VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

McEliece viešojo rakto kriptografinės sistemos saugumo tyrimas

Study of the security of McEliece public-key cryptosystem

Magistro baigiamasis darbas

Atliko: Andrius Versockas (parašas)

Darbo vadovas: lekt. dr. Gintaras Skersys (parašas)

Recenzentas: doc. dr. Vilius Stakėnas (parašas)

Vilnius – 2018

Santrauka

Šio darbo tikslas – ištirti McEliece viešojo rakto kriptografinės sistemos saugumą. Darbo uždaviniai: (1) Išanalizuoti McEliece kriptosistemos sandarą, veikimo principus ir atskirų sudedamųjų dalių įtaką bendram sistemos saugumui. (2) Ištirti ir realizuoti apibendrintos informacijos dekodavimo atakos algoritmą su mažais parametrais. (3) Ištirti ir realizuoti žinomo dalinio teksto atakos algoritmą. (4) Ištirti pranešimo persiuntimo atakos ir susijusių pranešimų saugumo spragą ir realizuoti efektyvų algoritmą atakoms įgyvendinti. (5) Įvertinti, kokie būtų atakų rezultatai su didesniais parametrais arba modifikuoti kriptosistemos operacijas, kad ji būtų atspari atakoms. (6) Nustatyti tokius kriptosistemos parametrus, su kuriais kriptosistema būtų saugiausia prieš pateiktas atakas.

Tyrimo metu buvo išanalizuotos ir realizuotos atakos prieš McEliece kriptografinę sistemą, pateikiami gauti vykdymo su asmeniniu kompiuteriu rezultatai ir saugiausi parametrai m ir t siekiant apsisaugoti nuo šių atakų. Taip pat pateikiamos kriptosistemos ir atakų modifikacijos, skirtos jų saugumui ir efektyvumui pagerinti, bei atakų vidutinių vykdymo laikų prognozės su didesniais m ir t saugumo parametrais.

Raktiniai žodžiai: McEliece viešojo rakto kriptosistema, kriptoanalizė, kriptografinės sistemos saugumas, kriptografija, apibendrinta informacijos dekodavimo ataka, žinomo dalinio teksto ataka, pranešimų persiuntimo ataka, susijusių pranešimų ataka

Summary

The aim of this paper is to investigate the security of McEliece public-key cryptographic system. Main tasks of the work are: (1) To analyze the McEliece cryptographic system, it's structure, operation principles and the influence of individual components on the overall security of the system. (2) To investigate and implement the generalized information-set decoding attack algorithm with small parameters. (3) To investigate and implement the known partial plaintext attack algorithm. (4) To investigate message resend and related message security flaw and implement an efficient attack algorithm for it. (5) To evaluate attack results with larger security parameters or modify cryptosystem operations to make it resistant to attacks. (6) Identify the most secure cryptosystem parameters against the analyzed attacks.

In this study the attacks against the McEliece cryptographic system were analyzed and implemented, the results of execution time with a personal computer were shown, the safest parameters m and t were found. Cryptosystem and attack modifications are also provided to improve their security and effectiveness, as well as predictions of average attack runtime with higher m and t security parameters.

Keywords: McEliece public-key cryptosystem, cryptanalysis, cryptographic system security, cryptography, generalized information-set decoding attack, known partial plaintext attack, related message attack, message resend attack

Turinys

Įvadas

Šiuolaikinės visuomenės gyvenimas sunkiai įsivaizduojamas be elektroninės erdvės suteikiamų informacijos įvedimo, saugojimo, apdorojimo ir pateikimo patogumų. Šioje erdvėje yra atliekamos itin svarbios operacijos tokios kaip elektroniniai bankiniai pavedimai, rinkimų balsavimai, svarbių konfidencialių dokumentų perdavimai ir kt., dėl to svarbu užtikrinti informacijos saugumą. Duomenų saugumui užtikrinti yra naudojamos matematiniais metodais paremtos kriptografinės sistemos. Sparčiai tobulėjant kvantiniams kompiuteriams kriptografinių sistemų saugumas tampa vis svarbesnis, kadangi yra atrastas kvantinis algoritmas galintis įveikti plačiausiai paplitusias RSA, Diffie–Hellman, El Gamal ir elipsinių kreivių viešojo rakto kriptografines sistemas. Iki šiol viešojo rakto kriptografinės sistemos, kurios remiasi klaidas taisančiais kodais, yra laikomos saugiomis ir geriausiu kandidatu „post kvantinei“ kriptografijai [DMR11]. Viena iš labiausiai žinomų klaidas taisančius kodus naudojančių sistemų yra McEliece viešojo rakto kriptografinė sistema.

Pagrindinis darbo tikslas yra ištirti McEliece kriptografinės sistemos saugumą. Norint pasiekti šį tikslą darbe buvo iškelti šie uždaviniai:

1. Išanalizuoti McEliece kriptosistemos sandarą, veikimo principus ir atskirų sudedamųjų dalių įtaką bendram sistemos saugumui.
2. Ištirti ir realizuoti apibendrintos informacijos dekodavimo atakos algoritmą su mažais parametrais.
3. Ištirti ir realizuoti žinomo dalinio teksto atakos algoritmą.
4. Ištirti pranešimo persiuntimo atakos ir susijusių pranešimų saugumo spragą ir realizuoti algoritmą atakoms įgyvendinti.
5. Įvertinti, kokie būtų atakų rezultatai su didesniais parametrais arba modifikuoti kriptosistemos operacijas, kad ji būtų atspari atakoms.
6. Nustatyti tokius kriptosistemos parametrus, su kuriais kriptosistema būtų saugiausia prieš pateiktas atakas.

Sąvokų apibrėžimai

- Jeigu sveikųjų skaičių a ir b skirtumas $a - b$ dalijasi iš m, $m \in \mathbb{N}$, tai sakoma, kad a lygsta b moduliu m ir rašoma $a \equiv b \pmod{m}$. Priešingu atveju, jei $a - b$ nesidalija iš m, sakoma, kad a nelygsta b moduliu m ir rašoma $a \not\equiv b \pmod{m}$. Kai $a \equiv 0 \pmod{m}$, tai skaičius a dalijasi iš m. Reiškiniai $a \equiv b \pmod{m}$ vadinami **lyginiais**.

- **Kūnas** yra matematinė struktūra, kurioje apibrėžtos sudėties, atimties, daugybos ir dalybos operacijos.

- Kūną, turintį baigtinį skaičių elementų, vadinsime **baigtiniu kūnu** ir žymėsime $GF(q)$ arba $\mathbb{F}_q$, kur q – elementų skaičius. Baigtinis kūnas iš q elementų egzistuoja tada ir tik tada, kai $q = p^m$, kur p yra pirminis ir $m \geq 1$.

- Šiame darbe visada bus naudojama abėcėlė $\mathcal{A} = \{0, 1\}$. Vektoriaus x žodžio, sudaryto iš abėcėlės $\mathcal{A}$ simbolių, **Hamingo svoris** (angl. *Hamming weight*) yra jo nenulinių koordinačių skaičius ir žymimas $wt(x)$.

- Dviejų žodžių vektorių u, v sudarytų iš abėcėlės $\mathcal{A}$ **Hamingo atstumu** vadinsime pozicijų, kuriose vektorius u skiriasi nuo vektoriaus v, skaičių ir žymėsime $d(u, v)$.

- Matricą, kurios eilutės ir stulpeliai iš matricos A sukeisti vietomis vadinsime matricos A **transponuota matrica** ir žymėsime $A^\intercal$.

- **Kvadratine matrica** vadinsime matrica, kurios eilučių skaičius n lygus stulpelių skaičiui m.

- **Vienetine matrica** vadinsime kvadratine matrica, kurios visi elementai pagrindinėje įstrižainėje lygūs 1, o likusieji elementai lygūs 0 ir žymėsime I.

- **Gauso metodu** vadinamas tiesinių lygčių sistemos sprendimo būdas, kai elementariaisiais pertvarkymais eliminuojant nežinomuosius siekiama gauti laiptuotos formos pavidalo sistemą. Pertvarkymų metu atsiradus nulinei eilutei, ši eliminuojama. Elementarieji pertvarkymai gali būti:

  - eilučių sukeitimas vietomis
  - kelių eilučių sudėtis
  - matricos eilutės daugyba iš skaičiaus, nelygaus nuliui

  Kadangi šiame darbe yra naudojama dvejetainė aritmetika, dėl to daugybos veiksmai nebus taikomi.

- Kvadratinės matricos A **atvirkštine** vadinama A^{-1} matrica, kuri tenkina lygybes $AA^{-1} = A^{-1}A = I$.

- **Matricos determinantas** yra skaliarinė reikšmė, kurią galima apskaičiuoti iš kvadratinės matricos elementų ir žymima $\det(A)$. Šios skaliarinės reikšmės pagalba galima pasakyti, ar egzistuoja matricai atvirkštinė.

1. Kriptografijos temų apžvalga

Pagal Oksfordo anglų kalbos žodyną [Dic89] kriptografija yra apibrėžiama kaip menas šifruoti ir dešifruoti šifrus. Šis apibrėžimas – istoriškai tikslus, tačiau neaprėpia visos modernios kriptografijos esmės. Pirmiausia, jis yra orientuotas tik į slaptą duomenų perdavimo problemą, tačiau šiuolaikinė kriptografija apima įrankius konfidencialumo išsaugojimui, metodus keistis slaptais raktais, protokolus vartotojų atpažinimui, jų validavimui, elektroninių pinigų, elektroninių aukcionų veikimui ir kita. Taigi galima teigti, kad moderni kriptografija apima matematinių metodų tyrimą skirtą užtikrinti skaitmeninės informacijos, sistemų ir paskirstytų skaičiavimų saugumui prieš priešininkų išpuolius. Taip pat, Oksfordo anglų kalbos žodyno apibrėžimas kriptografiją pateikia kaip meno formą. Iš tiesų, iki XIX amžiaus pabaigos, kriptografija didele dalimi buvo menas, nes šifravimo ir dešifravimo metodai rėmėsi kūrybingumu ir išvystyta nuojauta kaip šios metodai galėtų veikti. Požiūris į kriptografiją kaip į mokslą ir matematinę discipliną pasikeitė tik 1970-ųjų metų pradžioje [KL14; Sta07].

Šiuolaikinė kriptografija (iš gr. *κρυπτός, kryptós* „paslėptas“ + *γράφειν, graphein* „rašyti“) yra paremta matematika ir kompiuterių mokslu. Tai mokslas apie matematinius metodus, kurie užtikrina informacijos slaptumą, duomenų vientisumą, subjekto ir duomenų kilmės autentiškumą. Kriptografijos algoritmų pagalba yra kuriamos kriptografinės sistemos arba kriptosistemos, kurios suteikia jos naudotojui žinučių šifravimo ir dešifravimo metodus bei užtikrina šių operacijų saugumą. Norint apibrėžti kriptografiją, būtina pristatyti informacijos saugumo uždavinius. Šie saugumo uždaviniai turi būti įgyvendinti visiems informacijos apsikeitimo dalyviams – slaptos informacijos siuntėjui, jos gavėjui ir kriptoanalitikui, kuris perima siunčiamą šifrą. Nors šių uždavinių kiekis dažnai priklauso nuo situacijos, galimybių ir reikalavimų, tačiau visos kriptografinės sistemos turi išspręsti keturis pagrindinius uždavinius. Toliau kiekvienas jų yra aptariamas detaliau [KMV+96]:

1. **Duomenų konfidencialumo** (angl. *data confidentiality*) sąvoka dažnai yra tapatina su siunčiamos informacijos privatumu. Šio uždavinio tikslas yra užtikrinti, kad slapta informacija nebūtų pasiekiama teisės neturintiems subjektams, tuo pačiu užtikrinant, kad teisę turintys subjektai galėtų ją pasiekti, t. y. prieiga turi būti suteikiama tik tiems, kurie turi teisę peržiūrėti šiuos duomenis. Įprastu atveju duomenys yra suskirstomi pagal žalos dydį ir tipą, kurie gali būti padaryti, jei jie patenka į nenumatytas rankas. Tada pagal šias kategorijas įgyvendinamos daugiau arba mažiau griežtos priemonės. Dažnai itin svarbiai informacijai naudojami dviejų veiksnių autentiškumo apsaugos procesas (angl. *two–factor authentication*), kurio metu vartotojas turi pateikti dvi identifikavimo priemones, kurių viena paprastai yra fizinis ženklas, toks kaip kortelės ir kitos, ir kažkas įsimintino, kaip antai saugumo kodas. Daugybė metodų siūlo konfidencialumo paslaugą, pradedant nuo fizinės apsaugos, baigiant matematiniais algoritmais, kurie paverčia duomenis nesuprantamais.

2. **Duomenų vientisumo** (angl. *data integrity*) uždavinio tikslas yra nustatyti, ar neturint prieigos teisės duomenyse buvo padaryti pakeitimai. Vientisumas apima duomenų nuoseklumo, tikslumo ir patikimumo palaikymą per visą jų gyvavimo ciklą. Siekiant užtikrinti duomenų vientisumą, reikia gebėti aptikti pašalinių asmenų atliktas duomenų manipuliacijas. Daž-

nai siekiant patikrinti duomenų vientisumą prieš ir po jos perduodamo yra paskaičiuojamos kontrolinės sumos (angl. *checksum*) arba maišos funckijos reikšmės. Vientisumui užtikrinti turi būti nustatytos tam tikros priemonės, leidžiančios nustatyti bet kokius duomenų pakeitimus, kurie gali atsirasti dėl ne žmogaus sukeltų įvykių, tokių kaip elektromagnetinis impulsas (angl. *EMP*) ar serverio avarija. Atsarginės kopijos turi būti prieinamos, kurių pagalba galima būtų atkurti paveiktų duomenų teisingą būseną.

3. **Autentiškumo** (angl. *authentication*) uždavinio tikslas yra susijęs su identifikavimu, kuris yra taikomas tiek subjektui, tiek perduodamai informacijai. Abi pusės, kurios komunikuoja tarpusavyje, turi identifikuoti viena kitą. Perduodama informacija turėtų būti autentifikuojama, t. y. nustatoma jos kilmė, duomenų turinys, siuntimo laikas ir t. t. Dėl šių priežasčių šis uždavinys paprastai yra skirstoma į dvi pagrindines klases: subjekto autentiškumo ir duomenų kilmės autentiškumo. Duomenų kilmės autentiškumas taip pat netiesiogiai užtikrina duomenų vientisumą, pavyzdžiui, jei žinutė yra pakeista, tai bus pasikeitęs ir jos siuntimo šaltinis.

4. **Veiksmų neišsižadėjimas** (angl. *non-repudiation*) neleidžia subjektui neigti anksčiau atliktus veiksmus ar įsipareigojimus. Kylant ginčams dėl subjekto tam tikrų veiksmų, būtina imtis priemonių ginčui išspręsti. Pavyzdžiui, vienas subjektas gali leisti kitam subjektui įsigyti turtą, o vėliau neigti, kad šis leidimas buvo duotas. Šiam ginčui išspręsti reikalinga procedūra su patikima trečiąja šalimi. Svarbių elektroninių dokumentų ar pranešimų veiksmų neišsižadėjimui užtikrinti yra naudojami skaitmeniniai parašai, kurių pagalba užtikrinamas pasirašytų duomenų autentiškumas ir jų apsauga nuo klastojimo.

Pagrindinis kriptografijos tikslas yra tinkamai išspręsti šiuos uždavinius tiek teoriškai, tiek praktiškai. Šių sprendimų pagalba yra nustatomi sukčiavimai ir kitos piktybinės veiklos, atliekama jų prevencija.

1.1. Matematiniai uždaviniai

Kriptografinių sistemų veikimas remiasi matematiniais uždaviniais. Šiame poskyryje aprašomi matematiniai uždaviniai, kuriais remiasi plačiai paplitusios viešojo rakto kriptosistemos bei McEliece kriptosistema. Šie uždaviniai remiasi vienakryptės funkcijos principu, t. y. žinant funkcijos argumentus, jos reikšmė yra apskaičiuojama efektyviu polinominiu algoritmu, o iš funkcijos reikšmės rasti jos argumentus efektyvaus algoritmo nėra žinoma [Sta07]. Kriptografinėje aplinkoje yra protinga daryti prielaidą, kad kriptoanalitikas arba atakuotojas yra galingas ir turi pakankamai skaičiavimo resursų, kad galėtų išspręsti matematinį uždavinį, kuriam yra žinomas algoritmas, kuris per polinominį laiką gali išspręsti didžiąją dalį jo atvejų. Kriptosistemos, kurių saugumas yra grindžiamas tokiais uždaviniais yra laikomos nesaugiomis [KMV+96]. Žemiau pateikti uždaviniai yra laikomi sunkiai išsprendžiami ir šiuo metu efektyvaus būdo išspręsti juos praktikoje nėra žinoma.

1.1.1. Didelių skaičių faktorizavimo uždavinys

Didelių skaičių pirminių daugiklių radimas yra vadinamas skaičiaus faktorizavimu. Pagrindinė aritmetikos teorema teigia, kad bet kuris sveikas skaičius, didesnis už 1, gali būti išreikštas pirminių skaičių sandauga (faktorizuotas) vieninteliu būdu, pavyzdžiui: $999 = 3^3 \cdot 37$. Jei sveikasis skaičius yra pirminis, tada jis gali būti atpažintas per polinominį laiką, tačiau, jei jis yra sudėtinis, tai išskaidyti jį į natūraliuosius pirminius daugiklius yra sunkus skaičiavimo uždavinys. Pavyzdžiui, norint rasti p ir q iš $p \cdot q = N$, kur p, q yra pirminiai skaičiai ir $p < q$, naudojant paprastą dalybos bandymo algoritmą reikėtų skaičių N dalinti iš pirminių skaičių nuo 2 iki $\sqrt{N}$ [Sta07]. Kriptosistemos, kurios remiasi šiuo uždaviniu ir naudoja didelius pirminius skaičius, šiuo metu yra laikomos saugiomis, tačiau tobulėjant kvantiniams kompiuteriams kyla grėsmė jų saugumui. Jau dabar yra žinomas Shoro kvantinis algoritmas [BV97], kuris įveiktų tokias kriptosistemas per polinominį laiką.

1.1.2. Diskretaus logaritmo uždavinys

Tegul p yra pirminis skaičius, o a ir b yra sveiki skaičiai, kurie mod p nėra lygūs nuliui. Tarkime, kad egzistuoja sveikas skaičius x, tada diskretaus logaritmo uždaviniu yra laikoma lygtis:

$$a^x \equiv b \pmod{p}, \tag{1}$$

kur žinant a, b ir pirminį skaičių p reikia rasti x [Sta07]. Nors diskretaus logaritmo ir sveikųjų skaičių faktorizavimo uždaviniai yra skirtingi, tačiau jie turi matematinių sąsajų ir dėl to dažnai praktikoje vieno uždavinio algoritmai pritaikomi kitam [Gre03]. Taip pat, kaip ir sveikųjų skaičių faktorizavimo uždavinio atveju, diskretaus logaritmo uždavinys gali būti efektyviai išspręstas Shoro kvantinio algoritmo pagalba [BV97], tačiau efektyvių metodų išspręsti jį su įprastu kompiuteriu nėra žinoma.

1.1.3. Elipsinės kreivės uždavinių aritmetika

Pirmieji elipsinių kreivių kriptografiniai algoritmai buvo pasiūlyti 1985 metais [Kob87; Mil85], tačiau pradėti plačiau naudoti tik nuo 2004 metų, dėl problemų su patentais, kurios yra aktualios iki šiol [Res09]. Elipsinė kreivė – taškų rinkinys, atitinkantis matematinę lygtį $y^2 = x^3 + ax + b$, kur a,$b \in K$ ir K gali būti realusis, racionalusis, kompleksinis ar kitas algebrinis kūnas. Lygtis pasižymi horizontaliu simetriškumu, pagal Ox ašį, kitaip tariant, jeigu (x,y) yra kreivės taškas, tada ir $(x, -y)$ irgi yra kreivės taškas. Taip pat, bet kuri ne vertikali linija einanti per šią kreivę susikirs ne daugiau kaip trijose vietose [Kob87]. Šios lygties sprendiniais yra laikomos visos galimos x, y poros ir papildomas taškas O, dar vadinamas „begalybės taškas“, nes yra be galo nutolęs nuo Oy ašies. Elipsinės kreivės pagrindinė operacija tarp dviejų kreivės taškų yra vadinama taškų sudėtimi. Tegul E yra elipsinė kreivė ir P,$Q \in E$, tada taškų suma $P + Q$ elipsinėje kreivėje E apibrėžiama pagal šias taisykles [Kob87]:

1. Jei P ir Q – realūs taškai turintys skirtingas x koordinates, tai jų suma vadinamas taškas,

gaunamas brėžiant tiesę per $\overline{PQ}$ iki tol, kol ji susikirs elipsinėje kreivėje E ir rastam šio susikirtimo taškui paimant simetrišką Ox ašies atžvilgiu tašką.

