VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
STUDIJŲ PROGRAMA: INFORMATIKA

McEliece viešojo rakto kriptografinės sistemos saugumo tyrimas
Study of the Security of McEliece Public-key Cryptosystem

Baigiamasis magistro darbas

Atliko: Paulius Drabužinskis
VU el. p.: paulius.drabuzinskis@mif.stud.vu.lt

Vadovas: Gintaras Skersys
Recenzentas: Vilius Stakėnas

Vilnius
2022

SANTRAUKA

Šiame darbe yra atliekamas McEliece kriptografinės sistemos saugumo tyrimas. Pateikiama tyrimo teorija ir naudojant JAVA programavimo kalbą parašytos atakos nukreiptos prieš McEliece kriptosistemą, tokios kaip: apibendrintoji informacijos aibės dekodavimo ataka, mažo svorio žodžių radimo ataka, pakartotinai siųstos žinutės ataka. Šios atakos atliekamos mažiems parametrams ir iš gautųjų rezultatų nustatoma, kiek truks tos atakos su realiais parametrais, taip pat kiek jos truks naudojant superkompiuterį Fujitsu Fugaku, ir nustatomi saugiausi McEliece sistemos parametrai šioms atakoms. Darbe taip pat pasiūlyta McEliece kriptosistemos modifikacija, skirta atremti pakartotinai siųstos žinutės ataką. Šiai modifikacijai taip pat parašyta ir atlikta modifikuota susijusių pranešimų ataka.

Raktiniai žodžiai: McEliece kriptografinė sistema, kriptografinės atakos, apibendrintoji informacijos aibės dekodavimo ataka, mažo svorio žodžių radimo ataka, pakartotinai siųstos žinutės ataka, superkompiuteris Fujitsu Fugaku, McEliece kriptografinės sistemos modifikacija, modifikuota susijusių pranešimų ataka.

SUMMARY

In this work McEliece cryptographic system security has been studied. Research‘s theory has been provided and by using JAVA programming language attacks against McEliece cryptosystem have been written, such as: generalized information-set-decoding attack, low-weight codeword attack, message-resend attack. These attacks are executed for small McEliece cryptosystem parameters and from the results the time it takes to do these attacks with larger parameters are found. Also attack execution times for supercomputer Fujitsu Fugaku are also determined and the most secure parameters for McEliece cryptosystem against these attacks are found. In this work McEliece cryptosystem modification which can withstand message-resend attacks has been proposed. For this modification it was also written and executed the modified related-message attack.

Key-words: McEliece cryptographic system, cryptographic attacks, generalized information-set-decoding attack, low-weight codeword attack, message-resend attack, supercomputer Fujitsu Fugaku, McEliece cryptographic system modification, modified related-message attack.

TURINYS

ĮVADAS

Šiuolaikiniame pasaulyje technologijos pasistūmėjo į priekį taip, kad dauguma žmonių teikia pirmenybę naudoti internetą, kaip pagrindinę terpę siųsti duomenims iš vienos pasaulio vietos į kitą [CPB+17]. Pvz. elektroniniais laiškais ar per susirašinėjimus. Duomenų siuntimas taip atliekamas labai paprastai ir lengvai naudojant internetą. Čia atsiranda viena iš pagrindinių problemų tokiam duomenų perdavimui – saugumas. Siunčiami duomenys gali būti jautrūs, slapti ar privatūs, t.y. jų perskaityti neturėtų galėti bet kas. Kyla grėsmė, kad tokie duomenys gali būti perimti ar „nulaužti“ (angl. hack) programišių ar kitų priešiškų individų. Tam, kad apsaugoti šiuos duomenis yra naudojama kriptografija.

Kriptografija yra būdas apsaugoti svarbius duomenis nuo neleistinos prieigos [Goy12]. Ji atsirado kaip saugus būdas perduoti informaciją. Ji užtikrina duomenų konfidencialumą, vientisumą, elektroninius parašus, bei aukšto lygio vartotojų identifikavimą. Kriptografijos metodai naudoja matematiką tam, kad būtų apsaugoti duomenys. Duomenys paprastai yra matematiškai užšifruojami, o vėliau dešifruojami. Tam, kad tai būtų atlikta yra reikalingos kriptografinės sistemos.

Viena iš labiausiai naudojamų kriptografinių sistemų yra RSA viešojo rakto kriptosistema [Kir15]. Ši kriptosistema yra saugi, nes ją yra per daug sudėtinga įveikti naudojant dabartines kompiuterines sistemas. Vis dėlto kvantiniai kompiuteriai, kelia problemų šiai kriptosistemai, nes jų skaičiavimo galia ateityje gali tapti pakankama įveikti šią RSA kriptosistemą. Stipriai išaugus kvantinių kompiuterių skaičiavimo galiai, arba atradus greitesnius algoritmus sistemos įveikimui kyla pavojus, kad ši naudojama kriptosistema taps nebesaugi ir ją gali tekti pakeisti.

Viena iš daugiausiai žadančių viešojo rakto kriptosistemų, kuri sugeba atremti kvantinių kompiuterių atakas ir galėtų pakeisti RSA yra McEliece kriptografinė sistema [BBC+14]. Ši viešojo rakto kriptosistema remiasi tiesinio kodo dekodavimo sudėtingumu. Kadangi visoms kriptosistemoms svarbiausias dalykas yra jų saugumas, tai taip pat yra svarbu, kuo geriau ištirti ir McEliece kriptosistemos saugumą. Šiame magistriniame darbe bus siekiama tai padaryti panaudojant įvairias atakas nukreiptas prieš McEliece viešojo rakto kriptografinę sistemą. Kad atakos neužtruktų, jos bus atliktos su mažais McEliece kriptosistemos parametrais. Iš gautųjų rezultatų tada yra galima prognozuoti, kiek šios atakos užimtų laiko, kad sugebėtų įveikti

kriptosistemą su realiais parametrais. Tokiu būdu galima įvertinti McEliece kriptografinės sistemos saugumą.

Informatikos magistro studijų programos mokslo baigiamajam darbui atlikti yra keliami šie uždaviniai:

- aprašyti su tyrimo tikslu susijusią teoriją, atlikti tyrimo literatūros apžvalgą;
- surasti programavimo bibliotekas, kurios generuotų McEliece kriptosistemos raktus ir užšifruotų žinutes pasinaudodamos jais;
- parašyti programą - algoritmą, kuris praktiškai realizuotų apibendrintąją informacijos aibės dekodavimo ataką;
- pasinaudojant parašytuoju atakos algoritmu atlikti apibendrintąją informacijos aibės dekodavimo ataką mažiems McEliece kriptosistemos parametrams;
- parašyti kitą algoritmą, kuris praktiškai realizuotų mažo kodo žodžių radimo ataką;
- taip pat, pasinaudojant parašytuoju mažo svorio kodo žodžių radimo atakos algoritmu atlikti atakas mažiems McEliece kriptosistemos parametrams;
- galiausiai parašyti trečiąjį algoritmą, kuris praktiškai realizuotų pakartotinai siųstos žinutės ataką;
- naudojantis parašytuoju pakartotinai siųstos žinutės atakos algoritmu, atlikti atakas įvairiems McEliece kriptosistemos parametrams;
- iš atliktų eksperimentų rezultatų, nustatyti, kiek laiko užtruktų pasirinktos atakos, naudojant realius McEliece kriptografinės sistemos parametrus;
- nustatyti McEliece kriptografinės sistemos parametrus, su kuriais sistema yra atspari nagrinėtoms atakoms;
- pateikti rekomendacijas kaip išvengti silpnųjų McEliece kriptosistemos vietų, bei kitas išvadas;
- nustatyti, kiek laiko užtruktų nagrinėtos atakos naudojant šiuo metu galingiausią pasaulyje superkompiuterį Fugaku;
- palyginti tarpusavyje parašytųjų apibendrintosios informacijos aibės dekodavimo, mažo svorio kodo žodžių radimo ir pakartotinai siųstos žinutės atakų implementacijų vykdymo laikus.

Pirmuosiuose keturiuose šio darbo skyriuose aprašyta tyrimo teorija ir literatūros apžvalga (kriptografijos temų apžvalga, klaidas taisantys kodai, McEliece kriptografinės sistemos sandara,

atakos prieš McEliece kriptosistemą). Tuo tarpu 5-iajame skyriuje yra aptariamas atliktas tyrimas ir pateikiami atlikto darbo rezultatai.

1. KRIPTOGRAFIJOS TEMŲ APŽVALGA

1.1. KRIPTOGRAFIJA IR JOS UŽDAVINIAI

Kriptografija yra apibrėžiama, kaip mokslo šaka, kuri studijuoja skirtingus metodus slaptai siųsti žinutes (paprastai užšifruota ar užslėpta forma), taip kad tik numatytas gavėjas galėtų panaikinti maskuotę ir perskaityti siųstą žinutę [Mol06]. Originali žinutė yra vadinama pradiniu tekstu (angl. *plaintext*), o užmaskuota žinutė vadinama šifru. Procesas, kurio metu pradinis tekstas paverčiamas šifru yra vadinamas šifravimu. Dešifravimas yra priešingas procesas šifravimui. Jo metu gavėjas, kuris turi informaciją, kaip panaikinti maskuotę iš šifro gauna pradinį tekstą. "Pagrindinis šiuolaikinės kriptografijos reikalavimas, kad šifravimo ir dešifravimo operacijų rezultatai priklausytų nuo tam tikrų dydžių, paprastai vadinamų raktais" [Sta06]. Taigi, jeigu individui, norinčiam perimti žinutę, taptų žinomos visos ryšio slaptumą saugančios detalės (išskyrus raktus), jam perimti originalią žinutę nepasidarytų lengviau.

Tuo tarpu, matematinių metodų skirtų įveikti kriptografiją mokslas vadinamas kriptoanalize. T.y. ji yra bandymas įveikti kriptografinę duomenų apsaugą. Taigi, kriptoanalizė – duomenų kriptografinių apsaugos algoritmų vertinimo mokslas.

Sugretinę kriptografiją su kriptoanalize gauname kriptologiją – mokslą apie duomenų apsaugą naudojant matematikos ir informatikos priemones.

Pagrindinis, fundamentalus ir klasikinis kriptografijos uždavinys yra užtikrinti duomenų konfidencialumą (angl. *confidentiality*) naudojant šifravimo metodus [DK07]. Konfidencialumas apibrėžia taisykles, kurios riboja prieigą prie slaptos informacijos arba prideda papildomus ribojimus pačiai informacijai [BEH18]. T.y. konfidencialumas užtikrina, kad saugomi duomenys niekada nebus pasiekiami neturintiems leidimo žmonėms ar programoms. Taip kriptografija padeda informacijai išlikti slaptai.

Taip pat, nemažiau svarbūs kriptografijai yra kiti trys uždaviniai:

- duomenų vientisumas (angl. *data integrity*);
- autentiškumas (angl. *authentication*);
- duomenų nepaneigiamumas (angl. *non-repudiation*).

Duomenų vientisumo uždavinyje žinutės gavėjas turėtų galėti patikrinti ar atsiųsta žinutė yra iškraipyta, tiek dėl atsitiktinių veiksnių, tiek dėl tyčinių. Taip pat niekas neturėtų galėti pakeisti netikros žinutės ar dalies jos originalia.

Autentiškumo atveju žinutės gavėjas turėtų galėti patvirtinti žinutės siuntėją. Jeigu žinutės yra siunčiamos siuntėjos Alisos, gavėjui Bobui, tai niekas neturi galėti apsimesdamas Alisa nusiųsti žinutės Bobui. Prieš prasidedant Alisos ir Bobo tarpusavio komunikacijai, tiek Alisa, tiek Bobas turėtų galėti vienas kitą identifikuoti.

Duomenų nepaneigiamumas padeda užtikrinti, kad žinutės siuntėjas ateityje niekada negalės paneigti, kad jis siuntė tą žinutę.

1.2. KRIPTOGRAFINĖ SISTEMA IR JOS RŪŠYS

Duomenų apsaugos sistemą, naudojančią šifravimą, vadinsime kriptografine sistema arba tiesiog kriptosistema [Sta06]. Toliau pateikiamas labiau matematinis šios sąvokos apibrėžimas.

Kriptografine sistema vadinsime trejetą $\langle \boldsymbol{M}, \boldsymbol{K}, \boldsymbol{C} \rangle$, čia $\boldsymbol{M}$ – nešifruotų tekstų, $\boldsymbol{K}$ – naudojamų raktų ir $\boldsymbol{C}$ – šifrų aibės, kartu su šifravimo ir dešifravimo operacijomis

$$e(\cdot|K_e) : \boldsymbol{M} \rightarrow \boldsymbol{C}, \qquad \boldsymbol{d}(\cdot|\boldsymbol{K_d}) : \boldsymbol{C} \rightarrow \boldsymbol{M}, \qquad \langle K_e, K_d \rangle \in \boldsymbol{K},$$

tenkinančioms sąlygą

$$\forall m \in \boldsymbol{M} \Rightarrow d(e(m|K_e)|K_d) = m. \tag{1}$$

Iš čia matyti, kad naudojamų raktų aibė susideda iš šifravimui skirto rakto K_e ir dešifravimui skirto rakto K_d.

Kriptosistemos, kai abu raktai sutampa, t.y. $K_e = K_d$, vadinamos simetrinėmis kriptosistemomis. Kai šie raktai nesutampa t.y. $K_e \neq K_d$, tai tokios kriptosistemos vadinamos asimetrinėmis kriptosistemomis.

Simetrinių kriptosistemų atvejų, jeigu gavėjas norėtų, kad jam siuntėjas atsiųstų užšifruotą žinutę, gavėjas turėtų perduoti siuntėjui slaptą raktą, kurio neturėtų sužinoti kiti individai ir tada galėtų laukti žinutės. Todėl simetrinėse kriptosistemose komunikacijos pradžiai reikia bent trumpalaikio saugaus kanalo, kuriame būtų galimybė perduoti slaptą raktą.

Iki 1976 metų visos kriptosistemos buvo simetrinės. Tais metais du matematikai W. Diffie ir M.E. Hellman paskelbė savo žymųjį straipsnį „Naujos kryptys kriptografijoje“ (angl. *New directions in cryptography*), kuriame aprašė naujos kartos kriptosistemų principus [DH76]. Šiame straipsnyje jie pristatė revoliucinę sąvoką viešojo rakto kriptografiją. Jie kartu išsprendė ilgai egzistavusią problemą dėl saugaus raktų apsikeitimo.

Viešojo rakto kriptosistemos yra asimetrinės. Jų esmė yra tokia: nors dešifravimui skirtas raktas K_d yra susijęs su šifravimui skirtu raktu K_e, tačiau praktiškai neturi būti įmanoma per „tikrovišką” laiką nustatyti K_d. Tokiu principu bandoma užtikrinti viešojo rakto kriptosistemų saugumą. Paprastai šifravimui skirtas raktas K_e gali būti perduodamas ir nesaugiu kanalu. Toks raktas vadinamas viešuoju raktu. Tuo tarpu dešifravimui skirtas raktas K_d išlieka slaptu, todėl vadinamas privačiuoju raktu.

Taigi viešojo rakto kriptosistemose kiekvienas žinutės gavėjas turi savo raktą k, kuris susideda iš privataus rakto K_d ir viešojo rakto K_e, t.y. $k = (K_e, K_d)$ [DK07]. Viešasis raktas paskelbiamas viešai, o privatus lieka žinomas tik pačiam gavėjui. Jeigu Alisa nori siųsti žinutę m Bobui, tada ji naudodama Bobo viešąjį raktą K_e, kuris žinomas visiems, užšifruoja žinutę m. Galiausiai Bobas gavęs užšifruotą žinutę ją dešifruoja naudodamas savo privatųjį raktą K_d.

1.3. KRIPTOSISTEMŲ SAUGUMAS IR ATAKOS PRIEŠ JAS

Vykdyti kriptografinės sistemos kriptoanalizę reiškia atlikti jos atakas. Todėl kriptoanalizė dažnai apibrėžiama, kaip mokslas, kuris studijuoja atakas nukreiptas prieš kriptografines sistemas [DK07]. Sėkmingos atakos pvz. gali atkurti pagrindinį tekstą (ar dalį jo), pakeisti dalį originalios žinutės arba suklastoti elektroninius parašus.

Fundamentali prielaida apie kriptoanalizę vadinama Kerchhofo principu. Ji teigia, kad individas norintis pakenkti kriptografinei sistemai žino viską apie pačią kriptografinę sistemą, visus algoritmus ir jų implementacijas. Pagal šį principą, kriptosistemos saugumas privalo būti priklausomas tik nuo slaptųjų raktų.

Atakos prieš šifravimo schemų suteikiamą slaptumą paprastai stengiasi atkurti pagrindinį tekstą iš šifro, arba kartais net drastiškiau, atkurti slaptąjį raktą. Ataką vykdantis asmuo, pavadinkime jį Zigmu, nežino slaptojo rakto. Atlikdamas atakas jis paprastai naudojasi kitomis žiniomis, tokiomis kaip, žiniomis apie šifravimo ir dešifravimo operacijų struktūrą, kriptosistemos darbo duomenimis: šifrais ir juos atitinkančiais tekstais, viskuo išskyrus slaptuosius raktus.

Toliau galima panagrinėti galimų atakų tipus:

1. Pavienių šifrų ataka (angl. *ciphertext-only attack*). Zigmas sugeba gauti šifrus. Tai gana tikėtina daugumoje šifravimo situacijų. Net jeigu Zigmas negali atlikti sudėtingesnių atakų aprašytų apačioje, reikia manyti, kad jis gali gauti preigą prie užšifruotų žinučių. Šifravimo metodas, kuris negali atsilaikyti prieš šią ataką yra visiškai nesaugus.
2. Teksto-šifro porų ataka (angl. *known-plaintext attack*). Zigmas sugeba gauti pradinius tekstų-šifrų poras. Naudodamasis informaciją iš šių porų, jis bando dešifruoti šifrą, kuriam jis turi atitinkamą pradinį tekstą. Nors ir gali atrodyti, kad tokia informacija gali nebūti paprastai pasiekiama atakuotojui, tačiau iš tikrųjų ji yra labai dažnai pasiekiama. Žinutės gali būti siunčiamos standartiniais formatais, kuriuos Zigmas atpažįsta.
3. Pasirinktų teksto-šifro porų ataka (angl. *chosen plaintext attack*). Zigmas gali gauti šifrus iš savo pasirinktų pradinių tekstų. Tada jis bando dešifruoti šifrą, kuriam jis neturi pradinio teksto. Nors vėl atrodo, kad tokia situacija turėtų būti reta, tačiau yra daugybė atvejų kada Zigmas tai galėtų padaryti. Pavyzdžiui, jis išsiunčia kažkokią įdomią informaciją savo galimai aukai, kuri jo įsitikinimu tikrai užšifruos ir išsiųs. Šio tipo ataka daro prielaidą, kad Zigmas pirma privalo gauti bet kokią pasirinktą pradinio teksto-šifro porą ir tada atlikti savo analizę be jokios kitos sąveikos su auka. Tai reiškia, kad jam reikia pasiekti šifravimo prietaisą vieną kartą.
4. Adaptyvi pasirinktų teksto-šifro porų ataka (angl. *adaptively-chosen-plaintext attack*). Ji yra tokia pati kaip praėjusi ataka, tik kad dabar Zigmas gali atlikti šiek tiek analizės pradinio teksto-šifro poroms ir iš to gauti daugiau porų. Jis gali keisti savo veiklą tarp naujų porų gavimo ir analizės atlikimo kiek tik nori. Tai reiškia, kad jis turi arba ilgą prieigą prie šifravimo įrenginio arba gali kažkaip pakartotinai juo naudotis.
5. Pasirinktų ir adaptyvi pasirinktų šifrų ataka (angl. *chosen- and adaptively-chosen-cipertext attack*). Šios dvi atakos yra panašios į viršuje aprašytas pradinio teksto-šifro porų atakas. Zigmas gali pasirinkti šifrus ir gauti atitinkamus pradinius tekstus. Jis gali gauti prieigą prie dešifravimo įrenginio.

Yra keli požiūriai, kurie gali apibrėžti, ką galima laikyti saugia kriptosistema [Sta06].

"Kriptosistema yra vadinama besąlygiškai saugia (angl. *unconditional security*), jei net ir turėdamas beribius skaičiavimo išteklius kriptoanalitikas negali be rakto iš šifro nustatyti, koks pranešimas buvo siųstas. Tai griežčiausias saugios kriptosistemos apibrėžimas.

Kriptosistema vadinama saugia sudėtingumo teorijos požiūriu (angl. *complexity-theoretic security*), jei jos negali įveikti Zigmas, kurio skaičiavimo resursai leidžia jam taikyti tik polinominio laiko algoritmus (t.y. kai naudojamas laikas ir atmintis polinomiškais priklauso nuo įvedamų duomenų dydžio).

Sakoma, kad kriptosistemos saugumas yra įrodomas (angl. *provable security*), jeigu galima įrodyti, kad sistemos įveikimas yra tolygus matematinio (dažniausiai skaičių teorijos) uždavinio, kuris laikomas sunkiu, sprendimui.

Kriptosistema vadinama skaičiavimų požiūriu saugia (angl. *computational security*), jeigu pasiektas skaičiavimų resursų lygis yra pernelyg žemas, kad naudojant geriausias žinomas atakas, sistema būtų įveikta.

Pagaliau ad hoc saugia, arba euristiškai saugia, kriptosistema vadinama tokia sistema, kurios saugumą patvirtina tam tikri dažnai euristiniai argumentai. Suprantama, šis terminas tereiškia, kad specialistai atliko tam tikrą sistemos analizę, tačiau įveikti kriptosistemos nepavyko.”

2. KLAIDAS TAISANTYS KODAI

2.1. ĮVADAS Į KLAIDAS TAISANČIUS KODUS

Klaidas taisančių kodų teorija atsirado kaip sprendimas praktinėms problemoms, kurios atsiranda patikimame ryšyje, kai ryšiui yra naudojama skaitmeniškai užkoduota informacija [Ple98]. Joje nagrinėjama situacija, kad pranešimas *m* būna perduodamas nepatikimu ryšio kanalu, t.y. kanale jį gali iškraipyti triukšmas. Iš kanalo išeinantis pranešimas *m'* jau nebebūtinai sutaps su pradiniu pranešimu *m*. Tam ir reikalingi klaidas taisantys kodai.

Klaidas taisantys kodai naudoja metodus, padedančius aptikti ir ištaisyti kanale padarytas klaidas. Tam informacija prieš siunčiant į kanalą yra koduojama, o išėjusi iš kanalo dekoduojama [Ske20]. Paprastai kodavimo metu prie pradinio pranešimo *m* prijungiama papildoma informacija, kuri leidžia aptikti ir ištaisyti tam tikrą skaičių kanale padarytų klaidų. Gaunamas užkoduotas pranešimas *c*, kuris yra didesnės apimties negu *m*. c yra vadinamas kodo žodžiu (angl. *codeword*). Šis užkoduotas pranešimas *c* yra siunčiamas ryšio kanalu ir galbūt iškraipomas. Iš kanalo išeina *y,* kuris gali skirtis nuo *c*. Dekodavimo metu pranešime *y*, naudojantis kodavimo metu pridėta informacija, yra ištaisomos klaidos, bei gaunamas pranešimas *m',* kuris lygus pradiniam pranešimui *m*, jeigu pranešime *y* visos klaidos buvo ištaisytos.

2.2. KLAIDAS TAISANČIŲ KODŲ TAIKYMAS KRIPTOGRAFIJOJE

Klaidas taisantys kodai padeda rasti ir ištaisyti klaidas, kurios atsiranda ryšyje dėl kanalo triukšmo. Todėl kodai apsaugo siunčiamų žinučių vientisumą. Kriptografija, tuo tarpu, yra naudojama paslėpti žinutės informaciją nuo neleistinų individų ir suteikti žinutės siuntinėjui ir žinutės turiniui autentiškumą [IH08].

Nors klaidas taisančių kodų ir kriptografijos tikslai ir yra skirtingi, tačiau jie turi ir bendrų savybių. Abi šios teorijos pagrįstos kodavimu, t.y. transformuoja informaciją iš vieno formato į kitą. Taip pat jos yra pagrįstos panašiais matematiniais metodais. Galiausiai jos duoda pagrindą nepakeičiamoms technologijoms, kurios padaro žmonių gyvenimą patogesnį.

Šie panašumai ir skirtumai tarp klaidas taisančių kodų ir kriptografijos leidžia šias teorijas taikyti kartu. Yra daugybė pavyzdžių, kur kriptografija ir klaidas taisantys kodai eina išvien, pvz.

slaptas dalinimasis (angl. *secret sharing*), autentifikacijos kodai (angl. *authentication codes*), McEliece kriptosistemos, biometriškai pagrįsta asmens identifikacija (angl. *biometrics-based personal identification*), kriptoanalizė, kvantinė kriptografija ir kt. Šiame darbe plačiau bus aptariama tik McEliece kriptografinė sistema.

2.3. MCELIECE KRIPTOGRAFINĖS SISTEMOS VEIKIMAS

1978 m. R. J. McEliece remdamasis tiesinių Goppa kodų egzistavimu pristatė naują viešojo rakto kriptosistemą, kuri vadinama McEliece kriptografine sistema [Mce78]. Ši kriptosistema, kaip ir kitos viešojo rakto kriptosistemos turi savo viešąjį, bei privatųjį raktus, kartu šifravimo ir dešifravimo algoritmus. Binariniai Goppa kodai čia naudojami, nes šie kodai turi efektyvius dekodavimo algoritmus [Sti95]. Toliau yra aprašoma, kaip sudaroma ši kriptosistema.

