SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DEŠIFROVANIE V QC-LDPC MCELIECEOVOM KRYPTOSYSTÉME

TÍMOVÝ PROJEKT

2018 **Forks**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DEŠIFROVANIE V QC-LDPC MCELIECEOVOM KRYPTOSYSTÉME

TÍMOVÝ PROJEKT

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:	Ing. Viliam Hromada, PhD., Mgr. Tomáš Fabšič, PhD.

Bratislava 2018 **Forks**

Poďakovanie

Chceli by sme sa poďakovať vedúcim tímového projektu Mgr. Tomášovi Fabšičovi, PhD. a Ing. Viliamovi Hromadovi, PhD. za...

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek a značiek

GF - Galois Field

RSA - Rivest–Shamir–Adleman

Zoznam algoritmov

Úvod

1 Ponuka

Riešiteľský tím v zložení: Bc. Nikoleta Furičková, Bc. Juraj Karásek, Bc. Matej Ohradzanský, Bc. Peter Radvan a Bc. Lukáš Štrba na tomto mieste predkladá záväznú ponuku na riešenie problému s pracovným názvom: *Dešifrovanie v QC-LDPC McElieceovom kryptosystéme*. Projekt budeme riešiť pod vedením Mgr. Tomáša Fabšiča, PhD. a Ing. Viliama Hromadu, PhD. v rámci predmetu **Tímový projekt**.

V časti Riešiteľský kolektív stručne predstavíme členov tímu. Zameriame sa na ich schopnosti a skúsenosti, ktoré súvisia s problematikou tímového projektu. Zadanie a úlohy, ktoré z neho plynú budú opísané v časti Anotácia tímového projektu. Nasledovať bude časť, v ktorej vyjadríme našu dôveru v post-kvantovú kryptografiu a jej využitia v *blízkej* budúcnosti. Ponuku ukončíme popisom organizácie projektu.

1.1 Riešiteľský kolektív

Tím pozostáva z piatich študentov aplikovanej informatiky na Fakulte elektroniky a informatiky Slovenskej Technickej Univerzity. Členovia tímu sú spolužiaci už 4 roky a panujú medzi nimi priateľské vzťahy. Pracovný názov nášho kolektívu je *Forks*.

Bc. Nikoleta Furičková
Pozícia v tíme: Analytička

Je absolventkou bakalárskeho štúdia na Fakulte elektrotechniky a informatiky Slovenskej Technickej Univerzity v Bratislave, v študijnom programe Aplikovaná informatika, odbor Bezpečnosť informačných systémov. Bakalárske štúdium ukončila vypracovaním bakalárskej práce s názvom: *Invertovateľnosť blokovo cyklických matíc*. Bakalársku prácu robila pod vedením Mgr. Tomáša Fabšiča, PhD., pričom získala cenné skúsenosti a poznatky, ktoré sa dajú dobre využiť pri riešení tímového projektu.

Bc. Juraj Karásek
Pozícia v tíme: Developer

Je absolventom bakalárskeho štúdia na Fakulte elektrotechniky a informatiky Slovenskej Technickej Univerzity v Bratislave, v študijnom programe Aplikovaná informatika, odbor

Bezpečnosť informačných systémov. Bakalárske štúdium ukončil vypracovaním bakalárskej práce s názvom: *Porovnanie vybraných mechanizmov distribuovaného konsenzu z hľadiska IT bezpečnosti.*

Bc. Matej Ohradzanský
Pozícia v tíme: Developer

Je absolventom bakalárskeho štúdia na Fakulte elektrotechniky a informatiky Slovenskej Technickej Univerzity v Bratislave, v študijnom programe Aplikovaná informatika, odbor Bezpečnosť informačných systémov. Bakalárske štúdium ukončil vypracovaním bakalárskej práce s názvom: *Lúštenie substitučných šifrier.* Bakalársku prácu robil pod vedením prof. Ing. Pavla Zajaca, PhD. a v apríli tohto roku sa zúčastnil SVOČ-ky (*Študentská vedecká a odborná činnosť*), pričom sa umiestnil na prvom mieste vo svojej kategórii.