2. Jei P ir Q – realūs taškai turintys vienodas x koordinates ir skirtingas y koordinates, tai $P + Q = O$ (begalybės taškas).

3. Jei taškas P realus ir lygus taškui Q, tai jų suma laikomas taškas, gaunamas brėžiant liestinę elipsinei kreivei E, randant jos susikirtimo tašką su kreivėje E esančiu tašku ir paimant jam simetrišką tašką Ox ašies atžvilgiu.

4. Jei $P = 0$, tai $P + Q = Q$.

Kriptosistemose dažniausiai yra naudojamos elipsinės kreivės E virš baigtinio kūno F_q ir jos taškas P. Galima pastebėti, kad bet kuriam n rasti $n \cdot P$ yra nesudėtinga, pakanka naudoti greito dauginimo (n kartų sudėti P) metodą. Kita vertus, žinant $n \cdot P$ ir P rasti n yra sudėtingas uždavinys, dar vadinamas elipsinės kreivės (diskretaus logaritmo) uždaviniu. Elipsinėmis kreivėmis paremtos kriptografinės sistemos sugeneruoja žymiai mažesnius raktus lyginant su faktorizavimo ar diskretaus logaritmo sistemomis, tačiau dėl to atliekama kvantinė kriptoanalizė prieš šias sistemas bus žymiai veiksmingesnė [Ber09]. Be to, elipsinės kreivėmis paremtos kriptosistemos yra pažeidžiamos Shor algoritmo pagalba.

1.1.4. Tiesinių kodų dekodavimo uždavinys

Nagrinėsime baigtinį kūną, turintį q elementų – F_q, kur $q = p^m$, p – pirminis skaičius, $m \in \mathbb{N}$. Tada, tiesinis kodas C yra vektorinės erdvės F_q^n poerdvis, kur n yra jo žodžių ilgis, k – jo dimensija, o d – minimalus atstumas. Tiesinio kodo C parametrai žymimi $[n,k]$ arba $[n,k,d]$, o visi vektoriai $c_1, \ldots, c_n$ iš C vadinami tiesinio kodo C žodžiais (angl. *codeword*) arba vektoriais bei jie yra pateikiami vektorių pavidalu. Tiesinis kodas C gali turėti daug elementų, dėl to dažniausiai apibrėžimas minimaliu žodžių rinkiniu iš C, dar vadinamu kodo baze, kurio visų galimų rinkinio derinių tiesinės kombinacijos sudaro visus tiesinio kodo C žodžius. Tiesinio kodo C virš baigtinio kūno F_q generuojanti matrica G yra vadinama $k \times n$ matrica virš F_q, kurios eilutės sudaro C kodo bazę, t. y. $C = \{xG \colon x \in F_q^k\}$.

Tarkime, siunčiamas pranešimas $m \in F_q^k$ žodis $c = mG \in C$ per triukšmingą kanalą, kuris prideda prie jo klaidų vektorių $e \in F_q^n$ ir sukuria naują žinutę c'. Norint dekoduoti tokį pranešimą reikia rasti pridėtą klaidų vektorių e, atimti jį iš gauto vektoriaus c', t. y. $c = c' - e$, ir išspręsti tiesinių lygčių sistemą $c = mG$ [MMT11; Ske05; Sta07]. Klaidos ir pranešimo vektorių radimui galime taikyti minimalaus atstumo dekodavimo taisyklę, t.y. lyginame visus kodo žodžius iš tiesinio kodo C su gautu pranešimo vektoriumi c', kol randamas kodo žodis $x \in C$ esantis arčiausiai iš kanalo išėjusiam vektoriui c' ir tenkina sąlygą $d(x,c') = \min_{z \in C} d(z,c')$. Čia funkciją d vadiname Hamingo atstumu. Rastas x – pranešimo žodis, o klaidų vektorių galime apskaičiuoti $e = c' - x$. Šiuo būdu ieškant artimiausio žodžio reikia patikrinti gautą c' su visais C kodo žodžiais. Jeigu pasirinktas didelis tiesinio kodo parametras k, tuomet reikės peržiūrėti didelį skaičių elementų [Sta02].

1.2. Pagrindinės kriptosistemos

Kriptografinės sistemos yra skirstomos į du pagrindinius tipus – simetrinio ir viešojo (arba asimetrinio) rakto kriptosistemas. Simetrinio rakto kriptosistemos pasižymi tuo, kad duomenys yra šifruojami ir dešifruojami tuo pačiu raktu. Simetrinio rakto algoritmai yra tinkami didelio duomenų kiekio šifravimui, kadangi, lyginant su viešojo rakto algoritmais, dažniausiai sukuria trumpesnį šifrą, o šifravimo ir dešifravimo operacijos vyksta daug greičiau. Pagrindinės simetrinio rakto kriptosistemų problemos – kaip su kiekvienu vartotoju saugiai pasidalinti sugeneruotu, idealiu atveju – skirtingu, raktu ir kaip suvaldyti visus padalintus raktus [DH76b; DK07]. Pavyzdžiui, iki viešojo rakto kriptografinių sistemų atsiradimo, raktas buvo perduodamas registruotu paštu, privačiu kurjeriu ar susitikimo būdu [DH76a]. 1976 metais Whitfield Diffie ir Martin Hellman pasiūlė viešojo rakto (angl. *public-key*) kriptosistemos principą, kuris naudoja du skirtingus, bet matematiškai susijusius raktus – viešąjį ir privatųjį, kur viešasis raktas yra prieinamas visiems ir jo pagalba kriptosistema šifruoja išsiunčiamus duomenis, o privatus raktas prieinamas tik gavėjui ir jo pagalba gautas šifras yra dešifruojamas. Taip pat, autoriai pabrėžė, kad tokia kriptosistema būtų laikoma saugia, jeigu jos algoritmas naudotųsi vienakrypte funkcija. Šios funkcijos pagalba, turint tik privatųjį raktą, dešifravimas vyksta greitai, tačiau nustatyti privatųjį raktą iš viešojo yra praktiškai neįmanoma [DH76b]. Kai kurios kriptosistemos, dėl savo savybių, tokių kaip šifravimo ir dešifravimo efektyvumas, saugumas, šifrų dydis ir kitų, yra naudojamos dažniau už kitas. StorageCraft [Bra14] savo mokslo populiarinimo straipsnyje išskiria penkias populiariausias kriptosistemas: Triple DES, Blowfish, Twofish, AES ir RSA. Taigi, toliau bus trumpai apžvelgtos šios penkios kriptosistemos.

DES (angl. *Data Encryption Standard*) yra simetrinės kriptosistemos šifravimo algoritmas, kurį praeito amžiaus aštuntajame dešimtmetyje sukūrė IBM įmonės mokslininkai pasiremiant Horst Feistel sugalvotu algoritmu. Šio algoritmo rakto dydis – 56 bitai. Tuo metu tokio dydžio raktai buvo pakankamai saugūs ir JAV vyriausybė 1977 metais DES priėmė kaip oficialų šifravimo standartą [PT01]. Tačiau sparčiai vystantis informacinėms technologijoms ir atsiradus galimybei vykdyti nuoseklią paiešką (angl. *brute-force*), kuomet iš eilės bandomos visos galimo rakto kombinacijos, DES sistemos raktų dydis kriptoanalitikams nekėlė jokių iššūkių juos dešifruoti, o kriptosistema tapo nebesaugi. Šiuo metu DES sistema yra beveik nenaudojama, tačiau jos pagrindu 1993 metais buvo sukurta Triple DES kriptosistema[PT01]. Ši sistema kaip ir DES yra remiasi simetriniu algoritmu, tačiau naudoja tris atskirus raktus, kurių kiekvieno ilgis – 56 bitai. Tokiu būdu yra padidinamas sistemos saugumas. Nors Triple DES kriptosistema yra vis dar laikoma saugia, kadangi nėra jokios žinomos atakos, kuri galėtų pilnai įveikti šios sistemos šifrus, tačiau dėl atrastos gimimo dienų paradoksu paremtos Sweet32 atakos, kuri efektyviai gali įveikti kai kuriuos šifrus, šis sistema praktikoje nebenaudojama [Uma16].

Blowfish kriptosistema kaip ir Triple DES buvo sukurta siekiant pakeisti DES sistemą. Šią sistemą 1993 metais sukūrė Bruce Schneier. Blowfish yra simetrinio rakto kriptosistema, kuri naudoja kintančio dydžio raktus – nuo 32 iki 448 bitų. Ji išsiskiria savo dideliu šifravimo ir dešifravimo greičiu ir saugumu. Ši kriptosistema yra naudojama įvairioje programinėje įrangoje, nuo elektroninių komercinių platformų, skirtų saugiam apmokėjimui iki slaptažodžių valdymo įrankių, skirtų

slaptažodžių saugumui užtikrinti. Blowfish yra vienas lankstesnių šifravimo metodų [Bra14].

Twofish kriptosistema sukurta prieš tai aprašytos Blowfish sistemos pagrindu. Ši sistema dalyvavo AES (angl. *Advanced Encryption Standard*) konkurse, kurį organizavo JAV nacionalinis standartų ir technologijų institutas (angl. *National Institute of Standards and Technology*) siekiant surasti naują blokinio šifro standartą, galintį pakeisti prieš tai minėtą DES. Twofish buvo kurta pagal konkurso reikalavimus – turėjo 128 bitų simetrinį blokinį šifrą ir jos raktų ilgiai galėjo būti 128, 192 ir 256 bitų. Buvo parodyta, kad jos variantai yra efektyvūs tiek aparatinėje, tiek programinėje įrangoje [Cit12; SKW+98]. Ši kriptosistema konkurso nelaimėjo, tačiau pateko tarp penkių finalistų [Cit12]. Šiuo metu ši kriptosistema yra susieta su tokiomis šifravimo programomis, kaip – PhotoEncrypt, GPG bei populiaria atviro kodo programine įranga TrueCrypt.

AES konkursą 2001 metais laimėjo Rijndael sistema ir tapo pažengusio šifravimo standartu arba pasaulyje žinoma kaip AES sistema. AES algoritmą naudoja JAV vyriausybė ir daugybė kitų organizacijų. AES yra simetrinio rakto algoritmas ir ypač efektyvus naudojant 128 bitų raktus, tačiau gali būti naudojami ir 192 ar 256 bitų raktai siekiant geresnio saugumo [Cit12]. AES sistema plačiai naudojama saugiuose failų perdavimo protokoluose: FTPS, HTTPS, SFTP, AS2, WebDAVS ir OFTP.

RSA kriptosistema yra viena pirmųjų ir iki šiol plačiausiai naudojama viešojo rakto kriptosistema. Jos pavadinimas kilo nuo jos kūrėjų vardų – R. Rivest, A. Shamir ir L. Adleman. RSA gali būti naudojama tiek slaptumo užtikrinimui, tiek skaitmeniniams parašams. Šios sistemos algoritmo saugumas remiasi dviejų sudėtingų matematinių uždavinių sprendimu: didelių skaičių faktorizavimo uždaviniu (žr. 1.1.1 skirsnį) ir RSA uždaviniu, t.y. žinant $a^k \mod n$ rezultatą, k ir n reikia rasti a [Bon+99; KMV+96]. Privatūs raktai viešojo rakto sistemose turi būti didesni (pvz.: 1024 bitai RSA) nei slapti raktai simetrinio rakto sistemose (pvz.: 64 ar 128 bitai), kadangi dažnai efektyviausi išpuoliai prieš simetrinio rakto kriptosistemas remiasi visų galimų variantų perrinkimo ataka, o išpuoliai prieš viešojo rakto sistemas remiasi efektyvesniais, pvz.: faktorizavimo algoritmu.

1.3. Kriptosistemų saugumas

Kriptografiniai algoritmai, priklausomai nuo paties algoritmo, naudojamų parametrų ir raktų dydžių, lemia skirtingą kriptosistemos saugumo lygį. Saugumo lygiai yra naudojami įvertinant kriptosistemos gebėjimą apsaugoti duomenis nuo potencialių sukčių. Literatūroje yra įvardinami penki pagrindiniai kriptosistemų saugumo lygiai [KMV+96; Sta07]:

- Kriptosistema vadinama **besąlygiškai saugia** (angl. *unconditional security*), jei turėdamas beribius skaičiavimo išteklius kriptoanalitikas negalėtų dešifruoti pranešimo šifro be rakto. Šio lygio kriptosistemų perimtas siunčiamas šifras nesuteikia jokios informacijos priešininkui ir todėl tokios sistemos dar vadinamos puikaus slaptumo (angl. *perfect secrecy*). Viešojo rakto kriptosistemos negali būti besąlygiškai saugiomis, kadangi, atsižvelgiant į gautą šifrą, galima atkurti pradinį tekstą, šifruojant visus įmanomus tekstus, kol bus gautas toks pats šifras.

- Kriptosistema vadinama **saugia sudėtingumo teorijos požiūriu** (angl. *complexity-theoretic*

security), jei jos šifro negali įveikti asmuo, kurio skaičiavimo ištekliai leidžia jam taikyti tik polinominio laiko algoritmus. t. y. kai naudojamas laikas ir atmintis polinomiškai priklauso nuo įvedamų saugumo parametrų.

- Kriptosistemos **saugumas yra įrodomas** (angl. *provable security*), jeigu ją įveikti yra taip pat sudėtinga, kaip ir išspręsti sudėtingą ir gerai žinomą matematinį (dažniausiai skaičių teorijos) uždavinį, pavyzdžiui, didelių skaičių faktorizavimo, diskretaus logaritmo ar kitais.

- Kriptosistema vadinama **skaičiavimų požiūriu saugia** (angl. *computational security*), jeigu pasitelkus geriausias žinomas atakas priešininkui neužtektų išteklių jai įveikti. Dažniausiai, tokių sistemų algoritmai siejami su sudėtingais uždaviniais, tačiau skirtingai nuo sistemų, kurių saugumas yra įrodomas, nėra lygiavertiškumo įrodymų. Tokios sistemos dar vadinamos praktiškai saugiomis.

- Galiausiai, kriptosistema vadinama **euristiškai saugia** (angl. *ad hoc*), jeigu jos saugumą patvirtina tam tikri, dažnai euristiniai, argumentai. Tokios sistemos dažnai yra atsparios įprastoms žinomoms atakoms ir laikomos praktiškai saugiomis, tačiau išlieka grėsmė nenumatytoms atakoms, todėl jų saugumas laikomas silpnu.

Kriptosistemos saugumo lygio nustatymas yra sudėtinga užduotis, kadangi matematika pagrįsti vertinimo modeliai dažnai neaprėpia visų praktinių aspektų, bei kai kurios teorinės prielaidos neatitinka realybės, pavyzdžiui: nenuspėjamo atsitiktinių skaičių generatoriaus egzistavimas kompiuteriuose negalimas, kadangi tai yra deterministinis įrenginys. Dėl šių priežasčių dažnai kriptosistemos pagrindinis vertinimo kriterijus yra operacijų skaičius, reikalingas ją įveikti, taikant šiuo metu geriausius ir žinomiausius kriptoanalizės metodus [AS11; KMV+96]. Laikui bėgant, kompiuterių našumas didėja ir atrandami greitesni algoritmai kriptosistemoms įveikti, dėl to kriptosistemų saugumas prastėja ir yra reikalingos pakartotinės saugumo analizės bei įvertinimai.

1.4. Kriptosistemų atakų tipai

Kriptosistemos (kriptoanalizės) atakos – tai metodai, kuriais šifruota informacija yra paverčiama pradine, nešifruota, nenaudojant kriptosistemos slaptojo rakto ar kitų jos slaptųjų parametrų. Atakos tikslas gali būti ne tik dešifruoti duomenis, bet ir nustatyti slaptuosius raktus, parametrus. Kriptoanalitikas gali naudotis tik šifru ir žiniomis apie kriptosistemą: šifravimo ir dešifravimo operacijų struktūra, istoriniais šifrais ir juos atitinkančiais pradiniais duomenimis [Sta07]. Dažnai atakų metu kriptoanalitikas bando nustatyti ir atakuoti kriptosistemų algoritmų silpnąsias vietas [Wil06]. Toliau apibendrinami penki atakų tipai, kurie yra surūšiuoti nuo silpniausio iki stipriausio pagal daromas prielaidas ir jiems keliamus reikalavimus. Papildomai, visiems tipams yra daroma prielaida, kad atakuojantysis žino šifravimui ir dešifravimui naudojamą algoritmą ir pagrindinius kriptosistemos veikimo principus [Sta07].

- **Pavienių šifrų ataka** (angl. *ciphertext-only attack*). Šį atakos tipą naudojantis kriptoanalitikas žino šifruotą tekstą, kurį nori dešifruoti, tačiau neturi jo atitinkančio pradinio teksto.

Ataka yra laikoma sėkminga, jeigu jos metu pavyksta dešifruoti bent mažą dalį pradinio teksto. Šio tipo atakų nesugeba atlaikyti tik silpni algoritmai [Sta07; Sti05; Wil06].

- **Teksto-šifro porų ataka** (angl. *known-plaintext attack*). Atakuojantysis turi pradinio ir šifruoto tekstų poras ir pagal jas bando rasti slaptąjį raktą arba dešifruoti kitą, tuo pačiu raktu šifruotą, informaciją. Dažniausiai tokie raktai randami perimant komunikacijos metu siunčiamą šifrą ir įrašant tikrąjį pokalbį [Sta07; Sti05; Wil06].

- **Pasirinktų teksto-šifro porų ataka** (angl. *chosen-plaintext attack*). Kriptoanalitikas gali vieną kartą nurodyti savo pradinių tekstų rinkinį kriptosistemai ir gauti atitinkamus šifrus. Šios atakos tikslas – surinkti daugiau informacijos, kuri padėtų sumažinti kriptosistemos saugumą [Sta07; Sti05; Wil06].

- **Adaptyvi pasirinktų teksto-šifro porų** ataka (angl. *adaptive chosen-plaintext attack*). Kriptoanalitikas gali pakartotinai teikti pradinius tekstus kriptosistemai ir gauti atitinkamus šifrus. Kiekvieną kartą atlikęs ataką kriptoanalitikas gali analizuoti gautus šifrus, palyginti juos su prieš tai gautais ir taip mėginti surinkti kuo daugiau informacijos apie kriptosistemos silpnąsias vietas [Sta07].

- **Pasirinktų šifrų ataka** (angl. *chosen-ciphertext attack*). Atakuojantysis gali pasirinkti bet kurį šifruotą tekstą ir gauti pradinį tekstą, gautą dešifravimo metu [Sta07; Sti05; Wil06].

Svarbu paminėti, kad simetrinio ir viešojo rakto kriptoanalizės metodų principas šiek tiek skiriasi. Simetrinio rakto sistemų kriptoanalizė remiasi prielaida, kad pradinio teksto struktūra ar tam tikras šablonas gali būti aptinkamas šifruotame tekste. Tuo tarpu viešojo rakto sistemų kriptoanalizė remiasi prielaida, kad matematinės raktų poros savybės gali padėti iš vieno žinomo rakto nustatyti nežinomą.

2. Klaidas taisantys kodai

Klaidų nustatymo ir jų taisymo metodai pradėti naudoti atsiradus pirmosioms telekomunikacijoms. Žinant, kad siunčiant informaciją per triukšmingą kanalą gali atsirasti klaidų, prie jos buvo pridedama papildoma informacija ir klaidas taisančių kodų algoritmų pagalba buvo užtikrinamas duomenų teisingumas [Hud13].