Turime [*n*, *k*, *d*] tiesinį dvejetainį Goppa kodą C. Čia Goppa kodo parametrai: *n* – Goppa kodo ilgis, *k* – Goppa kodo dimensija, *d* – Goppa kodo minimalus atstumas. Tam, kad sudaryti kriptosistemą reikia parinkti *n*, *k*, *d* parametrus. Goppa kodams, galintiems ištaisyti *t* klaidų šie parametrai apskaičiuojami taip:

$$n = 2^m; \tag{2.3.1}$$

$$d = 2t + 1; \tag{2.3.2}$$

$$k = n - mt. \tag{2.3.3}$$

Iš aukščiau esančių formulių matyti, kad Goppa kodų parametrai priklauso nuo *m* ir *t* parametrų, todėl:

1. Parenkami *m* ir *t* parametrai ir pagal juos apskaičiuojami *n*, *k*, *d* parametrai remiantis 2.3.1-3 formulėmis.
2. Sudaroma $k \times n$ kodą generuojanti matrica G, kuri gali būti standartinio pavidalo - eilutės kanoninės (angl. *row canonical*) formos.
3. Parenkama atsitiktinė $k \times k$ matrica S, kurios determinantas nelygus nuliui.
4. Parenkama atsitiktinė $n \times n$ perstatų matrica P, kuri gaunama iš vienetinės matricos atlikus joje perstatas.
5. Sudaromas viešasis raktas $K_v = \langle \mathrm{G}' \rangle$, kur G' yra viešoji generuojanti matrica gaunama $G' = SGP$.
6. Privačiu raktu K_p paskelbiamos visos trys S, G ir P matricos $K_p = \langle S, G, P \rangle$.

Toliau, kad vykdyti šifravimą, duomenys, kuriuos norima užšifruoti, yra padalinami į k – bitų blokus. Toliau vykdomas vieno tokio bloko x šifravimas:

1. Parenkamas atsitiktinis $\leq t$ svorio (kai kuriuose McEliece kriptosistemos variantuose svoris yra tiksliai t) ir n ilgio klaidos vektorius e.
2. Naudojant sugeneruotą G' matricą ir klaidos vektorių e, užšifruojamas k – bitų ilgio blokas x naudojant formulę:

$$c = xG' + e,$$

kur c – žinutės x šifras.

Šifro c dešifravimo algoritmas:

1. Apskaičiuojamas $c' = cP^{-1}$, kur P^{-1} yra atvirkštinė matrica perstatų matricai P.
2. Naudojant Goppa kodų greitąjį dekodavimo algoritmą yra dekoduojamas c' ištaisant kode padarytas klaidas ir gaunant u.
3. Galiausiai apskaičiuojamas pradinis dvejetainis k – bitų pranešimas $x = uS^{-1}$.

3. MCELIECE KRIPTOGRAFINĖS SISTEMOS SANDARA

McEliece kriptosistema susideda iš kelių sudedamųjų dalių: Goppa kodų, S ir P matricų, bei klaidos vektoriaus e. Yra dvi pagrindinės klasikinės McEliece kriptosistemos saugumo prielaidos [RZ14]:

- Yra sudėtinga atsitiktinio tiesinio kodo žodžių aibėje surasti žodį, mažiausiai nutolusį nuo duoto vektoriaus (bendroji dekodavimo problema (angl. general decoding problem)). Tai NP-hard tipo problema [BMT78].
- Nėra jokio algoritmo, kuris galėtų dekoduoti kodus, kurie sugeneruoti matricos G' (be žinių apie slaptuosius parametrus) efektyviau negu atsitiktiniame tiesiniame kode.

Toliau yra aprašomos McEliece kriptosistemos dalys ir jų įtaka jos saugumui.

3.1. GOPPA KODAI

Visi Goppa kodai čia aprašomi yra tik dvejetainiai ir neredukuojami. Būtent tokie jie yra patogiausi naudojimui kompiuterinėse sistemose. Čia yra keletas priežasčių, kodėl jie yra svarbūs kriptografijai [EOS07]:

- Apatinė riba minimaliam atstumui yra lengvai apskaičiuojama.
- Žinios apie generuojantį polinomą leidžia efektyviai taisyti klaidas.
- Be žinių apie generuojantį polinomą, nėra žinomi jokie efektyvūs algoritmai galintys taisyti klaidas.

Goppa kodai, kurie priklauso klaidas taisančių kodų šeimai, buvo apibrėžti V. D. Goppa 1970 metais. Toliau pateikiami apibrėžimai [Gop70].

Tegul m ir t yra teigiami sveikieji skaičiai, $\mathbb{F}_{2^m}$ kūnas, $\mathbb{F}_{2^m}^n$ tiesinė erdvė virš to kūno, o $\mathbb{F}_{2^m}[X]$ daugianarių aibė virš $\mathbb{F}_{2^m}$. Tada t laipsnio Goppa polinomu $g(X)$ vadinama:

$$g(X) = \sum_{i=0}^{t} g_i X^i \in \mathbb{F}_{2^m}[X].$$

Tada

$$\boldsymbol{L} = (\gamma_0, \dots, \gamma_{n-1}) \in \mathbb{F}_{2^m}^n$$

yra tokia baigtinė surikiuota elementų seka, kad:

$$g(\gamma_i) \neq 0, \quad \text{visiems } 0 \leq i < n.$$

Bet kokiam vektoriui $c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_2^n$, apibrėžiamas *c* sindromas (angl. *syndrome of c*):

$$S_c(X) = -\sum_{i=0}^{n-1} \frac{c_i}{g(\gamma_i)} \frac{g(X)-g(\gamma_i)}{X-\gamma_i} \ \ mod\ g(X).$$

Dvejetainis Goppa kodas *G(**L**, g(X))* virš kūno $\mathbb{F}_2$ yra aibė visų vektorių $c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_2^n$ taip, kad lygybė

$$S_c(X) = 0$$

galioja polinominiame žiede $\mathbb{F}_{2^m}[X]$ arba ekvivalenčiai, jeigu galioja lygybė:

$$S_c(X) \equiv \sum_{i=0}^{n-1} \frac{c_i}{X-\gamma_i} \equiv 0 \ \ mod\ \ g(X).$$

Jeigu g(X) yra neredukuojamas virš $\mathbb{F}_{2^m}$, tada *G(**L**, g(X))* vadinamas neredukuojamu dvejetainiu Goppa kodu. Pastarasis ir naudojamas McEliece kriptografinėje sistemoje.

Šio kodo kontrolinę matricą H galima apskaičiuoti:

$$H = XYZ,$$

kur X, Y ir Z yra matricos:

$$X = \begin{pmatrix} g_t & 0 & 0 & \cdots & 0 \\ g_{t-1} & g_t & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & g_3 & \cdots & g_t \end{pmatrix},$$

$$Y = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \gamma_0 & \gamma_1 & \cdots & \gamma_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_0^{t-1} & \gamma_1^{t-1} & \cdots & \gamma_{n-1}^{t-1} \end{pmatrix},$$

$$Z = \begin{pmatrix} \frac{1}{g(\gamma_0)} & 0 & \cdots & 0 \\ 0 & \frac{1}{g(\gamma_1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{g(\gamma_{n-1})} \end{pmatrix}.$$

Generuojančiąją matricą G galima apskaičiuoti pasinaudojant jos ir kontrolinės matricos H sąryšiu:

$$GH^T = 0.$$

Pats R. J. McEliece pristatydamas šią kriptosistemą būtent ir rėmėsi, kad kiekvienam neredukuojamam t laipsnio polinomui virš kūno $\mathbb{F}_{2^m}$ egzistuoja dvejetainis neredukuojamas Goppa kodas, kurio ilgis $n = 2^m$, dimensija $k \geq n - mt$ ir gebantis ištaisyti t arba mažiau klaidų [Mce78]. Papildomai yra parodyta, kad tokiems kodams dekoduoti egzistuoja greitas dekodavimo algoritmas – Patterson'o algoritmas [Mce77].

McEliece kriptosistemos parametrai [*n, k, d*] yra apibrėžiami pagal Goppa kodų parametrus *m* ir *t*. R.J. McEliece pasiūlė naudoti $m = 10, t = 50$. Tada remiantis 2.3.1-3 formulėmis gaunami McEliece kriptosistemos parametrai [1024, 524, 101].

1986 metais Niederreiter pasiūlė kitokią schema, kuri vietoje Goppa kodų naudotų bendruosius Reed-Solomon (angl. *Generalized Reed-Solomon*) kodus. Deja, 1992 metais buvo parodyta, kad tokių kodų naudojimas kriptosistemoje yra nesaugus [SS92].

Kaip ir Niederreiter atveju yra buvę ir kitų pasiūlymų kriptosistemoje pakeisti Goppa kodus kitais kodais, tačiau dauguma jų pasirodė esantys nesaugūs arba neefektyvus lyginant su McEliece pasiūlytu variantu. Tuo tarpu, McEliece kriptosistema vis dar išlieka neįveikta naudojant tinkamus sistemos parametrus.

3.2. S IR P MATRICOS

Matricos *S* ir *P* yra labai svarbios McEliece kriptosistemos saugumui. Jos sumaišo ir paslepia Goppa kodą generuojančios matricos *G* struktūrą paversdamos ją viešuoju raktu *G'*. Nors R. Heiman'as 1987 metais ir teigė, kad atsitiktinė matrica *S* McEliece kriptosistemoje neatlieka visiškai jokios saugumo funkcijos, nes nepakeičia kodų, tačiau A. Canteaut parodė, kad matrica *S* gali būti svarbi paslepiant sistematinę Goppa kodo struktūrą ir dėl to turi svarbią rolę kriptosistemos saugume [Hei87] [Can96] [Lou01].

Žinant tiek *S*, tiek *P* kriptoanalitikui nesudėtinga gauti Goppa kodą generuojančią matricą *G*, o iš to radus Goppa polinomą *g(X)* jis sėkmingai gali pats taikyti dekodavimo algoritmą ir

dešifruoti pranešimą. Dėl šios priežasties viena iš galimų atakų galėtų būti S ir P matricų spėjimas. Bet pats R. J. McEliece savo originaliame straipsnyje pasiūlęs šią kriptosistemą savo parametrams paminėjo, kad S ir P matricų pasirinkimų skaičius yra astronomiškai didelis, todėl tokia ataka tampa beveik neįmanoma. Galima įvertinti kiek gi yra skirtingų variantų S ir P matricų spėjimams [Joc02].

Matrica S yra $k \times k$ dydžio ir jai galima rasti atvirkštinę matricą (t.y. jos determinantas negali būti lygus nuliui). Dėl pastarosios priežasties visos jos eilutės turi būti tiesiškai nepriklausomos viena nuo kitos. Iš čia aišku, kad pirmoji eilutė r_1 gali turėti $2^k - 1$ skirtingų variantų (visi įmanomi variantai išskyrus nulinė eilutė). Antroji eilutė negali turėti pirmosios eilutės tiesinių kombinacijų $0 \cdot \mathrm{r}_1$ (ta pati nulinė eilutė) ir $1 \cdot r_1$ (pirmoji eilutė), todėl viso galimų pasirinkimų antrajai eilutei yra $2^k - 2$. Trečioji eilutė nebegali turėti 0, r_1, r_2, $r_1 + r_2$ variantų, todėl jai lieka $2^k - 4$ variantai. Taigi viso variantų skaičius S matricai yra:

$$\prod_{i=0}^{k-1}\left(2^k - 2^i\right).$$

McEliece pasiūlytiems parametrams, kai $k = 524$, tikimybė atsitiktinai atspėti S matricą:

$$\frac{1}{\prod_{i=0}^{523}(2^{524} - 2^i)} = 0.846 \cdot 10^{-82655},$$

taigi ši tikimybė yra nykstamai maža.

Permutacijų matricai P rasti galimų variantų skaičių paprasčiau. Kadangi jos dydis yra $n \times n$ ir ji yra gaunama iš vienetinės matricos, tai skirtingų permutacijos matricų yra $n!$. Su McEliece pasiūlytais parametrais, kai $n = 1024$, tai skirtingų permutacijos matricų skaičius yra $1024!$, todėl tikimybė atspėti šią matricą taip pat yra nykstamai maža ir lygi $0.185 \cdot 10^{-2639}$.

3.3. KLAIDOS VEKTORIUS

Klaidos vektorius e yra esminė dalis McEliece kriptosistemos saugume. Jeigu klaidos vektoriaus nebūtų tada pranešimas x būtų užšifruojamas $c = xG'$. Kadangi G' yra viešas ir žinomas, tai rasti pranešimą x būtų labai paprasta, užtektų panaudoti Gauso metodą tiesinių lygčių sistemai spręsti.

Dažnai atakos, kurios nukreiptos prieš McEliece kriptografinę sistemą naudojasi šiuo faktu. Dėl to atakos stengiasi arba rasti klaidos vektorių e, arba rasti šifro koordinates, kurių klaidos

vektorius nepakeitė, taip palengvinant sistemos įveikimo procesą [EOS07]. Todėl klaidos vektorių galima laikyti viena silpnesnių McEliece kriptografinės sistemos vietų.

3.4. SILPNOSIOS VIETOS

Originalios McEliece kriptosistemos pagrindiniai trūkumai, kodėl ją sunku pritaikyti yra jos ilgas viešasis raktas, bei mažas pralaidumas [BBC08]. Yra pasiūlyta būdų išspręsti šią problemą. Paprastai stengiamasi pakeisti Goppa kodus, kitomis kodų šeimomis, tačiau jie susilpnina sistemos saugumą [BBC16]. Net pakankamai naujų pasiūlymų kriptosistemos, naudojančios kvazi-ciklinius (angl. *Quasi-Cyclic*) ir kvazi-diadinius (angl. *Quasi-Dyadic*) kodus, buvo įveiktos [UL10]. Kompaktinius raktus gali pasiūlyti LDPC (angl Low-Density Parity-Check) kodai, tačiau ir jie turi rimtų trūkumų [BBC16]. Vis dėlto įmanoma išnaudoti kvazi-ciklinius LDPC kodus sukurti sistemai, kuri būtų atspari visoms žinomoms atakoms [BBC13a] [BBC13b] [BBM+13] [BBC08]. Panašiai įmanoma pritaikyti ir bendruosius Reed-Solomon kodus [BBC16].

Viena didžiausių saugumo silpnybių originalioje McEliece viešojo rakto kriptosistemoje yra tai, kad ji nesugeba apsaugoti žinutės, kuri yra užkoduota daugiau negu vieną kartą, bei kartu negali apsaugoti panašios žinutės, t.y. tokios žinutės, kuri turi žinomą tiesinį sąryšį su kita žinute. Yra pasiūlyta McEliece kriptosistemos variantų, kurie gali sėkmingiau atremti atakas, kurios naudojasi šiomis saugumo silpnybėmis [Sun98].

4. ATAKOS PRIEŠ MCELIECE KRIPTOSISTEMĄ

Yra daug atakų nukreiptų prieš McEliece kriptografinę sistemą. Šiame skyriuje dalis jų yra aptariamos.

4.1. MAŽO SVORIO KODO ŽODŽIŲ RADIMO ATAKA

Tegul C yra n ilgio tiesinis binarinis kodas, kurio dimensija yra k, o minimalus atstumas d. Apie šį kodą niekas nėra žinoma išskyrus generuojančią matricą. Toliau aprašomas algoritmas, kuris skirtas rasti žodį, kurio svoris yra w kode C ir kur w vertė yra arti d [CS98]. Šis algoritmas gali taip pat būti naudojamas dekoduoti kodus, kuriuose padaryta iki $t = \left\lfloor \frac{\mathrm{d}-1}{2} \right\rfloor$ klaidų. Jeigu žinutė x yra sudaryta iš kodo žodžio, kuriame yra padarytos klaidos, panaudojant klaidos vektorių e, kurio svoris $w \leq t$, tai e gali būti rastas su šiuo algoritmu, nes tai yra vienintelis minimalaus svorio žodis tiesiniame kode $C \oplus x$, t.y. šis žymėjimas parodo, kad šio tiesinio kodo generuojančioji matrica sudaroma kaip matrica:

$$\begin{bmatrix} G' \\ c \end{bmatrix},$$

kur $c = xG' + e$. Tai reiškia, kad dekoduojant $[n, k]$ tiesinį kodą tereikia rasti minimalaus svorio kodo žodį $[n, k+1]$ kode.

Tegul $N = \{1, \dots, n\}$ yra koordinačių aibė. Bet kokiam N poaibiui I, $G = (V, W)_I$ žymėjimas reiškia G matricos stulpelių padalinimą pagal I aibę. T.y. $V = (G_i)_{i \in I}$ ir $W = \left(G_j\right)_{j \in N \setminus I}$, kur G_i yra i-tasis matricos G stulpelis. n-bitų vektoriaus x apribojimas pagal koordinačių aibę I žymimas $x_{|I} = (x_i)_{i \in I}$. Kaip ir visur $wt(x)$ žymi binarinio vektoriaus x Hamingo svorį.

Tegul I yra k elementų N poaibis. Tada I yra vadinamas informacijos aibe kodui C tada ir tik tada, jeigu $G^* = (Id_k, Z)_I$ yra sistematinė generuojančioji matrica kodui C. Čia Id_k - $k \times k$ vienetinė matrica. Kadangi yra keletas skirtingų algoritmo realizacijų, čia jis aprašomas taip, kaip jį aprašė Canteaut ir Chabaud [CS98].

Algoritmas taip ir pradedamas, kai atsitiktinai būna pasirenkama informacijos aibė ir panaudojama Gauso eliminacija, tam kad gauti sistematinę generuojančią matricą $G^* = (Id_k, Z)_I$. Toliau, iki tol kol randamas kodo žodis, kurio svoris w atliekami tokie veiksmai:

1. Atsitiktinai padalinama aibė I į dvi aibes I_1 ir I_2, kur $|I_1| = \lfloor k/2 \rfloor$ ir $|I_2| = \lceil k/2 \rceil$. Pagal jas Z matricos eilutės padalinamos į dvi dalis Z_1 ir Z_2. Atsitiktinai pasirenkamas σ elementų poaibis L iš aibės $J = N \backslash I$.
2. Kiekvienai tiesinei kombinacijai Λ_1 (analogiškai Λ_2), kuri gaunama iš matricos Z_1 (analogiškai Z_2) p eilučių, apskaičiuojami $\Lambda_{1|L}$ (analogiškai $\Lambda_{2|L}$) ir visos šios vertės yra sudedamos į maišos lentelę (angl. *hash table*), kurioje viso yra 2^σ verčių.
3. Panaudojant šiomis maišos lentelėmis, paimti visas tiesinių kombinacijų poras (Λ_1, Λ_2), tokias, kad $\Lambda_{1|L} = \Lambda_{2|L}$ ir patikrinti ar galioja lygybė $\mathrm{wt}((\Lambda_1 + \Lambda_2)_{|J \backslash L}) = w - 2p$. Jeigu galioja tada klaidos vektorius e yra tiesinė kombinacija matricos G^* eilučių, kai yra pasirenkamos tos pačios eilutės, kaip ir skaičiuojant $\Lambda_1 + \Lambda_2$ [EOS07].
4. Jeigu e nerandamas, tada atsitiktinai pasirenkami $\lambda \in I$ ir $\mu \in J$ tokie, kad $Z_{\lambda,\mu} = 1$. I pakeičiamas į $(I \backslash \{\lambda\}) \cup \{\mu\}$ ir atitinkamai atnaujinamas Z.

Parametrai p ir σ turi būti pasirenkami taip, kad algoritmo veikimo laikas būtų minimalus. Canteaut ir Chabaud apskaičiavo šiuos parametrus McEliece kriptografinei sistemai su standartiniais pasiūlytais parametrais $p = 2$ ir $\sigma = 18$. Tada vidutinis iteracijų skaičius sėkmingai atlikti šiai atakai yra 9.85×10^{11}. Darbo faktorius (angl. *work factor*) yra lygus $2^{64.2}$.

4.2. STATISTINIO DEKODAVIMO ALGORITMO ATAKA

Tiesinis kodas $[n, k]$ gali būti charakterizuojamas ne tik savo generuojančia matrica G, bet ir savo dualaus kodo generuojančia matrica H. H vadinama tiesinio kodo $[n, k]$ kontroline matrica ir jos sąryšiai su generuojančia matrica ir kodo žodžiu u yra:

$$GH^T = 0, \qquad uH^T = 0.$$

Tegul C ir $\mathbb{H}$ yra aibės visų kodo žodžių sugeneruotų kodo generuojančios matricos G ir jo dualaus kodo generuojančios matricos H [Jab01]. Kiekvienam kodo žodžiui $u \in C$ ir $h \in \mathbb{H}$ galioja lygybė:

$$uh^T = 0.$$

Tegul gautas vektorius c yra kodo žodžio u ir klaidos vektoriaus e, kurio svoris $\leq t$ suma:

$$u = xG,$$

$$c = u + e.$$

Jeigu $ch^T = 1$, tada šiam konkrečiam h yra sakoma, kad jis atrado nelyginį skaičių klaidų padarytų c. Panašiai, jeigu $ch^T = 0$, tai h sakoma, kad rado lyginį skaičių klaidų padarytų c. Pastarasis atvejis taip pat galioja ir tuo atveju, jeigu klaidų padaryta nebuvo.

Dabar apžvelgiamas atvejis, kai nelyginių skaičiaus klaidų radimo procesas yra apribotas tik tokiems $\mathbb{H}$ elementams, kurių svoris yra didelis. Tegul šis poaibis būna žymimas $\mathbb{H}_w$. Maži svoriai taip pat gali būti naudojami vietoje didelių. Bendru atveju, vektorius h, procese ch^T, elgiasi kaip kaukė arba filtras vektoriui c. Jeigu $ch^T = 1$ arba ekvivalenčiai $eh^T = 1$, tada yra labai tikėtina, kad vektoriaus h komponentės ant klaidų koordinačių bus vienetai.

Jeigu visi vektoriai $h \in \mathbb{H}_w$, kuriems galioja sąlyga ch^T duotajam c, sudedami, tada koordinatės su klaida dažniausiai turės didesnius dažnius negu pozicijos be klaidų. Tegul gautasis vektorius yra v:

$$v = \sum_{h \in \mathbb{H}_w} (ch^T)h. \tag{4.2.1}$$

Operacija $ch^T = 1$ išskaido $\mathbb{H}_w$ į du poaibius. Abu suteikia informacijos apie klaidų padėtis. Viename poaibyje yra klaidų koordinačių informacija tarp m $(\geq t)$ didžiausių v reikšmių, tuo tarpu kitame poaibyje yra klaidų koordinačių informacija tarp m $(t \leq m \leq n - k)$ mažiausių reikšmių. Pasirinkimas tarp didžiausių ir mažiausių priklauso atitinkamai ar t yra nelyginis ar lyginis. m čia yra pasiūlomas, tam kad atsižvelgti į neaiškumus lokalizuojant t klaidų koordinates. Faktiškai, dėl statistinės v skaičiavimo prigimties, klaidos nebus izoliuotos iki t maksimalių verčių, bet bendrai bus tarp didesnių koordinačių skaičiaus; šiuo atveju m. Ši vertė m turėtų garantuoti, kad visi klaidų modeliai yra ištaisomi.

Kadangi $wt(e)$ nežinomas iš anksto, dekoduotojas turi apgalvoti du scenarijus. Pirmasis, kad visoms nelyginės $wt(e)$ vertės ir klaidos yra tarp vektoriaus v didžiausių m koordinačių. Antrasis, kad $wt(e)$ vertės lyginės ir kad klaidos yra tarp m mažiausių vektoriaus v koordinačių. Kai m skaičiaus klaidų koordinačių kandidatės yra nustatomos, tada pasirenkamas subvektorius c_k, sudarytas iš k likusiųjų c koordinačių, kur klaidos nepadarytos. Atitinkamai galima paimti

submatricą G_k iš G ir jeigu egzistuoja suskaičiuoti $c_k G_k^{-1}$, arba bandyti kitą subvektorių. Tegul x_1 ir x_2 yra atitinkami sprendimai abiems scenarijams:

$$x_i = c_{ki} G_{ki}^{-1} \quad i = 1,2. \tag{4.2.2}$$

Dekoduotojas gali rasti teisinga sprendimą patikrindamas $x_1 G + c$ ir $x_2 G + c$ svorius ir tada pasirinkdamas tą x, kuris sugeneruoja svorį mažesnį arba lygų t.

Taigi statistinio dekodavimo algoritmą galima sutrumpintai suskirstyti į šiuos etapus:

1. Paskaičiuojami klaidas lokalizuojantys vektoriai v pagal (4.2.1) formulę.
2. Paskaičiuojami x_1 ir x_2 pagal (4.2.2) formulę.
3. Patikrinamos vertės:

$$wt(x_i G + c) \qquad i = 1,2,$$

ir pasirenkamas tas x_i, iš kurio seka, kad tikrinamas svoris yra $\leq t$. Ši vertė ir yra prilyginama ieškomam kodo žodžiui x.