Bc. Peter Radvan
Pozícia v tíme: Web Developer

Je absolventom bakalárskeho štúdia na Fakulte elektrotechniky a informatiky Slovenskej Technickej Univerzity v Bratislave, v študijnom programe Aplikovaná informatika, odbor Bezpečnosť informačných systémov. Bakalárske štúdium ukončil s vyznamenaním. Pod vedením Ing. Viliama Hromadu, PhD. vypracoval bakalársku prácu s názvom: *ABC kryptosystém*, pri tvorbe ktorej získal množstvo vedomostí z oblasti post-kvantovej kryptografie, ktoré predstavujú značný náskok v analytickej časti projektu.

Bc. Lukáš Štrba
Pozícia v tíme: Analytik a vedúci riešiteľského kolektívu?

Je absolventom bakalárskeho štúdia na Fakulte elektrotechniky a informatiky Slovenskej Technickej Univerzity v Bratislave, v študijnom programe Aplikovaná informatika, odbor Bezpečnosť informačných systémov. Bakalárske štúdium ukončil s vyznamenaním. Pod vedením Ing. Viliama Hromadu, PhD. vypracoval bakalársku prácu s názvom: *Kubický ABC kryptosystém*, pri tvorbe ktorej získal množstvo vedomostí z oblasti post-kvantovej kryptografie. Tieto poznatky sú stále aktuálne a predstavujú dobrý základ potrebný na úspešné vypracovanie projektu.

1.2 Anotácia tímového projektu

Pokrok vo vývoji kvantového počítača má vážne dôsledky aj pre kryptografiu. Je známe, že dostatočne vykonné kvantové počítače budú vedieť efektívne riešiť problém faktorizácie čísla na prvočísla a problém diskrétneho logaritmu. Na náročnosti riešenia týchto problémov je založená bezpečnosť v súčasnosti používaných asymetrických kryptosystémov (napríklad RSA). To znamená, že v prípade existencie dostatočne výkonného kvantového počítača by súčasné asymetrické kryptosystémy už neboli bezpečné. Niektoré odhady hovoria, že takto výkonné kvantové počítače by mohli existovať už o 10 rokov. Je preto dôležité, pracovať na vývoji nových asymetrických kryptosystémov, ktoré budú odolné voči útokom kvantového počítača, a ktoré by mohli nahradiť súčasné asymetrické kryptosystémy. Na dôležitosť tejto témy upozornil aj americký inštitút pre štandardy a technológiu *NIST* v správe Report on Post-Quantum Cryptography. NIST zároveň vyhlásil súťaž Post-Quantum Cryptography Standardization Process s cieľom navrhnúť nové kryptografické štandardy odolné voči kvantovým počítačom. Do súťaže prišlo vyše 60 návrhov kryptosystémov. Tieto návrhy budú v najbližších rokoch verejne analyzované vedeckou komunitou s cieľom vybrať najlepších kandidátov.

Viacero návrhov zaslaných do súťaže je založených na QC-LDPC McElieceovom kryptosystéme. Jedná sa o variant McElieceovho kryptosystému, v ktorom sa využívajú QC-LDPC kódy. Pri šifrovaní sa správa prevedie na kódové slovo QC-LDPC kódu a ku kódovému slovu sa pridá chyba. Pri dešifrovaní je potrebné túto chybu odstrániť. Chybu je možné odstrániť použitím tajného kľúča, ktorý obsahuje matice H a Q. Na odstránenie chyby je možné použiť dve metódy: buď sa použije dekódovací algoritmus pre QC-LDPC kódy, ktorý využíva maticu H, alebo sa použije dekódovací algoritmus pre QC-MDPC kódy, ktorý používa maticu H*Q. Cieľom práce je porovnať efektívnosť týchto dvoch metód. Úlohy, ktoré z toho vyplývajú sú nasledovné:

- naštudovanie si princípov fungovania QC-LDPC McElieceovho kryptosystému
- oboznámenie sa s implementáciou QC-LDPC McElieceovho kryptosystému v knižnici BitPunch (implementácia je v jazyku C)
- porovnanie efektívnosti dvoch vyššie spomenutých metód na odstránenie chyby zo správy zašifrovanej QC-LDPC McElieceovym kryptosystémom

1.3 Motivácia

Pre projekt sme sa rozhodli preto, že si uvedomujeme hrozbu, ktorú by predstavovalo zostrojenie dostatočne silného kvantového počítača pre dnešný svet. Dôkazom aktuálnosti tejto problematiky je aj fakt, že existujú viaceré medzinárodné projekty organizované napr. *NATO* alebo *VEGA*, do ktorých je zapojený aj Ústav informatiky a matematiky FEI STU. Skúmanie rôznych post-kvantových kryptosystémov je nesmierne dôležité a sme radi, že sa toho môžeme zúčastniť. Získané vedomosti môžu predstavovať cenný zdroj nových informácií pre členov tímu, ktorí sa už podrobnejšie stretli s post-kvantovou kryptografiou a zároveň dobrý základ pre členov, ktorí sú noví a vedomosti iba zbierajú. Veríme, že naše výsledky z tímového projektu pomôžu k napredovaniu výskúmu v tejto oblasti.

1.4 Organizácia projektu

Z diskusie členov riešiteľského kolektívu so zadávateľmi tímového projektu vyplynulo, že jediný možný termín pravidelných konzultácií je utorok ráno (cca 08:00). S návrhom súhlasili všetci prítomní. Neprítomný Bc. Peter Radvan s časom stretnutia dodatočne súhlasil. Z pravidelných stretnutí sa priebežne tvoria zápisnice, ktoré sú spolu s ostatnými materiálmi a informáciami dostupné na stránke projektu. Súčasťou vypracovania projektu je aj odovzdanie dokumentácie, v ktorej bude opísaný matematický model QC-LDPC McElieceovho kryptosystému, ako aj nami zistené výsledky meraní. Na vypracovanie analytickej časti projektu neuvažujeme nad žiadnymi špeciálnymi hardvérovými požiadavkami, ale v prípade spúšťania väčšieho množstva výpočtov uvažujeme nad využitím klastra STU v Bratislave.

2 Teoretická časť

2.1 Úvod do problematiky

Pri prenose informácií vznikajú prirodzene rôzne otázky. Nezachytáva niekto našu komunikáciu, nemení nám niekto obsah našich správ? Ako odpoveď na tieto otázky sa dáta začali rôzne modifikovať do podoby, ktorej by tretia strana nebola schopná rozumieť t.j. *šifrovať*. Pomocou šifrovania vieme informáciu bezpečne dostať od odosielateľa k prijímateľovi. Kryptografia ako vedná disciplína sa časom vyvíjala. Postupne vznikali nové a bezpečnejšie kryptosystémy. Avšak zostrojením dostatočne silného kvantového počítača by bolo možné tieto kryptosystémy prelomiť. Súčasná asymetrická kryptografia sa spolieha na problémy z teórie čísel, ako sú napríklad *faktorizácia prvočísel* a výpočet *diskrétneho logaritmu*. Tieto problémy by boli podľa P. W. Shora [5] riešiteľné v polynomiálnom čase na kvantovom počítači. Tým sa stávajú asymetrické kryptosystémy a podpisové schémy, ako napr. RSA a DSA zraniteľné a prelomiteľné.