Klaidų radimo ir taisymo kodais paremtos schemos yra skirstomos į sistemines arba nesistemines. Naudojant sisteminę schemą siųstuvas siunčia pradinius duomenis kartu su tam tikru skaičiumi patikros bitų, kurie deterministinio algoritmo pagalba yra išvedami iš pradinių duomenų. Jei reikalingas tik perduodamų duomenų klaidų aptikimas, imtuvas gali paprasčiausiai taikyti tą patį algoritmą gautiems duomenų bitams ir palyginti jo išvestį su gautais patikros bitais. Jei vertės nesutampa – gauti klaidingi duomenys. Naudojant nesisteminę schemą pradiniai duomenys yra paverčiami koduotu pranešimu, kuris yra bent pradinių duomenų dydžio, dažniausiai mažesnio negu sugeneruoti sisteminės schemos, tačiau pranešimui reikalingas papildomas dekodavimas [For70; Gio12]. Geras klaidas taisantis kodas turėtų sugebėti ištaisyti kuo daugiau klaidų, neužkertant kelio greitam koduotos informacijos perdavimui bei nepakenkiant užkodavimo ir dekodavimo etapų efektyvumui [Gio12].

Tiesiniai kodai yra klaidas taisančių kodų poaibis su papildomai apibrėžta struktūra ir operacijomis, kurių bet kokia žodžių kombinacija yra žodis iš abėcėlės (žr. 1.1.4 skirsnį). Jos naudoja generuojančią matricą G, kad paverstų siunčiamos žinutės vektoriaus msg turinį į tiesinio kodo žodį m, ir kontrolinę kodo matricą H, kad patikrintų gauto žodžio teisingumą. Čia tiesinio kodo kontrolinę matricą H vadinsime jo dualaus kodo generuojančią matricą, t.y. kurią tenkina lygybė $GH^\top = 0$. Jeigu $s = m \cdot H = 0$, tai gautas žodis m neturi jokių klaidų, kitu atveju s, dar vadinamas sindromu, gali būti naudojamas nustatyti klaidos vietą ir klaidos reikšmes [Hud13; Sta07].

2.1. Tiesinių kodų kriptosistemos

Tiesiniais kodais paremtos viešojo rakto kriptosistemos šifravimo etapo metu prie koduojamos informacijos prideda sugeneruotų atsitiktinių klaidų, kitaip tariant informacija yra užkoduojama naudojant tam tikrą klaidų šabloną. Dešifravimo metu klaidos yra pašalinamos ir dekoduojant atkuriamas pirminis tekstas [Gio12]. Klaidas taisančiais kodais paremtos kriptografijos sistemos remiasi faktu, kad bendrojo tiesinio kodo sindromo dekodavimas yra žinomas kaip NP-sunki problema, o kai kuriems konkretiems tiesiniams kodams dekoduoti yra žinomi efektyvūs algoritmai.

McEliece ir Niederreiter kriptosistemos yra pirmosios ir geriausiai žinomos tiesiniais kodais paremtos viešojo rakto kriptosistemos. Pirminis McEliece kriptosistemos autoriaus variantas buvo aprašytas kartu su neredukuojamais dvejetainiais Goppa kodais ir saugumo parametrais $n = 1024$, $k = 524$ ir $t = 50$ iki šiol dar laikomas neįveikiamu [BC07; BCG+09] su vienintele išimtimi – Bernstein, Lange ir Peters ataka [BLP08], kurios metu sugebėjo įveikti šios sistemos sukurta šifrą per 1400 dienų naudojantis keturių branduolių kompiuteriu bei išlygiagretintu ir optimizuotu Sterno algoritmu. Tiek Niederreiter, tiek McEliece gali naudoti ir kitus tiesinius kodus, kurie galėtų sistemai suteikti kitokių privalumų, pavyzdžiui, trupesnius raktus, deja, daugelis pasiūlymų,

naudojant kitokius kodus, sumažindavo sistemos saugumą [EOS07; Min07; OS09]. Pagrindinės atakos yra vykdomos būtent prieš šių sistemų su Goppa kodais variantus, kadangi jos net iki šiol yra laikomos atspariomis kriptoanalizės metodams [Bal14; KMV+96]. Tarpusavyje McEliece ir Niederreiter kriptosistemos yra laikomos lygiavertės saugumo atžvilgiu [LDW94].

2.2. McEliece kriptosistemos veikimas

R. McEliece 1978 metais pasiūlė naują viešojo rakto kriptosistemą, kuri remiasi tiesinių kodų teorija. Ši sistema dabar žinoma kaip McEliece kriptosistema (sutrump. MECS). Šios sistemos principas – pirmiausia parinkti tam tikrą kodą, kurio efektyvus dekodavimo algoritmas yra žinomas, ir tada šį kodą užmaskuoti kaip įprastą tiesinį kodą. McEliece yra pirmoji viešojo rakto sistema, kuri šifravimo procese naudoja atsitiktinumus [KMV+96; Mce78].

Toliau aprašomi klasikinės McEliece kriptosistemos pagrindiniai etapai: viešojo ir privataus rakto kūrimas, pradinių duomenų šifravimas ir dešifravimas [Gio12; Hud13; Mce78].

2.2.1. Raktų kūrimas

Pradiniai duomenys: McEliece kriptosistemos saugumo n, k, t parametrai, kur n yra žinutės šifro ilgis, k – žinutės ilgis ir t klaidų skaičius, kurias galima ištaisyti naudojant efektyvų Patterson algoritmą.

Rezultatas: sugeneruoti McEliece $R_{vie.}$ ir $R_{priv.}$ sistemos raktai.

Algoritmas:

1. Atsitiktiniu būdu parenkamas neredukuojamas dvejetainį $[n, k]$-tiesinį Goppa kodą C generuojantis polinomas, kuriam taikant Patterson algoritmą galima ištaisyti t klaidų [LLC14].
2. Polinomo koeficientai yra išrašomi į kontrolinę matricą H ir ji transformuojama į $k \times n$ generuojančią matricą G.
3. Parenkama atsitiktinė neišsigimusi $k \times k$ matrica S.
4. Parenkama atsitiktinė $n \times n$ perstatų matrica P, kuri gaunama iš vienetinės sukeičiant stulpelius vietomis.
5. Apskaičiuojama viešojo rakto generuojanti matrica $G' = SGP$.
6. Gaunami viešasis $R_{vie.} = (G',t)$ ir privatusis $R_{priv.} = (G, S, P)$ raktai

2.2.2. Šifravimas

Pradiniai duomenys: norima išsiųsti žinutė msg ir jos gavėjo viešasis raktas $R_{vie.} = (G',t)$.

Rezultatas: šifras c, kurio ilgis n.

Algoritmas:

1. Žinutė msg yra užkoduojama į dvejetainę k ilgio vektorių m.

2. Sugeneruojamas atsitiktinis n-bitų klaidos vektorius e, turintis tiksliai t vienetų, t. y. kurio ilgis yra n ir Hamingo svoris $wt(e) = t$

3. Suskaičiuojamas žinutės msg šifras: $c = mG' + e$.

2.2.3. Dešifravimas

Pradiniai duomenys: gautas žinutės šifras c, kurio ilgis n, ir jo gavėjo privatusis raktas $R_{priv.} = (G,S,P)$.

Rezultatas: dešifruota žinutė msg.

Algoritmas:

1. Apskaičiuojamas $c' = cP^{-1}$, kur c' – nėra žodis iš Goppa kodo, kadangi gauto vektoriaus struktūra yra paslėpta S matricos pagalba, tačiau galime pastebėti, kad pridėtų klaidų skaičius šiame vektoriuje nepasikeitė $wt(eP^{-1}) = t$, čia P^{-1} matrica tik sukeičia gauto vektoriaus pozicijas vietomis, ir galime ištaisyti pridėtas klaidas.

2. Naudojamas Patterson dekodavimo $D_{Goppa}(c')$ algoritmas, kurio pagalba randamas klaidų vektorius e ir apskaičiuojamas $m' = c' - e$.

3. Apskaičiuojamas Goppa kodo generuojančios matricos G žodis $m = m'S^{-1}$.

4. Transformuojamas gauto žodžio vektorius m iš dvejetainės išraiškos į žinutę msg.

Goppa kodo dekodavimo algoritmo taikymas yra daugiausiai laiko užtrunkantis etapas dešifravime ir dėl to dešifravimas yra žymiai lėtesnis nei šifravimas.

2.3. Niederreiter kriptosistemos veikimas

Praėjus aštuoniems metams po McEliece sistemos publikavimo, Niederreiter aprašė kitokią McEliece kriptosistemos variaciją, kurioje vietoje generuojančios matricos G pasiūlė šifravimui naudoti kontrolinę tiesinio kodo matricą H. Taip pat savo publikacijoje aprašė būdą kaip susisteminti šią kontrolinę matricą, tokiu būdu sumažinant jos dydį. Sistemine kontroline matrica H vadinsime matricą, kurią Gauso metodu galima pertvarkyti į $H = \left[-P^\top | I_{n-k}\right]$. Čia galima pastebėti, kad tokią matricą galima išsaugoti kaip $-P^\top$ ir tokiu būdu sutaupyti vietos, o prireikus pilnos kontrolinės matricos, pridėti vienetinę matricą iš dešinės. Šis būdas gali būti pritaikomas ir kai kuriems McEliece variantams [Nie86; WSN17]. Niederreiter aprašyta sistema užkoduoja visą pranešimą klaidos vektoriuje. Tokiu būdu išvengiama pradinio teksto bitų, kurių nepaveikė klaidos vektoriaus pridėjimas (kaip McEliece kriptosistemoje), nutekėjimo. Daugiau apie šią McEliece kriptosistemos saugumo spragą 4.3 skyriuje. Vietoje kodo žodžio Niederreiter naudoja sindromą kaip šifrą, tokiu būdu perkeliamas tam tikras dešifravimo krūvis į šifravimą, kuris vis tiek yra greitesnis nei McEliece.

Skirtingai nei McEliece, Niederreiter savo publikacijoje neapsistoja ties specifine tiesinių kodų klase, tačiau iškelia jiems du reikalavimus: kodas turi turėti didelį klaidų taisymo pajėgumą ir kodo

dekodavimo etapas turi būti atliktas per priimtiną laiką. Taip pat pasiūlė naudoti šiuos reikalavimus atitinkančius RS arba Goppa kodus. Deja, bet buvo įrodyta, kad su RS kodais kriptosistema yra nesaugi [SS92], todėl binariniai Goppa kodai vis dar laikomi saugiausiu pasirinkimu Niederreiter kriptosistemai.

Toliau bus aprašyti Niederreiter kriptosistemos su Goppa kodais pagrindiniai etapai: viešojo ir privataus rakto kūrimas, šifravimas ir dešifravimas [Gio12; Hud13; Nie86].

2.3.1. Raktų kūrimas

Pradiniai duomenys: Niederreiter kriptosistemos saugumo n, k, t parametrai, kur n yra žinutės šifro ilgis, k – žinutės ilgis ir t klaidų skaičius, kurias galima ištaisyti naudojant efektyvų Patterson algoritmą.

Rezultatas: sugeneruoti Niederreiter $R_{vie.}$ ir $R_{priv.}$ sistemos raktai.

Algoritmas:

1. Atsitiktiniu būdu parenkamas dvejetainis $[n, k]$-tiesinį Goppa kodą C generuojantis polinomas, galintis ištaisyti t klaidų. Šis kodas turi efektyvų dekodavimo algoritmą.
2. Polinomo koeficientai yra išrašomi į $(n - k) \times n$ kontrolinę matricą H.
3. Parenkama atsitiktinė neišsigimusi dvejetainė $(n - k) \times (n - k)$ matrica S.
4. Parenkama atsitiktinė $n \times n$ perstatų matrica P, kuri gaunama iš vienetinės sukeičiant stulpelius vietomis.
5. Apskaičiuojama viešojo rakto $(n - k) \times n$ matrica $H' = SHP$.
6. Gaunami viešasis $R_{vie.} = (H',t)$ ir privatusis $R_{priv.} = (H, S, P)$ raktai

2.3.2. Šifravimas

Pradiniai duomenys: norima išsiųsti žinutė msg ir jos gavėjo viešasis raktas $R_{vie.} = (H',t)$.

Rezultatas: šifras c, kurio ilgis $n - k$.

Algoritmas:

1. Žinutė msg yra užkoduojama į dvejetainę n ilgio vektorių m, kurio Hamingo svoris t.
2. Suskaičiuojamas žinutės šifras: $c = H'm^\top$.

2.3.3. Dešifravimas

Pradiniai duomenys: gautas žinutės šifras c, kurio ilgis $n - k$ ir jos gavėjo privatusis raktas $R_{priv.} = (H,S,P)$.

Rezultatas: dešifruota žinutė msg.

Algoritmas:

1. Apskaičiuojamas $c' = S^{-1}c = HPm^\top$.

2. Naudojant sindromo dekodavimo algoritmą $D_Goppa(c')$ iš c' gaunamas $m' = Pm^\top$.

3. Apskaičiuojamas $m^\top = P^{-1}m'$, kurio ilgis n ir svoris t.

4. Dekoduojamas iš dvejetainės išraiškos $m^\top$ į žinutę msg.

3. McEliece kriptosistemos sandara ir saugumas

Klasikinė McEliece kriptosistema susideda iš kelių sudedamųjų dalių: Goppa kodų, S ir P matricų bei klaidos vektoriaus e. McEliece kriptosistemos saugumas remiasi šiomis prielaidomis [LS01]:

- Kriptosistemos sugeneruojama viešojo rakto $R_{vie.}$ generuojanti matrica G' yra pakankamai didelė ir šifro užšifruoto šia matrica dekodavimas yra neefektyvus, naudojant bendrą tiesinio kodo dekodavimo algoritmą.
- Žinant tik viešąjį raktą $R_{vie.}$ sunku sukurti greitą (polinominio laiko) dešifravimui skirtą algoritmą.

Visos šios sistemos dalys jau buvo minėtos aptariant bendrą sistemos veikimą, o šiame skyriuje kiekviena iš šių dalių bus aptarta detaliau.

3.1. Goppa kodai

1970 metais V. D. Goppa publikavo straipsnį, kuriame pranešė apie naują tiesinių klaidas taisančių kodų klasę [Ber73; Gop70]. Dėl greitų dekodavimo algoritmų šios klasės kodai labai išpopuliarėjo ir buvo pavadinti kūrėjo garbei – Goppa kodais [Gab95].

Goppa kodas $\Gamma(L, g(z))$ yra apibrėžiamas pora – Goppa polinomu $g(z)$ ir seka L [Ber73; Joc02]. Goppa kodo polinomą $g(z)$ laikysime polinomą esantį virš išplėstinio kūno $GF(q^m)$. Šį polinomą galime užrašyti:

$$g(z) = g_0 + g_1 z + \cdots + g_t z^t = \sum_{i=0}^{t} g_i z^i, \tag{2}$$

kur kiekvienas $g_i \in GF(q^m)$, q yra pirminis skaičius ir $m \in \mathbb{N}$.

Seka L laikysime baigtinį poaibį iš $GF(q^m)$, kurio elementai nėra Goppa polinomo $g(z)$ šaknys ir užrašysime:

$$L = \{\alpha_1,\ldots,\alpha_n\} \subseteq GF(q^m), \tag{3}$$

kur visi $\alpha_i \in L$ nariai turi tenkinti sąlygą $g(\alpha_i) \neq 0$.

Apibrėžkime vektoriaus $c = (c_1, \ldots, c_n)$ su elementais virš kūno $GF(q^m)$, funkciją:

$$R_c(z) = \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i}, \tag{4}$$

kurioje $\dfrac{1}{z-\alpha_i}$ yra unikalus polinomas pagal $(z - \alpha_i) \cdot \dfrac{1}{z-\alpha_i} \equiv 1 \pmod{g(z)}$. Tada, Goppa kodas $\Gamma(L,g(z))$ susideda iš visų vektorių c, kuriems teisinga lygybė:

$$R_c(z) \equiv 0 \pmod{g(z)}. \tag{5}$$

Klasikinė McEliece sistema remiasi neredukuojamais dvejetainiais Goppa kodais, kurie yra smulkesnė pagrindinių Goppa kodų klasė. Goppa kodo polinomą $g(z)$ virš $GF(2^m)$ vadinsime neredukuojamu, jeigu jo elementai neturi bendro daliklio. Dvejetainė struktūra lyginant su ne dvejetainiais variantais šiai kodų klasei suteikia matematinių pranašumų bei yra geriau pritaikyta bendram naudojimui kompiuteriuose ir telekomunikacijoje.

Pagrindiniai McEliece kriptosistemos parametrai yra (n, k, t), tačiau juos taip pat galime apibrėžti per Goppa kodo parametrus m ir t, kadangi kiekvienam neredukuojamam laipsnio t polinomui $g(z)$ virš $GF(2^m)$, egzistuoja neredukuojamas dvejetainis Goppa kodas, kurio ilgis $n = 2^m$, dimensija $k \geq n - mt$, minimalus atstumas $d = 2t+1$, galintis ištaisyti t klaidų naudojant efektyvų dekodavimo algoritmą. Čia m vadinsime neredukuojamo dvejetainio Goppa kodo baigtinio kūno laipsniu [Ber73]. McEliece šiai kriptosistemai pasiūlė naudoti $(n = 2^m, k = n - mt, d = 2t+1) = (1024, 524, 101)$ Goppa kodus virš $GF(2^m)$, kur $m = 10$ ir $t = 50$ [Mce78].

Kriptografijoje Goppa kodai buvo prisitaikyti dėl kelių priežasčių: lengva įvertinti minimalų atstumą iš apačios, esant žinioms apie generuojantį polinomą galima efektyviai taisyti klaidas, o nesant joms – klaidų taisymui nėra žinomų efektyvių algoritmų [EOS07]. Goppa kodai yra viena iš nedaugelio kodų klasių, kurios iki šiol atsilaikė prieš visas kriptoanalizės atakas [Hud13].

3.2. S ir P matricos

McEliece kriptosistemoje naudojamos dvi pagalbinės matricos S ir P. Matricos S paskirtis yra paslėpti Goppa kodo generuojančios matricos G struktūrą, dėl to ji dar vadinama maskavimo matrica. Perstatų matricos P tikslas sukeisti matricos G stulpelius ir tokiu būdu dar labiau paslėpti šios struktūrą. S ir P matricos yra būtinos McEliece kriptosistemos saugumui, kadangi jei kriptoanalitikas galėtų nustatyti S ir P matricas, jis nesunkiai apskaičiuotų generuojančią G matricą, o vėliau ir Goppa polinomą $g(z)$. Turint šiuos sistemos komponentus, kriptoanalitikas nesunkiai įveiktų kodą ir atkurtų užšifruotą informaciją [Joc02]. 1978 m paskelbtoje McEliece kriptosistemos schemoje buvo pasiūlyti naudoti $n = 1024$, $k = 524$, $t = 50$ saugomo parametrai [Mce78]. Kokia tikimybė nustatyti S matricą? S matricos dydis – $k \times k$ elementų. Pirmąjai eilutei sudaryti yra $2^k - 1$ galimybių, antrajai – $2^k - 2$, trečiajai – $2^k - 4$ ir t.t. Taigi galimybių S matricai sudaryti yra:

$$\prod_{i=0}^{k-1}(2^k - 2^i) = (2^{524} - 2^0) \cdot (2^{524} - 2^1) \cdot \cdots \cdot (2^{524} - 2^{523}), \tag{6}$$

o tikimybė, kad atsitiktinis atakuotojo bandymas suras teisingą S matricą yra:

$$\frac{1}{\prod_{i=0}^{523}(2^{524} - 2^i)} = 0{,}8459238718 \cdot 10^{-82655}. \tag{7}$$

Tuo tarpu P matricai, kurios dydis yra $n \times n$, sudaryti reiktų $n!$ galimybių. Taigi tikimybė rasti teisingą perstatų matricą P yra:

$$\frac{1}{1024!} = 0{,}1845519398 \cdot 10^{-2639} \tag{8}$$

Taigi, S ir P matricų radimas priklauso nuo saugumo parametrų ir jeigu šie yra tokie, kokius

pasiūlė McEliece kriptoanalitikui atsitiktinai atspėti šias matricas yra beveik neįmanoma užduotis [Joc02].