4.3. PAKARTOTINAI SIŲSTOS ŽINUTĖS ATAKA

Tarkime, kad dėl kažkokios priežasties dvi kriptogramos:

$$c_1 = xSGP + e_1$$

ir $$e_1 \neq e_2$$

$$c_2 = xSGP + e_2$$

yra išsiunčiamos [Ber97]. Tokia sąlyga vadinama pakartotinai siųstos žinutės (angl. *message-resend*) sąlyga. Šiuo atveju, kriptoanalitikui yra lengva atkurti žinutę x iš sistemos c_i. Galima pastebėti, kad $c_1 + c_2 = e_1 + e_2$. Tai reiškia, kad dviejų vienodų kriptogramų sumos svoris yra $\leq 2t$. Jis yra žymiai mažesnis negu turėtų būti, jeigu šios sąlygos nebūtų. Tokiu būdu galima aptikti pakartotinai siųstos žinutės sąlygą ir ja pasinaudoti atakuojant sistemą.

Toliau skaičiuojamos dvi aibės iš $c_1 + c_2$. Aibė L_0 yra tokia aibė koordinačių pozicijų, kur $c_1 + c_2$ koordinatės yra nuliai. Tuo tarpu, aibė L_1 bus tokia, kur pozicijose $c_1 + c_2$ yra vienetai.

Tegul

$$L_0 = \{l \in \{1,2,\dots,n\}: c_1(l) + c_2(l) = e_1(l) + e_2(l) = 0\}$$

ir

$$L_1 = \{l \in \{1,2,\dots,n\}: c_1(l) + c_2(l) = e_1(l) + e_2(l) = 1\}.$$

Toliau, stengiamasi pasinaudoti faktais, kad:

- kai $l \in L_0$, tai greičiausiai nei $c_1(l)$, nei $c_2(l)$ koordinatės nėra paveiktos klaidos vektoriumi, o
- kai $l \in L_1$, tai tiksliai vienas iš $c_1(l)$ ar $c_2(l)$ koordinatės yra paveiktos klaidos vektoriumi.

Kiekvienas $l \in L_0$ reiškia, kad arba $e_1(l) = e_2(l) = 0$, arba $e_1(l) = e_2(l) = 1$. Darant prielaidą, kad e_1 ir e_2 yra pasirenkami atsitiktinai, tada bet kokiam l tikimybė P:

$$P(e_1(l) = e_2(l) = 1) = \left(\frac{t}{n}\right)^2.$$

Pagal McEliece pasiūlytus parametrus $n = 1024$, o padarytų klaidų skaičius $t = 50$, gaunama, kad $P(e_1(l) = e_2(l) = 1) \approx 0.0024$. Kitais žodžiais, dauguma $l \subset L_0$ parodo, kad $e_1(l) = e_2(l) = 0$. Taip kriptonalistas gali lengvai atspėti $k = 524$ stulpelius, kuriuose nėra klaidų iš tų, kurie indeksuoti, kaip priklausantys aibei L_0.

Galima taip pat nustatyti, kiek veiksminga yra ši strategija. Tegul p_i yra tikimybė, kad tiksliai i koordinatės yra tuo pačiu metu paveiktos klaidos vektoriais e_1 ir e_2:

$$p_i = P(|\{l: e_1(l) = 1\} \cap \{l: e_2(l) = 1\}| = i) = \frac{\binom{50}{i}\binom{974}{50-i}}{\binom{1024}{50}}.$$

Tuo tarpu tikėtinas L_1 kardinalumas yra:

$$E(|L_1|) = \sum_{i=0}^{50}(100 - 2i)p_i \approx 95.1,$$

nes kiekvienas l, kuriam $e_1(l) = e_2(l) = 1$ sumažina $|L_1|$, per reikšmę 2. Pvz. tarkim $|L_1| = 94$, tada $|L_0| = 930$ iš kurių tik 3 yra paveikti klaidos vektoriaus. Matyti, kad tikimybė, kad atspėti 524 nepaveiktus stulpelius iš tų, kurie pažymėti L_0 yra:

$$\frac{\binom{927}{524}}{\binom{930}{524}} \approx 0.0828,$$

taigi kriptoanalitikas tikisi sėkmingai atlikti ataką šiuo atveju tik su 12 spėjimų. Kai $|L_1| = 96$ tai užtenka tik apie 5 spėjimų.

4.4. APLAIDŽIOS ALISOS ATAKA

Aplaidžios Alisos ataka (angl. *sloppy Alice attack*) yra atakos, kuriose kenksmingas tarpininkas Zigmas gauna žinių, kad ar užšifruotos žinutės yra atkoduojamos sėkmingai, ar ne.

Šiam algoritmui aprašyti reikalinga sąvoka maksimalių klaidų savybė (angl. *maximum error property* (MEP)) [VDT02]. Tegul turime McEliece kriptosistemą, kurios dekodavimo algoritmas yra $\mathbb{A}_t$. Tada persiunčiamas vektorius c, kuriame padarytos $\leq t$ klaidų. Sakoma, kad algoritmas $\mathbb{A}_t$ turi MEP savybę, jeigu dekoduojant c šis algoritmas arba gražins kodo žodį $u = xG'$ tiesiniame kode C, kurio atstumas iki c yra $\leq t$ (jeigu toks kodo žodis egzistuoja) arba grąžins klaidos pranešimą. Tokiu atveju $\mathbb{A}_t$ niekada negrąžins kodo žodžio, kuris yra didesnio negu t atstumu nuo c.

Toliau galima aprašyti algoritmą. Tegul McEliece kriptosistemos dekodavimo algoritme ir yra naudojamas toks $\mathbb{A}_t$, kuris turi maksimalių klaidų savybę. Taip pat c yra šifras siunčiamas Alisos ir perimtas Zigmo ($c = xG' + e$). Tada Zigmas gali atlikti tokius veiksmus:

1. Padidinamas klaidų skaičius, kuris buvo atliktas Alisos šifre c iki t. Kad tai būtų atlikta, Zigmas vis keičia atsitiktinai pasirinktas vektoriaus c koordinates, kiekvieną koordinatę pakeisdamas daugiausiai vieną kartą, ir vis nusiunčia gautą vektorių Bobui. Jis tą daro tol kol iš Bobo gauna klaidos pranešimą. Tada Zigmas žino, kad dabar klaidų kodo žodyje yra padaryta per daug ir prieš tai siųstame vektoriuje ir buvo maksimalus klaidų skaičius. Šis vektorius yra pavadinamas c'.

2. Nustatomas pakankamas skaičius koordinačių, kuriose nėra padarytos klaidos. Kai Zigmas žino, kad turi vektorių su tiksliai t klaidų, jis gali pradėti atidžiau žiūrėti į atsitiktines koordinates (skirtingas negu visi prieš tai buvę pasirinkimai, įskaitant ir iš pirmo žingsnio). Jis gali pasirinkęs atskirą koordinatę ją pakeisti vektoriuje c' ir gautąjį rezultatą vėl išsiųsti Bobui. Jeigu klaidos pranešimas yra grąžinamas, tai reiškia, kad šioje koordinatėje nebuvo padaryta klaida. Kai pakankamai tokių koordinačių yra nustatoma Zigmas gali atlikti trečiąjį žingsnį.
3. Gaunamas pradinis tekstas. Kai Zigmas žino pakankamai koordinačių, kuriose nepadarytos klaidos, tada jis gali išspręsti matricų lygtį $r' = xG'$ pradiniam tekstui x, naudojant Gauso metodą tiems stulpeliams, kuriuose žinoma, kad nebuvo padarytos klaidos.

4.5. APIBENDRINTOJI INFORMACIJOS AIBĖS DEKODAVIMO ATAKA

Apibendrintoji informacijos aibės dekodavimo ataka (angl. *Generalized Information-Set-Decoding Attack*) veikia taip: tegul ${G'}_k$ žymi k nepriklausomų stulpelių pasirinktų iš viešosios matricos G' ir c_k kartu su e_k žymi atitinkamas k coordinates šifre c ir klaidos vektoriuje e. Juos sieja atitinkamas ryšys [KI03]:

$$c_k = xG'_k + e_k.$$

Jeigu $e_k = 0$ ir G'_k kvadratinė matrica, kurios determinantas nelygus nuliui, tai žinutė x gali būti randama:

$$x = (c_k + e_k){G'_k}^{-1}.$$

Net jeigu $e_k \neq 0$, x gali būti gaunamas spėjant e_k tarp mažos aibės $\{e_k | wt(e_k) \leq j\}$ mažiems j. Pradinio teksto x teisingumas yra patikrinamas, tikrinant ar reiškinio Hamming'o svoris:

$$c + x \cdot G' = c + c_k G'^{-1}_k \cdot G' + e_k {G'}^{-1}_k \cdot G'$$

yra t ar ne. Taigi, galiausiai algoritmas atrodo taip:

- Turimi duomenys: šifras c, viešasis raktas (G', t) ir atakos parametras $j \in \mathbb{Z}$.

- Randamas pradinis tekstas x.

Algoritmo žingsniai:

1. Pasirenkami k nepriklausomi stulpeliai iš G' ir tada apskaičiuojama $\widehat{G'_k} := G'^{-1}_k G'$. Tegul I žymi aibę indeksų iš k pasirinktų stulpelių ir tada J žymi aibę iš likusiųjų stulpelių.
2. Kartojami šie žingsniai iki tol kol žinutė x yra randama:

2.1. (Procesas jeigu $wt(e'_k) = 0$). Suskaičiuojama $\hat{e} := c + c_k \widehat{G'_k}$. Jeigu $wt(\hat{e}) = t$, tai $x := c_k G'^{-1}_k$.

2.2. (Procesas jeigu $1 \leq wt(e'_k) \leq j$). Kiekvienam i_1 nuo 1 iki j atliekami šie veiksmai:

Visoms $\binom{n}{i_1}$ vektoriaus e'_k kombinacijoms tokioms, kad $wt(e'_k) = i_1$, atlikti tai:

A) Sugeneruoti naują e'_k kombinaciją. Jeigu $wt(\hat{e} + e'_k \widehat{G'_k}) = t$, tai $x := (c_k + e'_k) G'^{-1}_k$.

2.3. Pakeičiama viena koordinatė aibėje I į koordinatę aibėje J ir tada atnaujinama matrica $\widehat{G'_k} := G'^{-1}_k G'$, panaudojant Gauso eliminaciją.

4.6. LAIKINĖ ATAKA PRIEŠ PATERSONO ALGORITMĄ

Laikinė ataka prieš Patersono algoritmą (angl. *timing attack against Patterson algorithm*) remiasi tuo, kad dekoduojant Goppa kodus naudojant Patersono algoritmą, klaidų lokatoriaus polinomas yra palyginamas n kartų. Šio polinomo laipsnis daro įtaką dekodavimo trukmei. Kriptoanalitikas gali bandyti pakeisti tą trukmę, pakeisdamas Hamming'o svorį klaidos vektoriuje pasirinktam šifrui. Čia toliau aprašomas tik pačios atakos algoritmas, papildomos informacijos apie šią ataką galima rasti [SSM+10].

Nagrinėjama ataka remiasi vieno bito c_i turimame šifre c apvertimu ir dekodavimo trukmės matavimu. Kai apverstasis bitas nėra klaidos vektoriaus teisingas bitas, tai šio bito apvertimas tik padidins padarytų klaidų skaičių. Jeigu tas bitas yra klaidos vektoriaus teisingas bitas, tai šio bito apvertimas sumažins klaidų skaičių šifre ir dekodavimas užtruks trumpiau negu klaidingu atveju.

Algoritme naudojami: šifras c, McEliece kriptosistemos parametrai t – klaidų vektoriaus svoris, n – šifro ilgis, bei tikslumo parametras M. Toliau pateikiamas algoritmo žingsniai, kai t yra nelyginis:

1. Kiekvienam i nuo 0 iki $n - 1$ atliekama:

1.1. Šifre c apverčiama i koordinatė ir rezultatas prilyginamas z.

1.2. Kiekvienam j nuo 0 iki M atliekami veiksmai:

1.2.1. Išprovokuojama dešifruoti šifrą z, tokiu būdu išmatuojant dešifravimo laiką $T_{i,j}$.

1.3. Apskaičiuojama visų $T_{i,j}$ vidutinė reikšmė T_i ir įdedama į sąrašą L.

2. Sąrašas L yra surikiuojamas T_i reikšmių didėjimo tvarka.

3. Toliau kiekvienam k nuo 0 iki $t-1$ atliekama:

3.1. Gaunamas i indeksas k elementui sąraše L.

3.2. Šis indeksas parodo kurioje klaidų vektoriaus vietoje yra klaida, t.y. $e_i := 1$.

4. Atlikus veiksmus gaunamas klaidos vektorius e.

Algoritmą galima modifikuoti ir lyginiam t. Tokiu atveju reikia modifikuoti bent du bitus šifre c, atliekant žingsnį 1.1. Tada tereikia apvertus i-ąją koordinatę, kartu apvertinėti visas kitas koordinates nuo $i+1$ iki $n-1$ ir kiekvienai dviejų apvertimų koombinacijai išmatuoti dešifravimo laikus ir tęsti visą procesą, kaip ir nelyginio t atveju.

4.7. SUSIJUSIŲ PRANEŠIMŲ ATAKA

Susijusių pranešimų ataka (angl. *Related-Message attack*) yra apibendrintasis 4.3. pakartotinai siųstos žinutės atakos atvejis [Ber97]. Tarkime turime dvi kriptogramas:

$$c_1 = x_1 SGP + e_1$$

ir $$e_1 \neq e_2$$

$$c_2 = x_2 SGP + e_2$$

Jeigu yra žinomas tiesinis ryšys tarp žinučių x_1 ir x_2, tarkime $x_1 + x_2$, tada tokia sąlyga yra vadinama susijusių pranešimų sąlyga. Turint šią sąlygą galima rasti x_i panašiai, kaip 4.3. atakos atveju. Sudedant abu šifrus gaunama:

$$c_1 + c_2 = x_1 SGP + x_2 SGP + e_1 + e_2.$$

Čia galima pastebėti, kad $x_1SGP + x_2SGP = (x_1 + x_2)SGP$ ir šią reikšmę galima suskaičiuoti remiantis susijusių pranešimų sąlyga ir viešuoju raktu G'. Tada galima apskaičiuoti:

$$c_1 + c_2 + (x_1 + x_2)SGP = e_1 + e_2,$$

ir tęsti atakos vykdymą, kaip 4.3. atakos atveju pakeičiant $c_1 + c_2$ sąlygą į $(c_1 + c_2 + (x_1 + x_2)SGP)$.

5. ATAKŲ PRIEŠ MCELIECE KRIPTOGRAFINĘ SISTEMĄ TYRIMAS

5.1. TYRIMO TECHNINĖ IR PROGRAMINĖ ĮRANGA

Šiame darbe atakų realizacijai reikia ne tik pačių atakos realizacijų, tačiau ir pačios McEliece kriptografinės sistemos realizacijos, nes be jos šių atakų nebuus įmanoma atlikti. Kadangi tyrimo metu yra fokusuojamasi daugiausiai į pačias atakas ir sistemos saugumą, tai yra naudojamasi jau suprogramuota McEliece kriptografinės sistemos realizacija.

Kadangi atakas norima realizuoti pasinaudojant JAVA programavimo kalba, tai McEliece kriptografinės sistemos realizacija yra gaunama pasinaudojant Bouncy Castle JAVA kriptografiniu paketu (angl. *The Bouncy Castle Crypto Package For Java* https://github.com/bcgit/bc-java). Šis Bouncy Castle kriptografinis paketas yra kriptografinių algoritmų JAVA implementacija, kuri buvo sukurta Legion of the Bouncy Castle. Šiame pakete tarp skirtingų kriptografinių šifravimo algoritmų taip pat yra ir McEliece kriptografinės sistemos implementacija, kuria ir yra naudojamasi šiame darbe viešųjų raktų generavimui, bei žinučių šifravimui.

McEliece kriptografinės sistemos algoritmai yra realizuojami naudojant GF2Matrix ir GF2Vector klases, kurios taip pat yra aprašytos pakete. Tai atitinkamai dvejetainių matricų ir vektorių implementacijos, su kuriomis galima atlikti matematinius veiksmus būdingus matricoms ir vektoriams. Šios klasės pasižymi tuo, kad kiekvienas bitas, kiekviena matricos ar vektoriaus laukelio reikšmė yra saugoma int duomenų tipuose. JAVA programavimo kalboje int duomenų tipas atmintyje viso užima 32 bitus, todėl GF2Matrix vienos eilutės 32 stulpelių reikšmes atitinka vienas int duomenų tipo sveikasis skaičius. Kadangi McEliece kriptografinė sistema naudoja šias matricos ir vektoriaus implementacijas, tai atliekant atakas nukreiptas prieš McEliece kriptografinę sistemą taip pat yra naudojamasi jomis.

Atakų programinis kodas yra rašomas ir pačios atakos yra atliekamos naudojantis asmeniniu kompiuteriu, kurio techniniai parametrai pateikti 1 lentelėje, bei naudojantis Eclipse integruota kūrimo aplinka (angl. *IDE*) ir JAVA 15-aja versija. Kad pasinaudoti Bouncy Castle bibliotekomis yra naudojamas Maven kūrimo įrankis (angl. *build tool*), kuris suranda ir instaliuoja šias bibliotekas, tarp kurių yra McEliece kriptografinės sistemos realizacija.

1 lentelė. Asmeninio kompiuterio techniniai parametrai.

Procesorius (CPU)	8 branduolių Intel 10700F 4.6 GHz (visų branduolių turbo-boost)
Operatyvioji atmintis (RAM)	16GB 3600MHz CL16
Operacinė sistema	64 bitų Windows 10 Pro

Realizuojant atakas, buvo parašytas programinis kodas, kuriame yra interfeisas pavadinimu Attack ir kuris turi vieną abstraktų metodą pavadinimu decrypt. Kiekviena sukurta ataka turi realizuoti Attack intefeisą ir jo metodą decrypt. Realizuotas metodas decrypt kiekvienoje sukurtoje atakos klasėje turi gauti McEliece kriptosistemos šifrą, metodo viduje atlikti konkrečios atakos algoritmą ir pasinaudodamas juo iššifruodamas gautąjį šifrą, grąžinti iššifruotą žinutę. Visuose toliau pateiktuose atakų bandymuose atakų vykdymo laikas yra matuojamas kaip realizuoto Attack interfeiso decrypt metodo vykdymo laikas, t.y. laikas nuo šifro gavimo iki iššifruotos žinutės grąžinimo. McEliece kriptosistemos raktai yra sugeneruojami parenkant m ir t reikšmes. Vykdymo laikai ir kita papildoma informacija yra rašoma į failus pasinaudojant loginimo biblioteka logback.

Paprastai vienai atakai su vienodais parametrais yra atliekamas tam tikras skaičius bandymų ir rezultatai yra suvidurkinami, nubraižomi grafikai. Naudojantis šiais rezultatais ir grafikais yra nagrinėjamos atakos, šifrai ir jų saugumas, atsparumas atakoms. Tyrime šifras laikomas saugiu prieš konkrečią ataką, jeigu ta ataka su superkompiuteriu užtrunka ilgiau negu 20 metų [Woo10].

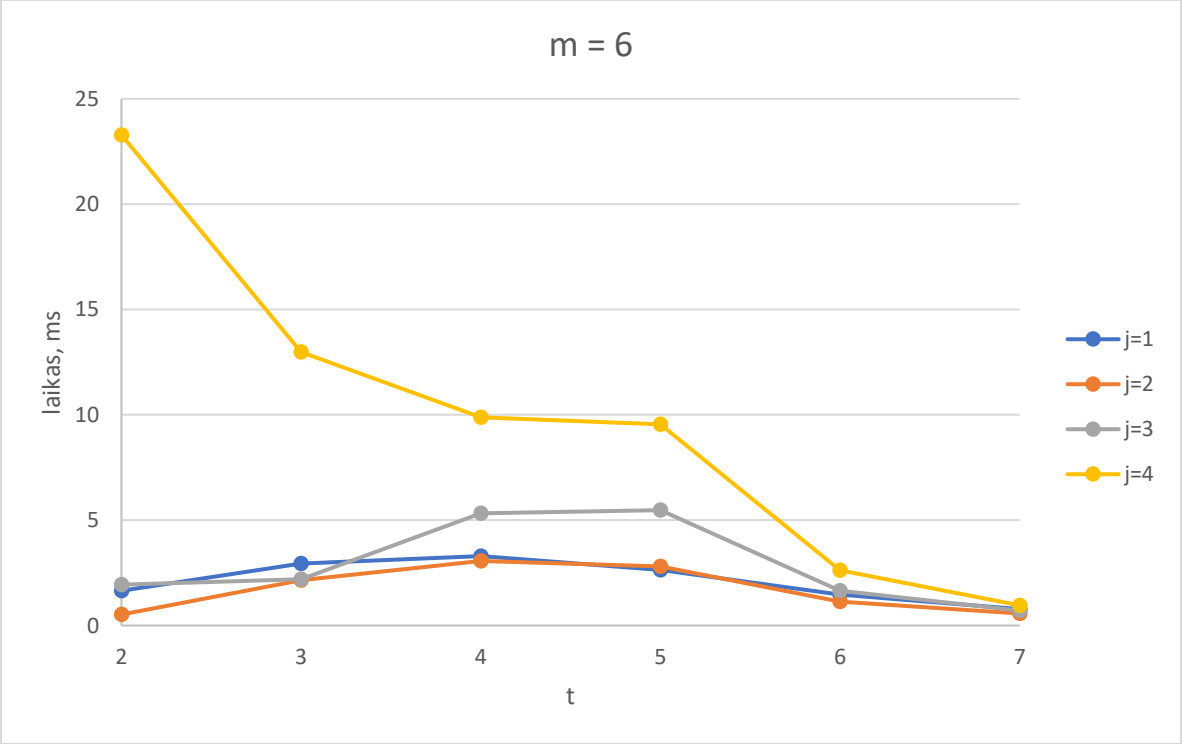
5.2. APIBENDRINTOSIOS INFORMACIJOS AIBĖS DEKODAVIMO ATAKOS TYRIMAS

Darbo metu buvo suprogramuota klasė realizuojanti apibendrintąją informacijos aibės dekodavimo ataką. Ši klasė viešąjį raktą (G', t), gali gauti tiek per konstruktorių, tiek per seterius. Tuo tarpu atakos parametras j yra gaunamas tik per seterį.

Šiai atakai, kiekvienam i_1 nuo 1 iki j reikia visų $\binom{n}{i_1}$ vektoriaus e'_k kombinacijų. Nuo šių kombinacijų generavimo efektyvumo labai priklauso pačios atakos efektyvumas. Parašytoje atakos implementacijoje kombinacijų generavimas vykdomas naudojant rekursinę funkciją. Deja, bet rekursinė funkcinė nėra efektyvi JAVA programavimo kalboje. Kad ataka būtų efektyvesnė ir

vyktų greičiau, galbūt reikėtų sugalvoti kaip padaryti pačią rekursiją kiek įmanoma efektyvesnę, o galbūt net pakeisti rekursiją į iteraciją. Tai galbūt galėtų pagreitinti vykdomą ataką. Tai ir buvo bandoma padaryti. Pirmąjį realizuoto algoritmo variantą buvo bandoma optimizuoti, optimizuojant rekursiją, bet skaičiavimo laikai arba pailgėdavo, arba nepasikeisdavo. Taip pat buvo bandoma rekursiją pakeisti į iteraciją panaudojant steką, bet šiuo atveju atakos laikas tik nežymiai pailgėjo. Galiausiai buvo nuspręsta tyrimą tęsti su pirmuoju realizuoto algoritmo variantu.