Našťastie, existuje skupina matematických problémov, označované ako *NP-úplné* problémy, ktoré by sa nedali rýchlejšie vyriešiť ani s použitím kvantovného počítača. Je teda rozumné skúmať takéto problémy a na nich zakladať nové kryptosystémy odolné voči útokom na kvantovom počítači. Veda, ktorá sa zaoberá touto problematikou sa nazýva *post-kvantová kryptografia*. Jedným z takýchto problémov je aj tvz. *dekódovací problém*, ktorý sa zaoberá dekódovaním náhodného lineárneho kódu. Vyhodnotenie úspešnosti rôznych spôsobov dekódovania bude predmetom skúmania tohto tímového projektu.

2.2 Lineárne kódy

Pri prenose správy prostredníctvom komunikačného kanálu môže nastať chyba prenosu a preto príjemnca nedostane pôvodnú odoslanú správu. Riešením tohto problému sa zaoberá *teória kódovania*. Pridaním *redundantnej* informácie je možné *detekovať*, alebo dokonca *opraviť* určitý počet chýb. Takto sa správa dĺžky k rozšíri o r redundantných bitov a celková dĺžka správy narastie na $k + r$. Ďalej budú vysvetlené základné pojmy a definície.

Uvažujme, že prenášané správy, *kódové slová*, sú tvorené bitmi, t.j. len hodnotami 0 alebo 1. Každé kódové slovo binárneho lineárneho kódu C má dĺžku n a samotný lineárny kód C má dimenziu k. Tento lineárny kód C je lineárnym podpriestorom vektorového priestoru

F_q^n. Konečné pole F_q obsahuje q prvkov. Počet kódových slov je q^k.

Lineárny kód vieme reprezentovať prostredníctvom generujúcej alebo kontrolnej matice. Generujúca matica G má veľkosť $k \times n$. Riadky matice sú tvorené bázou pre lineárny kód C. Báza obsahuje k lineárne nezávislých vektorov z vektorového priestoru F_2^n. Vieme, že existuje $n-k$ lineárne nezávislých vektorov, ktoré sú kolmé na všetky vektory z lineárneho kódu C.

Lineárny kód vieme vygenerovať predpisom $C = \{aG|a \in F_q^k\}$. Štandardná forma generujúcej matice G má tvar $G = [I_k|P]$, kde I_k reprezentuje jednotkovú maticu veľkosti $k \times k$ a matica P má rozmery $k \times r$. Kontrolná matica H predstavuje generujúcu maticu pre *duálny kód* C' ku kódu C. Jej tvar vieme zapísať predpisom $H = [-P^T|I_{n-k}]$ a platí $HG^T = -P^T + P^T = 0$.

Uveďme príklad generujúcej a kontrolnej matice binárneho lineárneho kódu [7,4]. Generujúca matica G lineárneho kódu je definovaná štyrmi bázovými vektormi, ktoré sa nachádzajú v 7-rozmernom binárnom priestore. Počet všetkých možných kódových slov je 2^4.

Generujúca matica $G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$.

Kontrolná matica $H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$.

Lineárny kód máme definovaný a môžeme vytvoriť kódové slovo. Napríklad, nech vektor $v = (1, 0, 1, 0)$. Takto vytvorený vektor vynásobíme maticou G a dostaneme kódové slovo $k = (1, 0, 1, 0, 1, 0, 1)$ lineárneho kódu C. Správnosť vytvorenia kódového slova k si vieme overiť práve pomocou kontrolnej matice H, tak že po vynásobení dostaneme nulový vektor.

$$H \cdot k' = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Nižšie uvádzame ďalšie potrebné informácie dôležité pre problematiku tímového projektu. Kontrolná matica H pre binárny kód, môže mať rôznu hustotu. Ak má nízku hustotu, tak kód nazývame *LDPC kód* (z anglického Low Density Parity-check Code). Naopak, ak má kontrolná matica H miernu hustotu, kód nazývame *MDPC kódom* (z anglického Moderate Density Parity-check Code). Kontrolná matica MDPC obsahuje viac jednotiek ako matica pre LDPC kód.

Pod pojmom *cyklická matica* chápeme maticu, ktorá vznikne postupnou rotáciou prvého riadku.