3.3. Klaidos vektorius

Klaidų vektorius e yra esminis sistemos komponentas, kuriuo remiasi visas kriptosistemos saugumas. Jeigu šifruojant pranešimą m klaidos vektorius e nebūtų pridedamas arba dešifravimo metu jis būtų žinomas, tada kriptoanalitikas iš perimto šifro c ir viešojo rakto generuojančios matricos G' galėtų lengvai rasti m. Jam reikėtų tik išspręsti tiesinę lygčių sistemą $c = mG'$, jei klaidos vektorius nėra pridedamas, arba $c = mG' + e$, jei klaidos vektorius yra žinomas. Klaidos vektorius yra parenkamas pagal kriptosistemos t parametrą taip, kad tiesinio kodo dekodavimo algoritmas galėtų jį dekoduoti, t. y. ištaisyti jo klaidas.

Literatūroje klaidos vektorius dažnai minimas kaip pirminės McEliece kriptosistemos silpnoji vieta, kadangi kriptoanalitikui naudojant adaptyvią pasirinktų teksto-šifro porų ataką ir siunčiant tokį patį pradinį tekstą galima sumažinti sistemos saugumą. Tam, kad informacija nenutekėtų yra sukurtos McEliece modifikacijos, kurių pagalba sugeneruojami šiai atakai atsparūs klaidų vektoriai [FO99; KI01; Nie86].

3.4. McEliece kriptosistemos saugumas: silpnosios vietos ir kaip jų išvengti

S. Au ir kiti savo publikacijoje [AEE03] nurodo pranešimo persiuntimo ir susijusių pranešimų atakas, kaip vienas iš pagrindinių klasikinės McEliece sistemos saugumo trūkumų. Straipsnyje yra parodoma, kad jeigu kanalu, kurio klausosi kriptoanalitikas, yra siunčiamas toks pats tekstas kelis kartus arba keli tiesišką priklausomybę turintys tekstai, tai atakuotojas gali ne tik juos identifikuoti, bet ir atkurti pradinį tekstą. Atakuotojui toks dešifravimas užtruktų αk^3 laiko, kur α yra maža konstanta, o k yra siunčiamo teksto ilgis [Ber97]. Yra žinomos paprastos McEliece kriptosistemos modifikacijos, kurios be didelių trūkumų užkerta kelią tokioms atakoms [BBC13; EOS07; Sun98].

4. Atakos prieš McEliece kriptosistemą

Šiame skyriuje sutelkiamas dėmesys į labiausiai žinomas atakas prieš klasikinę McEliece kriptosistemą su neredukuojamais dvejetainiais Goppa kodais. Kadangi McEliece ir Niederreiter sistemų saugumas yra lygiavertis [LDW94], visos aptariamos atakos yra netiesiogiai susijusios ir su Niederreiter sistemos saugumu. Pagal McEliece, kriptoanalitikas gali atakuoti McEliece kriptosistemą dviejų tipų atakomis [Mce78]:

- Struktūrine ataka, kurios tikslas atkurti privatųjį raktą $R_{priv.}$ iš viešojo rakto $R_{vie.}$ ir tokiu būdu dešifruoti informaciją.
- Dekodavimo ataka, kurios tikslas dekoduoti informaciją tiesiogiai iš perimto šifruoto teksto, nesigilinant į Goppa kodo struktūrą.

Dažniausiai struktūrinės atakos yra žymiai sunkiau įgyvendinamos, kadangi saugumo parametrai būna per dideli, o dekodavimo atakos yra lankstesnės ir daugiau žadančios kriptoanalitikui, nes jis gali naudoti ne vieną dekodavimo metodą. Dekodavimo atakos toliau yra skirstomos į kritines ir nekritines atakas, priklausomai nuo to ar padidinant parametrų dydį atakos gali būti išvengiamos, ar ne. Jei atakos gali būti išvengiamos – jos yra nekritinės, ir atvirkščiai. Visos kritinės atakos reikalauja papildomos informacijos, tokios kaip, pradinio teksto dalies, ar numanomo rakto, kuris galėtų dešifruoti kitus nei pradinius šifruotus tekstus. Tai yra pagrindinės McEliece kriptosistemos atakos, daugiau nėra žinoma efektyvių algoritmų, galinčių dešifruoti šifruotus tekstus [KI01]. Toliau bus pateikiami žinomų struktūrinių, kritinių ir nekritinių dekodavimo atakų algoritmai.

4.1. Struktūrinės atakos

Loidreau ir Sendier 2001 metais pasiūlė šiuo metu geriausiai žinomą struktūrinę ataką prieš McEliece kriptosistemą [LS01]. Pateiktoje publikacijoje atakos metu yra naudojamas Sendier atramos skaidymo (angl. *support-splitting*) algoritmas, kuris leidžia apskaičiuoti perstatas tarp dviejų ekvivalenčių tiesinių ar netiesinių kodų [Sen00] ir tikrinama, ar generavimo matrica G' priklauso silpnųjų raktų klasei, patikrinant ar Goppa kodas buvo sugeneruotas polinomo, turinčio dvejetainius koeficientus – $\mathbb{F}_2$ vietoje $\mathbb{F}_{2^m}$. Taigi, šios atakos vykdymui yra paimama iš McEliece kriptosistemos gauta viešojo rakto matrica G' ir ji yra lyginama su visomis (m, t) saugumo parametrus tenkinančiomis Goppa kodus generuojančiomis matricomis, kol randama perstatų atžvilgiu ekvivalenti matrica ir randamas privatus raktas. Konkrečiu atveju, kai Goppa polinomo koeficientai yra iš F_2, ataka gali netrukti ilgai, tačiau vis tiek yra atliekama daug skaičiavimų siekiant atkurti vieną raktą ir atakos galima išvengti paprasčiausiai nenaudojant tokių silpnų viešųjų raktų [Sen00; Sen99] Bendru atveju, ši ataka yra laikoma neefektyvia, kadangi taikant bet kokius Goppa kodo parametrus turi būti patikrinta maždaug $2^{m(t-3)}/mt$ Goppa kodų ir kiekvienam jų turi būti taikomas paramos skaidymo algoritmas, čia m ir t yra neredukuojamo dvejetainio Goppa kodo parametrai. Naudojant pirminius McEliece parametrus, kai $m = 10$ ir $t = 50$, šis skaičius padidėja maždaug iki 2^{461} ir ataka laikoma neįvykdoma [Sen00].

4.2. Nekritinės dekodavimo atakos

Literatūroje dažniausiai minimos nekritinės dekodavimo atakos [KI01]:

- Apibendrinta informacijos rinkinio dekodavimo ataka (angl. *generalized information-set decoding attack*).
- Mažo svorio kodo žodžių radimo ataka (angl. *finding low-weight-codeword attack*).

Abiejų atakų gali būti išvengta padidinant parametrų dydį arba nekeičiant parametrų dydžio ir pritaikant Loidreau modifikaciją [Loi00], kurios pagalba pasirenkami specialūs Goppa kodų polinomai, šifravimo metu prie žinutės pridedama daugiau klaidų negu Goppa kodas gali ištaisyti ir viešasis raktas papildomas dekodavimui skirtu rinkiniu, vadinamu „t-tower“. Taip pat, abi atakos remiasi informacijos rinkinio (angl. *information-set*) ir fiksuoto atstumo dekodavimo (angl. *fixed-distance-decoding*) metodais.

Informacijos rinkiniu vadinsime tiesinio $[n, k]$-kodo $C \subseteq F_q^n$ k koordinačių pozicijų rinkinį $i_1, \ldots, i_k$, kur k stulpeliai su indeksais $i_1, \ldots, i_k$ iš generuojančios matricos G' yra tiesiškai nepriklausomi virš F_q. Kitaip tariant, bet koks kodo žodis yra gaunamas perimant k informacijos pozicijas iš bet kokių galimų q^k neturinčių klaidų žodžių, kurių ilgis k, virš F_q, o likusios $n-k$ koordinatės yra paskaičiuojamos kaip tiesinės kombinacijos su k informacijos simbolių. Tiksli tiesinė kombinacija yra nustatoma naudojant tiesinio kodo kontrolinę matricą H [Pra62]. Fiksuoto atstumo dekodavimo algoritmai ieško kodo žodžio, kuris yra per fiksuotą atstumą nuo gauto vektoriaus [BLP+09]. Toliau plačiau aptariamos šios dvi nekritinės dekodavimo atakos.

4.2.1. Apibendrinta informacijos rinkinio dekodavimo ataka

1978 metais McEliece savo publikacijoje pasiūlė apibendrintą informacijos rinkinio dekodavimo ataką visiems kodams [Mce78], vėliau ją tobulino Lee ir Brickell [LB88; Pet10]. Šio tipo atakos naudoja bendrą algoritmą skirtą dekoduoti bet kurį klaidas taisantį tiesinį kodą ir tokiu būdu išsprendžia NP-sunkią dekodavimo problemą. Dažnai tokios atakos yra vykdomos eksponentiniame laike, tačiau jų rezultatų analizė gali būti naudinga ir jos pagalba galima aptikti efektyvesnius algoritmus atakai.

Tegul C yra $[n, k, 2t+1]$ tiesinis kodas virš $\mathbb{F}$, m yra žinutė, o G'_I, c_I ir e_I, kur I žymi $i_1, \ldots, i_k$ stulpelius paimtus iš gauto viešojo rakto G', gauto šifro c ir klaidų vektoriaus e atitinkamai. Tada juos tenkina lygybė:

$$c_I = mG'_I + e_I \tag{9}$$

Jeigu $wt(e_I) = 0$, t. y. prie šifro c pasirinktose pozicijose nebuvo pridėta jokių klaidų, ir G'_I neišsigimusi matrica, galime dešifruoti m:

$$m = c_I G_I'^{-1} \tag{10}$$

Jeigu $wt(e_I) \neq 0$, m gali būti dešifruotas spėliojant e vektorių.

Norint dešifruoti gautą šifrą c, kriptoanalitikas vykdo apibendrintą informacijos rinkinio dekodavimo algoritmą. Toliau pateikiame apbindrintą Lee-Brickell atakos algoritmą [LB88; Pet10]:

1. Pasirenkami k atsitiktinių stulpelių indeksų $I = i_1, \ldots, i_k$. Gauname G'_I.
2. Pasirenkamas nedidesnis nei leistinas klaidų skaičius t paieškos parametras p, kur $0 \leq p \leq t$.
3. Pasirinktą I pozicijų rinkinį tenkina lygybė: $c_I = mG'_I + e_I$. Bandome apskaičiuoti $Q = G'^{-1}_I G'$. Nepavykus grįžtame į pirmąjį žingsnį.
4. Pasirenkamas atsitiktinis e_I, kuris tenkina $wt(e_I) \leq p$. Kadangi iš trečio žingsnio matome, kad $c_I Q = mG' + e_I Q$, todėl galime paskaičiuoti $e' = c + c_I Q + e_I Q$.
5. Jeigu $wt(e') = t$, gauname $m = (c_I + e_I)G'^{-1}_I$, kitu atveju pereiname į ketvirtą žingsnį su kitokiu atsitiktiniu vektoriumi e_I, o nesėkmingai išbandžius visus įmanomus atsitiktinius vektorius, grįžtame į pirmą žingsnį ir pasirenkame dar nebandytą I rinkinį.

Detaliau šios atakos algoritmas ir rezultatai yra aptariami 5.2 skyriuje.

4.2.2. Mažo svorio kodo žodžių radimo ataka

Remiantis informacijos rinkinio dekodavimo idėja, daugelis autorių pateikė veiksmingesnius atakų algoritmus, kurie remiasi mažo svorio kodo žodžių radimu. Canteaut ir Chabaud pažymėjo, kad pradinio teksto ieškojimo iš šifruoto $y \in \mathbb{F}_{2^n}$, kuris yra užkoduotas tiesiniu kodu C, uždavinys gali būti supaprastintas iki mažo svorio kodo žodžių ieškojimo uždavinio, šiek tiek didesniame tiesiniame kode. Šis supaprastinimas yra susijęs su NP-sunkios dekodavimo problemos atakavimu, ieškant tam tikro svorio kodo žodžių tiesiniame kode [CC98]. Ši ataka nėra skirta tik Goppa kodams, tačiau gali būti pritaikyta bet kuriam klaidas taisančiam tiesiniam kodui.

Tarkime C yra $[n, k, 2t + 1]$ tiesinis kodas virš $\mathbb{F}_2$. Tada žodis $y \in F_2^n$ yra mažiausiai nutolęs nuo kodo žodžio $c \in C$ per atstumą $d(y, c) = t$, tada galime parodyti, kad $y - c$ yra unikalus svorio t kodo žodis praplėstame kode $C + y$, čia suma reiškia y eilutės pridėjimą prie C kodo generuojančios matricos. Kadangi tiesinio kodo C minimalus atstumas tarp kodo žodžių yra $2t + 1$, tai y negali būti kodo žodžiu iš C dėl to pridedant jį prie C generuojančios matricos sukuria naują tiesinį kodą C', kuriame $y - c$ yra vienintelis t svorio kodo žodis. Jei kriptoanalitikas gali rasti c kodo žodį, jis gali lengvai dešifruoti šifrą ir gauti pradinį tekstą [BLP08].

Visos mažo svorio kodo žodžių radimo tipo atakos pirmiausiai bando rasti mažo svorio kodo žodžius mažesniame tiesiniame kode. Pasirenkamas generuojančios matricos stulpelių pogrupis ir patikrinamas ar rastas kodo žodis turi svorio atitikmenį originaliame kode. Šio tipo atakos tarpusavyje skiriasi tik būdu, kuriuo jos pasirenka smulkesnius tiesinio kodo segmentus ir strategija, kuria randa žemo svorio kodo žodžius segmente [EOS07]. Šiuo metu geriausiai žinomą šio tipo praktiškai įgyvendintą ataką 2008 metais publikavo Bernstein [BLP08]. Publikacijoje rėmėsi išlygiagretinta Stern [Ste88] mažo svorio kodo žodžių radimo ataka ir parodė, kad jai reikėtų $2^{60.55}$ operacijų, kad galėtų įveikti McEliece kriptosistemą su saugumo parametrais $n = 1024$, $k = 524$ ir $t = 50$. Savo straipsnyje jis taip pat pateikė saugumo parametrų rekomendacijas, kaip išvengti šio

mažo svorio kodo radimo atakų [BLP08]. Vis dėl to, Torres ir Sendrier parodė [TS16], kad visi iki šiol žinomi mažo svorio kodo žodžių radimu paremti atakų algoritmai $\mathbb{A}$ turi tokį patį asimptotinį sudėtingumą, t. y.

$$WF_{\mathbb{A}}(n,k,t) = 2^{ct(1+o(1))}, \tag{11}$$

kur c maža konstanta priklausanti nuo tiesinio kodo tankio $R = k/n$ ir klaidų tankio t/n. Esant taisomam klaidų skaičiui $t = o(n)$, visų žinomų algoritmų konstanta yra $c = \log_2 \frac{1}{1-R}$. Taigi, galime pastebėti, kad atakų sudėtingumas yra eksponentiškai priklausomas nuo kodo tankio. Didinant saugumo parametrus tokios atakos tampa vis mažiau efektyvios prieš McEliece kriptosistemą. Taip pat, manoma, kad ateityje šio tipo atakų algoritmai tik nedaug pagerins konstantą c [TS16].

4.3. Kritinės dekodavimo atakos

Kritinės atakos, tai tokios atakos, kurios negali būti išvengtos didinant parametrų dydį ar pritaikant Loidreau modifikaciją [Loi00]. Šios atakos tarpusavyje skiriasi pagal tai, kiek kriptoanalitikas turi informacijos, kai atakuoja šifrą. Toliau bus aptartos dažniausiai pasitaikančios kritinės McEliece atakos [CGN+15].

4.3.1. Žinomo dalinio teksto ataka

Žinomo dalinio teksto atakų (angl. *known partial plaintext attack*) grupei priskiriamos tokios atakos, kurias vykdant yra žinoma dalis pradinio teksto ir perimtas šifras. Šios atakos tikslas yra sumažinti McEliece kriptosistemos saugumo parametrų dydžius.

Tegul m_l žymi kairiąją bitų dalį k_l, o m_r atitinkamai likusius k_r bitus iš pradinio teksto m, t. y. $k = k_l + k_r$ ir $m = (m_l||m_r)$. Tarkime, kad kriptoanalitikas žino m_r. Tada McEliece kriptosistemos su saugumo parametrais (n,k) žinutės m dešifravimo uždavinio sudėtingumas yra supaprastinamas iki nežinomos pradinio teksto dalies m_l radimo su (n,k_l) saugumo parametrais, kadangi [KI01]:

$$c = mG' + e \tag{12}$$

$$c = m_l G'_l + m_r G'_r + e \tag{13}$$

$$c - m_r G'_r = m_l G'_l + e \tag{14}$$

$$c' = m_l G'_l + e \tag{15}$$

čia G_l yra matricos G' viršutinės l eilutės, o G_r yra apatinės r eilutės atitinkamai. Jeigu k_l yra labai nedidelis, ataka tampa labai pavojinga, kadangi kriptosistemos saugumo parametrai tampa labai maži. Tokio tipo ataka gali būti vykdoma ir žinant ne iš eilės esančius pradinio teksto bitus. Įvykdžius šią ataką, galima taikyti kitą ataką skirtą pradiniam tekstui atkurti, pavyzdžiui, mažo svorio kodo žodžių radimo. Detaliau šios atakos algoritmas ir rezultatai yra aptariami 5.3 skyriuje.

4.3.2. Pranešimo persiuntimo ataka

1997 metais Berson [Ber97] pristatė pranešimo persiuntimo ataką (angl. *message-resend attack*). Šios atakos tikslas yra dešifruoti pranešimą. Siekiant įvykdyti šią ataką, kriptoanalitikas turi perimti bent du kartus siunčiamo tokio pačio pranešimo m tam pačiam gavėjui šifrus. Taip pat, abu kartus pranešimas turi būti šifruojamas su skirtingais atsitiktiniais klaidos vektoriais. Tarkime vykdome šią ataką su saugumo parametrais $(n, k, t) = (1024, 524, 50)$ ir siunčiame pranešimą m du kartus ir gauname šifrus $c_1 = mG' + e_1$ ir $c_2 = mG' + e_2$, kur $e_1 \neq e_2$. Tada galime paskaičiuoti $c_1 + c_2 = e_1 + e_2$.

Galima pastebėti, kad tikimybė turėti bitą, kuris lygus „1“ toje pačioje pozicijoje e_1 ir e_2, yra labai nedidelė, o tiksliau [Ber97]:

$$P(e_1(l) = e_2(l) = 1) \leq \left(\frac{t}{n}\right)^2 = \left(\frac{50}{1024}\right)^2 \approx 0{,}00238, \tag{16}$$

kur $l \in \{1, \ldots, n\}$.

Apibrėžkime klaidos vektoriaus pozicijų rinkinius L_0 ir L_1:

$$\begin{aligned} L_0 &= \{l \in \{1, \ldots, n\} \colon c_1(l) + c_2(l) = e_1(l) + e_2(l) = 0\} \\ L_0 &= \{l \colon e_1(l) = 0 = e_2(l)\} \cup \{l : e_1(l) = 1 = e_2(l)\} \\ L_0 &\approx \{l \colon e_1(l) = 0 \text{ ir } e_2(l) = 0\} \end{aligned} \tag{17}$$

ir

$$L_1 = \{l \in \{1, \ldots, n\} : c_1(l) + c_2(l) = e_1(l) + e_2(l) = 1\} \tag{18}$$

Rinkinys L_0 dažniausiai susidės tik iš klaidų pozicijų, kuriose nebuvo pridėtas klaidos vektorius. Taip pat galima matyti, kad jeigu pozicija l:

- $l \in L_0$, tada tikėtina, kad $c_1(l)$ ir $c_2(l)$ nebuvo paveikti klaidos vektoriaus.
- $l \in L_1$, tada vienas iš $c_1(l)$ arba $c_2(l)$ buvo pakeistas klaidos vektoriaus.