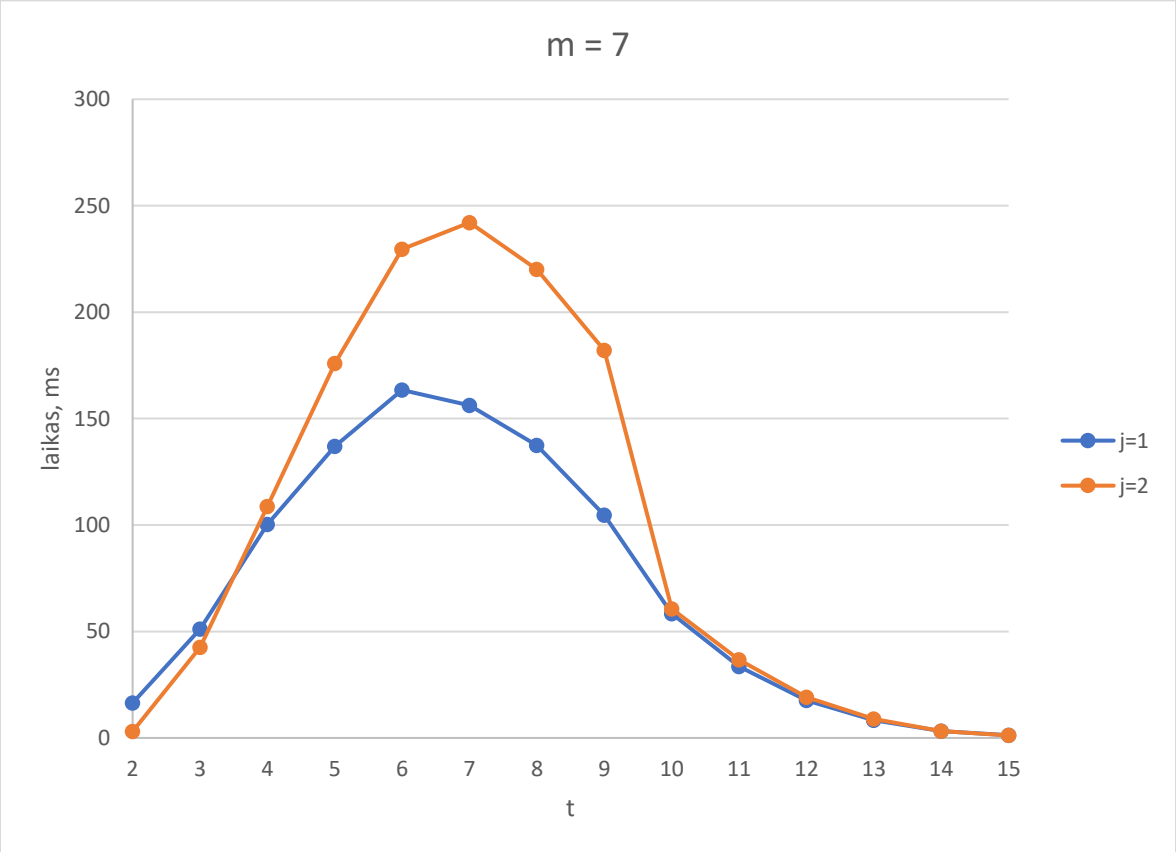
Atakos atliekamos su įvairiais parametrais, tačiau kai $m < 6$ atakos trunka mažiau negu 1 ms, todėl su tokiais parametrais jos neatliekamos. Kai $m > 8$ atakos užtrunka per ilgai, todėl su tokiais paremtrais jos irgi neatliekamos. 1 pav. grafiko kreivės rodo atliktas atakas su $m = 6$, visiems t nuo 2 iki 7 ir j atakos parametrais nuo 1 iki 4. Vienam grafiko taškui atidėti buvo atlikti viso 100000 bandymų su naujai sugeneruotais viešaisiais raktais. Galima pastebėti iš kreivių, kad kai šifro parametras t yra mažesnis už j tada ataka užtrunka ilgiau. Taip yra todėl, nes teisingas e'_k niekad nebus didesnio svorio negu t, todėl kai kurios kombinacijos, kurių svoriai didesni yra generuojamos be reikalo. Net, jeigu $j < t$ vis tiek matyti, kad atakos laikai trumpiausi yra prie mažesnių j parametrų, t.y., kai $j = 1$ ir $j = 2$, o su didesniais atakos jau užtrunka žymiai ilgiau ir



1 pav. Apibendrintosios informacijos aibės dekodavimo atakos grafikas, kai m = 6.

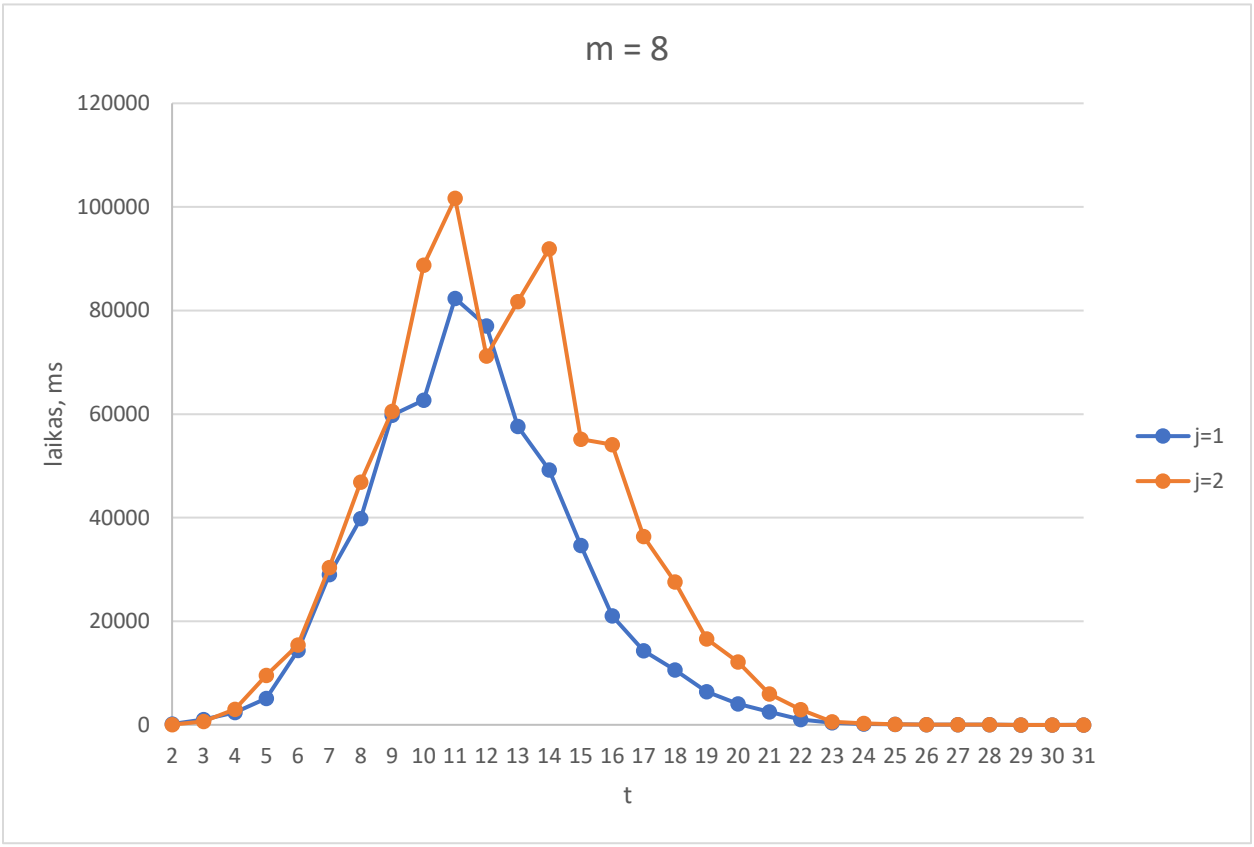
šie parametrai nebėra tokie optimalūs. Matavimai su $j > 4$ nebebuvo vykdomi, nes atakos su vis didesniais j parametrais trunka vis ilgiau. Optimaliausias atakos parametras, kai $m = 6$ vis dėlto yra $j = 2$. Tuo tarpu iš grafiko matyti, kad su šiuo optimaliausiu parametru ilgiausiai atakos užtrunka, kai padarytų klaidų skaičius kode $t = 4$ ir trunka vidutiniškai $3{,}06395\ ms$. Taigi saugiausia McEliece kriptografinė sistema, kai $m = 6$ yra, kai $t = 4$.

2 pav. grafike pavaizduotas atliktas analogiškas tyrimas, tik šįkart jau su $m = 7$. Kadangi šifras ilgesnis, jame galima padaryti ir daugiau klaidų, todėl matavimai atliekami t vertėms nuo 2 iki 15. j vertės šįkart imamos tik dvi optimaliausios $j = 1$ ir $j = 2$. Atakos truko ilgiau, todėl vienam taškui grafike atidėti buvo atlikta 5000 bandymų. Iš grafikų matyti, kad $j = 1$ yra optimaliausias parametras realizuotai atakos implementacijai, o kai padarytų klaidų skaičius $t = 6$ yra gaunamas saugiausias šifras, kai $m = 7$. Su optimaliausiu parametru šio saugiausio šifro įveikimui vidutiniškai reikia $163{,}4194\ ms$. Taigi ši ataka trunka 54 kartus kartus ilgiau su $m = 7$ saugiausiu šifru lyginant su $m = 6$ saugiausiu šifru.



2 pav. Apibendrintosios informacijos aibės dekodavimo atakos grafikas, kai m = 7.

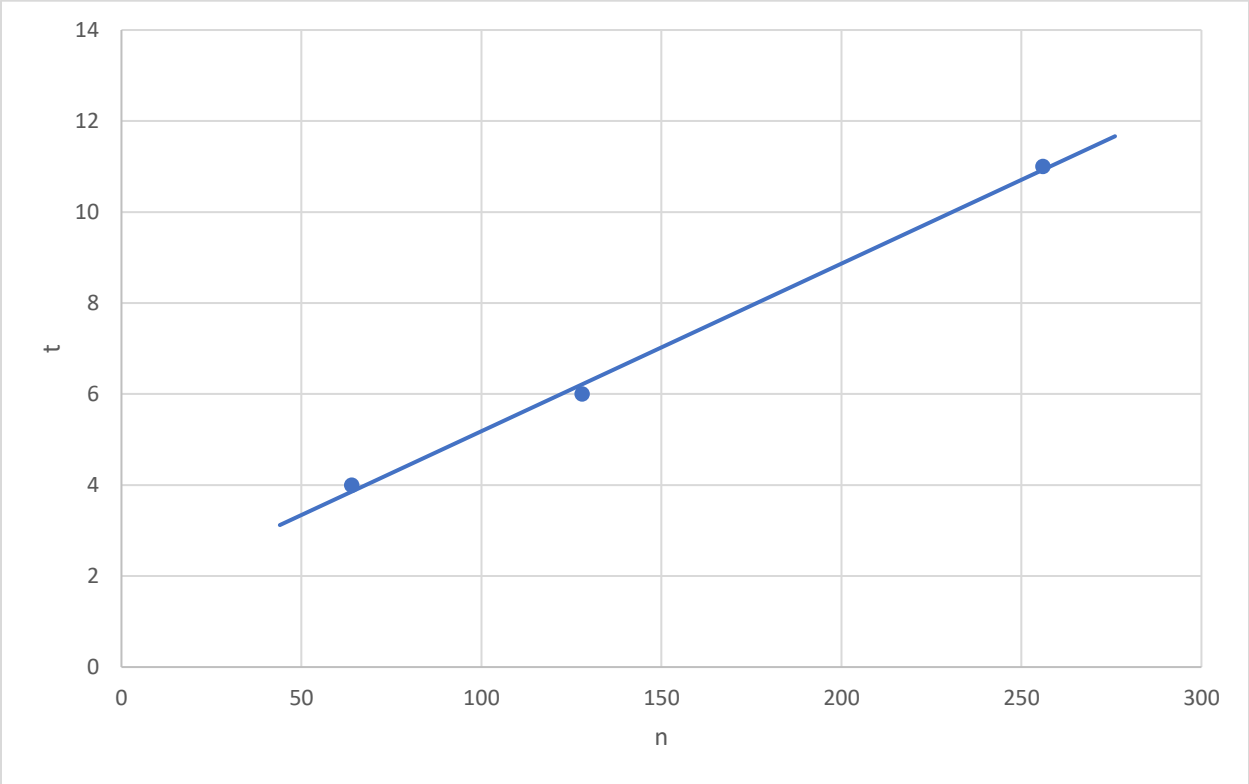
3 pav. grafike pavaizduoti atakų laikai įveikti kriptosistemai, kurios $m = 8$ skirtingiems t nuo 2 iki 31. Vėl atliekami bandymai tik dviem optimaliausiems atakos parametrams $j = 1$ ir $j = 2$. Kadangi šįkart atakos užtruko žymiai ilgiau negu prieš tai buvusiais atvejais, su kai kuriais parametrais 1 bandymas net daugiau negu vidutiniškai 100 sekundžių, tai vienam taškui grafike buvo atlikta tik 50 bandymų. Iš jų matyti, kad $j = 1$ vėl yra optimaliausias parametras, o ilgiausiai trunkanti ataka prie šio parametro yra, kai $t = 11$. Tada ataka trunka $82350{,}64\ ms$ arba $82{,}35064$ sekundžių. Ši ataka jau užtrunka apie 504 kartus ilgiau negu ataka su $m = 7$.



3 pav. Apibendrintosios informacijos aibės dekodavimo atakos grafikas, kai $m = 8$.

Iš 1-3 pav. grafikų matyti, kad atakos trumpiausiai užtrunka, kai t yra arba mažas, arba didelis. Tai galima paaiškinti iš atakos algoritmo struktūros. Algoritmas paprastai stengiasi atspėti klaidų vektoriaus k stulpelių, kuriuose nėra padaryta klaidų. Kai t mažas tai yra labai lengva padaryti, nes dauguma stulpelių neturi klaidų. Tuo tarpu iš (2.3.3) formulės matyti, kad didėjant t reikšmei, k mažėja žymiai greičiau negu auga t. Dėl šios priežasties reikia atspėti mažiau stulpelių, kuriuose nepadarytos klaidos ir prie didelių t verčių atakos trunka trumpiau.

Atlikus bandymus su mažais McEliece kriptosistemos parametrais iš gautųjų rezultatų galima apytiksliai nustatyti saugiausią kriptosistemos t parametrą įvairiems šifro ilgiams $n = 2^m$. Tam 4 pav. yra braižoma šio eksperimento metu rastų saugiausių atakos t parametrų šiais atakai priklausomybė nuo šifro ilgio n. Kadangi iš atidėtų taškų priklausomybė panaši į tiesinę, tai pats grafikas yra aproksimuojamas tiesine priklausomybe. (5.2.1) formulė yra šios tiesinės priklausomybės lygtis. Naudojantis šia lygtimi galima apytiksliai apskaičiuoti, kokie yra saugiausi šifro parametrai t, kiekvienam m arba šifro ilgiui n. Iš formulės gaunama, kad kai $m = 9$, tai saugiausias parametras $t = 20$, kai $m = 10$, tai atitinkamai $t = 39$, o $m = 11$ atveju - $t = 77$.



4 pav. Saugiausių apibendrintosios informacijos aibės dekodavimo atakos šifro parametrų t priklausomybė nuo n.

$$t = 0{,}0368n + 1.5 \tag{5.2.1}$$

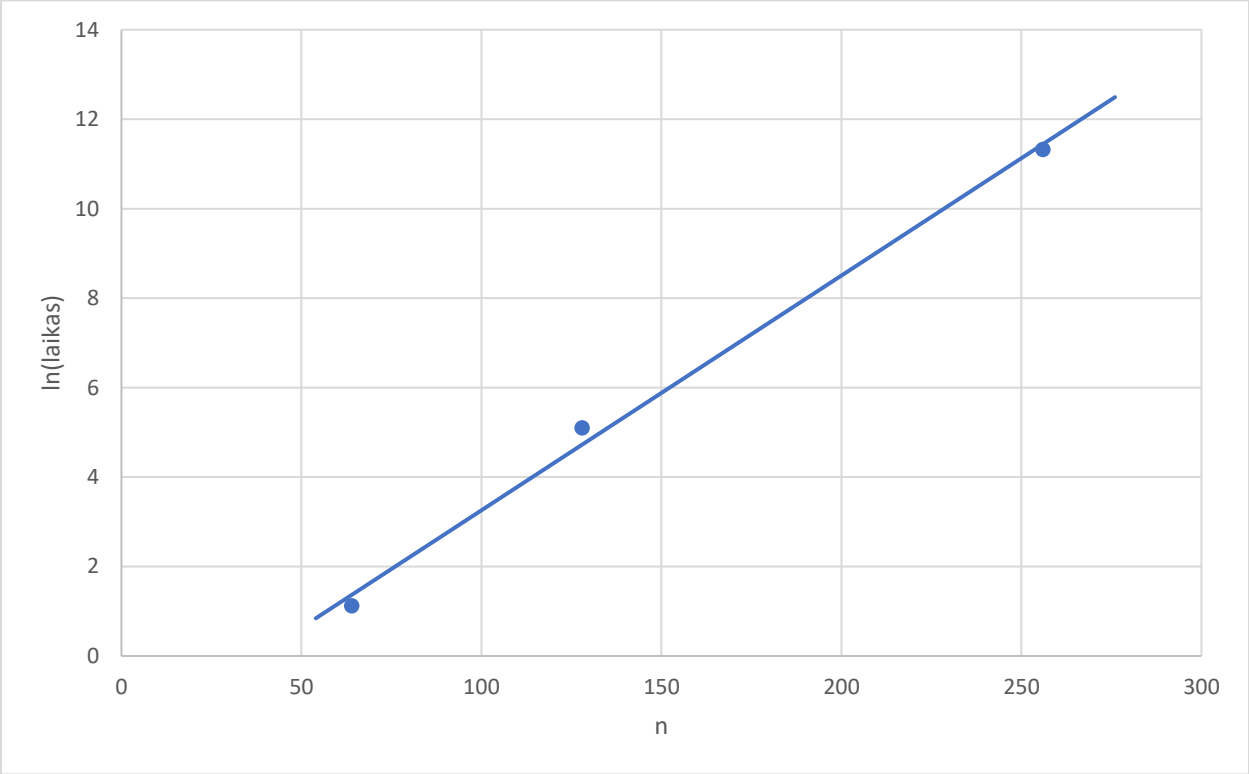
Analogišką analizę galima atlikti nustatant kiek vidutiniškai laiko užtruktų apibendrintosios aibės dekodavimo atakos įvairiems šifro ilgiams n, naudojant saugiausius t parametrus. Iš atliktų eksperimentų matyti, kad didėjant šifro ilgiui n vidutinis atakos laikas saugiausiam t parametrui didėja žymiai greičiau ir įgauna panašią į eksponentinę formą. Kad tuo

įsitikinti apskaičiuojami eksperimente išmatuotų saugiausių parametrų vidutinių atakos vykdymo laikų natūralūs logaritmai (2 lentelė) ir 5 pav. grafike atidedama jų priklausomybė nuo šifro ilgio n. Čia vėl gautasis grafikas yra aproksimuojamas tiesine priklausomybe, kurios lygtis:

$$\ln(laikas) = 0{,}0525n - 1{,}9915. \tag{5.2.2}$$

2 lentelė. 5 pav. grafiko taškų duomenys.

laikas, ms	3,06395	163,4194	82350,64
ln(laikas)	1,1197	5,0963	11,3187
m	6	7	8
n	64	128	256



5 pav. Saugiausių apibendrintosios informacijos aibės dekodavimo atakos saugiausių šifro parametrų vidutinių atakos laikų priklausomybė nuo n.

Iš šios lygties galima apytiksliai apskaičiuoti, kiek vidutiniškai laiko užtruktų atakos esant saugiausiems šifro parametrams t, kiekvienam m arba šifro ilgiui n. Iš formulės gaunama, kad kai $m = 9$, tai vidutinis šios atakos vykdymo laikas saugiausiam t parametrui yra 745 dienos arba

apytiksliai 2 metai, kai $m = 10$ atveju vidutinis atakos vykdymo laikos siekia net 963 mlrd. metų, o $m = 11$ - $2{,}15 \times 10^{35}$ metų. Jau kai $m = 9$ šią ataką įvykdyti asmeniniu kompiuteriu yra sudėtinga, o $m = 10$, bei $m = 11$ juo labiau, ir neįmanoma.

Toliau būtų įdomu bent apytiksliai įvertinti, kiek vidutiniškai laiko truktų ši ataka naudojant galingiausią šiuo metu pasaulyje superkompiuterį Fujitsu Fugaku. Remiantis Linpack etalonu[1] (angl. *benchmark*) yra nustatyta, kad šio superkompiuterio našumas šiuo metu yra $442\,010\,\frac{TFlop}{s}$. Tuo tarpu kompiuterio procesoriaus su kuriuo atlikti eksperimentai remiantis Geekbench 4 etalonu[2] yra $473{,}8\,\frac{GFlops}{s}$. Remiantis šiais našumais yra apytiksliai apskaičiuota, kiek laiko užtruktų realizuota ataka įvairiems šifro m parametrams, naudojant saugiausius t parametrus. Kai $m = 6$ Fugaku ataką atliktų per vidutiniškai $3{,}28 \times 10^{-6}\ ms$, $m = 7$ - $1{,}75 \times 10^{-4}\ ms$, $m = 8$ - $0{,}0882\ ms$. Tuo tarpu iš prieš tai atliktų skaičiavimų žinant kiek apytiksliai užtruktų atakos su didesniais parametrais, yra analogiškai nustatomi vidutiniai atakų laikai, kai $m = 9$ apie $1{,}15\ min$, $m = 10$ apie 1 milijoną ir 33tūkst. metų, o $m = 11$ apie $2{,}30 \times 10^{29}$ metų. Taigi skirtumas tarp eksperimente naudoto kompiuterio ir superkompiuterio, kad superkompiuteris gali labai lengvai įvykdyti atakas, kai parametras $m = 9$, tačiau net ir jam $m = 10$ ir ilgesni šifrai su saugiausiais t parametrais šiuo metu vis dar lieka neįveikiami. Eksperimentų ir detalesnės analizės pagrindiniai rezultatai taip pat yra pateikiami 3 lentelėje.

3 lentelė. Apibendrintosios informacijos aibės dekodavimo atakos tyrimo pagrindiniai rezultatai.

m	n	$t_{eksperimentinis}$	$laikas_{eksperimentinis}$	$t_{analizės}$	$laikas_{analizės}$	$laikas_{superkompiuterio}$
6	64	4	3,06395 ms	3,8552	3,929 ms	$3{,}28 \times 10^{-6}$ ms
7	128	6	163,4194 ms	6,2104	113,125 ms	0,000175 ms
8	256	11	82350,64 ms	10,921	93760,599 ms	0,08827 ms
9	512	-	-	20,342	2,04 metai	1,15 min
10	1024	-	-	39,183	$9{,}63 \times 10^{11}$ metų	1 033 071,16 metų
11	2048	-	-	76,866	$2{,}15 \times 10^{35}$ metų	$2{,}30 \times 10^{29}$ metų

5.3. MAŽO SVORIO KODO ŽODŽIŲ RADIMO ATAKOS TYRIMAS

Mažo svorio kodo žodžių radimo ataka buvo realizuojama programiniu būdu parašant šią ataką realizuojančią klasę. Ši klasė taip pat realizuoja aprašytąjį Attack interfeisą ir jo metodą

1 - https://www.top500.org/system/179807/
2 - https://gadgetversus.com/processor/intel-core-i7-10700f-gflops-performance/

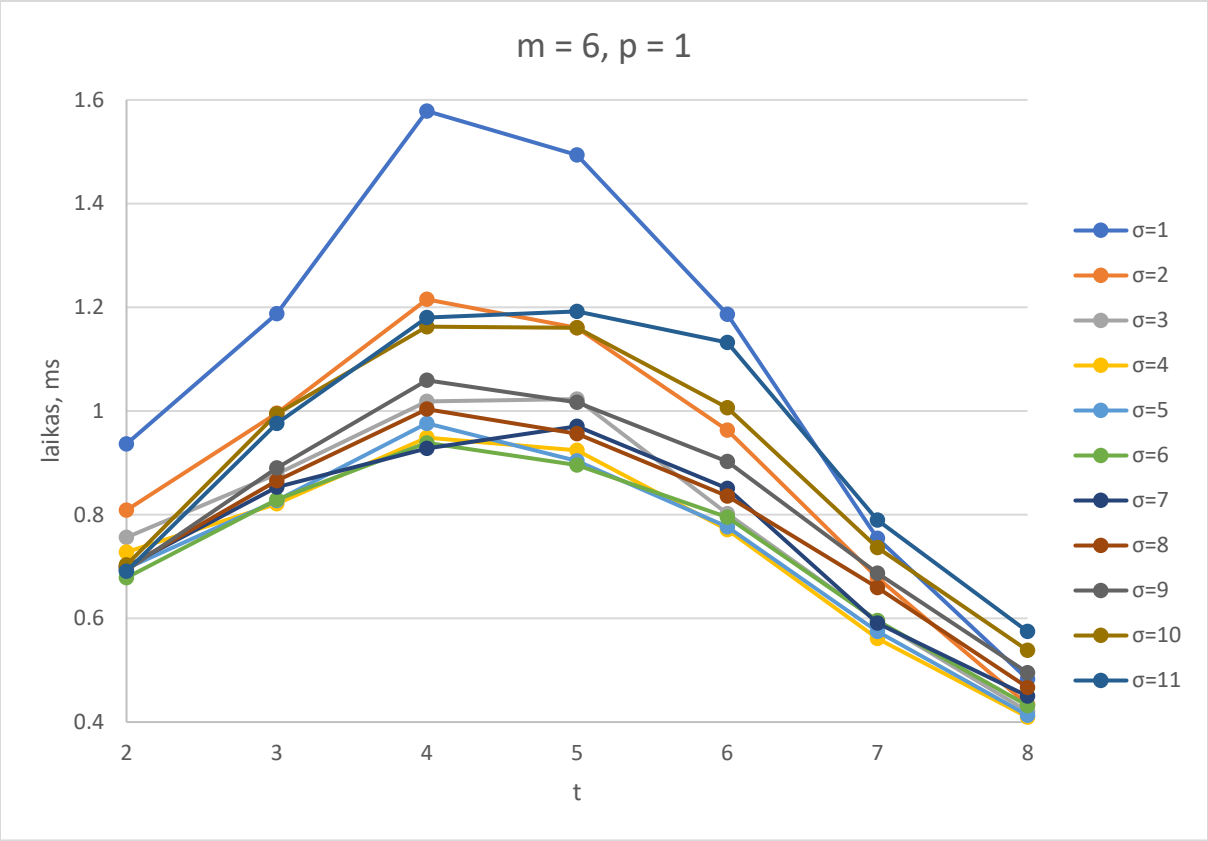
decrypt. Metodas decrypt ir realizuoja mažo svorio kodo žodžių radimo ataką. Taip pat klasė viešąjį raktą (G', t) ir atakos parametrus p ir σ gali gauti tiek per konstruktorių, tiek per seterius.

