	33.4
8	33.4
46	33.0
34	33.0
38	33.0
41	33.0
23	32.7
47	32.6
4	32.5
17	32.4

V tabuľke 5 sú zobrazené veľkosti blokov, pre ktoré je úspešnosť najvyššia. Je zaujímavé pozorovať, že sa v nej nachádzajú všetky mocniny dvojky, ktoré boli súčasťou testovania. Pre veľkosti blokov, ktoré sú v tvare 2^n, má polynóm $x^{2^n} - 1$ nad $GF(2)$ rozklad $(x-1)^{2^n}$, teda malý počet rôznych faktorov a preto vyššia šanca na úspech.

Tabuľka 6: Výsledky pre najneúspešnejšie veľkosti blokov

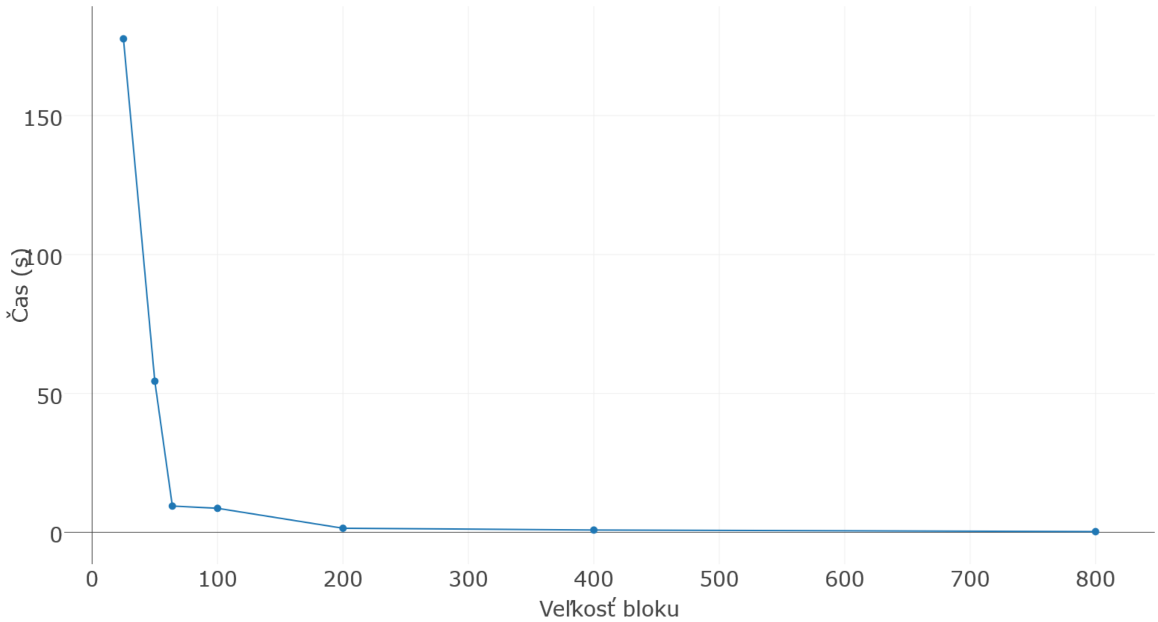
Veľkosť bloku	**Úspešnosť (%)**
21	2.8
42	3.0
45	4.0
15	4.6
30	4.7
36	6.0
27	7.1
35	7.2
7	7.3
28	7.4
18	7.6
24	7.7
49	7.7

V tabuľke 6 sú zobrazené výsledky pre bloky, ktoré mali najnižšiu úspešnosť. Vidieť, že vo výsledkoch sú hodnoty, pre ktoré majú polynómy veľa faktorov a ich spoločné násobky.

4.2 Výkonnosť LDGM

Pri testovaní LDGM implementácie sme sa zamerali hlavne na testovanie s odporúčanými parametrami, ale testovali sme aj ako sa prejavia zmeny v parametroch na časovej zložitosti. Merali sme časovú zložitosť generovania kľúčov, podpisovania a overovania. Podľa odporúčaných parametrov je veľkost generujúcej matice 9800×4900 a veľkosť blokov je 50. Generujúcu maticu teda tvorí 196×98 blokov. Testovali sme však aj iné násobky počtu blokov s veľkosťou blokov. Kedže pri generovaní kľúčov sa počítajú inverzné matice a časová zložitosť počítania inverzie závisí kvadraticky od počtu blokov, čím menej blokov tým rýchelšie je generovanie. Toto je vidieť aj na výsledkoch testov. Ďalšie zrýchlenie generovania vieme dosiahnuť, ak zvolíme veľkosť blokov takú, pre ktorú má polynóm $x^n - 1$ málo faktorov. Na druhú stranu, kedže chceme maximálne využiť zmenšenie veľkosti kľúča, je vhodné zvoliť veľkosť bloku v okolí $\sqrt{k}$, kde k je veľkosť generujúcej matice.