Kadangi žinome, kad $e_1 + e_2$ Hamingo svoris yra nemažesnis $2t$, tai $|L_0| \approx 1024 - wt(e_1 + e_2) \geq 1024 - 2 \cdot 50 = 924$ ir $|L_1| \leq 2t = 100$. Kriptoanalitikas, turėtų mėginti atspėti 524 neiškraipytus stulpelius iš galimų 924 iš L_0. Savo straipsnyje Berson parodo, kad ši ataka įgyvendinama per βk^3 laiką, kur β yra maža konstanta, o k – žinutės vektoriaus ilgis. Kadangi ši dekodavimo ataka nepriklauso nuo kodo struktūros, Goppa kodo pakeitimas į kitą kodą arba saugumo parametrų padidinimas efektyviai nepagerintų McEliece kriptosistemos saugumo prieš ją.

Detaliau šios atakos algoritmai ir rezultatai yra aptariami 5.4 skyriuje.

4.3.3. Susijusių pranešimų ataka

Susijusių pranešimų ataka (angl. *related-message attack*) – apibendrinta prieš tai aprašyta pranešimo persiuntimo ataka. Šios atakos metu kriptoanalitikas turi kelis šifruotus tekstus, kurie yra susieti su atitinkamais pradiniais tekstais. Šie šifruoti tekstai gali kriptoanalitikui padėti gauti naudingos informacijos apie slaptus pradinius tekstus. Tarkime dvi žinutės m_1 ir m_2, kur $m_1 \neq m_2$,

buvo siųstos tam pačiam gavėjui. Susijusios pranešimų atakos sąlyga – tiesinis ryšys tarp žinučių m_1 ir m_2, pavyzdžiui, jei mes žinome, kad $\sigma m = m_1 + m_2$. Kriptoanalitikas žino $c_1 = m_1 G' + e_1$ ir $c_2 = m_2 G' + e_2$. Tada jis galės suskaičiuoti:

$$c_1 + c_2 = m_1 G' + m_2 G' + e_1 + e_2 = (\sigma m) G' + e_1 + e_2. \tag{19}$$

Gauname $c_1 + c_2 + (\sigma m) G' = e_1 + e_2$ ir pritaikome žinutės persiuntimo atakos algoritmą naudodami $c_1 + c_2 + (m_1 + m_2) G'$ vietoje $c_1 + c_2$ [Ber97].

5. Atakų prieš McEliece kriptosistemą tyrimas

Remiantis 4 skyriaus apžvalgoje atlikta atakų analize yra realizuotos ir ištirtos apibendrintos informacijos dekodavimo, žinomo dalinio teksto, pranešimo persiuntimo bei susijusių pranešimų atakos. Kadangi pranešimo persiuntimo atakos algoritmai beveik nesiskiria nuo susijusių pranešimų atakos algoritmų dėl to jų rezultatai su tokiais pačiais pradiniais duomenimis yra vienodi ir nėra pateikiami.

Darbo metu buvo realizuota pirminė McEliece kriptosistema ir jos pagrindinės operacijos. Kriptosistemai įgyvendinti buvo pasiremta vienos iš populiariausių atvirojo kodo kriptografinių bibliotekų „Bouncy Castle“[1] programiniu kodu. Ši biblioteka yra pasirinkta dėl kodo patikimumo (didžioji programinio kodo dalis yra padengta testais), esamos dokumentacijos, pavyzdžių bei aktyvios šią biblioteką tobulinančios bendruomenės.

Dėl asmeninės patirties ir žinių kriptosistema bei atakos buvo įgyvendintos naudojant Scala programavimo kalbą. Didelėmis vertybėmis laikomi Scala suteikiamos galimybės realizuoti atakų algoritmus objektiškai orientuota paradigma griežtai tipizuotoje aplinkoje (angl. *strongly typed*). Šios kalbos programinis kodas veikia Java virtualioje mašinoje (angl. *JVM*) ir dėl to gali būti vykdomas įvairiose platformose. Lyginant su kai kuriomis kitomis programavimo kalbomis, pavyzdžiui, C, Scala atakų vykdymo trukmės bus nežymiai lėtesnis, tačiau įgyvendintų atakų analizei ir efektyvumui tai didelės įtakos neturės.

Atakoms vykdyti sukurta taikomoji programa, kurios pagalba vartotojas gali sugeneruoti McEliece kriptosistemos raktus, raktų pagal užšifruoti atsitiktines žinutes ir įgyvendinti pasirinktą ataką. Pagal vartotojo pasirinktis fiksuojami darbo su programa faktai, atakų statistiniai duomenys ir rezultatai bei atspausdinami vartotojui. Atakų algoritmams yra sukurti automatiniai testai, kurie užtikrina teisingą jų veikimą. Ši taikomoji programa bei jos programinis kodas yra patalpinti viešai prieinamoje atvirojo kodo projektų talpykloje „GitHub“: `https://github.com/myDisconnect/mceliece-security`. McEliece raktų generavimui, žinučių šifravimui, atakų vykdymui ir statistinių duomenų rinkimui buvo naudojamas nešiojamas kompiuteris, kurio parametrai:

Centrinis procesorius (CPU)	2.2 GHz quad-core Intel Core i7
Operatyvioji adresuojama atmintis (RAM)	16 GB 1600 MHz DDR3
Operacinė sistema (OS)	64-bit macOs High Sierra
Java versija (JDK)	1.8.0_144
Scala versija	2.12.4

Visi atakoms realizuoti algoritmai nėra išlygiagretinti, dėl to kiekvienos atakos metu bus naudojamas tik vienas centrinio procesoriaus branduolys. Taip pat, kadangi operacinei sistemai veikti reikalinga vidinė atmintis, taikomajai programai suteikiama 15 gigabaitų (GB) iš 16 galimų.

Pagrindiniai McEliece kriptosistemos saugumo parametrai yra laikomi (n, k, t), tačiau šiame darbe naudojami neredukuojamų dvejetainių Goppa kodų parametrai: m ir t. Šiuos parametrus galima paversti į standartinius naudojantis $(n, k, t) = (2^m, 2^m - m * t, t)$. Taip pat, kai kurioms

[1] https://www.bouncycastle.org/

atakoms darbe yra naudojami sumažinti kriptosistemos saugumo parametrai ir prognozuojami rezultatai su didesniais, kadangi praktiškai įgyvendinti visas atakas su McEliece siūlomais saugumo parametrais $(n, k, t) = (1024, 524, 50)$ arba $(m, t) = (10, 50)$ yra sudėtinga užduotis skaičiavimo resursų atžvilgiu. Gauti atakų vykdymo rezultatų statistiniai duomenys yra pateikiami lentelių ir grafikų pagalba.

Toliau kiekvienai atakai yra pateikiami optimalūs atakų parametrai ir prognozuojami rezultatai su didesniais parametrais, pavyzdžiui, $(m, t) = (10, 50)$, bei McEliece saugumo parametrai, su kuriais kriptosistema būtų saugi. Kai kuriems atvejams pateikiamos rekomendacijos kaip išvengti atakų.

Eksperimentuose buvo tiriami atakų algoritmai ir bandoma juos optimizuoti siekiant gauti geresnius rezultatus. Taip pat, ieškomi optimalūs atakų parametrai ir saugiausi m bei t parametrai. Tiriami atakų, kurių vidutinė vykdymo trukmė buvo ne mažesnė nei 1 milisekundė ir ne didesnė nei 24 valandos, rezultatai. Taip pat, pateikiami preliminarūs rezultatai su didesniais parametrais arba rekomendacijos kaip šių atakų išvengti.

5.1. Taikomosios programos realizacija

Šiame skyriuje apžvelgiama sukurtos taikomosios programos realizacija, kurios pagrindinė paskirtis yra vykdyti atakas prieš McEliece kriptosistemą. Tam, kad būtų įvykdytos atakos prieš šią kriptosistemą, pirmiausia reikia realizuoti kriptosistemos viešojo ir privataus raktų generavimo, žinučių šifravimo ir dešifravimo operacijas. Šių operacijų pagalba pagal pateiktus vartotojo pasirinktus saugumo parametrus m ir t kuriami viešasis ir privatusis raktai bei šifruojamas atsitiktiniu būdu sukuriamos žinutės. Toliau vartotojui pateikiamas atakos pasirinkimas iš įgyvendintų sąrašo, informacija apie galimus atakų parametrus bei galimybė pasirinkti juos. Galiausiai, įvykdžius atakas yra pateikiama informacija apie statistinius vykdymo duomenis, kurių pagalba galima analizuoti šių atakų efektyvumą.

5.1.1. McEliece kriptosistemos realizacija

Šio darbo eigoje buvo tyrinėjama „Bouncy Castle“ McEliece kriptosistemos realizacija. Tyrimo metu buvo pastebėta, kad „Bouncy Castle“ sistema privataus rakto generavimo metu visada pasirenka tik patį pirmą neredukuojamą Goppa polinomą. Todėl nuspręsta sukurti naują, pataisytą McEliece kriptosistemos klasę „McElieceCryptosystem“, kurios privataus rakto generavimo funkcionalumas atitiktų pirminę McEliece sistemą ir naudotų atsitiktinius neredukuojamus Goppa polinomus.

Sukurtoje McEliece kriptosistemos realizacijoje buvo pasinaudota „Bouncy Castle“ vektoriaus ir matricos klasėmis, kuriose yra įgyvendinti tiesinės algebros daugybos, sudėties operacijos, bei matricos transponuotos ir atvirkštinės skaičiavimai. Abi klasės pateiktus dvejetainius matricos ir vektoriaus duomenis išsaugo sveikųjų skaičių masyvo pavidalu vidinėje atmintyje grupuojant eilučių reikšmes po 32 logines reikšmes, vietoje loginio tipo masyvo. Tokiu būdu sutaupoma naudojama masyvo adresavimui reikalinga vidinė atmintis, nepadarius jokios įtakos operacijų efektyvumui,

kadangi kiekvienam loginio tipo masyvo įrašui JVM aplinkoje yra išskiriami 8 bitai, vietoje 1 bito.

Norint sukurti naują McEliece klasę vartotojui reikia pateikti m ir t parametrus jos konstruktoriui, kuris sukuria viešąjį ir privatų raktus. Taip pat, galima pateikti papildomus parametrus, pagal kuriuos bus registruojami šios klasės veiksmai, pavyzdžiui, privataus ir viešojo rakto generavimo žingsniai arba šifravimo žingsniai. Ši klasė turi du pagrindinius metodus: šifravimo ir dešifravimo. Patogumo dėlei šiems metodams sukurti pagalbiniai metodai, kurie gali šifruoti ir dešifruoti trijų tipų žinutes: užkoduotas baitais, dvimačiu sveikųjų skaičių masyvu susidedančiu iš nulio ar vieneto reikšmių arba pateikta „Bouncy Castle“ vektoriaus abstrakcija.

5.1.2. Pasiruošimas atakoms ir statistinių duomenų rinkimas

Atakų tyrimui realizuota taikomoji programa. Taikomosios programos duomenys įvedami ir išvedami terminale.

Vartotojui pateikiami pasirinkimai:

1. Pasirinkti ataką iš įgyvendintų atakų sąrašo.
2. Pasirinkti norimus m ir t saugumo parametrus, norimai McEliece kriptosistemai tirti.
3. Pasirinkti kokiam kiekiui tokių atsitiktinių raktų porų bus vykdomos atakos.
4. Pasirinkti, kiek atsitiktinių žinučių bus bandoma dešifruoti vienai raktų porai.
5. Pasirinkti atakos algoritmui reikalingus parametrus, jeigu tokie yra.
6. Pasirinkti kokius darbo su programa faktus fiksuoti. Čia galima pasirinkti tarp: viešojo ir privataus raktų generavimo, šifro kūrimo, kiekvienos raktų poros statistinių duomenų išvedimo, visų raktų porų statistinių duomenų išvedimo bei vidinės atminties naudojimo atakos metu.

Toliau programa vykdo atakas pagal vartotojo pasirinkimus. Atakų rezultatas – išvedami vykdymo laikai ir surinkti statistiniais duomenys. Pagrindiniai renkami ir išvedami statistiniai duomenys: atakos vykdymo trukmių vidurkis ir mediana, mažiausiai trukusi ataka, ilgiausiai trukusi ataka, tikimybė atakai įveikti šifrą iš pirmo karto bei tikėtinas bandymų skaičius.

5.2. Apibendrinta informacijos rinkinio dekodavimo ataka

Šiai atakai įgyvendinti kriptoanalitikas privalo perimti siunčiamą žinutės šifrą c ir gauti gavėjo viešąjį raktą (G', t). Ataka laikoma sėkminga, jeigu pavyksta dešifruoti siųstą pranešimą m.

Šios atakos metu perimtame šifre c bandoma atspėti k pozicijų rinkinį I, kuris nebūtų paveiktas klaidų vektoriaus arba paveiktas labai nedaug, bei kurio pozicijas atitinkantys generuojančios matricos G' stulpeliai sudarytų naują matricą G'_I, kuriai galima būtų paskaičiuoti atvirkštinę ${G'_I}^{-1}$. Tokį I rinkinį vadinsime informacijos rinkiniu. Radus informacijos rinkinį bandoma atspėti prie šifro pridėtą klaidų vektorių e, kurio Hamingo svoris $wt(e)$ būtų lygus t. Radus informacijos rinkinį I ir klaidų vektorių e tiesinės algebros pagalba lengvai dešifruojamas perimtas šifras c.

5.2.1. Algoritmo realizacija

Remiantis P. J. Lee ir E. F. Brickell straipsniu [LB88] bei literatūroje išnagrinėtais šaltiniais [EOS07; Pet10] įgyvendintas apibendrintos informacijos rinkinio dekodavimo atakos algoritmas. Algoritmui įgyvendinti realizuotas trijų žingsnių metodas, kurio įvestis n ilgio žinutės šifro vektorius c, paieškos dydžio parametras p, kur $0 \leq p \leq t$, bei viešojo rakto parametrai $k \times n$ matmenų generuojanti matrica G' ir pridedamų klaidų skaičius t.

Pirmame žingsnyje iš visų galimų G' stulpelių pozicijų $0, \ldots, n-1$ sukuriamas n ilgio sąrašas L. Tada atsitiktiniu būdu iš jo parenkame k pozicijų ir sukuriame naują sąrašą I.

Antrajame žingsnyje iš I sąrašo pozicijas atitinkančių stulpelių iš G' sudaroma nauja $k \times k$ matmenų matricą G'_I ir naudojant Gauso metodą bandoma paskaičiuoti šios matricos atvirkštinę ${G'_I}^{-1}$. Jeigu pasirinktam rinkiniui I galima apskaičiuoti atvirkštinę ${G'_I}^{-1}$, vadinasi visi pasirinkti stulpeliai yra tiesiškai nepriklausomi ir galima paskaičiuoti $Q_I = {G'_I}^{-1} G'$, bei tikėtiną klaidų vektorių $e' = c - c_I Q_I$. Nepavykus rasti atvirkštinės ${G'_I}^{-1}$, grįžtama į pirmą žingsnį ir atsitiktiniu būdu parenkamas naujas I rinkinys.

Trečiajame žingsnyje pagal pasirinktą paieškos dydžio parametrą p atliekamas klaidų vektoriaus e paieškos algoritmas. Jeigu pasirinktas $p = 0$, tada tikrinamas gauto klaidų vektoriaus Hamingo svoris $wt(e')$ ir jeigu jis yra ne didesnis už klaidų skaičių t, tada pasirinktos I pozicijos yra informacijos rinkinys nepaveiktas klaidų vektoriaus ir dešifruojama žinutė $m = c_I {G'_I}^{-1}$. Jeigu $wt(e') > t$, pereinama į pirmą žingsnį. Čia galima pastebėti, kad tikimybė išrinkti klaidų nepaveiktą rinkinį I yra:

$$Pr_{p=0}(n, k, t) = \frac{\binom{n-t}{k}}{\binom{n}{k}}. \tag{20}$$

Pavyzdžiui, su pirminiais McEliece pasiūlytais parametrais $(n, k, t) = (1024, 524, 50)$ tikimybė yra labai maža – $7{,}26 \times 10^{-17}$.

Jeigu pasirinktas $p > 0$, tada bandoma atspėti pridėtą klaidų vektorių e. Siekiant kompensuoti iki p leidžiamų klaidų pasirinktame informacijos rinkinyje I, klaidų vektoriaus radimui reikia išnagrinėti visas galimas $0 \ldots p$ eilučių sumas iš Q_I ir jas atimti iš antrajame žingsnyje apskaičiuoto e'. Šiai paieškai yra naudojami du ciklai. Pirmasis priskiria p_I klaidų paieškos dydžio reikšmes $0 \ldots p$, o antrasis priskiria a_I klaidų pozicijų sąrašui visus įmanomus p_I ilgio kombinatorinius derinius iš $0 \ldots k-1$ galimų pozicijų. Tokių kombinatorinių derinių skaičius yra $\binom{k}{p}$. Po kiekvieno naujo sukurto a_I pozicijų sąrašo yra sudedamos a_I pozicijų eilutes iš Q_I ir gaunamas vektorius e'_n, bei apskaičiuojamas naujas potencialiai teisingas klaidų vektorius $e = e' - e'_n$. Jeigu vektoriaus e Hamingo svoris yra lygus klaidų skaičiui t, tuomet galime dešifruoti pranešimą $m = (c_I + e_I) {G'_I}^{-1}$. Išbandžius visus p galimybių derinius ir neradus vektoriaus, kuris tenkintų lygybę $wt(e) \leq t$, pereinama į pirmą žingsnį.

Tikimybė rasti tokį klaidų vektorių e, kurio Hamingo svoris yra lygus p iš pasirinkto rinkinio I yra:

$$Pr(p, n, k, t) = \frac{\binom{k}{p}\binom{n-k}{t-p}}{\binom{n}{t}}. \tag{21}$$

Kadangi algoritmo vykdymo metu pagal pasirinktą paieškos parametrą p tikrinamos visos

$0, \ldots, p$ eilučių sumos, tikimybė yra:

$$Pr_{0\leq p\leq t}(p, n, k, t) = \sum_{i=0}^{p} \frac{\binom{k}{i}\binom{n-k}{t-i}}{\binom{n}{t}}. \tag{22}$$

Taigi, $Pr_{0\leq p\leq t}(p, n, k, t)$ yra tikimybė, kad ataka informacijos rinkinyje I ras klaidų vektorių ir įveiks šifrą pirmuoju bandymu. Nurodžius paieškos parametrą p didesnį nei ieškomų pozicijų skaičius k, laikome paieškos parametrą $p = k$, kadangi pasirinktuose k stulpeliuose negali būti daugiau nei k klaidų.

Galima pastebėti, kad šis algoritmas gali būti išlygiagretintas kiekviename žingsnyje ir tokiu būdu sumažintų vykdymo trukmę ir taptų efektyvesnis. Pavyzdžiui, pirmo žingsnio atsitiktinių derinių kūrimą galėtų vykdyti vienas procesorius, kuris sukurtą naują derinį perduotų kitam procesoriui. Šis procesorius vykdytų antrąjį žingsnį ir tikrintų, ar pasirinktas derinys yra informacijos rinkinys, jeigu taip – perduotų dar kitam procesoriui vykdyti klaidos vektoriaus paieškai naudojamą trečiąjį žingsnį.

5.2.2. Rezultatai

Šiame skyriuje pateikiami apibendrintos informacijos dekodavimo atakos rezultatai su parametrais m, t ir p.