Šios atakos implementaciją buvo bandoma suprogramuoti dvejais panašiais būdais. Iš pradžių, pirmuoju būdu, parašytas atakos algoritmas, kuris ketvirtajame teoriškai aprašyto algoritmo žingsnyje (4.1 poskyris) vietoje po I aibės elementų atnaujinimo tik Z matricos atnaujinimo yra atnaujinama ne tik Z matrica, o visa informacijos aibės matrica G^*, t.y. matricos stulpeliai yra sukeičiami vietomis, o originali stulpelių tvarka yra saugoma papildomose aibėse. Ši originali tvarka panaudojama tam, kad gauti klaidos vektorių e su bitu reikšmėmis teisingose vietose. Stulpelių sukeitimas vietomis G^* matricoje padeda labai lengvai ir greitai programai gauti atnaujintą Z matricą.

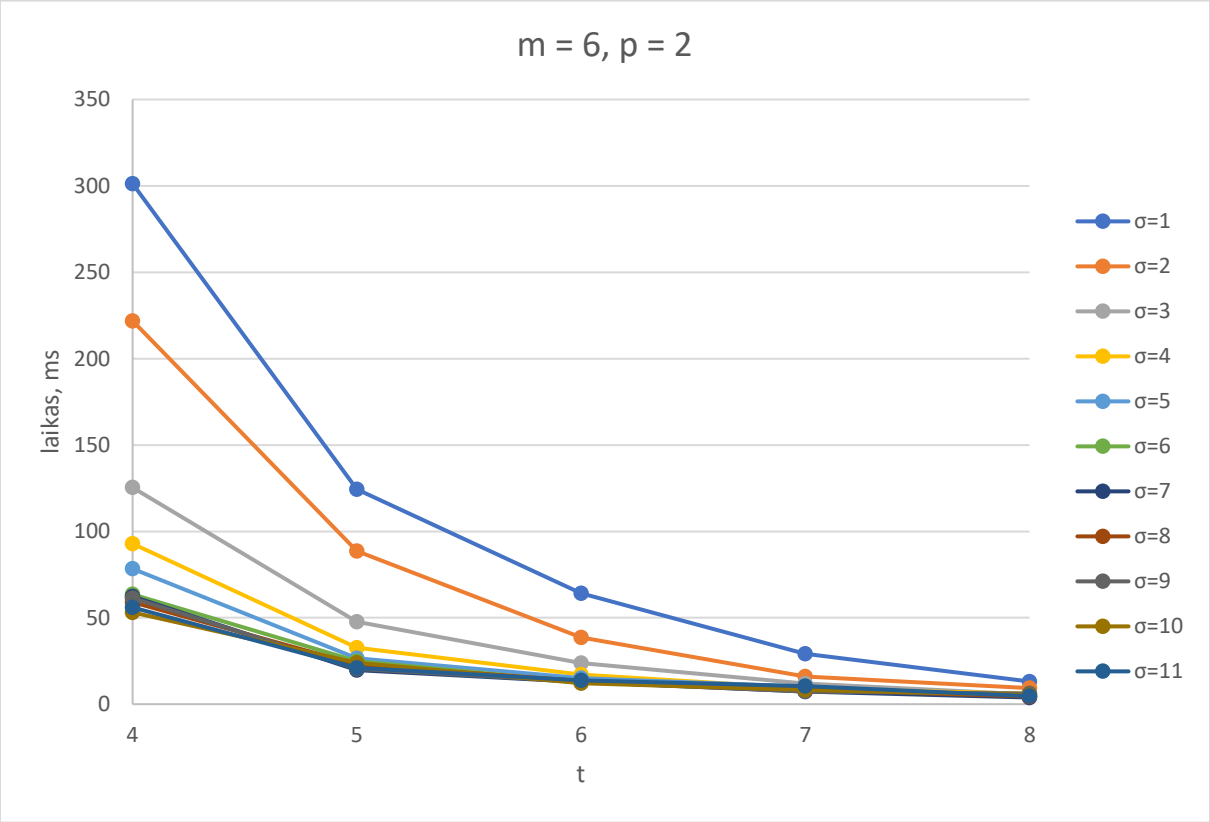
Vėliau buvo pagalvota, kad galbūt nereikėtų daryti G^* matricos stulpelių sukeitimų, o visus stulpelius palikti jų originaliuose vietose, kaip ir viešajame rakte G'. Tokiu būdu nereikia sekti originalių stulpelių vietų ir prieš gaunant e vektorių jame dar atkurti originalią bitų tvarką. Deja, nors pačios informacinės aibės Z atnaujinimas tikriausiai ir neužtrunka ilgiau, tačiau atnaujinus G^* matricą kiekvienos iteracijos metu iš jos išgauti Z matricą, naudojant GF2Matrix matricų programinę implementaciją patampa žymiai sudėtingesniu procesu ir užtrunka ilgesnį laiko tarpą, lyginant su pirmuoju variantu. Taip gaunasi, kad atakos algoritmas antruoju būdu yra lėtesnis negu pirmasis suprogramuotas variantas. Dėl šios priežasties buvo atsisakyta antrojo algoritmo varianto ir visi tyrimai atlikti naudojant pirmąją atakos algoritmo implementaciją.

Kadangi, ši mažo svorio kodo žodžių radimo ataka turi du atakos parametrus p ir σ, tai tyrimas kiek pasunkėja, todėl tyrimo metu stengiamasi apsiriboti tik efektyviausiais atakos parametrų rinkiniais. Čia taip pat atakos neatliekamos su McEliece kriptosistemos parametrais $m < 6$, nes atakos užtrunka per trumpai, bei $m > 8$, nes jos užtrunka per ilgai.

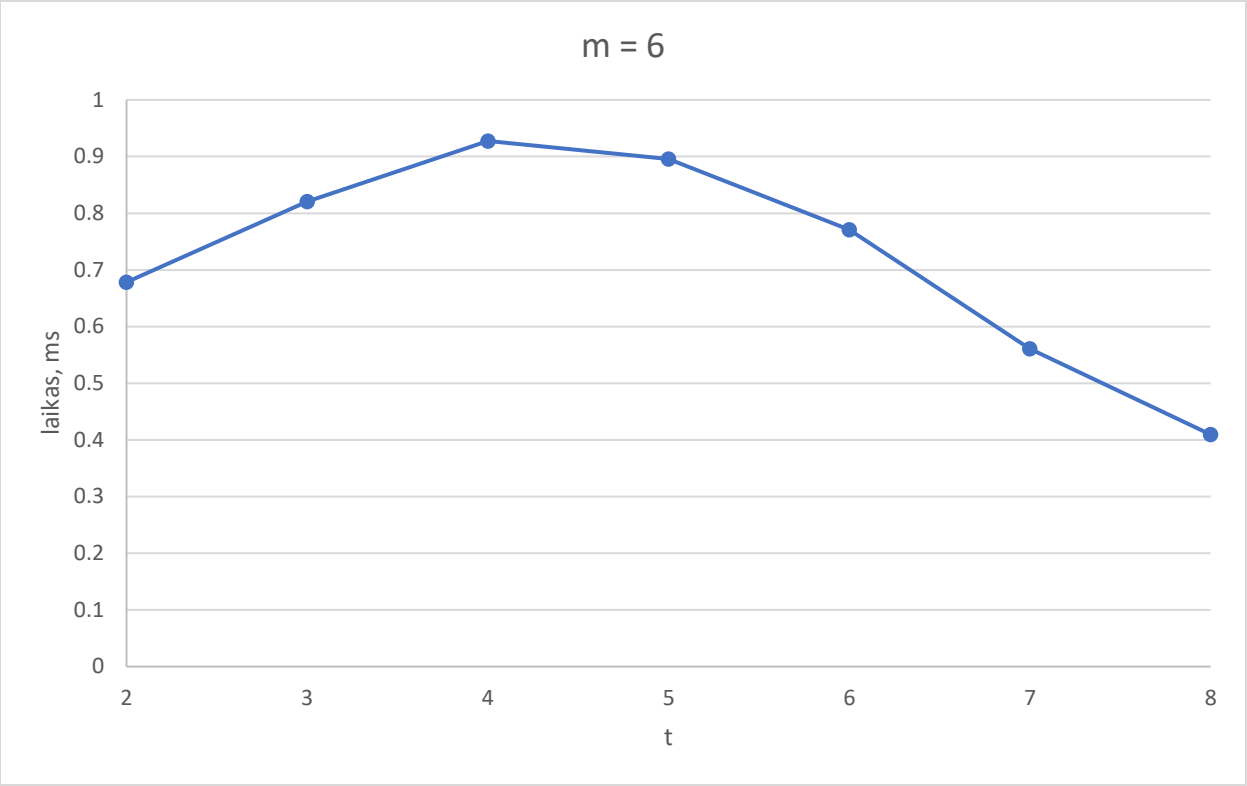
6 pav. nubraižytas grafikas su išmatuotais vidutiniais atakų laikais, kai McEliece kriptosistemos parametras $m = 6$, o atakos parametras $p = 1$, visiems galimiems atakos parametrams σ nuo 1 iki 11. Vienam grafiko taškui atidėti buvo atlikta 10000 bandymų. Tuo tarpu 7 pav. grafike jau pavaizduotos atakos atliktos su atakos parametru $p = 2$. Kadangi atakos truko ilgiau, tai vienam grafiko taškui atidėti jau buvo atliekama 500 bandymų. Jau iš kreivių grafike verčių matyti, kad pvz. su $t = 4$ atakos užtrunka bent apie 50 kartų ilgiau už atakas, kai $p = 1$. Iš čia matyti, kad parametro p parinkimas daro didelę įtaką atakos vykdymo laikui, žymiai didesnę negu parametras σ. Kadangi, $p = 1$ yra efektyviausias p atakos parametras, kai $m = 6$, tai toliau



6 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai m = 6, p = 1, pagal t.



7 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai m = 6, p = 2, pagal t.

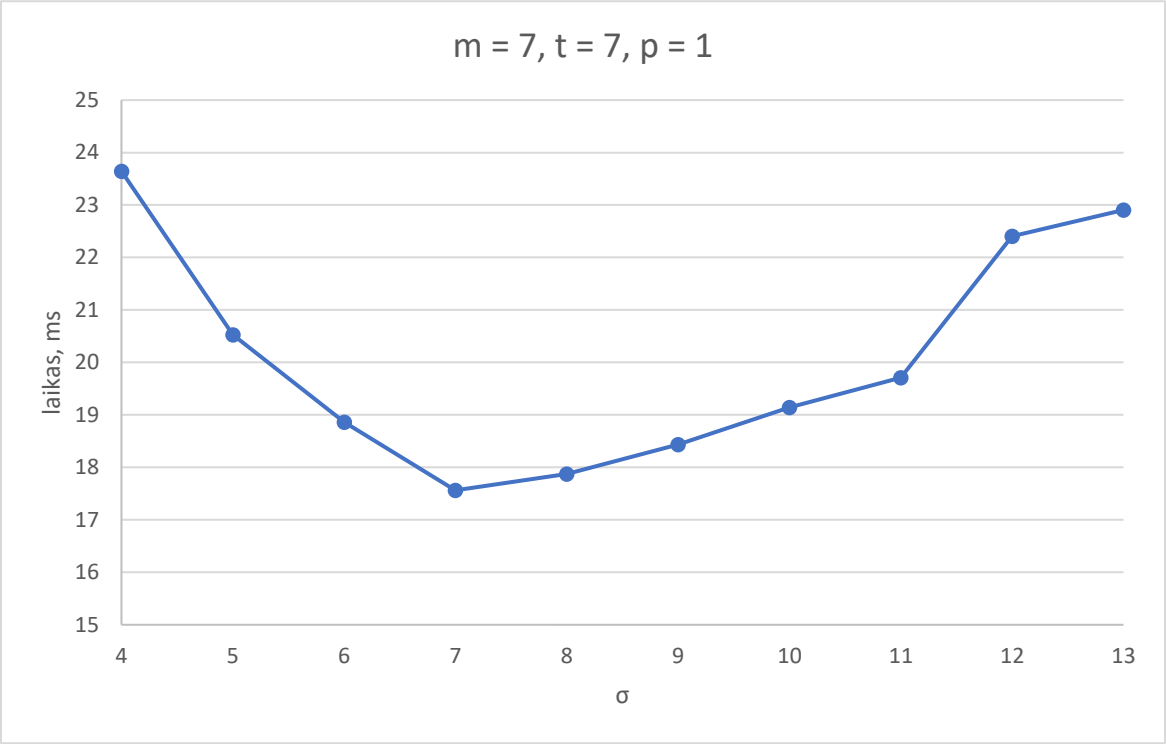


8 pav.Minimumų grafikas pagal 6 pav. grafiką.

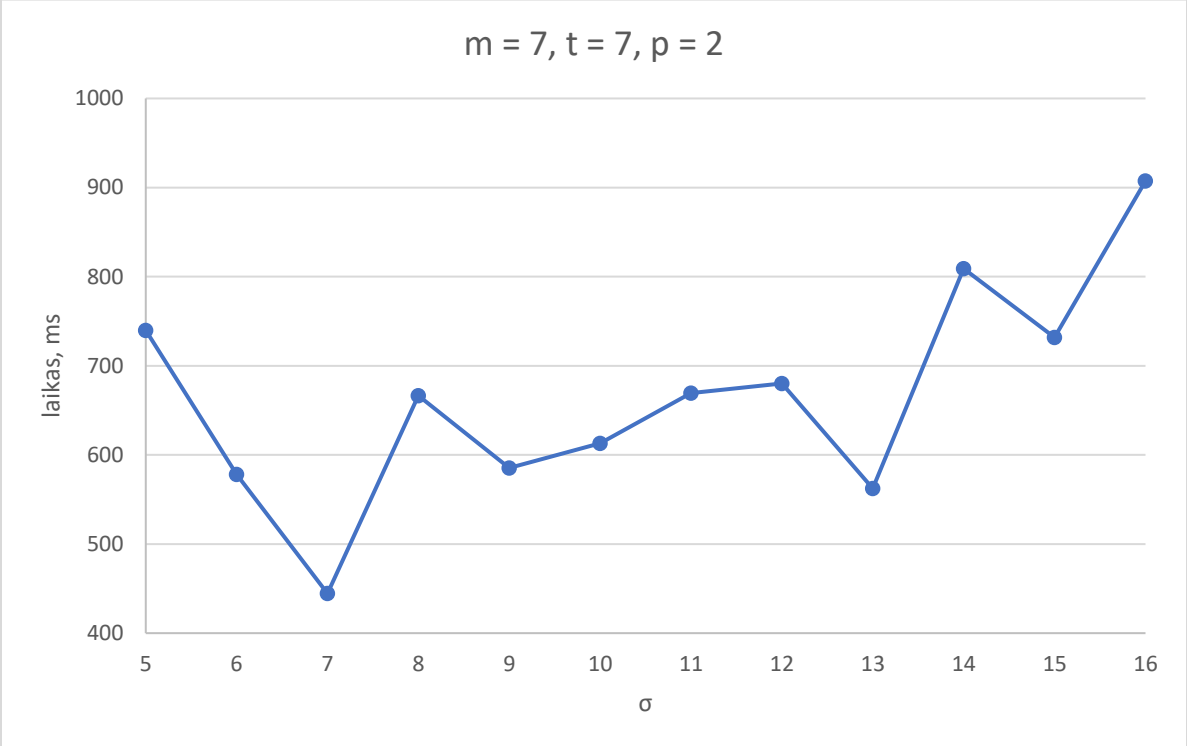
nagrinėjamas 6 pav. grafikas.

Tam, kad rasti šiai atakai saugiausią šifro t parametro vertę, kai $m = 6$ ir šiam tokiam saugiausiam šifrui efektyviausią atakos parametrą σ yra braižomas 8 pav. grafikas. Šiame grafike yra paimami 6 pav. grafiko kiekvienos t vertės minimalus išmatuotas taškas, t.y. taškas, kai su tuo šifro parametrų rinkiniu ataka užtruko trumpiausiai. Tai reiškia, kad kiekvienas naujojo grafiko taškas taškas parodo efektyviausių atakų laikus konkrečiam šifrui. Tuo tarpu šio grafiko maksimumas parodo, kuri McEliece kriptosistemos šifro t reikšmė yra saugiausia šifro parametrui $m = 6$. Iš grafiko matyti, kad kai $m = 6$, tai $t = 4$ yra saugiausias šifro parametras, o optimaliausi atakos parametrai yra $p = 1$, $\sigma = 7$ ir tokia ataka užtrunka vidutiniškai $0{,}9275\ ms$. Taigi šios atakos implementacija yra apie 3,3 karto greitesnė už analogišką apibendrintąją informacijos aibės dekodavimo ataką.

Kadangi ilgėjant šifrui daugėja ir atakos parametrų ir atakų laikas, tai stengiamasi atlikti atakos tyrimą tik su efektyviausiais parametrais. Dėl šios priežasties, remiantis toliau atliktais eksperimentais, pasirenkamas vienas saugiausių $m = 7$ šifro parametrų, kai kode padarytų klaidų skaičius $t = 7$ ir atliekami tyrimai su didesne efektyvių σ atakos parametrų aibe, esant kitam atakos parametrui $p = 1$ ir $p = 2$. Rezultatai pavaizduoti 9 pav. ir 10 pav. grafikuose. Bandymų skaičiai yra atitinkamai 1000 ir 100. Iš grafikų matyti, kad kai $p = 2$, atakos trunka žymiai ilgiau.



9 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai $m = 7$, $t = 7$, $p = 1$, pagal σ.

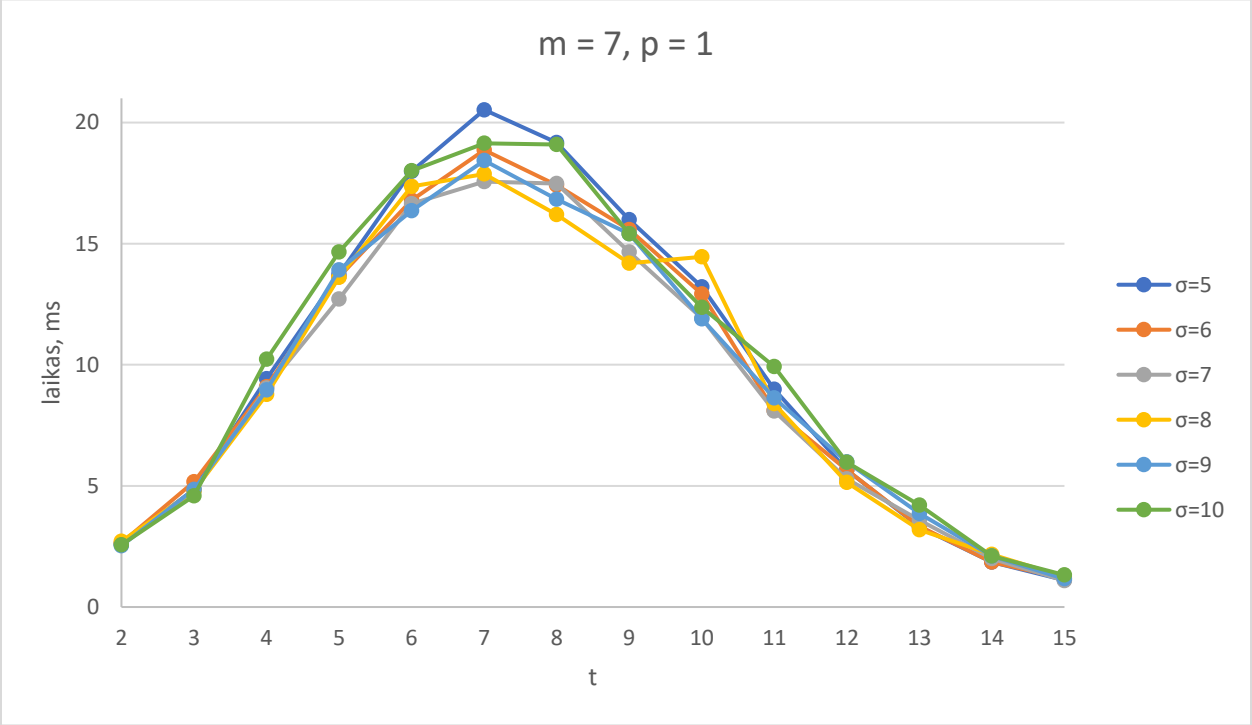


10 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai $m = 7$, $t = 7$, $p = 2$, pagal σ.

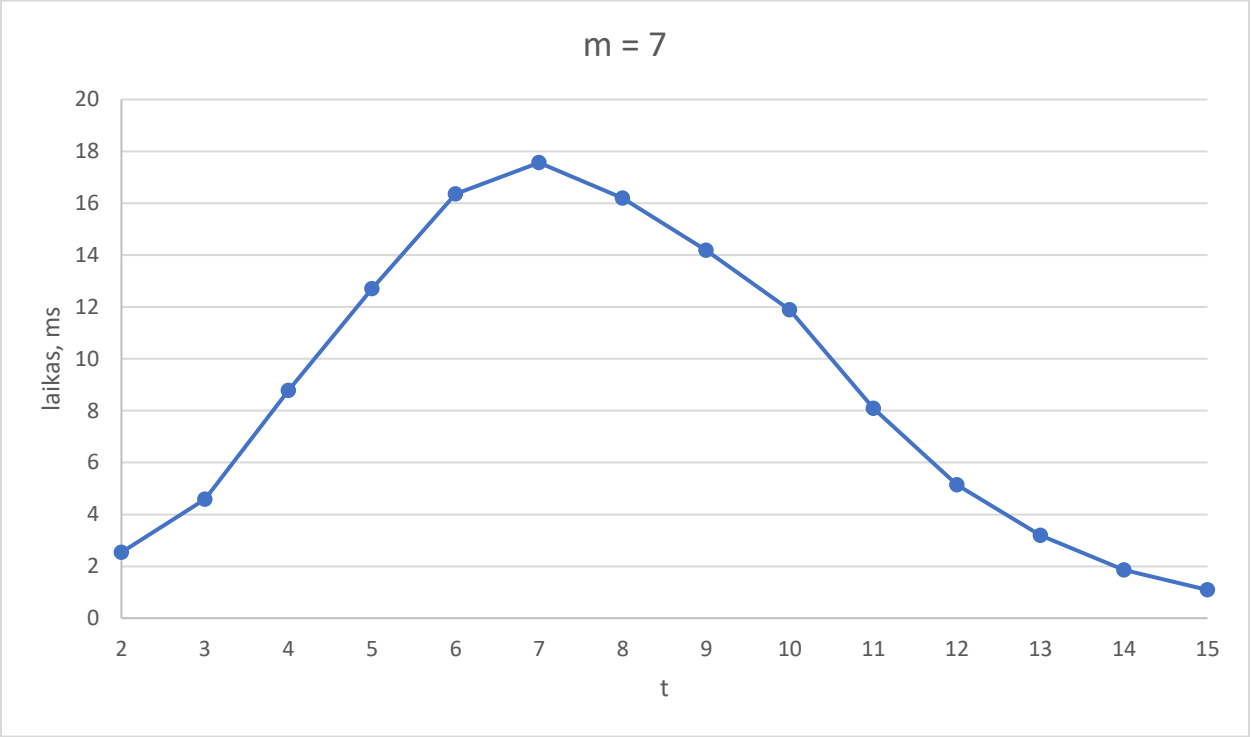
Dėl to toks variantas nenagrinėjamas. Tuo tarpu iš 9 pav. grafiko pasirenkami 6 σ atakos parametrai netoli kreivės minimumo ir su jais toliau atliekamas tyrimas.

11 pav. grafiko kreivėse ir pavaizduotas šis atliktas tyrimas. Bandymų skaičius čia yra 1000 vienam taškui atidėti. Iš čia matyti, kad kai $t = 7$ ir $t = 8$ atakos trunka ilgiausiai. Čia vėl analogiškai, kaip $m = 6$ atveju, kad nustatyti saugiausią šifrą ir efektyviausius atakos parametrus braižomas 12 pav. minimumų grafikas. Iš jo randama, kad saugiausias šifras $m = 7$ atveju yra, kai $t = 7$, o optimaliausi atakos parametrai prieš šį šifrą yra $p = 1$, $\sigma = 7$, o pati ataka vidutiniškai užtrunka 17,563 ms. Tai yra apie 18,94 karto lėtesnis rezultatas lyginant su $m = 6$, tačiau ši ataka yra net 9,3 karto greitesnė lyginant su $m = 7$ saugiausiu šifru ir optimaliausia ataka apibendrintosios informacijos aibės dekodavimo atakos.