Príklad cyklickej matice: $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$.

Príklad blokovo cyklickej matice: $\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$.

Pri QC (kvázi–cyklických) kódoch existuje kontrolná bloková matica H zložená z cyklických matíc. Cyklickým posunom v kódovom slove sa vytvorí taktiež kódové slovo. Ak pre kód existuje takáto matica, tak lineárny kód označujeme *QC-LDPC* resp. *QC-MDPC*.

Hammingova váha vektora predstavuje počet nenulových prvkov vektora. Hammingova vzdialenosť dvoch vektorov predstavuje počet prvkov, ktoré sa odlišujú. Môžeme to chápať aj ako počet nutných zmien na to, aby sa jeden vektor pretransformoval na druhý.

2.3 McEliece kryptosystém

McEliece kryptosystém je asymetrický šifrovací algoritmus vyvinutý v roku 1978 profesorom Robertom McElieceom [4]. Bol to prvý kryptosystém svojho druhu, ktorý pri šifrovacom procese využíva náhodnosť. Napriek tomu, že si v kryptografickej komunite nikdy nenašiel príliš veľké prijatie, je jedným z kandidátov na post-kvantovú kryptografiu vďaka jeho odolnosti voči útokom, ktoré používajú Shorov algoritmus. Jeho bezpečnosť je založená na probléme dekódovania lineárnych kódov, ktorý je klasifikovaný ako *NP-úplný* problém [1]. Pri jeho konštrukcii je potrebné zvoliť dekódovací algoritmus, ktorý dokáže opraviť t chýb, ktoré sú zanesené do kódu pri šifrovaní. Pôvodný algoritmus využíva tzv. *Goppa kódy*.

McEliece kryptosystém pozostáva z troch častí a to konkrétne generovanie klúčov, šifrovanie a deterministický dešifrovací algoritmus. Všetci používatelia takéhoto kryptosystému musia zdielať rovnaké bezpečnostné parametre n, k, t, kde n je dimenzia priestoru, k je dimenzia kódu (podpriestoru) a t je minimálny počet opraviteľných chýb.

2.3.1 Generovanie klúčov

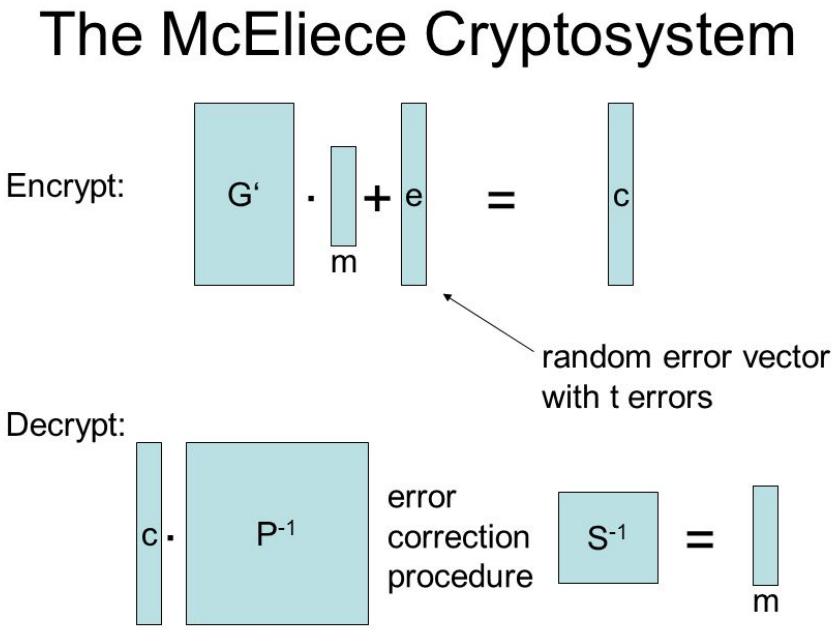
Alica si zvolí generujúcu maticu G s rozmermi $k \times n$ a lineárny kód C, ktorý je generovaný touto maticou. Náhodne vygeneruje maticu S s rozmermi $k \times k$, ktorá nie je singulárna. Náhodne vygeneruje permutačnú maticu P s rozmermi $n \times n$. Vypočíta maticu $G' = SGP$ s rozmermi $k \times n$. Potom jej verejný kľúč je (G', t) a súkromný kľúč sú matice (S, G, P).