Darbo metu buvo bandoma optimizuoti pirmąjį žingsnį, kadangi šio žingsnio metu atsitiktiniu būdu pasirenkamos k pozicijos gali kartotis su prieš tai nepasisekusiais pirmojo žingsnio bandymais, o esant mažai $Pr_{0\leq p\leq t}(p, n, k, t)$ tikimybei tokių pasikartojimų gali labai būti daug. Taip pat, esant parametrui $p = 0$, tikimybė rasti k nepriklausomų stulpelių iš G' yra labai nedidelė (žr. 20 formulę) arba dėl pridėto klaidos vektoriaus tokių stulpelių gali nebūti ir ataka nebaigs darbo. Tyrimo metu išbandytos optimizacijos:

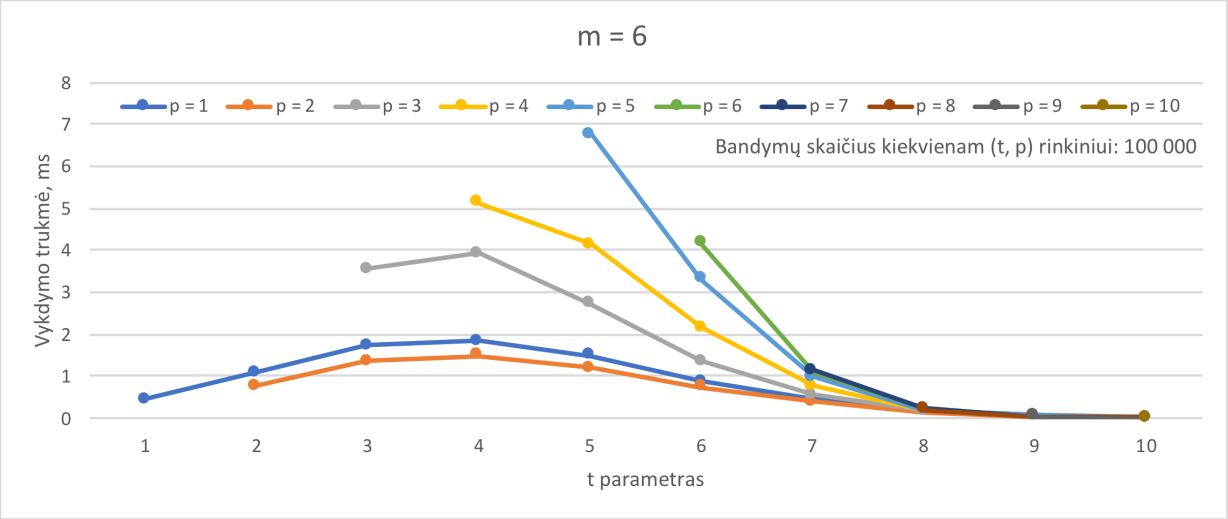
1. Prieš pradedant ataką galima sukurti visų galimų informacijos rinkinių kombinatorinių derinių sąrašą ir šio sąrašo narius išmaišyti atsitiktine tvarka. Tada kiekvieno pirmojo žingsnio metu bandyti sąrašo elementus iš eilės. Šios optimizacijos pagalba užtikrinamas atakos baigtinumas atvejais, kai neįmanoma rasti tiesiškai nepriklausomų stulpelių iš G'. Deja, eksperimento metu buvo pastebėta, kad tokio kombinatorinių derinių sąrašo kūrimas trukdavo žymiai ilgiau negu atakos vykdymas, net esant nedideliems kriptosistemos parametrams, pvz.: esant pradiniams parametrams $(m, t) = (6, 2)$, ir paskaičiavus $n = 64$, $k = 53$, reikalingų derinių skaičius: $\binom{64}{52} = 64!/52!(64 - 52)! = 3284214703056$. Taip pat, šio būdo metu kuriamas sąrašas užėmė labai daug vidinės atminties ir su saugumo parametrais $(m, t) = (8, 1)$, kompiuteriui neužteko 15 gigabaitų vidinės atminties.

2. Po kiekvieno nepavykusio atsitiktinio I rinkinio bandymo galima šį rinkinį pridėti į nesėkmingų bandymų sąrašą ir kiekvieno naujo I rinkinio kūrimo metu patikrinti, ar jo nėra nesėkmingų bandymų sąraše. Jeigu sąraše jo nėra, algoritmas gali tęsti darbą ir pereiti į antrą žingsnį, o jeigu toks rinkinys egzistuoja sąraše, algoritmui reikėtų ieškoti naujo atsitiktinio

I rinkinio. Tokių būdu visi pirmojo žingsnio atsitiktiniai bandymai yra visada be pasikartojimų. Nesėkmingų bandymų skaičiui augant tolimesni bandymai turės didesnę $Pr_{0\leq p\leq t}$ tikimybę rasti teisingą pozicijų rinkinį I. Deja, didėjant nepasisekusių bandymų skaičiui, didėja ir nepasisekusių pozicijų sąrašas, bei laikas, kurio reikia patikrinti ar atsitiktinis I nebuvo bandytas. Eksperimento metu, naudojant šią optimizaciją buvo pastebėta, kad atakų vidutiniai vykdymo laikai labai nedaug viršydavo pirminio algoritmo vidutinius laikus ir sunaudodavo daugiau vidinės atminties. Bandymams su šia optimizacija ir $m = 9$ parametru neužteko suteiktos 16 GB vidinės atminties atakai įgyvendinti.

Kadangi 1 optimizacija buvo labai neefektyvi, dėl to jos buvo atsisakyta, o kadangi 2 optimizacijos rezultatai nedaug skyrėsi nuo originalaus algoritmo rezultatų, jie yra pateikiami priede Nr. 2. Toliau pateikiami rezultatai su pirminiu algoritmu.

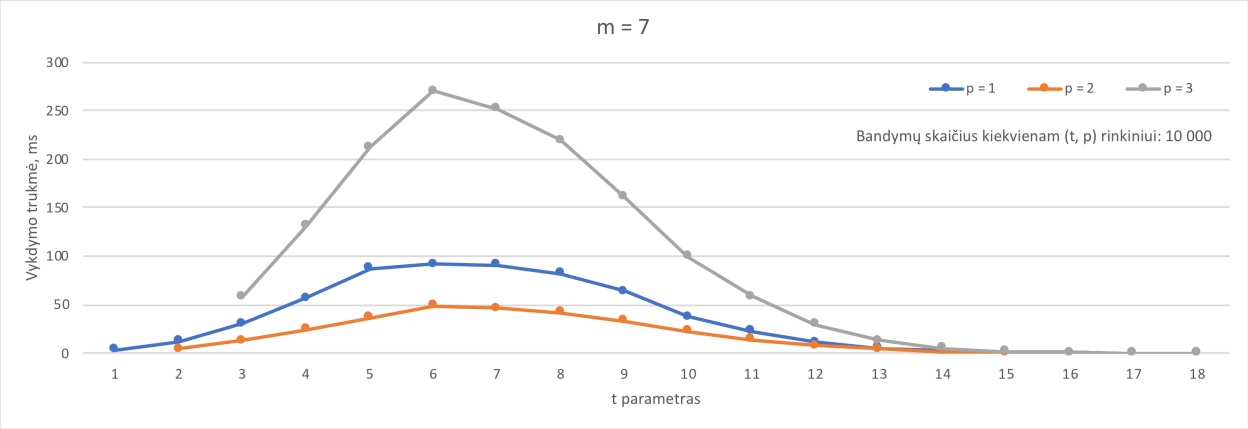
Darbo metu buvo ieškoma optimalaus p parametro, su kuriuo ataka bendruoju atveju būtų efektyviausia (truktų mažiausiai laiko), bei saugiausi m, t parametrai, su kuriais ataka užtruktų ilgiausiai. Pavyzdžiui, pasirinkus $p = t$, tikimybė radus teisingą informacijos rinkinį I dešifruoti šifrą yra lygi 1, t.y. žinutė visada randama, tačiau didėjant saugumo parametrams (m, t) tikrinamų p derinių iš k pozicijų skaičius sparčiai auga, dėl to reikia rasti optimalų p. Eksperimento metu nebuvo tiriamas paieškos parametras $p = 0$, kadangi tokiu atveju algoritmas ne visais atvejais gali įveikti šifrą ir prilygsta visų galimų variantų perrinkimo atakai (angl. *brute force*), kuri yra neefektyvi. Esant parametrui $m < 6$ ir visais galimais t, p kiekvienam m atakų vidutiniai laikai trukdavo mažiau negu 1 milisekundę ($1 * 10^{-3}$s), dėl to jų rezultatai nepateikiami. Žemiau grafiškai pateikiami įgyvendintų atakų vidutiniai vykdymo laikai. Visi atakų surinkti duomenys yra pateikiami priede Nr. 1.



1 pav. Apibendrinto informacijos rinkinio dekodavimo atakų rezultatai milisekundėmis su $m = 6$ ir visais galimais t ir p.

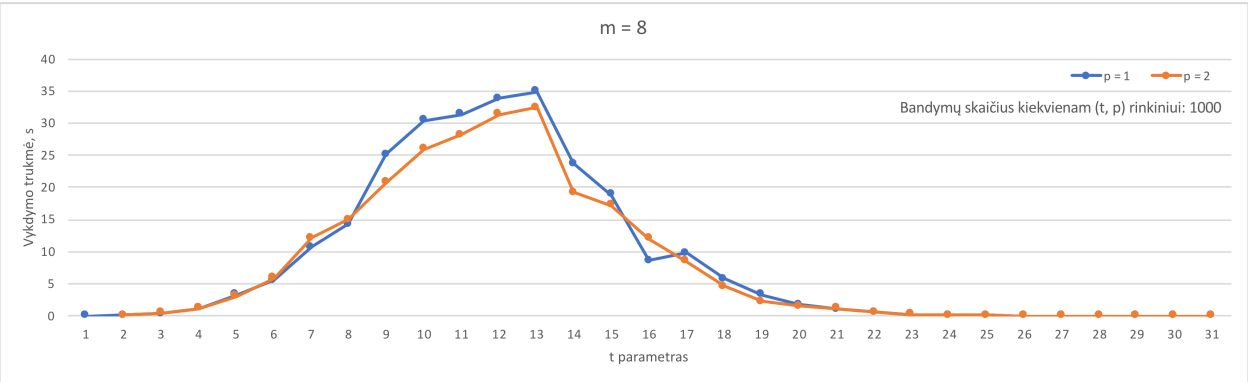
1 paveikslėlio grafike pateikti tyrimo su $m = 6$ parametru rezultatai. Čia galima matyti, kad atakos su paieškos parametru $p = 2$ įveikė šifrus greičiausiai, dėl to laikysime jį optimaliu algoritmo parametru $m = 6$ parametrui. Atakos su parametrais $p > 3$ buvo net iki 6 kartų lėtesnės negu su $p = 2$, kadangi didinant parametrą p sparčiai didėja tikrinamų derinių skaičius, kuris yra reikalingas

klaidos vektoriaus paieškai. Kadangi vykdymo laikai su $p = 1$ ir $p = 3$ atsiliko nedaug nuo $p = 2$, dėl to tolimesniame tyrime su parametru $m = 7$ naudojami $p = 1, \ldots, 3$ paieškos parametrai.



2 pav. Apibendrinto informacijos rinkinio dekodavimo atakų rezultatai milisekundėmis su $m = 7$, visais galimais t ir $p = 1, \ldots, 3$ parametrais.

2 paveiksle galima matyti, kad esant paieškos parametrui $p = 3$ atakos įveikia šifrus žymiai lėčiau negu su $p = 1$ ar $p = 2$. Abu parametrai $p = 1$ ir $p = 2$ yra efektyvūs ir vidutiniai vykdymo laikai su jais skiriasi nedaug, tačiau beveik visais atvejais atakos su $p = 2$ buvo optimalus. Tolimesniame tyrime su parametru $m = 8$ naudojami efektyvūs $p = 1$ ir $p = 2$ paieškos parametrai.



3 pav. Apibendrinto informacijos rinkinio dekodavimo atakų rezultatai sekundėmis su $m = 8$, visais galimais t, $p = 1$ ir $p = 2$ parametrais jiems.

3 paveiksle galima matyti, kad bendruoju atveju $p = 2$ įveikė šifrus greičiau negu $p = 1$. Kadangi bandymų skaičius buvo tik 1000, todėl šiame grafike matomi didesni šuoliai, negu grafikuose su mažesniais m parametrais.

Tyrimo metu buvo tiriamas McEliece kriptosistemos saugumas su $m = 9$, $p = 2$ ir visomis galimomis $t = 1 \ldots 56$ reikšmėmis, tačiau šios atakos truko pernelyg ilgą laiką, pavyzdžiui vienam šifrui dešifruoti su parametru $t = 9$ prireikė daugiau nei 2 valandų, su $t = 13$ – daugiau nei 5 valandų, o su $t = 17$ ataka truko daugiau nei 24 valandas, dėl to jų grafikai nėra pateikiami ir su didesniais m parametrais tyrimai nebebuvo vykdomi.

Visuose grafikuose $p = 2$ bendruoju atveju buvo efektyviausias, dėl to galime teigti, kad su didesniais m parametrais šis p išliks optimalus.

1 lentelė. Saugiausi praktinių tyrimų m ir t parametrų rezultatai

m	t	Bandymų skaičius	Vidutinė trukmė, ms
6	4	1000000	1,492
7	6	100000	48,606
8	13	1000	34866,75

2 lentelėje pateikiami atakų prieš McEliece kriptosistemas rezultatai su saugiausiu parametru t kiekvienam m, naudojant optimalų $p = 2$.

2 lentelė. Saugiausių m ir t parametrų praktinių rezultatų palyginimas su teoriniais

m	Praktinis optimalus t	Teorinis optimalus t
6	4	5
7	6	8
8	13	13

3 lentelėje pateiktas gautų saugiausių m, t praktinių rezultatų palyginimas su teoriniais. Teoriniai rezultatai yra suskaičiuoti pagal atakos vidutinio laiko išlaidų modelio formulę:

$$Cost(p, n, k, t) = \frac{\frac{1}{2}(n-k)^2(n+k) + \binom{k}{p}p(n-k)}{Pr_{0 \le p \le t}(p, n, k, t)}, \tag{23}$$

kur $\frac{1}{2}(n-k)^2(n+k)$ yra antrojo žingsnio matricos atvirkštinės skaičiavimo Gauso metodu vidutinio laiko įvertis, $\binom{k}{p}$ yra trečiojo žingsnio mažo Hamingo svorio klaidos vektoriaus paieškos operacijų skaičius ir $p(n-k)$ – trečiojo žingsnio naujo klaidų vektoriaus atimties operacijų skaičius. Galima matyti, kad esant $m = 6$ ir $m = 7$ praktiniai t parametro rezultatai nesutapo su teoriniais. Kadangi su šiais saugumo parametrais m buvo atlikta didelis skaičius bandymų, t. y. su $m = 6$ buvo atlikta 1000000 ir su $m = 7$ – 100000 bandymų, galima daryti išvada, kad jie yra saugesni negu pateikti teoriniai ir toliau darbe tokie t parametrai bus laikomi saugiausiais pateiktiems m. Tokius rezultatus galėjo lemti daug faktorių, pavyzdžiui, gerų atsitiktinių stulpelių pasirinkimo sėkmė esant mažesnei tikimybei, blogų stulpelių pasirinkimo nesėkmė esant didesnei tikimybei arba greiti neteisingų pasirinktų stulpelių atpažinimai Gauso metodu.

Pagal gautus 23 modelio ir praktinių atakų vykdymo laikų rezultatus 4 lentelėje pateikiamos vidutinių atakų vykdymo laiko prognozės su didesniais m parametrais. Prognozei apskaičiuoti buvo dalinami praktinių laikų rezultatai su modelio prognozuojamais ir iš gautų konstantų paimamas vidurkis.

3 lentelė. Apibendrintos informacijos dekodavimo atakų su $m = 9, 10, 11, 12$ vidutinių vykdymo laikų prognozavimas

m	Optimalus t	Vykdymo laikas
9	22	10,17 dienų
10	39	2043,12 metų
11	70	2,69 * 10^{17} metų
12	128	1,59 * 10^{43} metų

5.3. Žinomo dalinio teksto ataka

Žinomo dalinio teksto atakos tikslas yra sumažinti McEliece kriptosistemos perimto šifro saugumą. Šiai atakai įgyvendinti kriptoanalitikas privalo žinoti dalį siunčiamo pradinio teksto m, perimti siunčiamą žinutės šifrą c ir gauti gavėjo viešojo rakto generuojančią matricą G'. Ataka laikoma sėkminga, jeigu pavyksta sumažinti šifro saugumą, t.y. sumažinti generuojančios matricos G' matmenis ir dalinai dešifruoti šifrą c.

Įvykdžius šią ataką, galima taikyti kitą ataką iš pavienių šifrų atakų grupės, kurių tikslas atkurti pradinį tekstą. Šiame darbe sumažintiems saugumo parametrams taikysime anksčiau minėtą apibendrintą informacijos rinkinio dekodavimo ataką (žr. 5.2 poskyrį).

5.3.1. Algoritmo realizacija

Remiantis [CGN+15; CS98; KI01] literatūros šaltiniais įgyvendintas žinomo dalinio teksto atakos algoritmas. Algoritmui įgyvendinti realizuotas metodas, kurio įvestis m_r – žinoma žinutės m bitų dalis iš dešinės vektoriaus pavidalu, perimtas šios žinutės šifro vektorius c bei žinutės gavėjo viešasis raktas (G', t). Šis metodas iš pateikto m_r suskaičiuoja žinomų bitų skaičių r ir pagal šį skaičių padalina generuojančią matricą G' į G'_r ir G'_l matricas, kur G'_r yra matricos G' viršutinė k eilučių dalis, o G'_l – apatinė $k - r$ eilučių dalis. Tada paskaičiuoja skaliarinę sandaugą $m_l * G'_l$ ir rezultatą atima iš pateikto šifro c, taip gaunant naują dalinai dešifruotą šifrą c'. Čia rezultatu laikomi c' ir G'_r, kurie toliau perduodami apibendrintą informacijos rinkinio dekodavimo atakos algoritmui, kuri bus vykdomas su optimaliu paieškos parametru $p = 2$.

Galima pastebėti, kad kriptoanalitikas norėdamas pasinaudoti šia ataka ir žinodamas pozicijas ne iš eilės, galėtų perstatų būdu sukeisti c pozicijas ir atitinkamai G' eilučių pozicijas taip, kad žinomi bitai būtų iš eilės, ir gautą žinutės rezultatą c iš pavienių šifrų atakos atkeisti pagal perstatas atgal.

5.3.2. Rezultatai

Šiame poskyryje apibendrinami žinomo dalinio teksto atakų rezultatai įvykdyti kartu su apibendrinto informacijos rinkinio dekodavimo algoritmu.

Tyrimo metu atsitiktinių žinučių šifravimui buvo naudojami saugiausi (m,t) McEliece parametrai. Atakos įgyvendintos žinant 10, 25, 50 ir 75 procentus pradinio žinutės teksto. Lentelėse pateikiami atakų vidutiniai vykdymo laikai pagal pasirinktas dalinio teksto dalis bei vykdymo laikų

pagreitėjimas procentais lyginant su apibendrinto informacijos rinkinio dekodavimo atakų rezultatais.

4 lentelė. Žinomo dalinio teksto atakų vidutiniai vykdymo laiko rezultatai (ms) su saugumo parametrais $m = 6$ ir $t = 4$

Žinomo teksto dalis	Vykdymo laikas (ms)	Bandymų skaičius	Atakos pagreitėjimas (%)
10% (4 iš 40)	1,049	1000000	142
25% (10 iš 40)	0,425	1000000	351
50% (20 iš 40)	0,147	1000000	1015
75% (30 iš 40)	0,053	1000000	2815

5 lentelė. Žinomo dalinio teksto atakų vidutiniai vykdymo laiko rezultatai (ms) su saugumo parametrais $m = 7$ ir $t = 6$

Žinomo teksto dalis	Vykdymo laikas (ms)	Bandymų skaičius	Atakos pagreitėjimas (%)
10% (9 iš 86)	16,823	100000	289
25% (22 iš 86)	10,111	100000	481
50% (43 iš 86)	1,475	100000	3295
75% (65 iš 86)	0,341	100000	14252

6 lentelė. Žinomo dalinio teksto atakų vidutiniai vykdymo laiko rezultatai (ms) su saugumo parametrais $m = 8$ ir $t = 13$

Žinomo teksto dalis	Vykdymo laikas (ms)	Bandymų skaičius	Atakos pagreitėjimas (%)
10% (15 iš 152)	8771,15	1000	398
25% (38 iš 152)	328,165	1000	10625
50% (76 iš 152)	15,14	1000	230296
75% (114 iš 152)	0,827	1000	4216052

Iš pateiktų rezultatų 4, 5 ir 6 lentelėse, galima matyti, kad net žinant 10% pradinio teksto atakų vykdymo laikas pagreitėja daugiau nei 140%, o žinant 75% – daugiau nei 14000% ir atakos įveikia šifrus per mažiau nei 1 milisekundę. Toliau pateikiami rezultatai su didesniais m parametrais, kurių tyrimo metu nepavyko įveikti su apibendrinto informacijos rinkinio dekodavimo algoritmu.