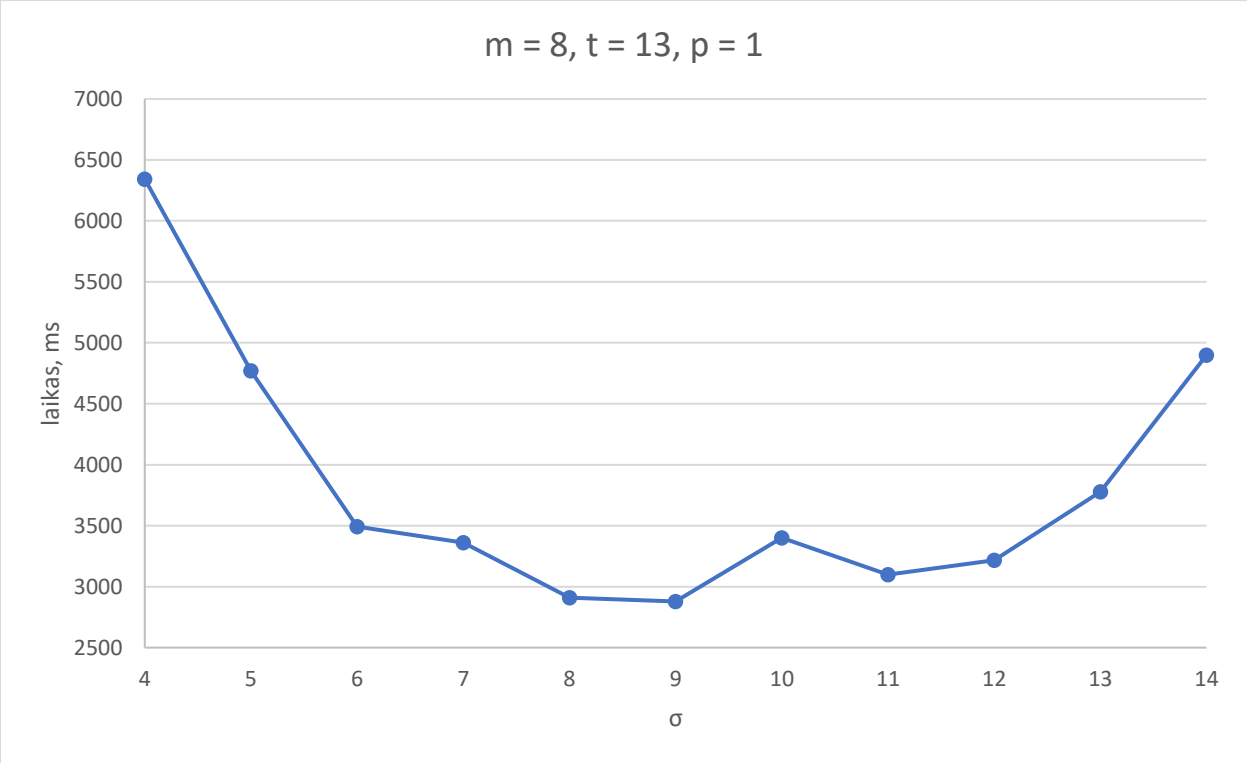
Toliau analogiškai tyrimas atliekamas $m = 8$ šiframs. Šįkart pasirenkamas šifras σ parametrams nustatyti yra su $t = 13$. Pirmiausiai, kai $p = 1$ ir kai $p = 2$ atitinkamai 13 pav. ir 14 pav. pavaizduoti vidutiniai atakų laiko grafikai su atitinkamais bandymų skaičiais 100 ir 5. Kadangi, 14 pav. grafike buvo atlikti tik 5 bandymai, grafikas nėra labai tikslus, bet jau ir iš to lyginant su $p = 1$ 13 pav. grafiku galima pastebėti, kad atakos trunka bent kelias dešimtis kartų il-



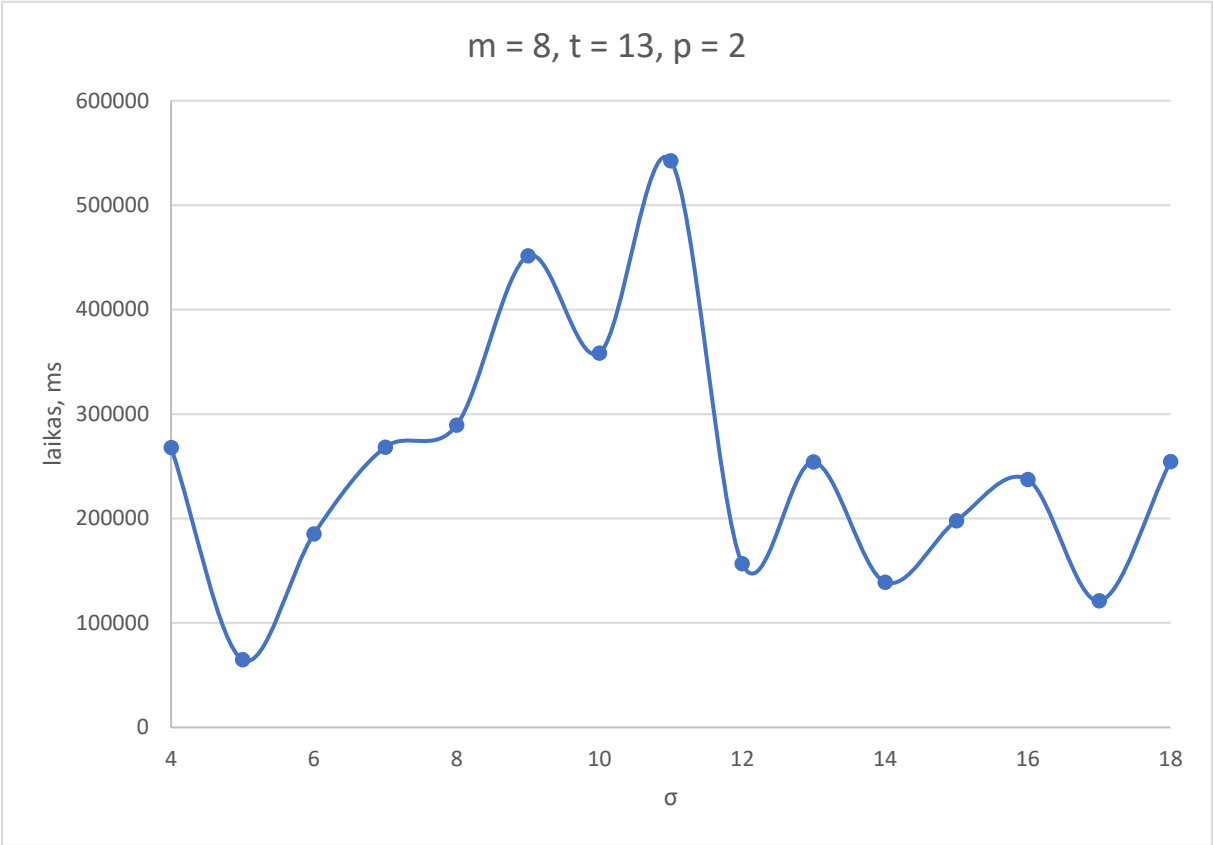
11 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai $m = 7$, $p = 1$, pagal t.



12 pav.Minimumų grafikas pagal 11 pav. grafiką.



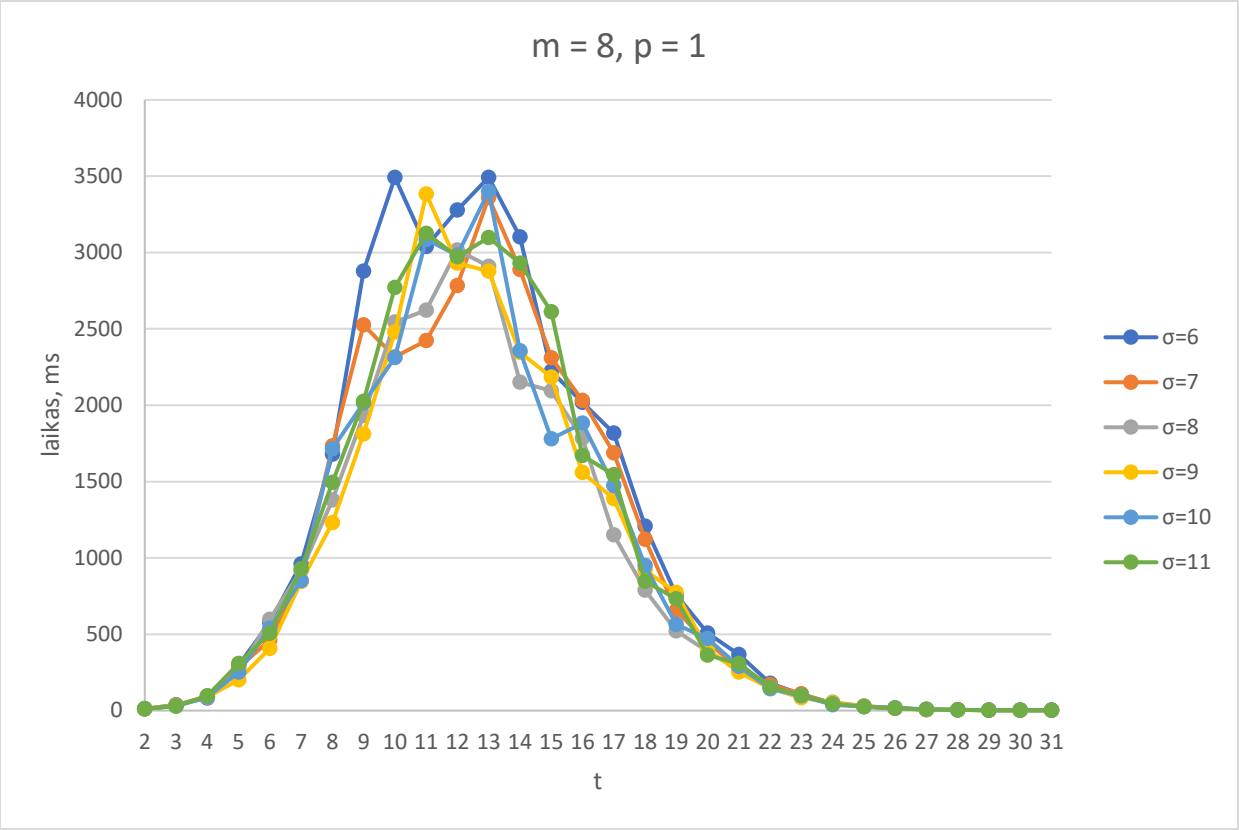
13 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai m = 8, t=13, p = 1, pagal σ.



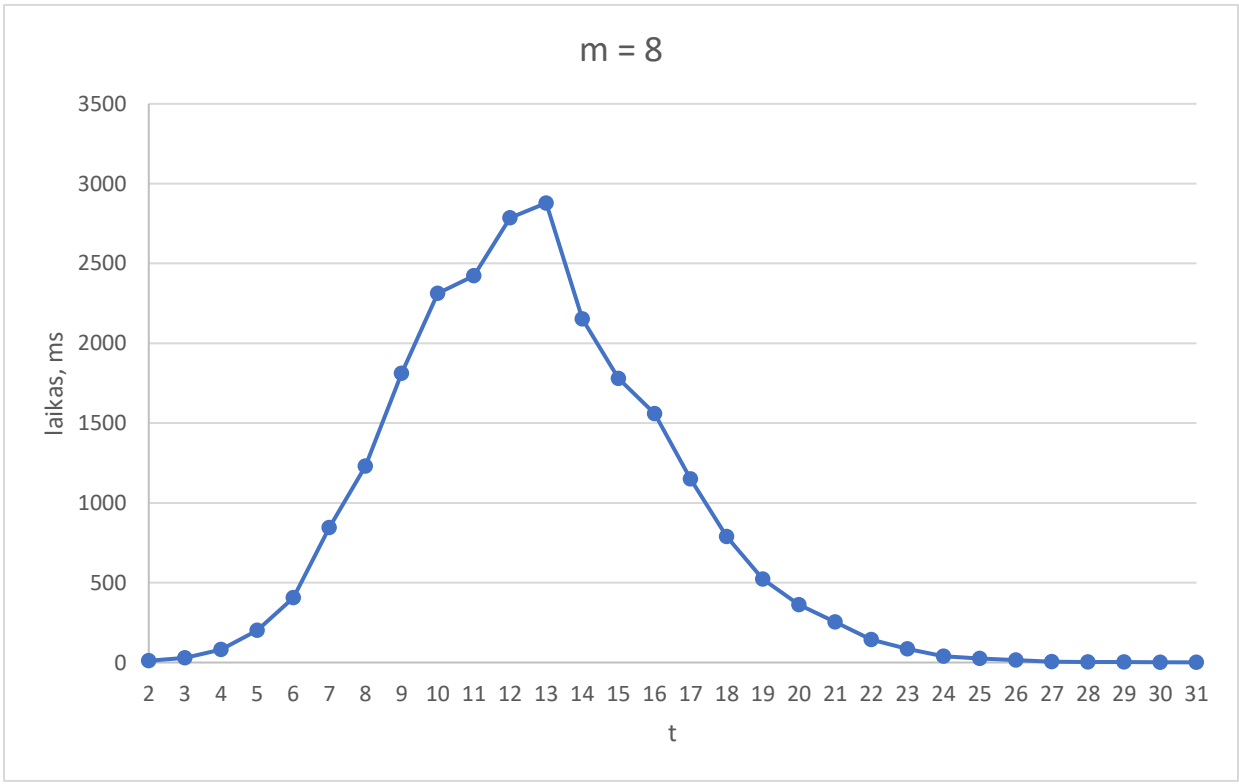
14 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai m = 8, t=13, p = 2, pagal σ.

giau. Dėl šios priežasties $p > 1$ atvejai nenagrinėjami. Vėl, šįkart aplink 13 pav. grafiko minimumo tašką pasirenkamos šešios σ vertės ir tyrimas tęsiamas jau su visais McEliece kriptosistemos $m = 8$ t parametrais.

15 pav. nubraižytas grafikas, kai $m = 8$ visoms t vertėms su šešiais pasirinktais σ parametrais. Bandymų skaičius kiekvieno taško grafike yra 100. Iš jo toliau galima atlikti analizę nustatant saugiausius šifro parametrus ir efektyviausius atakos parametrus. 16 pav. pavaizduoti tie patys duomenys tik kiekvienai $m = 8$ ir t porai paimta minimali laiko vertė. Iš šio grafiko maksimumo matyti, kad saugiausias šifro parametras prieš mažo svorio kodo žodžių radimo ataką, kai $m = 8$ yra $t = 13$. Grafiko maksimumas yra kartu ir efektyviausios atakos prieš saugiausią šifrą, kai $m = 8$ taškas. Šį tašką atitinką ir efektyviausi atakos parametrai prieš saugiausius parametrus yra $p = 1$ ir $\sigma = 9$, o pati ataka vidutiniškai užtrunka $2878{,}55\ ms$ arba $2{,}87855$ sekundžių. Ši ataka trunka apie 163,9 karto ilgiau negu su parametru $m = 7$, tačiau rezultatas yra apie 28,61 karto greitesnis negu analogiškas gautasis apibendrintosios informacijos aibės dekodavimo atakos rezultatas.



15 pav. Mažo svorio kodo žodžių radimo atakos grafikas, kai m = 8, p = 1, pagal t.



16 pav. Minimumų grafikas pagal 15 pav. grafiką.

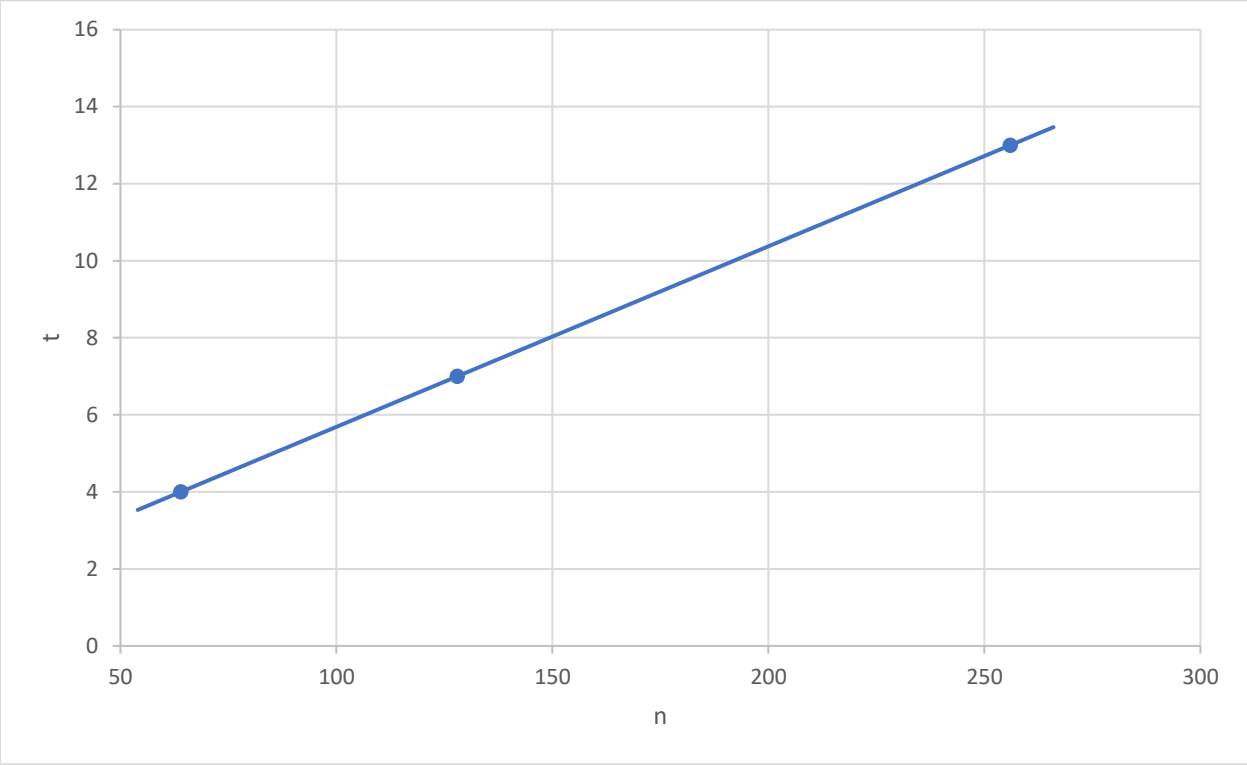
Iš šios atakos tyrimo grafikų matyti, kad mažo svorio kodo žodžių atakos užtrunka trumpiausiai, kai t yra didelis arba mažas. Tai galima paaiškinti dėl atakos algoritmo struktūros. Visų pirma, kai t yra mažas, tai Z matricos stulpelių skaičius yra žymiai mažesnis, todėl 4.1. poskyrio algoritmo 2 ir 3 žingsniai yra atliekami su mažesniais trumpesniais vektoriais, o veiksmai su trumpesniais vektoriais užtrunka trumpiau. Taip pat tokiu atveju yra ir pačių tiesinių kombinacijų daugiau, o su daug tiesinių kombinacijų surasti tokią porą, kurios svoris mažesnis (sąlyga $t - 2p$) yra paprasčiau. Tuo tarpu, kai t yra didelis tiesinių kombinacijų yra žymiai mažiau, dėl mažo Z matricos eilučių skaičiaus. Dėl šios priežasties algoritmui prireikia patikrinti mažiau tiesinių kombinacijų variantų per vieną iteraciją, galiausiai gaunama, kad ataka trunka trumpiau.

Kaip ir apibendrintosios informacijos aibės dekodavimo atakos atveju taip ir ši mažo svorio kodo žodžių radimo ataka buvo eksperimentiškai atlikta mažiems parametrams, o toliau yra atliekama analizė rasti saugiausius parametrus ir atakos laikus didesniems parametrams. Tam, kad nustatyti saugiausius atakos parametrus įvairiems šifro ilgiams n yra braižomas 17 pav. grafikas, kuriame atidėti mažo svorio kodo žodžių radimo atakos eksperimento metu rastų saugiausių atakos t parametrų priklausomybė nuo šifro ilgio n. Kadangi, kaip ir apibendrintosios informacijos aibės dekodavimo atakos atveju taip ir čia iš atidėtų taškų priklausomybė yra panaši į tiesinę, todėl pats grafikas yra aproksimuojamas tiesine priklausomybe, kurios lygtis:

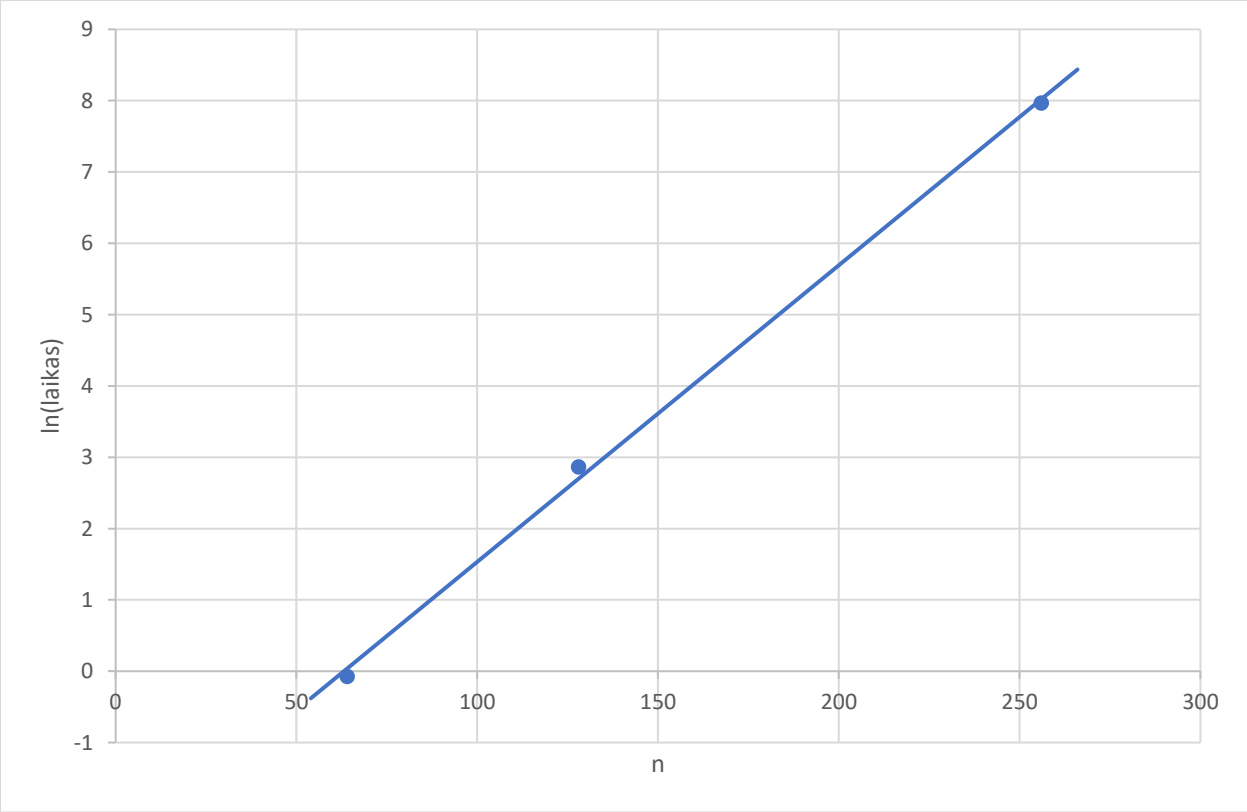
$$t = 0{,}0469n + 1. \tag{5.3.1}$$

Naudojantis šia lygtimi galima apytiksliai apskaičiuoti, kokie yra saugiausi šifro parametrai t, kiekvienam m arba šifro ilgiui n. Iš formulės gaunama, kad kai $m = 9$, tai saugiausias parametras $t = 25$, o kai $m = 10$, tai atitinkamai $t = 49$, kas yra labai artima McEliece pasiūlytiems $m = 10$ ir $t = 50$ parametrams. Tuo tarpu, kai $m = 11$, tai $t = 97$.

Analogiškai analizė atliekama nustatant kiek vidutiniškai laiko užtruktų mažo svorio kodo žodžių radimo atakos įvairiems šifro ilgiams n, naudojant saugiausius t parametrus. Kaip ir apibendrintosios informacijos aibės dekodavimo atakos atveju, taip ir čia iš atliktų eksperimentų matyti, kad didėjant šifro ilgiui n vidutinis atakos laikas saugiausiam t parametrui didėja labai greitai ir įgauna panašią į eksponentinę formą. Kad tuo įsitikinti apskaičiuojami eksperimente išmatuotų saugiausių parametrų vidutinių atakos vykdymo laikų natūralūs logaritmai, kurie pateikti 4 lentelėje ir 18 pav. grafike atidedama jų priklausomybė nuo šifro ilgio n. Čia vėl gautasis



17 pav. Saugiausių mažo svorio kodo žodžių radimo atakos šifro parametrų t priklausomybė nuo n.



18 pav. Saugiausių mažo svorio kodo žodžių radimo atakos saugiausių šifro parametrų vidutinių atakos laikų priklausomybė nuo n.

4 lentelė. 18 pav. grafiko duomenys.

m	6	7	8
n	64	128	256
laikas	0,9275	17,563	2878,55
ln(laikas)	-0,07526	2,865794416	7,965042

grafikas yra aproksimuojamas tiesine priklausomybe, kurios lygtis:

$$\ln(laikas) = 0{,}0416n - 2{,}6249. \tag{5.2.2}$$

Iš šios lygties galima rasti, kad kai $m = 9$, tai vidutiniškai ataka prieš saugiausią t parametrą turėtų trukti $35{,}79$ valandų arba beveik $1{,}5$ paros. Taigi su tokiais parametrais naudojant mažo svorio kodo žodžių radimo ataką vis dar yra įmanoma iššifruoti šifrą. Tuo tarpu, kai $m = 10$, ataka naudojant saugiausią šifro parametrą t, tai ataka užtruktų $7{,}268$ mln. metų, o $m = 11$ atitinkamai net $2{,}30 \times 10^{25}$ metų. Taigi šių atakų atlikti su asmeniniu kompiuteriniu šiuo metu tikrai neįmanoma.