2.3.2 Šifrovanie

Bob chce poslať správu m Alici. Vygeneruje zašifrovanú správu $x = mG' + e$, kde e je náhodný chybový vektor s dĺžkou n, ktorého Hammingova váha nie je väčšia ako t. Táto podmienka zabezpečuje, že kód bude možné pri dešifrovaní dekódovať.

2.3.3 Dešifrovanie

Alica vypočíta inverznú maticu P^{-1}. Ďalej vypočíta $x' = xP^{-1}$. Potom použije dekódovací algoritmus, aby z x' dostala m'. Posledný krok, ktorý musí urobiť je vypočítať $m = m'S^{-1}$, aby sa dostala k správe, ktorú jej Bob poslal.



Obrázok 1: Schéma McEliece kryptosystému [2]

McElieceov kryptosystém má viacero výhod. Jednou z nich je napríklad, že šifrovanie a dešifrovanie je rýchlejšie ako pri RSA. Vďaka *Niederreiterovej schéme* môže byť dokonca použitý na realizáciu podpisovania súborov. Hlavnou nevýhodou tohto kryptosystému je veľkosť kľúčov. To je dôvod prečo sa tento kryptosystém veľmi neuchytil v praxi. Jednou z mála aplikácií, ktorá používala toto šifrovanie bola aplikácia *Entropy*, čo je decentralizovaná *peer-to-peer* komunikačná sieť, ktorá bola navrhnutá, aby bola odolná proti akejkoľvek cenzúre.

2.4 QC-LDPC McEliece

Napriek tomu, že nie je známy praktický útok na McElieceov kryptosystém s Goppa kódmi, mal pôvodný návrh jednu nevýhodu - veľkosť kľúčov. Existuje však spôsob, ako túto veľkosť znížiť a to výber kódov z veľkej automorfnej grupy, akými sú kvázi-cyklické kódy. Avšak mnohé z nich sú napadnuteľné algebraickým útokom, pri ktorom môže útočník vytvoriť systém rovníc a tento systém riešiť pomocou Groebnerových báz. Úspešnosť takéhoto útoku vychádza z vlastností algebraickej štruktúry použitej rodiny kódov. Jedným z dôvodov prečo je takýto útok realizovateľný je, že kvázi-cyklická štruktúra umožňuje drastické zníženie počtu neznámych v systéme. Hrozbe takéhoto útoku môžeme predísť použitím kódov, ktoré nemajú algebraickú štruktúru.

Vhodnými kandidátmi pre tento účel sú LDPC kódy, ktoré nemajú algebraickú štruktúru. Rozdiel voči QC-MDPC spočíva v riedkej kontrolnej matici H, ktorá umožňuje efektívnu korekciu chýb pri dekódovaní.

Ak by chcela Alica dostávať šifrované správy od Boba, potrebuje si vytvoriť verejný kľúč K_V, ktorým Bob bude šifrovať správy adresované Alici a súkromný kľúč K_S, ktorým bude Alica správy dešifrovať. Pre vytvorenie súkromneho kľúča si Alica potrebuje zvoliť náhodnú riedku kontrolnú maticu H a maskovacie matice S a Q. Matice S a Q sú kvázi-cyklické štvorcové invertovateľné matice, pričom S je hustá matica a Q riedka. Tieto tri matice tvoria Alicin súkromný kľúč K_S. Pre vytvorenie verejného kľúča K_V potrebuje Alica z matice H odvodiť generujúcu maticu G. Matice $H_1 ... H_n$ sú riedke kvázi-cyklické štvorcové matice.