7 lentelė. Žinomo dalinio teksto atakų vidutiniai vykdymo laiko rezultatai (s) su saugumo parametrais $m = 9$ ir $t = 22$

Žinomo teksto dalis	Vykdymo laikas (s)	Bandymų skaičius
10% (31 iš 314)	-[2]	1
25% (79 iš 314)	278,933	1000
50% (157 iš 314)	2,576	1000
75% (236 iš 314)	0,045	1000

8 lentelė. Žinomo dalinio teksto atakų vidutiniai vykdymo laiko rezultatai (s) su saugumo parametrais $m = 10$ ir $t = 39$

Žinomo teksto dalis	Vykdymo laikas (s)	Bandymų skaičius
10% (63 iš 634)	-	1
25% (159 iš 634)	-	1
50% (317 iš 634)	3,423	100
75% (476 iš 634)	0,887	100

9 lentelė. Žinomo dalinio teksto atakų vidutiniai vykdymo laiko rezultatai (s) su saugumo parametrais $m = 11$ ir $t = 70$

Žinomo teksto dalis	Vykdymo laikas (s)	Bandymų skaičius
10% (128 iš 1278)	-	1
25% (320 iš 1278)	-	1
50% (639 iš 1278)	-	1
75% (959 iš 1278)	278,4527	100

Lentelėse 7, 8 ir 9 galima matyti, kad šių atakų kombinacija įveikė McEliece kriptosistemos šifrus su laikomais saugiais $m = 10, 11$ parametrais jai. Žinant 50% pradinio teksto galima įveikti pirminės McEliece kriptosistemos su saugumo parametru $m = 10$ šifrą per mažiau nei 5 sekundes. Tyrimo metu sukurtų šifrų su $m \geq 12$ nepavyko dešifruoti per 24 valandas net žinant 75% šifro dalies. Nors žinomo dalinio teksto atakos pagalba nėra sužinoma jokios informacijos apie pridedamą klaidos vektorių e, tačiau dėl sumažintos viešojo rakto generuojančios matricos G'_l ir dalinai dešifruoto šifro c' šių atakų kombinacija įveikia šifrus greičiau negu apibendrinto informacijos rinkinio dekodavimo ataka su tokiais pačiais parametrais, kadangi ieškomas mažesnis skaičius nepriklausomų stulpelių iš sumažintos G'_l matricos.

Norint apsisaugoti nuo žinomo dalinio teksto atakos būtina pakeisti McEliece kriptosistemos šifravimo ir dešifravimo etapus, kadangi tokiu būdu, kad šios atakos pagalba net žinant dalį pradinės žinutės negalima būtų dalinai dešifruoti šifro c. Vienas iš galimų modifikacijos būdų yra pasinaudoti Marco Baldi ir Franco Chiaraluce pasiūlyta kokia nors vienakrypte funkcija $h(e)$, kurios įvestis klaidų vektorius e, o rezultatas k ilgio vektorius [BC07]. Tuomet pakeitus šifravimo operaciją į $c = (m + h(e))G' + e$, šifravimo metu pridėtas klaidos vektorius pakeistų siunčiamą pranešimą ir nežinant jo būtų neįmanoma dalinai dešifruoti šifro. Norint dešifruoti tokį šifrą, radus pridėtą

[2] Su pasirinktais parametrais šifro nepavyko dešifruoti per 24 valandas.

klaidos vektorių e reikėtų paskaičiuoti jam $h(e)$ ir nesunkiai gauti dešifruotą žinutę $m = (m + h(e)) - h(e)$.

5.4. Pranešimo persiuntimo ataka

Pranešimo persiuntimo atakos tikslas yra dešifruoti pranešimą. Šiai atakai įgyvendinti kriptoanalitikas privalo perimti tam pačiam gavėjui bent du kartus siunčiamo tokio pačio pranešimo m šifrus $c_1, c_2, \ldots, c_i$ bei žinutės gavėjo viešąjį raktą (G', t). Ataka laikoma sėkminga, jeigu pavyksta teisingai dešifruoti pranešimą m, tačiau ne visada tai yra įmanoma. Pavyzdžiui, kai perimti $c_1, c_2, \ldots, c_i$ šifrai yra vienodi, ši ataka negali dešifruoti pranešimo, kadangi pridėti atsitiktiniai klaidų vektoriai yra vienodi ir nesuteikia jokios informacijos kriptoanalitikui. Taip pat, jeigu perimtų šifrų sugeneruoti klaidų vektoriai $e_1, e_2, \ldots, e_i$ sutampa daugumoje pozicijų, ši ataka gali grąžinti neteisingą žinutę. Tokiu atveju kriptoanalitikas gali naudodamas kokią nors gautos žinutės teisingumo validavimo euristiką, pavyzdžiui, tikrinti dešifruotų žinučių prasmę, ir kartoti šią ataką kol ji bus sėkminga. Sužinojęs bent kelias vieno šifro c klaidos vektoriaus e' pozicijas kriptoanalitikas gali pasinaudoti kita ataka iš pavienių šifrų atakų grupės su sumažintu saugumo parametru t bei pakeistu šifru $c' = c - e'$. Remiantis Berson [Ber97] straipsnyje pateikta spraga buvo realizuoti du skirtingi atakų algoritmai.

5.4.1. Algoritmų realizacijos

Abiejų algoritmų pradiniai duomenys – tos pačios žinutės du skirtingi šifro n ilgio vektoriai c_1, c_2 bei viešojo rakto parametrai $k \times n$ matmenų generuojanti matrica G' ir t. Pirmojo algoritmo tikslas yra rasti vieno iš perimtų šifro vektoriaus, pavyzdžiui, c_1, klaidos vektorių e_1 ir atėmus jį iš c_1 išspręsti tiesinę lygtį. Šios lygties rezultatas – dešifruotas žinutės vektorius. Antrojo algoritmo tikslas yra rasti tiesinę priklausomybę tarp šifrų pozicijų, kuriose tikėtina, kad nėra pridėtos klaidos, ir tų pačių viešojo rakto tiesiškai nepriklausomų stulpelių pozicijų. Radus šias pozicijas išsprendžiama nesudėtinga lygtis ir gaunama dešifruota žinutė.

5.4.1.1. Klaidų vektoriaus paieškos algoritmo realizacija

Sukurtoje programoje buvo realizuotas vienas esminis atakos metodas, kuris susidaro iš dviejų dalių. Pirmosios dalies metu bandoma rasti šifro vektorių be klaidų c' iš dviejų pateiktų šifrų c_1 ir c_2, o antrosios dalies metu išsprendžiama tiesinė lygtis $G'^{\intercal} m = c'$, kur m yra ieškomos žinutės vektorius.

Pirmiausia, metode apskaičiuojama gautų šifrų vektorių suma $c_{\text{sum}} = c_1 + c_2$ ir patikrinama, ar šios sumos Hamingo svoris $wt(c_{\text{sum}})$ yra nelygus nuliui bei ne didesnis nei padvigubintas viešojo rakto galimų klaidų skaičius t. Jeigu $wt(c_{\text{sum}}) = 0$, tada iš pateiktų šifrų neįmanoma sužinoti jokios papildomos informacijos apie jų klaidų vektorius, kadangi jeigu $c_1 = c_2$, tada $e_1 = e_2$, ir tokiu atveju ši ataka yra labai neefektyvi, bei prilygsta visų galimų variantų perrinkimo atakai (angl. *brute force*). Esant $wt(c_{\text{sum}}) > 2 \cdot t$ ataka yra neįmanoma, kadangi pateikti šifrai nėra užšifruoti tos pačios žinutės arba užšifruoti skirtingomis viešojo rakto generuojančios matricomis.

Toliau metode surenkamas sąrašas L_1 iš vektoriaus c_{sum} pozicijų, kuriose nesutampa perimtų šifrų c_1 ir c_2 reikšmės. Kitaip tariant, $c_{sum_{1\ldots n}} = 1$, kur n – vektoriaus c_{sum} ilgis. Likusios pozicijos priskiriamos sąrašui L_0. Tuomet tikrinama, ar Hamingo svoris $wt(c_{\text{sum}})$ yra lygus dvigubam galimų klaidų skaičiui. Jeigu lygus, tada prie abiejų šifrų pridėti klaidos vektoriai nesutampa nei vienoje pozicijoje ir galima spėti klaidos vektorių e atsitiktiniu būdu pasirenkant t skirtingas pozicijas iš surinkto L_1 sąrašo pozicijų. Čia galime pastebėti, kad tokių bandymų gali būti $\binom{2 \cdot t}{t}$ ir augant t parametrui, sparčiai auga galimų klaidų vektorių derinių skaičius, pavyzdžiui, kai $t = 5$ galimų derinių skaičius 252, o kai $t = 10$, derinių skaičius 184756. Kuomet Hamingo svoris $wt(c_{\text{sum}})$ nėra lygus dvigubam galimų klaidų skaičiui, suskaičiuojama kiek klaidos vektorių pozicijų sutampa $x = (2 \cdot t - wt(c_{\text{sum}}))/2$. Tada atsitiktiniu būdu parenkamos $t - x$ skirtingos pozicijos iš sąrašo L_1 ir x skirtingos pozicijos iš sąrašo L_0. Žinant šifro ilgį n galima suskaičiuoti galimų unikalių bandymų skaičių klaidos vektoriui atspėti – $\binom{2 \cdot (t-x)}{t-x} \cdot \binom{n-t}{x}$.

Antroje dalyje iš gautų pozicijų sukuriamas klaidos vektorius e ir apskaičiuojamas šifras $c' = c_1 - e$. Tada, transponuojama G' ir išsprendžiama tiesinė lygtis $G'^{\intercal} m = c'$. Pirmiausia, prie $G'^{\intercal}$ pridedamas sprendimo stulpelis c' ir sprendžiama lygtis Gauso metodu, kurio pagalba duotoji matrica naudojant eilučių sudėties ir sukeitimo operacijas yra suvedama į laiptuotą tiesinių lygčių sistemą. Gautas lygties sprendimas – potencialiai teisinga žinutė m.

5.4.1.2. Tiesiškai nepriklausomų stulpelių paieškos algoritmo realizacija

Sukurtoje programoje buvo realizuotas vienas esminis atakos metodas, kuriame ieškomos nepaveiktos klaidų vektoriaus šifro pozicijos, kurias atitinkantys viešosios matricos stulpeliai būtų tiesiškai nepriklausomi. Kaip ir pirmajame algoritme apskaičiuojama pateiktų šifrų vektorių suma c_{sum}, šios sumos Hamingo svoris $wt(c_{\text{sum}})$ ir patikrinama, ar ataka įgyvendinama. Toliau, metode surenkamas sąrašas L_0 iš vektoriaus c_{sum} pozicijų, kuriose sutampa perimtų šifrų c_1 ir c_2 reikšmės, kitaip tariant, $c_{sum_{1\ldots n}} = 0$, kur n – vektoriaus c_{sum} ilgis. Tikėtina, kad dauguma sąrašo L_0 pozicijų arba net visos pozicijos nebus paveiktos klaidų vektoriaus. Klaidingų pozicijų skaičių galime apskaičiuoti $p_e = t - wt(c_{\text{sum}})/2$.

Toliau, atsitiktiniu būdu išrenkama k pozicijų iš sąrašo L_0 ir sukuriamas naujas sąrašas i. Iš i sąrašo pozicijas atitinkančių stulpelių iš G' sudaroma nauja $k \times k$ matmenų matrica G'_i ir naudojant Gauso metodą bandoma rasti šios matricos atvirkštinę $G_i'^{-1}$. Jeigu atvirkštinę matricą pavyksta rasti, vadinasi visi pasirinkti stulpeliai yra tiesiškai nepriklausomi ir galime rasti potencialiai teisingą žinutę m. Tokiu atveju galime sudaryti naują k ilgio vektorių c'_1 iš pasirinktų i pozicijų ir žinutę $m = c'_1 G_i'^{-1}$. Jeigu atvirkštinės $G_i'^{-1}$ nepavyksta rasti, bandoma atsitiktiniu būdu išsirinkti kitokį k pozicijų rinkinį. Čia galima pastebėti, kad iš viso tokių skirtingų bandymų gali būtų $\binom{|L_0|}{k}$. Kadangi yra įmanoma, kad tiesiškai nepriklausomų stulpelių rinkinys iš L_0 pozicijų neegzistuoja, t.y. tokie stulpeliai yra pakeisti dėl pridėto klaidos vektoriaus, todėl kiekvieno atsitiktinio bandymo metu tikrinama, ar įmanomų bandymų skaičiuos nėra pasiektas. Jeigu pasiektas – grąžinama klaida, kad ataka neįmanoma.

Tikėtinas šios atakos bandymų skaičius yra lygus $\frac{\binom{|L_0|}{k}}{\binom{|L_0|-p_e}{k}}$, kur p_e yra klaidingų pozicijų skai-

čius L_0 sąraše. Galima pastebėti, kad esant $wt(c_{\text{sum}}) = 2 \cdot t$ ($p_e = 0$), ši ataka visada bus sėkminga iš pirmo karto.

5.4.2. Rezultatai

Šiame poskyryje apibendrinami pranešimo persiuntimo atakų algoritmų rezultatai.

Tyrimo metu šiframs kurti buvo naudojami sukurti McEliece kriptosistemos raktai su saugiausiais (m,t) saugumo parametrais. Kadangi abu algoritmai ne visada gali grąžinti teisingai dešifruotą žinutę jie yra vykdomi tol, kol teisinga žinutė yra randama. Atakų vykdymo metu kaupiami nesėkmingų bandymų ir tikėtinų bandymų skaičiai. Tyrimo metu abiems atakų algoritmams buvo pateikiami tokie patys šifrai.

10 lentelė. Klaidų vektoriaus paieškos algoritmo vykdymo rezultatai.

m	t	Vykdymo laikas (ms)	Bandymu skaičius	Tikėtinas bandymų skaičius	Algoritmo bandymų skaičius
6	4	0,53	1000	474105	665790
7	6	20,55	1000	10701600	18807402

11 lentelė. Tiesiškai nepriklausomų stulpelių paieškos algoritmo vykdymo rezultatai.

m	t	Vykdymo laikas (ms)	Bandymu skaičius	Tikėtinas bandymų skaičius	Algoritmo bandymų skaičius
6	4	0,647	1000	1562	2701
7	6	1,627	1000	2052	2873
8	13	2,737	1000	4279	3113
9	22	14,507	1000	6033	5474
10	39	46,595	1000	16255	15390
11	70	206,84	1000	10332	9932

Lentelėje 10 pateikti klaidų vektoriaus paieškos algoritmo rezultatai su $m = 6$ ir $m = 7$. Čia galime matyti, kad bandymų skaičius reikalingas įveikti šifrus sparčiai didėja, o kadangi matricos su šiais parametrais nėra didelės šių lygties sprendimo paieška yra efektyvi, tačiau palyginus rezultatus su 11 lentelėje gautais galima matyti, kad tiesiškai nepriklausomų stulpelių paieškos algoritmas yra žymiai efektyvesnis bei jam reikia žymiai mažiau pakartotinių bandymų. Kriptoanalitikas pasinaudojęs tiesiškai nepriklausomų stulpelių paieškos algoritmu gali greitai dešifruoti pakartotinai siųstus pranešimus su saugiais laikomais $m = 10$ ir $m = 11$.

Norint apsisaugoti nuo pranešimo persiuntimo ir susijusių pranešimų atakų būtina pakeisti McEliece kriptosistemos šifravimo ir dešifravimo etapus, kadangi pranešimo persiuntimo atakos remiasi pridėto klaidos vektoriaus pozicijų paieška. Vienas iš galimų modifikacijos jau ankščiau pateiktas vienakryptės funkcijos metodas (žr. 5.3.2 skirsnį). Šios optimizacijos pagalba klaidos vektoriaus neišeitų lengvai atpažint net ir siunčiant tokį patį pranešimą.

Rezultatai ir išvados

Pirmiausiai darbe išanalizuota McEliece kriptosistemos sandara, veikimo principai ir atskirų sudedamųjų dalių įtaką bendram sistemos saugumui. Pagrindiniai kriptosistemos saugumo parametrai yra laikomi m ir t, didinant šiuos parametrus saugumo lygis kyla, tačiau taip pat sparčiai didėja ir šios kriptosistemos viešojo ir privataus raktų ilgiai. Darbe parodoma, kad privataus rakto matricos S ir P bei pridedamas klaidos vektorius e yra svarbiausi šios sistemos elementai saugumui. Kriptoanalitikas turėdamas S ir P matricas galėtų nesudėtingai rasti Goppa kodo matricą bei patį polinomą, tokiu būdų įveikdamas kriptosistemą. Radęs klaidų vektorių kriptoanalitikas nesudėtingai įveiktų šifrą.

Darbe ištirtas ir realizuotas apibendrintos informacijos dekodavimo atakos algoritmas su mažais parametrais. Apibendrintos informacijos dekodavimo atakos tyrimo metu nustatytas optimalus paieškos parametras $p = 2$, saugiausi t parametrai ištirtiems m, kurie tyrimo metu buvo saugesni negu pateikiami teoriniai. Darbo metu gauti rezultatai parodė, kad net su asmeniniu kompiuteriu galima įveikti, bet kokį McEliece kriptosistemos šifrą su parametru $m < 9$, o kriptoanalitikui turint daugiau resursų arba išlygiagretinus algoritmo žingsnius sukurtus šifrus įveiktų ir su parametru $m = 9$. Pateikti apibendrintos informacijos dekodavimo atakos rezultatai parodo, kad informacijos rinkiniu paremtos atakos nors šifrus dešifruoja lėčiau negu kritinės atakos, tačiau jos kelia didelę grėsmę kriptosistemoms su McEliece pasiūlytu $m = 10$ parametru.

Darbe ištirtas ir realizuotas žinomo dalinio teksto algoritmas kartu su apibendrintos informacijos dekodavimo ataka. Pateiktuose rezultatuose parodoma, kad šių atakų kombinacija įveikia šifrus labai efektyviai. Tyrimo metu parodoma, kad žinant 75% pradinio žinutės teksto galima efektyviai įveikti McEliece kriptosistemos sukurtus šifrus su laikomais saugiais $m = 10$ ar $m = 11$ saugumo parametrais.

Darbe ištirta pranešimo persiuntimo atakos ir susijusių pranešimų saugumo spraga ir pasinaudojus šia spraga realizuoti du algoritmai atakoms įgyvendinti. Atakų tyrime parodoma šios atakų vykdymo laikas nepriklauso nuo saugumo parametro m ir įveikia kriptosistemą su parametrais $m = 10$ ir $m = 11$ per mažiau nei 1 sekundę, tačiau atakoms įgyvendinti reikalinga heuristika, dešifruoto turinio tikrinimui.

Darbe remiantis apibendrinta informacijos dekodavimo atakos rezultatais su optimaliu paieškos parametru $p = 2$ yra pateikiami praktinių tyrimu metu nustatyti saugiausi McEliece m, t saugumo parametrai bei vidutiniai vykdymo laikai su jais. Toliau remiantis šiais rezultatais ir operacijų skaičiumi reikalingo dešifruoti šifrui prognozuojami saugiausi m, t saugumo parametrai bei vidutiniai atakų vykdymo laikai naudojant asmeninį kompiuterį. Taip pat, siekiant apsisaugoti nuo žinomo dalinio teksto, pranešimo persiuntimo ir susijusių pranešimų atakų, pasiūlyta McEliece kriptosistemos modifikacija, kurios pagalba klaidos vektorius yra paslepiamas žinutėje ir šią žinutę galima dešifruoti tik radus šį vektorių.

Šaltinių sąrašas

[AEE03] Suanne Au, Christina Eubanks-Turner ir Jennifer Everson. The McEliece cryptosystem. *Unpublished manuscript*, 5, 2003.

[AS11] Mircea Andraşiu ir Emil Simion. Evaluation of cryptographic algorithms. *Journal of Information Systems & Operation Management*, 5(1):51–61, 2011.

[Bal14] Marco Baldi. The McEliece and Niederreiter cryptosystems. *QC-LDPC Code-Based Cryptography*, p.p. 65–89. Springer, 2014.

[BBC13] Marco Baldi, Marco Bianchi ir Franco Chiaraluce. Security and complexity of the McEliece cryptosystem based on quasi-cyclic low-density parity-check codes. *IET Information Security*, 7(3):212–220, 2013.

[BC07] Marco Baldi ir Franco Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, p.p. 2591–2595. IEEE, 2007.