Toliau remiantis 5.2. poskyryje paminėtais asmeninio kompiuterio ir Fujitsu Fugaku superkompiuterio našumais yra apytiksliai apskaičiuojama, kiek laiko truktų atakos su pastaruoju superkompiuteriu. Naudojantis eksperimento rezultatais: $m = 6$ atveju Fugaku saugiausią šifrą iššifruotų per $1{,}11288 \times 10^{-6}\, ms$, $m = 7$ per $1{,}59483 \times 10^{-5}\, ms$ ir $m = 8$ per $0{,}003275261\, ms$. Toliau remiantis analizės rezultatais nustatoma, kiek truktų atakos su didesniais parametrais. Kai $m = 9$ Fugaku atliktų ataką per vidutiniškai $138{,}1364863\, ms$. Taigi superkompiuteriui atlikti tokią ataką yra labai lengva. Tuo tarpu, kai $m = 10$, tai ataka apytiksliai užtruktų $7{,}79$ metų. Taigi superkompiuteriui įveikti tokį šifrą jau yra įmanoma su nagrinėjama JAVA implementacija, tačiau per gan ilgoką laiko tarpą - beveik 8 metus. Parašius greitesnę algoritmo implementaciją, pvz. naudojant kitą programavimo kalbą, galima būtų gauti ir mažesnius laikus. Toks šifras galėtų būti saugus nebent, kai nereikia laikyti užšifruotų duomenų daugelį metų. Deja, bet pagal išsikeltus saugumo kriterijus toks šifras nėra saugus. $m = 11$ atveju superkompiuterio ataka truktų $2{,}47 \times 10^{19}$ metų ir toks šifras jau gali būti laikomas saugiu.

Pagrindiniai mažo svorio kodo žodžių radimo atakos tyrimo rezultatai pateikti 5-oje lentelėje.

5 lentelė. Mažo svorio kodo žodžių radimo atakos tyrimo pagrindiniai rezultatai.

m	n	$t_{eksperimentinis}$	$laikas_{eksperimentinis}$	$t_{analizės}$	$laikas_{analizės}$	$laikas_{superkompiuterio}$
6	64	4	0,9275 ms	4,0016	1,038 ms	$1{,}113 \times 10^{-6}$ ms
7	128	7	17,563 ms	7,0032	14,878 ms	$1{,}594 \times 10^{-5}$ ms
8	256	13	2878,55 ms	13,0064	3055,504 ms	0,003275 ms
9	512	-	-	25,0128	35,797 h	138,136 ms
10	1024	-	-	49,0256	7 268 819 metų	7,792 metų
11	2048	-	-	97,0512	$2{,}30 \times 10^{25}$ metų	$2{,}47 \times 10^{19}$ metų

5.4. PAKARTOTINAI SIŲSTOS ŽINUTĖS ATAKOS TYRIMAS

Pakartotinai siųstos atakos implementacijai sukurti, buvo parašyta klasė realizuojanti šią ataką. Ši klasė kaip ir kitų atakų klasės taip pat implementuoja Attack interfeisą ir realizuoja jos abstraktų metodą decrypt. Šiame metode ir yra praktiškai realizuotas pakartotinai siųstos žinutės atakos algoritmas. Ši ataka skiriasi nuo prieš tai nagrinėtų, nes jai yra reikalinga pakartotinai siųstos žinutės sąlyga. T.y. reikia turėti ne tik žinutės šifrą, tačiau ir kitą tos pačios žinutės šifrą, kuris gautas naudojant tuos pačius raktus, tačiau kitą klaidos vektorių e. Tyrimo metu nėra nagrinėjama, kaip yra gaunamas šis šifras. Vietoje to tyrime yra užšifruojama ta pati žinutė du kartus naudojant tuos pačius raktus, bet skirtingus e vektorius ir vienas iš šių šifrų yra paduodamas ataką realizuojančiajai klasei kaip atakos parametras per konstruktorių arba per seterį.

Pats algoritmas yra realizuojamas, kaip aprašyta 4.3. poskyryje, lengvai randant stulpelius, kuriuose analogiškai e vektoriuje nėra padaryta klaidų. Toliau radus šiuos stulpelius iššifruoti pačią žinutę yra pritaikoma apibendrintosios informacijos aibės dekodavimo atakos (4.5. poskyris) algoritmo 2.1. žingsnis. Jeigu šis žingsnis nesėkmingas, tai pasirenkamas kitas stulpelių rinkinys iš L_0 aibės.

Tyrimo metu buvo pastebėta, kad ši realizuotoji ataka yra žymiai greitesnė už prieš tai dvi nagrinėtas atakas. Ši ataka tokia greita, kad net su asmeniniu kompiuteriu galima iššifruoti pakankamai greitai ir iki $m = 16$ sistemos šifrą, o ataka truktų 5,2 metų, o visi $m < 16$ trumpiau negu metus, taigi McEliece kriptografinė sistema yra nesaugi prieš šią ataką. Ir tai yra labai didelė silpnoji vieta. Taip pat buvo pastebėta, kad naudojant ilgesnius šifrus pats raktų generavimas ir žinutės užšifravimas užtrunka žymiai ilgesnį laiko tarpą negu trunka pati ataka.

Visos atakos buvo atliktos su McEliece kriptografinės sistemos šifrais nuo $m = 6$ iki įskaitant $m = 10$, kad būtų galima palyginti su prieš tai įvykdytomis kitomis dvejomis atakomis.

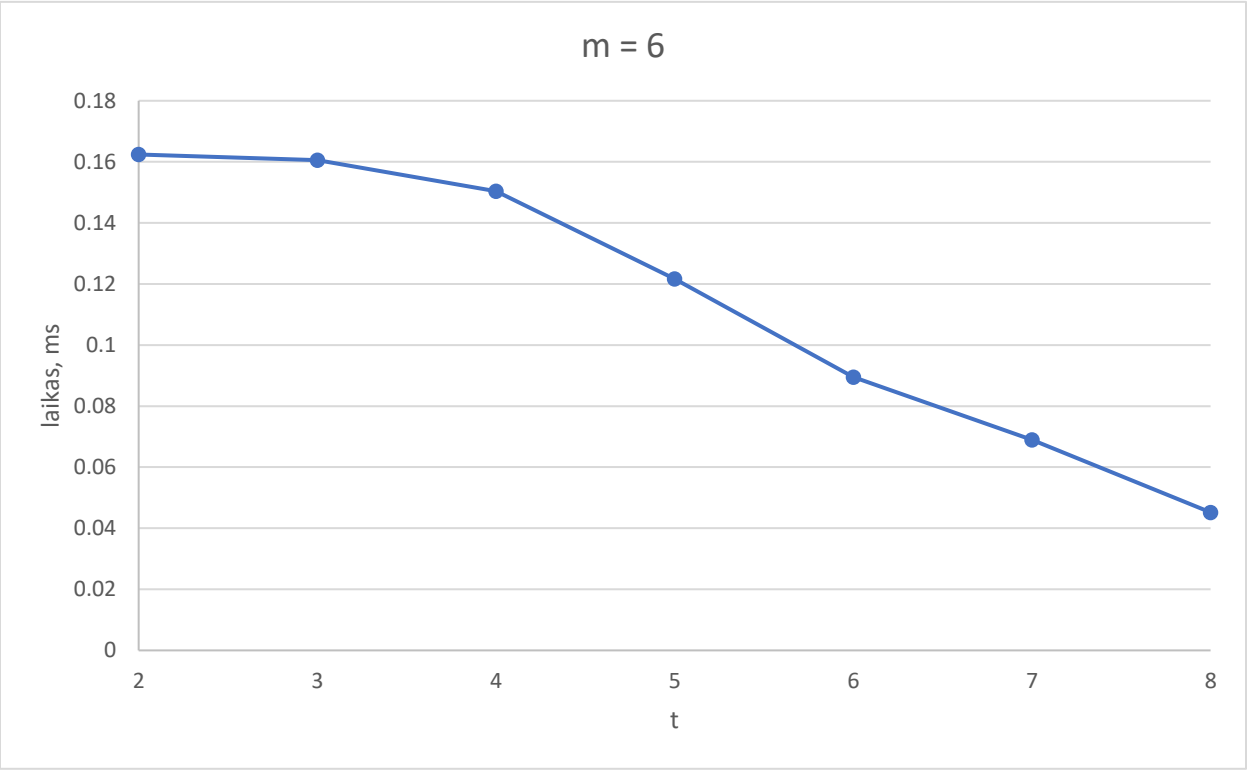
19 pav. grafike pavaizduotos pakartotinai siųstos žinutės atakos, kai $m = 6$. Vienam grafiko taškui atidėti buvo atlikta 10 tūkst. bandymų. Iš grafiko matyti, kad ataka ilgiausiais trunka $t = 2$. Tai yra saugiausias t parametras prieš pakartotinai siųstos žinutės ataką, kai $m = 6$ ir ataka trunka vidutiniškai $0{,}1624\ ms$. Šis rezultatas yra apie 18,87 karto greitesnis už apibendrintosios informacijos aibės dekodavimo atakos rezultatą ir apie 5,71 karto greitesnis už mažo svorio kodo žodžių radimo atakos atitinkamą rezultatą.

20 pav. grafike jau pavaizduotas atliktas tyrimas, kai $m = 7$. Čia vėl vienam grafiko taškui atidėti buvo atlikta 10 tūkst. bandymų. Iš grafiko matyti, kad ataka ilgiausiai trunka ir saugiausias parametras prieš šią ataką yra, kai $t = 4$. Tokia ataka trunka vidutiniškai $0{,}7615\ ms$. Gautasis laikas yra 4,69 karto lėtesnis už $m = 6$ atvejį, bet rezultatas yra 214,60 kartų greitesnis už analogišką apibendrintosios informacijos aibės dekodavimo atakos rezultatą ir apie 23,06 karto greitesnis už mažo svorio kodo žodžių radimo atakos atitinkamą rezultatą.

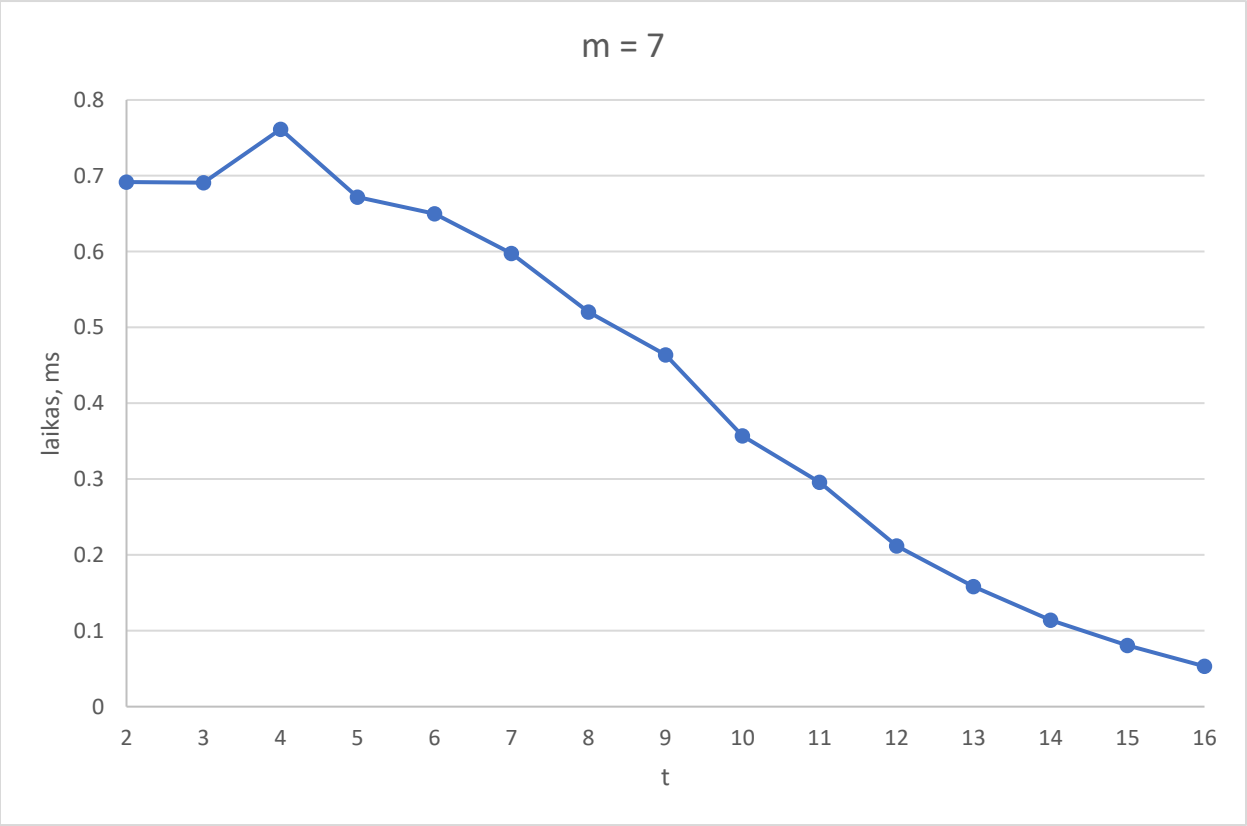
21 pav. grafike pavaizduotas atliktas tyrimas, kai $m = 8$. Čia vėl vienam grafiko taškui atidėti buvo atlikta 10 tūkst. bandymų. Iš grafiko matyti, kad ataka ilgiausiai trunka ir saugiausias parametras prieš šią ataką yra, kai $t = 7$. Tokia ataka trunka vidutiniškai $4{,}1994\ ms$. Gautasis laikas yra 5,51 karto lėtesnis už $m = 7$ atvejį, bet rezultatas yra 19 610,10 kartų greitesnis už analogišką apibendrintosios informacijos aibės dekodavimo atakos rezultatą ir apie 685,47 karto greitesnis už mažo svorio kodo žodžių radimo atakos atitinkamą rezultatą.

22 pav. grafike pavaizduotas atliktas tyrimas, kai $m = 9$. Čia jau šįkart vienam grafiko taškui atidėti buvo atlikta 1000 bandymų. Iš grafiko matyti, kad ataka ilgiausiai trunka ir saugiausias parametras prieš šią ataką yra, kai $t = 15$. Tokia ataka trunka vidutiniškai $38{,}684\ ms$. Gautasis laikas yra 9,21 karto lėtesnis už $m = 8$ atvejį, bet rezultatas šios atakos yra net 1 mlr.665mln. kartų greitesnis už analogišką analizės rezultatą apibendrintosios informacijos aibės dekodavimo atakos rezultatą ir apie 3 mln. kartų greitesnis už mažo svorio kodo žodžių radimo atakos atitinkamą analizės rezultatą.

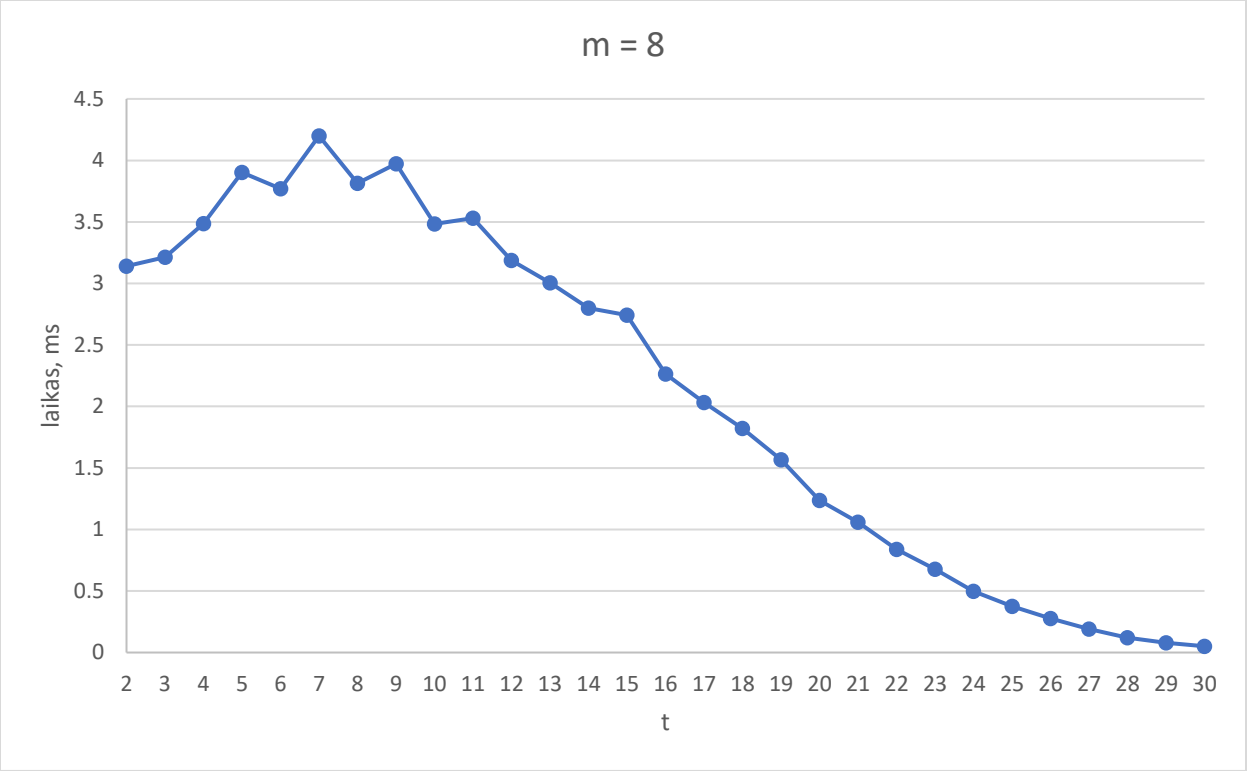
23 pav. grafike pavaizduotas atliktas tyrimas, kai $m = 10$. Čia taip pat vienam grafiko taškui atidėti buvo atlikta 1000 bandymų. Iš grafiko matyti, kad ataka ilgiausiai trunka ir saugiausias parametras prieš šią ataką yra, kai $t = 45$. Tokia ataka trunka vidutiniškai $352{,}391\ ms$. Gautasis laikas yra 9,11 karto lėtesnis už $m = 9$ atvejį, bet rezultatas šios atakos yra



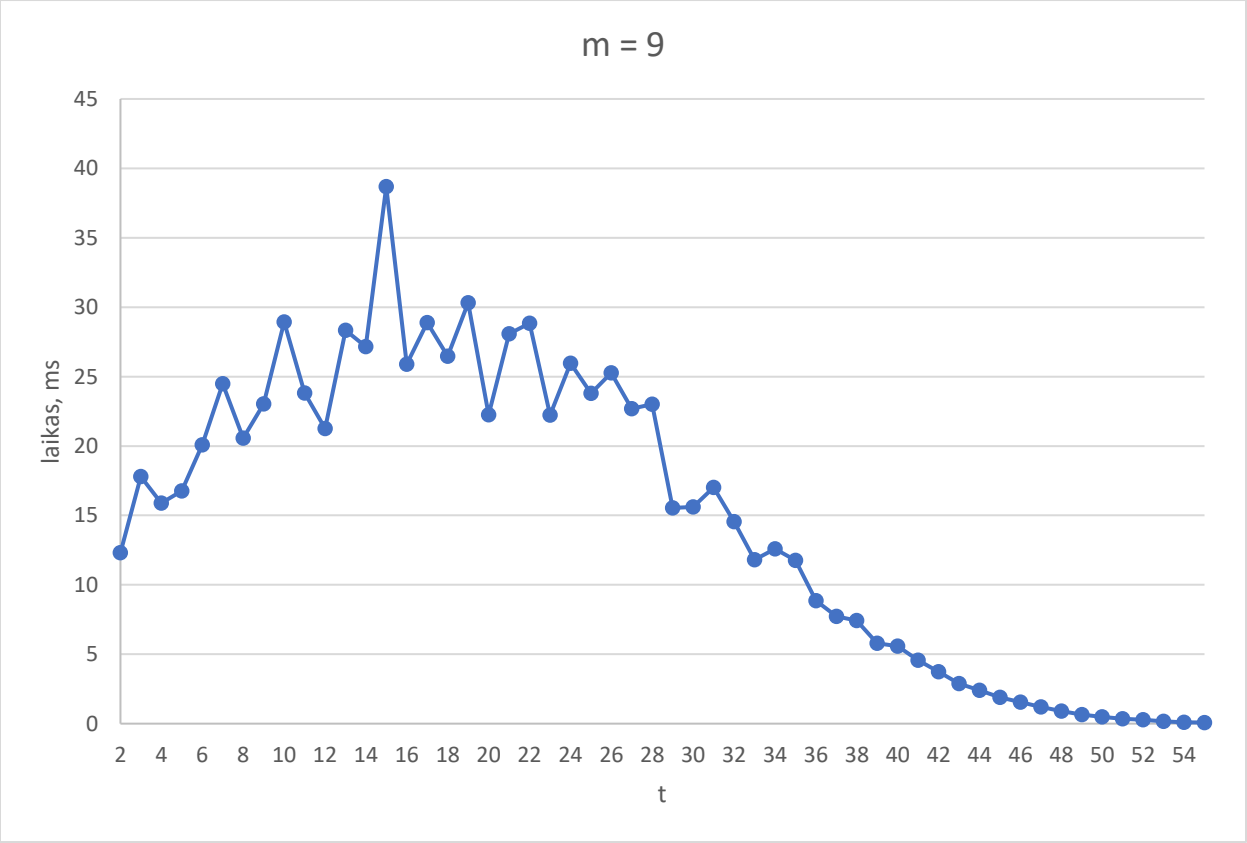
19 pav. Pakartotinai siųstos žinutės atakos grafikas, kai m = 6.



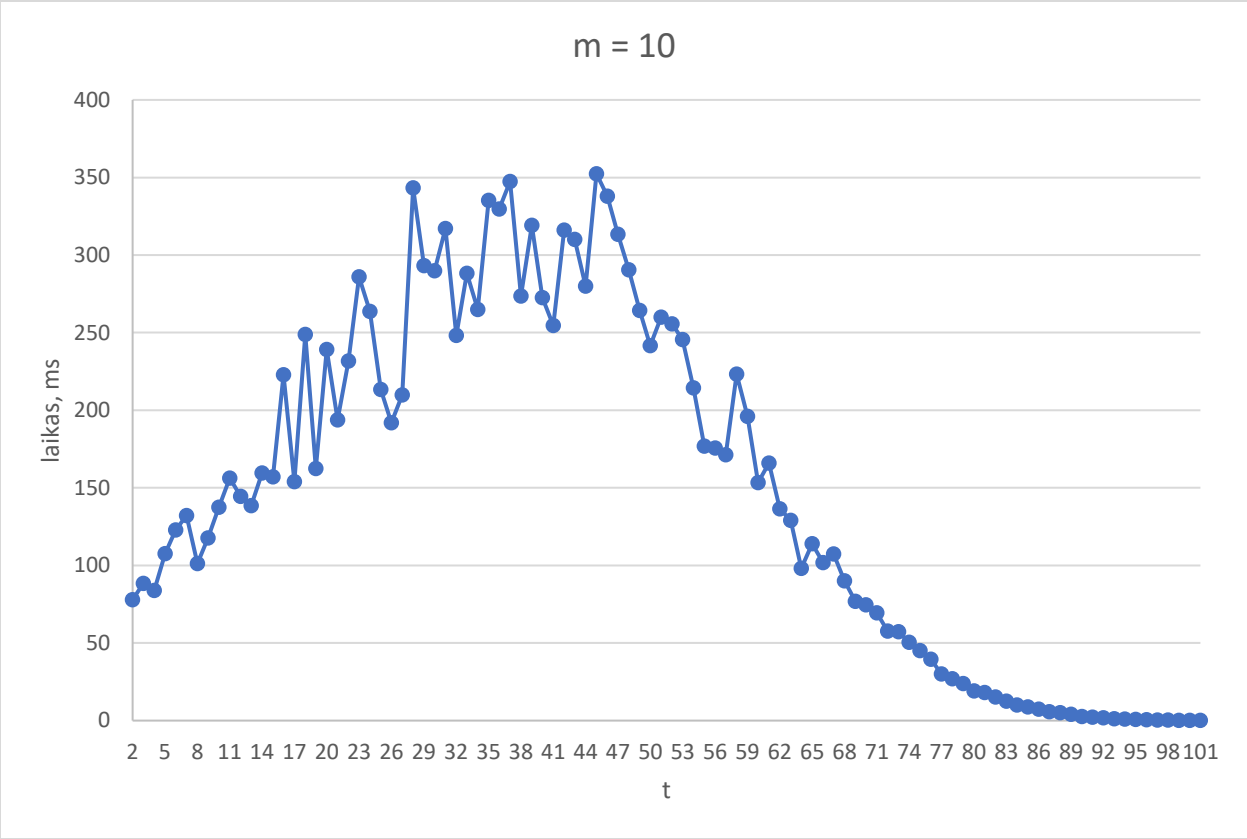
20 pav. Pakartotinai siųstos žinutės atakos grafikas, kai m = 7.