$$H = \left[\begin{array}{cccc} H_1 & |H_2 & |... & |H_n \end{array} \right]$$

Nech matica I je jednotková štvorcová matica, potom generujúcu maticu G vytvorí Alica nasledovne:

$$G = \left[\begin{array}{c|c} \mathrm{I} & (H_1 H_n^{-1})^T \\ & \vdots \\ & (H_n H_1^{-1})^T \end{array} \right]$$

Po dostaní generujúcej matice môže Alica vypočítať svoj verejný kľúč K_V ako $S^{-1}GQ^{-1}$. Platí, že $GH^T = 0$. Tak ako sa dá vypočítať generujúca matica z kontrolnej, dá sa vypočítať aj kontrolná matica z generujúcej. Dôvodom prečo je práve generujúca matica odvodzovaná, je kvôli tomu, že je problém dostať riedke H takýmto výpočtom.

Následne môže Bob zašifrovať správu m jej vynásobením Aliciným verejným kľúčom K_V a pridaním chyby e. Pričom Hammingova váha chybového vektoru nie je väčšia ako t, čo je počet chýb, ktorý dokáže kód generovaný maticou G opraviť.

$$x = mK_V + e$$

Alica prijatú správu x dešifruje nasledovne: šifrovanú správu x vynásobí maticou Q, čím dostane $x' = mS^{-1}G + eQ$ (Q je riedka a teda Hammingova váha eQ je malá). x' je kódové slovo QC-LDPC kódu + chyba. Ďalej pomocou kontrolnej matice určí syndróm slova a použitím vhodného dekódovacieho algoritmu odstráni chyby. Tak dostane správu m' rovnú mS^{-1}, ktorú stačí vynásobiť maticou S a získa pôvodnú správu m.

2.5 Bit-flippingové algoritmy

Bit-flippingové algoritmy pre dekódovanie LDPC kódov prešli dlhú cestu v oblasti výskumu a modifikácie [6]. Boli navrhnuté rôzne varianty, ako napríklad *WBF* (Weighted BF) alebo *MWBF* (Modified Weighted BF). Prvý algoritmus tohto typu vymyslel R. G. Gallager. Jeho ukážku si uvedieme nižšie. Z tohto základného algoritmu potom vznikali rôzny nástupcovia, ktorí však zachovávali hlavnú myšlienku a dedili stratégiu z Gallagerovho algoritmu. Ďalším typom sú *GDBF* (Gradient Descent BF), ktoré sú odvodené z formulácie Gradient Descent-u čo je optimalizačný algoritmus na nájdenie minima funkcie.

Gallagerov bit-flippingový algoritmus je iteratívny dekódovací algoritmus pre LDPC kódy. V prvom kroku sa vypočíta syndróm správy vynásobením kontrolnej matice H a transponovaného kódu x^T. Tento výsledok môžeme označiť ako *UPC* (Unsatisfied Parity-check Equations). Následne pre každý bit zo správy x vyrátame počet UPC rovníc, do ktorých prispieva. Pre tento výpočet môžeme použiť *Tannerov graf*. Bit, ktorý prispieva do najviac UPC rovníc preklopíme, opäť vypočítame syndróm správy ale teraz už s aktualizovaným kódom x'. Ak je výsledok nulový vektor, tak sme správne dekódovali. Ak nie, pokračujeme rovnako v ďalšej iterácii pokým syndróm nie je nulový vektor, alebo

sa nedosiahne maximálny počet iterácií. Pre ilustráciu si uvedieme jednoduchý príklad s jednou iteráciou.

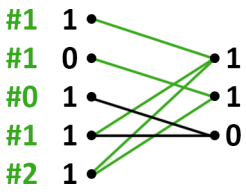
Majme kontrolnú maticu H a kódové slovo x:

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \; x = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Ďalej si vypočítame syndróm správy s:

$$s = Hx^T = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Pomocou Tannerovho grafu vieme určiť množstvo UPC rovníc, do ktorých jednotlivé bity prispievajú. Na ľavo je transponované kódové slovo x a na pravej strane je syndróm správy. Konkrétny bit prispieva do UPC rovnice, ak existuje spojenie v grafe z daného bitu do ktoréhokoľvek bitu syndrómu s hodnotou 1.