[BCG+09] Thierry P Berger, Pierre-Louis Cayrel, Philippe Gaborit ir Ayoub Otmani. Reducing key length of the McEliece cryptosystem. *International Conference on Cryptology in Africa*, p.p. 77–97. Springer, 2009.

[Ber09] Daniel J Bernstein. Introduction to post-quantum cryptography. *Post-quantum cryptography*, p.p. 1–14. Springer, 2009.

[Ber73] Elwyn Berlekamp. Goppa codes. *IEEE Transactions on Information Theory*, 19(5):590–592, 1973.

[Ber97] Thomas A Berson. Failure of the McEliece public-key cryptosystem under message-resend and related-message attack. *Annual International Cryptology Conference*, p.p. 213–220. Springer, 1997.

[BLP+09] Daniel J Bernstein, Tanja Lange, CP Peters ir Henk CA van Tilborg. Explicit bounds for generic decoding algorithms for code-based cryptography, 2009.

[BLP08] Daniel J Bernstein, Tanja Lange ir Christiane Peters. Attacking and defending the McEliece cryptosystem. *International Workshop on Post-Quantum Cryptography*, p.p. 31–46. Springer, 2008.

[Bon+99] Dan Boneh ir k.t. Twenty years of attacks on the RSA cryptosystem. *Notices-American Mathematical Society*, 46:203–213, 1999.

[Bra14] C. Bradford. 5 common encryption algorithms and the unbreakables of the future. `https://www.storagecraft.com/blog/5-common-encryption-algorithms`, 2014. Tikrinta 2017-11-19.

[BV97] Ethan Bernstein ir Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[CC98] Anne Canteaut ir Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

[CGN+15] Pierre-Louis Cayrel, Cheikh T Gueye, Ousmane Ndiaye ir Robert Niebuhr. Critical attacks in code-based cryptography. *International Journal of Information and Coding Theory*, 3(2):158–176, 2015.

[Cit12] Citizendium. AES competition. `http://en.citizendium.org/wiki/AES_competition`, 2012. Tikrinta 2017-11-19.

[CS98] Anne Canteaut ir Nicolas Sendrier. Cryptanalysis of the original McEliece cryptosystem. *International Conference on the Theory and Application of Cryptology and Information Security*, p.p. 187–199. Springer, 1998.

[DH76a] Whitfield Diffie ir Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[DH76b] Whitfield Diffie ir Martin E Hellman. Multiuser cryptographic techniques. *Proceedings of the June 7-10, 1976, national computer conference and exposition*, p.p. 109–112. ACM, 1976.

[Dic89] Oxford English Dictionary. OED online: Oxford University press, 1989.

[DK07] Hans Delfs ir Helmut Knebl. Symmetric-key encryption. *Introduction to Cryptography*, p.p. 11–31. Springer, 2007.

[DMR11] Hang Dinh, Cristopher Moore ir Alexander Russell. McEliece and Niederreiter cryptosystems that resist quantum Fourier sampling attacks. *Annual Cryptology Conference*, p.p. 761–779. Springer, 2011.

[EOS07] Daniela Engelbert, Raphael Overbeck ir Arthur Schmidt. A summary of McEliece-type cryptosystems and their security. *Journal of Mathematical Cryptology JMC*, 1(2):151–199, 2007.

[FO99] Eiichiro Fujisaki ir Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Annual International Cryptology Conference*, p.p. 537–554. Springer, 1999.

[For70] GD Forney. Error correction for partial response modems. *1970 Int. Symp. Information Theory*, p.p. 34–35, 1970.

[Gab95] Ernst M Gabidulin. *Public-key cryptosystems based on linear codes*. Citeseer, 1995.

[Gio12] Andrea Giorgi De. *McEliece-type Cryptosystems: Costs, Security and an Attack to a Recent Variant*. Disertacija, 2012.

[Gop70] Valerii Denisovich Goppa. A new class of linear correcting codes. *Problemy Peredachi Informatsii*, 6(3):24–30, 1970.

[Gre03] John Aaron Gregg. *On factoring integers and evaluating discrete logarithms*. Disertacija, Harvard University, 2003.

[Hud13] Hans Christoph Hudde. *Development and Evaluation of a Code-based Cryptography Library for Constrained Devices*. Disertacija, Citeseer, 2013.

[Joc02] Ellen Jochemsz. Goppa codes & the McEliece cryptosystem. *Doktorarbeit, Universiteit van Amsterdam*, 2002.

[KI01] Kazukuni Kobara ir Hideki Imai. Semantically secure McEliece public-key cryptosystems-conversions for McEliece PKC. *International Workshop on Public Key Cryptography*, p.p. 19–35. Springer, 2001.

[KL14] Jonathan Katz ir Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.

[KMV+96] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot ir Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[LB88] Pil Joong Lee ir Ernest F Brickell. An observation on the security of McEliece's public-key cryptosystem. *Workshop on the Theory and Application of of Cryptographic Techniques*, p.p. 275–280. Springer, 1988.

[LDW94] Yuan Xing Li, Robert H Deng ir Xin Mei Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.

[LLC14] Seongan Lim, Hyang-Sook Lee ir Mijin Choi. An efficient decoding of Goppa codes for the McEliece cryptosystem. *Fundamenta Informaticae*, 133(4):387–397, 2014.

[Loi00] Pierre Loidreau. Strengthening McEliece cryptosystem. *International Conference on the Theory and Application of Cryptology and Information Security*, p.p. 585–598. Springer, 2000.

[LS01] Pierre Loidreau ir Nicolas Sendrier. Weak keys in the McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, 2001.

[Mce78] Robert J Mceliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. *Conference on the theory and application of cryptographic techniques*, p.p. 417–426. Springer, 1985.

[Min07] Lorenz Minder. Cryptography based on error correcting codes, 2007.

[MMT11] Alexander May, Alexander Meurer ir Enrico Thomae. Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. *International Conference on the Theory and Application of Cryptology and Information Security*, p.p. 107–124. Springer, 2011.

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Control and Inf. Theory*, 15(2):159–166, 1986.

[OS09] Raphael Overbeck ir Nicolas Sendrier. Code-based cryptography. *Post-quantum cryptography*, p.p. 95–145. Springer, 2009.

[Pet10] Christiane Peters. Information-set decoding for linear codes over F_q. *International Workshop on Post-Quantum Cryptography*, p.p. 81–94. Springer, 2010.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[PT01] Vikram Pasham ir Steve Trimberger. High-speed DES and triple DES encryptor/decryptor. *Xilinx Application Notes*, 2001.

[Res09] Certicom Research. Standards for efficient cryptography group. sec 1: elliptic curve cryptography (version 2.0). `http://www.secg.org/sec1-v2.pdf`, 2009. Tikrinta 2017-10-08.

[Sen00] Nicolas Sendrier. Finding the permutation between equivalent linear codes: the support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4):1193–1203, 2000.

[Sen99] Nicolas Sendrier. *The support splitting algorithm*. Disertacija, INRIA, 1999.

[Ske05] G. Skersys. Klaidas taisančių kodų teorija. Paskaitų konspektai. `http://www.mif.vu.lt/katedros/cs/Asmen/KodavimoTeorija.pdf`, 2005. Tikrinta 2018-04-08.

[SKW+98] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall ir Niels Ferguson. Twofish: a 128-bit block cipher. *NIST AES Proposal*, 15:23, 1998.

[SS92] Vladimir M Sidelnikov ir Sergey O Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2(4):439–444, 1992.

[Sta02] V. Stakėnas. Kodavimo teorija. Paskaitų konspektai, 2002.

[Sta07] V. Stakėnas. *Kodai ir šifrai: informacijos kodavimo ir kriptografijos pagrindai*. Vaistų žinios, 2007.

[Ste88] Jacques Stern. A method for finding codewords of small weight. *International Colloquium on Coding Theory and Applications*, p.p. 106–113. Springer, 1988.

[Sti05] Douglas R Stinson. *Cryptography: theory and practice*. CRC press, 2005.

[Sun98] Hung-Min Sun. Improving the security of the McEliece public-key cryptosystem. *International Conference on the Theory and Application of Cryptology and Information Security*, p.p. 200–213. Springer, 1998.

[TS16] Rodolfo Canto Torres ir Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. *International Workshop on Post-Quantum Cryptography*, p.p. 144–161. Springer, 2016.

[Uma16] Muhammad Umair. Comparison of symmetric block encryption algorithms, 2016.

[Wil06] Stallings William. *Cryptography and network security: principles and practices*. Pearson Education India, 2006.

[WSN17] Wen Wang, Jakub Szefer ir Ruben Niederhagen. FPGA-based key generator for the Niederreiter cryptosystem using binary Goppa codes. *International Conference on Cryptographic Hardware and Embedded Systems*, p.p. 253–274. Springer, 2017.

Priedas Nr. 1

Apibendrintos informacijos rinkinio dekodavimo atakos rezultatai

12 lentelė. Atakų su parametrų $m = 5$ gauti vidutiniai vykdymo laikai (ms) iš 1 000 000 bandymų vienam m, t, p deriniui.

$m = 5$	p					
t	1	2	3	4	5	6
1	0,139	-	-	-	-	-
2	0,174	0,120	-	-	-	-
3	0,149	0,095	0,122	-	-	-
4	0,085	0,060	0,060	0,117	-	-
5	0,036	0,028	0,026	0,043	0,053	-
6	0,010	0,009	0,009	0,021	0,013	0,014

13 lentelė. Atakų su parametrų $m = 6$ gauti vidutiniai vykdymo laikai (ms) iš 100 000 bandymų vienam m, t, p deriniui.

$m = 6$	p									
t	1	2	3	4	5	6	7	8	9	10
1	0,438	-	-	-	-	-	-	-	-	-
2	1,094	0,778	-	-	-	-	-	-	-	-
3	1,727	1,343	3,554	-	-	-	-	-	-	-
4	1,849	1,492	3,919	5,137	-	-	-	-	-	-
5	1,492	1,197	2,734	4,145	6,774	-	-	-	-	-
6	0,881	0,749	1,367	2,154	3,31	4,156	-	-	-	-
7	0,47	0,383	0,544	0,757	0,974	1,103	1,143	-	-	-
8	0,2	0,164	0,173	0,196	0,209	0,221	0,223	0,209	-	-
9	0,068	0,055	0,049	0,055	0,062	0,058	0,057	0,054	0,05	-
10	0,025	0,018	0,018	0,021	0,021	0,018	0,022	0,018	0,017	0,022

14 lentelė. Atakų su parametrų $m = 7$ gauti vidutiniai vykdymo laikai (ms) iš 10 000 bandymų vienam m, t, p deriniui.

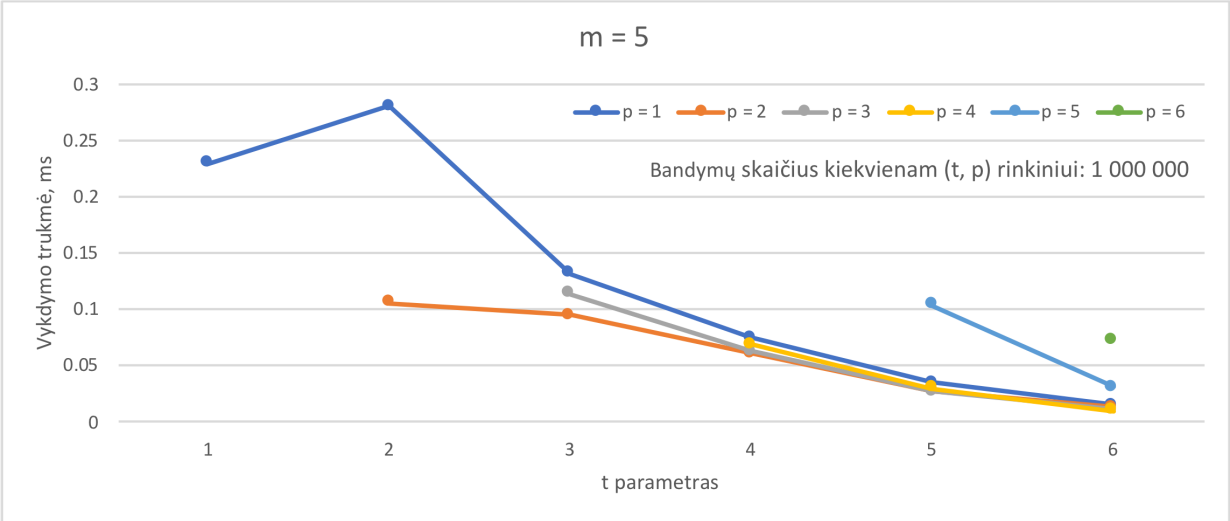
$m = 7$	p		
t	1	2	3
1	3,383	-	-
2	12,39	4,262	-
3	29,986	12,51	58,462
4	56,665	24,159	131,035
5	87,009	36,609	212,558
6	91,325	48,606	269,723
7	90,845	45,681	251,869
8	81,818	42,022	218,911
9	63,445	33,001	160,533
10	37,421	22,695	99,391
11	22,376	13,797	58,43
12	11,508	7,346	29,623
13	5,203	3,969	12,448
14	2,028	1,558	4,55
15	0,746	0,673	1,272
16	0,272	0,196	0,302
17	0,068	0,059	0,061
18	0,016	0,017	0,017

15 lentelė. Atakų su parametrų $m = 8$ gauti vidutiniai vykdymo laikai (s) iš 1000 bandymų vienam m, t, p deriniui.

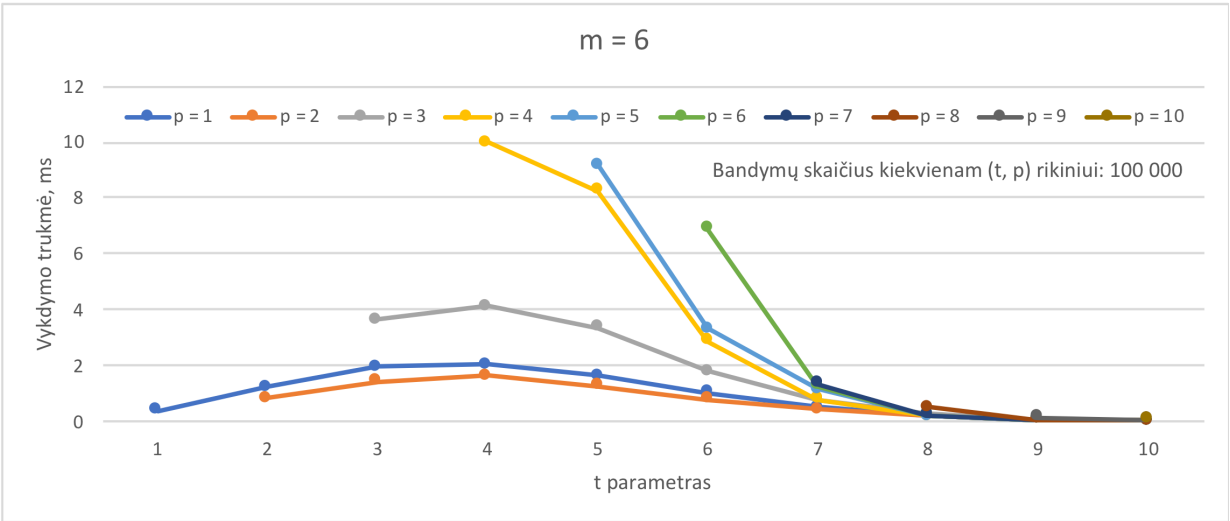
$m = 8$	p	
t	1	2
1	0,016	-
2	0,080	0,079
3	0,354	0,441
4	1,109	1,197
5	3,312	3,060
6	5,499	5,829
7	10,645	12,165
8	14,268	15,006
9	25,120	20,687
10	30,446	25,968
11	31,403	28,105
12	33,815	31,371
13	34,867	32,464
14	23,704	19,217
15	18,891	17,149
16	8,610	12,021
17	9,702	8,498
18	5,734	4,624
19	3,283	2,178
20	1,684	1,515
21	1,049	1,138
22	0,505	0,515
23	0,250	0,168
24	0,106	0,086
25	0,044	0,038
26	0,015	0,022
27	0,007	0,005
28	0,002	0,002
29	0,001	0,001
30	0,000	0,000
31	0,000	0,000

Priedas Nr. 2

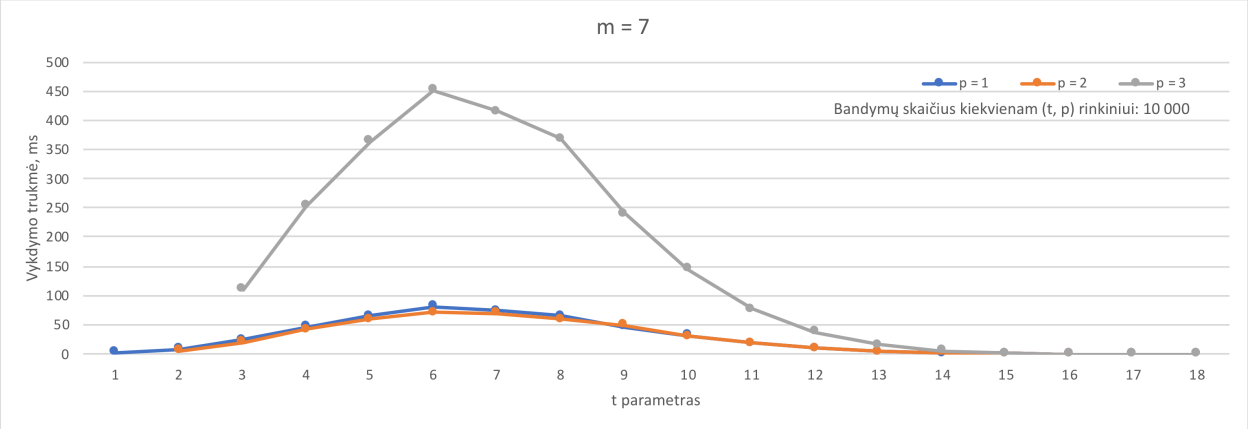
Apibendrintos informacijos rinkinio dekodavimo atakos rezultatai su 2 optimizacija



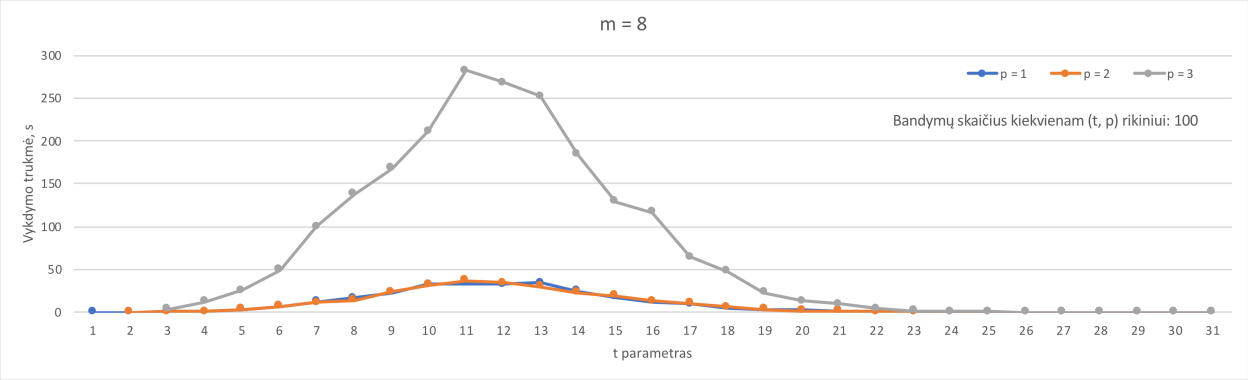
4 pav. Apibendrinto informacijos rinkinio dekodavimo atakų rezultatai milisekundėmis su $m = 5$ ir visais galimais t ir p.



5 pav. Apibendrinto informacijos rinkinio dekodavimo atakų rezultatai milisekundėmis su $m = 6$ ir visais galimais t ir p.



6 pav. Apibendrinto informacijos rinkinio dekodavimo atakų rezultatai milisekundėmis su $m = 7$, visais galimais t ir $p = 1, \ldots, 3$ parametrais.



7 pav. Apibendrinto informacijos rinkinio dekodavimo atakų rezultatai sekundėmis su $m = 8$, visais galimais t, $p = 1$ ir $p = 2$ parametrais jiems.