Obrázok 7: Graf závislosti času generovania od veľkosti blokov



Tabuľka 7: Výsledky meraní pre LDGM implementáciu

Počet blokov	**Veľkosť blokov**	**Generovanie (s)**	**Podpis (s)**	**Overovanie(s)**
12	800	0.253	0.022	0.291
25	400	0.827	0.010	0.275
49	200	1.461	0.005	0.224
98	100	8.641	0.004	0.239
153	64	9.449	0.009	0.225
196	50	54.393	0.004	0.221
392	25	177.634	0.007	0.157

Z analýzy výsledkov, ktoré sú v tabuľke 7 sme usúdili, že najlepšie výsledky a najefektívnejšie zmenšenie veľkosti kľúča je pre veľkosť bloku 64.

Pre porovnanie výkonnosti s RSA2048 uvádzame tabuľku s výsledkom merania OpenSSL.

	Podpis (s)	**Overovanie (s)**
RSA2048	0.012	0.0003
LDGM-SL80	0.004	0.2390
LDGM-SL120	0.017	2.3250

Záver

Táto diplomová práca ponúkla prehľad code-based kryptosystémov, code-based podpisových schém a úspešne implementovala LDGM podpisovú schému, čím splnila všetky vytýčené ciele. Výkonnosť podpisovania tejto implementácie prekonáva RSA2048 v knižnici OpenSSL. Do budúcnosti ponúka priestor na zlepšenie výkonnosti overovania, ktoré je momentálne pomalšie ako RSA v OpenSSL, treba však dodať, že OpenSSL je dlhoročný projekt, ktorý je vysoko optimalizovaný a RSA je lepšie preskúmaný kryptosystém ako LDGM. Testovanie implementácie ukázalo, že z praktického hľadiska je vhodnejšie používať iné vstupné parametre, ako odporúčajú autori.

Kapitoly, ktoré sa venovali prieskumu postkvantových kryptosystémov a podpisových schém spolu s ukážkovým príkladom LDGM schémy boli prezentované vo februári 2016 na konferencii Norwegian-Slovakian Workshop in Crypto v nórskom Bergene [5].

Počas písania práce boli oznámené nové teoretické útoky na LDGM podpisovú schému. Predpokladáme, že autori LDGM schémy v blízkej dobe prídu s úpravami kryptosystému, ktoré zvýšia odolnosť voči predstaveným útokom. Dovtedy môže naša implementácia slúžiť na testovanie a realizáciu možných útokov.

Zoznam použitej literatúry

[1] Baldi, M., Bianchi, M., Chiaraluce, F., Rosenthal, J., and Schipani, D. Using ldgm codes and sparse syndromes to achieve digital signatures. In Post-quantum cryptography. Springer, 2013, pp. 1–15.

[2] Bernstein, Daniel J and Buchmann, Johannes and Dahmen, Erik. Post-quantum cryptography. Springer Science & Business Media.

[3] Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., and Smith-Tone, D. Report on post-quantum cryptography. National Institute of Standards and Technology Internal Report 8105 (2016).

[4] Courtois, N. T., Finiasz, M., and Sendrier, N. How to achieve a mceliece-based digital signature scheme. In Advances in Cryptology—ASIACRYPT 2001. Springer, 2001, pp. 157–174.

[5] Dobročka, P. Code-base digital signatures. In Proceedings of Norwegian-Slovakian Workshop in Crypto (2016), Slovenská Technická Univerzita.

[6] Goldreich, Oded. Foundations of cryptography: volume 2, basic applications. Cambridge university press.

[7] McEliece, R. A public-key cryptosystem based on algebraic coding theory. The Deep Space Network Progress Report 42-44 (1978), 114–116.

[8] Niebuhr, R., Meziani, M., Bulygin, S., and Buchmann, J. Selecting parameters for secure mceliece-based cryptosystems. International Journal of Information Security 11, 3 (2012), 137–147.

[9] Sendrier, N. Encoding information into constant weight words. In Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on (2005), IEEE, pp. 435–438.

[10] Uhrecký, František. Implementácia kryptografickej knižnice s McEliece kryptosystémom, Diplomová práca. Slovenská Technická Univerzita.

[11] Vlissides, J., Helm, R., Johnson, R., and Gamma, E. Design patterns: Elements of reusable object-oriented software. Reading: Addison-Wesley 49, 120 (1995), 11.

Prílohy

A BitPunch s dokumentáciou

Kompletné zdrojové kódy ku knižnici BitPunch rozšírenej o LDGM podpisovú schému sa nachádzajú na priloženom CD v priečinku bitpunch. Dokumentácia v anglickom jazyku je súčasťou zdrojového kódu vo formáte JavaDocs.

B Výsledky meraní

Výsledky meraní invertovania kvázicyklických matíc a výkonnosť LDGM schémy sú na priloženom CD v priečinku measurements.

C Demo aplikácia

Spustiteľná aplikácia, ktorá ponúka generovanie kľúčov, podpisovanie a overovanie sa nachádza spolu s manuálom na priloženom CD v priečinku demo.