Obrázok 2: Tannerov graf na zistenie najpodozrivejšieho bitu

Z Tannerovho grafu vieme určiť najpodozrivejší bit, v našom prípade piaty, nakoľko zasahuje až do dvoch UPC rovníc. Preto ho preklopíme. Dostávame nové x' a opäť môžeme vypočítať syndróm s':

$$x' = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$s' = H(x')^T = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

V tomto prípade je výsledok nulový vektor, čo je ukazovateľom toho, že sme kódové slovo úspešne dekódovali.

Ako bolo spomenuté vyššie, existujú mnohé modifikácie bit-flippingových algoritmov. Najväčší rozdiel medzi jednotlivými modifikáciami možno pozorovať pri nastavovaní *prahovej hodnoty*, podľa ktorej sa určuje či sa má daný bit preklopiť alebo nie. Vo všeobecnosti, ak je táto hodnota nízka, algoritmus je rýchlejší ale kódové slovo nemusí byť správne dekódované. Naopak ak je táto hodnota vysoká, algoritmus je pomalší, no zväčšuje sa pravdepodobnosť, že povedie k správnemu výsledku. Preto by sa dalo považovať správne nastavenie tejto hodnoty za kritické.

Modifikáciou bit-flippingového algoritmu môžu byť napríklad, že sa prahová hodnota predpočítava podľa návrhu R. G. Gallagera z jeho práce *Low Density Parity-check Codes*. Ďalší prístup volí túto prahovú hodnotu ako maximálny počet UPC v každej iterácii. Treťou možnosťou je určenie prahovej hodnoty podľa vzorca, v ktorom sa od maximálneho počtu UPC rovníc odpočíta malá hodnota δ. Takýto krok má za následok preklopenie väčšieho počtu bitov na rozdiel od druhej modifikácie, kde sa preklopí jeden alebo len malý počet bitov [3].

3 Praktická časť

Záver

Zoznam použitej literatúry

[1] Berlekamp, E., McEliece, R., and Van Tilborg, H. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory 24*, 3 (1978), 384–386.

[2] Dowsley, R., van de Graaf, J., Müller-Quade, J., and Nascimento, A. C. Oblivious transfer based on the mceliece assumptions. In *International Conference on Information Theoretic Security* (2008), Springer, pp. 107–117.

[3] Andrej Gulyás. *Implementácia QC-MDPC McEliecovho kryptosystému*.

[4] Mceliece, R. J. A public-key cryptosystem based on algebraic. *Coding Thv 4244* (1978), 114–116.

[5] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review 41*, 2 (1999), 303–332.

[6] Wadayama, Tadashi and Nakamura, Keisuke and Yagita, Masayuki and Funahashi, Yuuki and Usami, Shogo and Takumi, Ichi. *Gradient descent bit flipping algorithms for decoding LDPC codes*.

Prílohy

A Štruktúra elektronického nosiča

\
\Tímový projekt.pdf
\Zdrojové súbory - *upravená knižnica BitPunch*

B Používateľská príručka

Elektronický nosič obsahuje dokumentáciu k Tímovému projektu a upravenú verziu knižnice BitPunch.

Používateľ musí mať nainštalovaný kompilátor jazyka C. Na kompilovanie knižnice *BitPunch* je možné použiť priložený *makefile*:

- *Kompilovanie knižnice* v príkazovom riadku: v priečinku *../BitPunch/lib* zadať príkaz **make**
- *Spustenie implementácie*: v priečinku *../BitPunch/lib/dist/test* spustiť novovytvorený binárny súbor pomocou príkazu **./BitPunch**