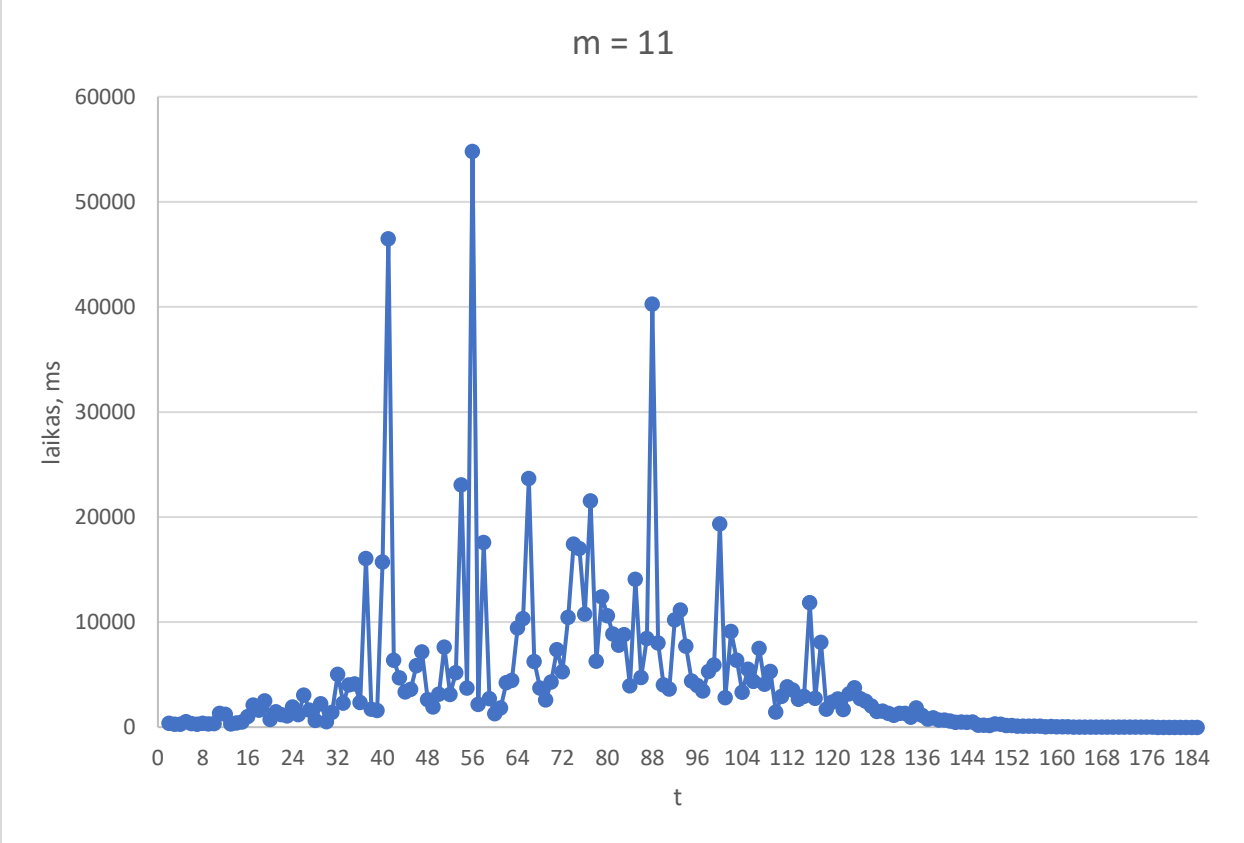
21 pav. Pakartotinai siųstos žinutės atakos grafikas, kai m = 8.



22 pav. Pakartotinai siųstos žinutės atakos grafikas, kai m = 9.



23 pav. Pakartotinai siųstos žinutės atakos grafikas, kai m = 10.



24 pav. Pakartotinai siųstos žinutės atakos grafikas, kai m = 11.

net $8{,}6248 \times 10^{19}$ kartų greitesnis už analogišką analizės rezultatą apibendrintosios informacijos aibės dekodavimo atakos rezultatą ir apie $6{,}50498 \times 10^{14}$ kartų greitesnis už mažo svorio kodo žodžių radimo atakos atitinkamą analizės rezultatą.

24 pav. grafike pavaizduotas atliktas tyrimas, kai $m = 11$. Vienam grafiko taškui atidėti buvo atlikta jau tik 50 bandymų, nes atakos užtruko žymiai ilgiau. Iš grafiko matyti, kad ataka ilgiausiai trunka ir saugiausias parametras prieš šią ataką yra, kai $t = 56$. Tokia ataka trunka vidutiniškai $54816{,}18\,ms$. Gautasis laikas yra 155,55 karto lėtesnis už $m = 10$ atvejį, bet rezultatas šios atakos yra net $1{,}23 \times 10^{41}$ kartų greitesnis už analogišką analizės rezultatą apibendrintosios informacijos aibės dekodavimo atakos rezultatą ir apie $1{,}32 \times 10^{31}$ kartų greitesnis už mažo svorio kodo žodžių radimo atakos atitinkamą analizės rezultatą.

Kadangi atlikti $m = 11$ šios atakos eksperimentus visoms t vertėms su 50 bandymų užtrunka ilgiau negu 24 valandas, nuspręsta eksperimentų su didesniais m parametrais toliau nebevykdyti.

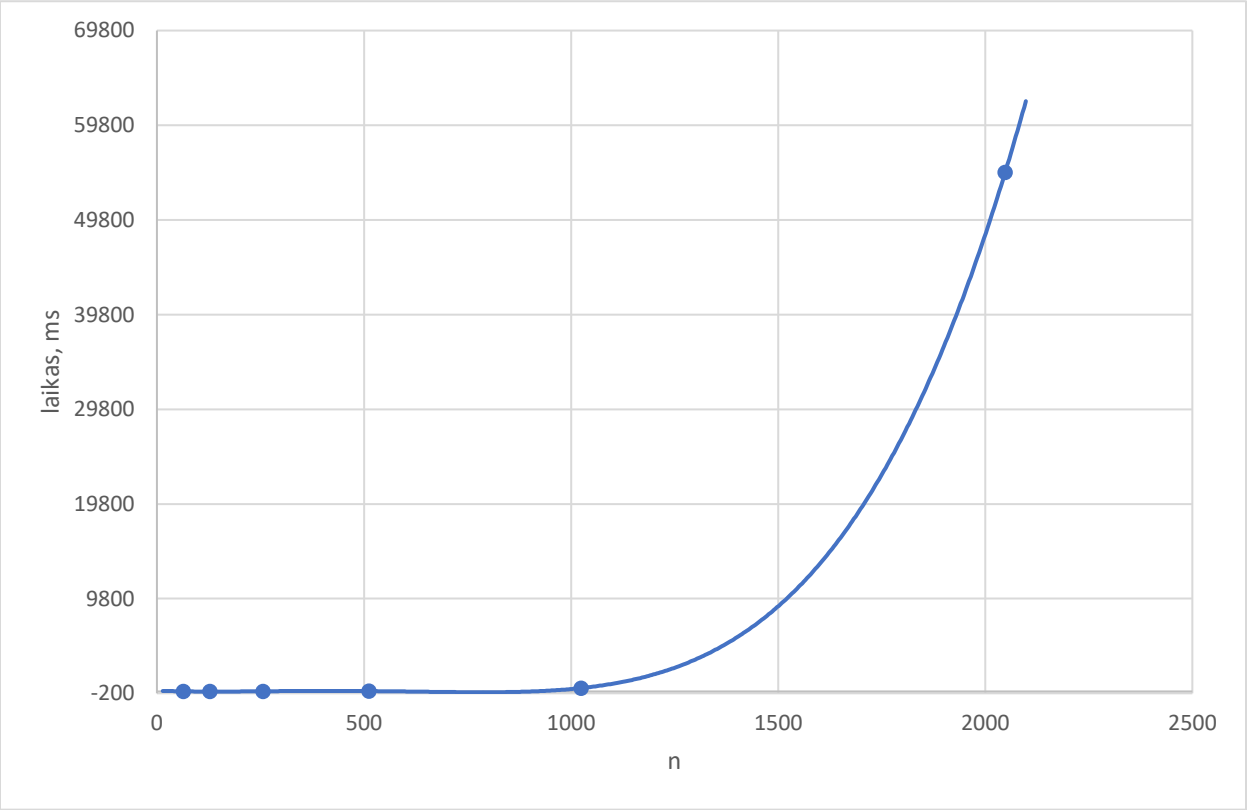
Iš šios atakos grafikų matyti, kad esant didelėms t vertėms atakos užtrunka žymiai trumpiau. Taip yra todėl, nes didėjant t vertei, k vertė sumažėja daug labiau negu padidėja t, dėl to atakos algoritmui reikia atspėti žymiai mažiau viešosios G' matricos stulpelių, kai atitinkamuose e vektoriaus koordinatėse nėra padaryta klaidų.

Toliau būtų įdomu sužinoti kokia priklausomybė sieja šifro ilgį n ir vidutinį atakos laiką prieš saugiausią to ilgio šifrą. Nubraižius 25 pav. grafiką matyti, kad tai ne tiesinė priklausomybė, tai taip negali būti ir eksponentinė priklausomybė, nes padvigubėjus šifro ilgiui ataka sulėtėtų žymiai labiau. Nuspręsta aproksimuoti gautąjį rezultatą 4-ojo laipsnio laipsnine funkcija, kuri pakankamai gerai atitinka gautąjį rezultatą, šios funkcijos lygtis yra:

$$y = 9{,}13611 \times 10^{-9}x^4 - 1{,}63698 \times 10^{-5}x^3 - 9{,}06977 \times 10^{-3}x^2 + 1{,}66797x + 81.9481. \quad (5.4.1)$$

Remiantis šia formule ir kituose atakos tyrimuose naudotomis našumo vertėmis galima apskaičiuoti kokia tūrėtų būti m vertė ir šifro ilgis n, kad sistema atitiktų iškeltus saugumo reikalavimus. Kai $m = 22$ ir atitinkamai šifro ilgis $n = 2^{22} = 4\,194\,304$ tada superkompiuteriui atlikti pakartotinai siųstos žinutės ataką prireiktų apie 96 metų, o asmeniniam apie 89 mln. metų. Aišku naudoti tokį šifro ilgį yra nepraktiška.

Tiek iš atakų tyrimo, tiek šios lygties ir grafiko galima pastebėti, kad tikrai didėjant šifro ilgiui n atakos trukmė ilgėja žymiai lėčiau negu apibendrintosios informacijos aibės dekodavimo atakos atveju, tiek mažo svorio žodžių radimo atakos atveju. Dėl to šią ataką galima atlikti net pakankamai dideliems parametrams su asmeniniu kompiuteriu ir ataka yra silpnoji McEliece kriptografinės sistemos vieta. Tačiau pačią ataką nėra visada įmanoma atlikti, nes jai reikalinga pakartotinai siųstos žinutės sąlyga. Dėl šios priežasties kriptosistemos silpnąją vietą galima panaikinti. Vienas iš pasiūlymų būtų tiesiog nesiųsti tos pačios žinutės. Kitas variantas, jeigu žinutę vis dėl to reikia išsiųsti pakartotinai, tai naudoti tą patį šifrą, tai pačiai žinutei išsiųsti. Tokiu būdu ataką norintis atlikti žmogus ar kompiuteris neturės galimybės atlikti šios atakos. 5.5. poskyryje nurodomas dar vienas, kitoks būdas.



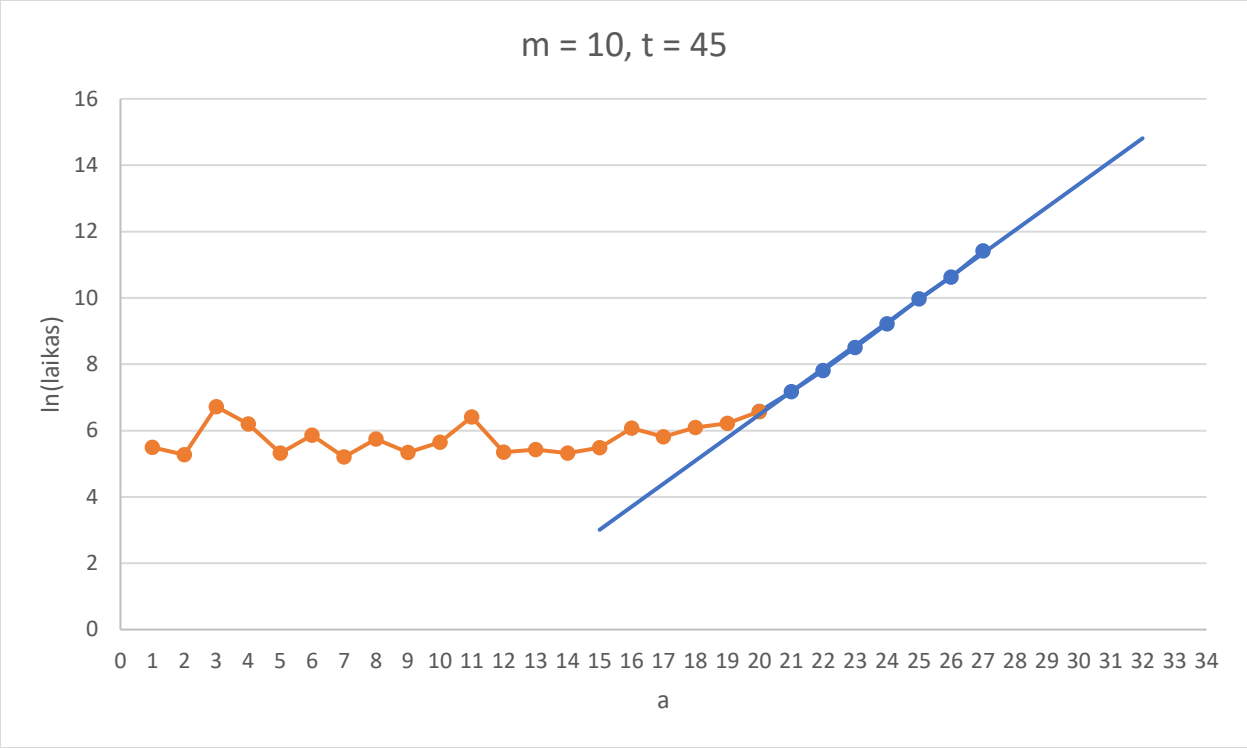
25 pav. Pakartotinai siųstos žinutės atakos saugiausių šifro parametrų vidutinių atakos laikų priklausomybė nuo n.

5.5. MCELIECE KRIPTOGRAFINĖS SISTEMOS MODIFIKACIJOS TYRIMAS

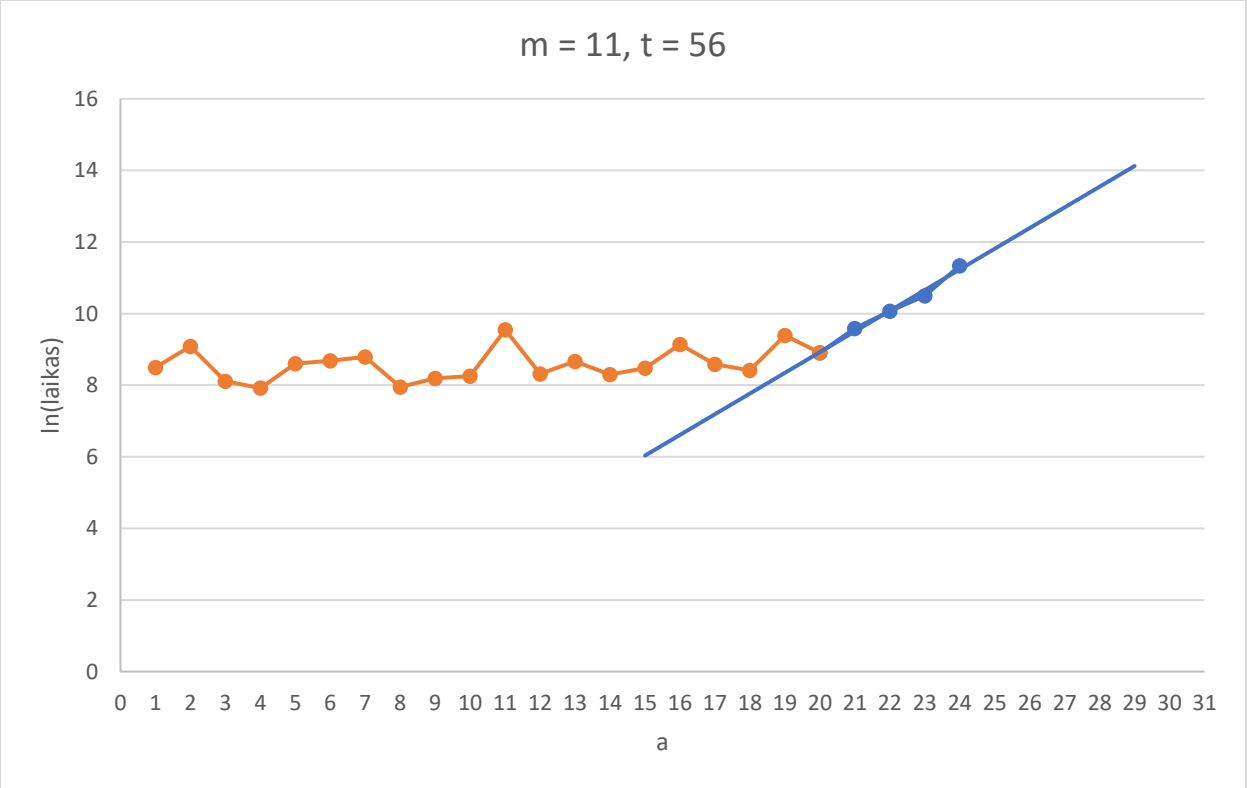
Iš pakartotinai siųstos žinutės atakos tyrimo buvo gauta, kad pakartotinai siųstos žinutės sąlyga yra silpnoji sistemos vieta. Tam, kad ją panaikinti yra pasiūloma McEliece kriptografinės sistemos modifikacija. Šios modifikacijos esmė yra, kad tam tikras žinomas kiekis bitų a žinutės pradžioje yra rezervuojami ir atsitiktinai sugeneruojami prieš šifravimą, šie atsitiktinai sugeneruoti bitai jokios naudingos informacijos neparodo, tuo tarpu žinutės informaciją parodo likusioji žinutės dalis. Tokiu būdu net siunčiant tą pačią informaciją du kartus iš eilės, bus užšifruojamos dvi skirtingos žinutės ir tokiai sistemai pakartotinai siųstos žinutės sąlyga negalios. Kadangi atsitiktinai sugeneruotų bitų skaičius a yra priskirtas prie viešojo rakto, tai šifro gavėjas dešifravęs šifrą gali lengvai gauti naudingą informaciją.

Nors ir prieš šią McEliece kriptosistemos modifikaciją ir negalima pritaikyti pakartotinai siųstos žinutės atakos, kai siunčiama ta pati informacija, tačiau pačios žinutės siunčiant tą pačią informaciją yra tarpusavyje susijusios. Dėl šios priežasties tokiai modifikacijai galima pritaikyti modifikuotą susijusių pranešimų ataką. Ši ataka yra modifikuota 4.7. poskyrio ataka. Kai siunčiama ta pati informacija naudojant, modifikuotą kriptosistemą, šią ataką reikia modifikuoti, nes žinoma, kad žinutės susijusios, tačiau nėra tiksliai žinoma pati susijusių pranešimų sąlyga $x_1 + x_2$. Akivaizdu, kad ten kur informacija sutampa reiškinio $x_1 + x_2$ bitų vertės bus nuliai, tuo tarpu pirmųjų a bitų vertės bus atsitiktinės, kaip ir žinučių x_1 ir x_2. Dėl šios priežasties reikia tikrinti visus 2^a $x_1 + x_2$ reiškinio galimus variantus, t.y. sugeneravus $x_1 + x_2$ galimą variantą tikrinama sąlyga $(c_1 + c_2 + (x_1 + x_2)SGP) \leq 2t$. Jeigu sąlyga galioja tada žinoma, kad $x_1 + x_2$ yra tikroji susijusių pranešimų sąlyga ir su ja toliau atliekama susijusių pranešimų ataka, kaip aprašyta 4.7. poskyryje.

Šiame tyrime buvo atlikta ši modifikuota ataka dviems saugiausiems šiframs gautiems iš pakartotinai siųstos žinutės atakos tyrimo, kai $m = 10$, $t = 45$ ir $m = 11$, $t = 56$, įvairiems a parametrams. 26 ir 27 pav. yra nubraižyti atakų laikų natūralių logaritmų priklausomybė nuo parametro a. Abiejų šifrų tyrime su ta pačia a verte atlikta po 100 skirtingų bandymų. Iš tyrimo grafikų matyti, kad pirmoji pusė grafiko iki maždaug $a = 20$ vertės yra pastovi, tuo tarpu didėjant parametrui a nuo $a = 20$ vertė pradeda augti tiesiškai logaritminiame grafike, o tai parodo, kad nuo tos vietos atakos laiko priklausomybė nuo parametro a tampa eksponentine. Šie eksperimentų



26 pav. Modifikuotos susijusių pranešimų atakos grafikas $m = 10, t = 45$.



27 pav. Modifikuotos susijusių pranešimų atakos grafikas $m = 11, t = 56$.

grafikai yra tokie, nes modifikuotos susijusių pranešimų atakos algoritmas susideda iš dviejų dalių. Iš susijusių pranešimų sąlygos paieškos ir susijusių pranešimų atakos algoritmo. Esant mažai a vertei susijusių pranešimų sąlyga yra randama žymiai greičiau negu užtrunka susijusių pranešimų ataka, o ji tiems patiems parametrams paprastai užtrunka tiek pat laiko, todėl grafike matoma pastovią priklausomybę. Tuo tarpu, a vertei pasiekus tam tikrą vertę susijusių pranešimų sąlygos paieška pradeda trukti ilgiau negu susijusių pranešimų ataka, todėl toje vietoje matome jau eksponentinę priklausomybę.

Yra randamos eksponentinių grafiko dalių tiesių priklausomybės. $m = 10$, $t = 45$ atveju gaunama priklausomybės lygtis:

$$\ln(laikas) = 0{,}6944x - 7{,}408, \tag{5.5.1}$$

o tuo tarpu $m = 11$, $t = 56$ lygtis:

$$\ln(laikas) = 0{,}5778x - 2{,}6331. \tag{5.5.2}$$

Remiantis šiomis lygtimis ir prieš tai naudotais našumo parametrais, galima nustatyti nuo kokios a vertės nagrinėti šifrai tenkins iškeltus saugumo reikalavimus.

Kai $m = 10$, $t = 45$, iškeltus saugumo reikalavimus tenkina nuo parametro vertės $a = 70$. Su tokiu parametru naudotas asmeninis kompiuteris atlikti modifikuotą ataką užtruktų 24 mln. 781 tūkst. metų, o superkompiuteris $26{,}56$ metų. Kadangi visas įmanomas užšifruoti žinutės ilgis yra su tokiais parametrais yra $k = 574$, tai iš jų 504 bitai gali būti skiriami naudingai informacijai.

Tuo tarpu, kai $m = 11$, $t = 56$, iškeltus saugumo reikalavimus tenkina parametrų vertės nuo $a = 76$. Su tokiu parametru asmeninis kompiuteris atliktų modifikuotą ataką per 26 mln. 837 tūkst. metų, o superkompiuteris $28{,}77$ metų. Kadangi visas įmanomas užšifruoti žinutės ilgis su tokiais parametrais yra $k = 1432$, tai iš jų 1356 bitai gali būti skiriami naudingai informacijai.

Taigi, iš šių abiejų šifrų tyrimų akivaizdu, kad modifikuota McEliece kriptografinė sistema gali sėkmingai panaikinti McEliece kriptosistemos pakartotinai siųstų žinučių silpnąją vietą ir su mažesniais McEliece kriptosistemos parametrais tenkinti išsikeltus sistemos saugumo reikalavimus.

6. PAGRINDINIAI REZULTATAI IR IŠVADOS

Šiame magistro baigiamajame darbe buvo atliktas McEliece kriptografinės sistemos saugumo tyrimas. Pirmiausia buvo atlikta su tyrimu susijusios teorijos ir literatūros apžvalga. Po to pasinaudota Bouncy Castle veiksmų su vektoriais ir matricomis funkcijų bibliotekomis ir parašyta programa, kuri realizuoja atakas nukreiptas prieš McEliece kriptosistemą: apibendrintąją informacijos aibės dekodavimo ataką, mažo svorio kodo žodžių radimo ataką, bei pakartotinai siųstos žinutės ataką. Pasinaudojant šiomis atakų implementacijomis buvo atlikti šių atakų tyrimai įvairiems McEliece kriptosistemos parametrams, nubraižyti grafikai, paskaičiuoti atakos laikai didesniems parametrams, bei atakos laikai jei jas atliktų superkompiuteris Fujitsu Fugaku, aptarti gautieji rezultatai, pasiūlyti būdai išvengti silpnųjų vietų. Taip pat pasiūlyta modifikuota McEliece kriptografinė sistema skirta išvengti pakartotinai siųstos žinutės silpnosios vietos ir atliktas tokios sistemos tyrimas.

Apibendrintosios informacijos aibės dekodavimo atakos tyrime gauta, kad kai $m = 6$ saugiausias šifras yra, kai $t = 4$ optimaliausias atakos parametras yra $j = 2$, o ataka trunka vidutiniškai $3.06395\ ms$, ant superkompiuterio truktų $3{,}28 \times 10^{-6}\ ms$. Analogiškai, kai $m = 7$, gautos vertės, kad $t = 6, j = 1$, o ataka trunka $163{,}4194\ ms$, ant superkompiuterio $0{,}000175\ ms$. O kai $m = 8$, tai $t = 11,\ j = 1$, o vidutinis atakos laikas $82350{,}64\ ms$, o ant superkompiuterio $0{,}8827\ ms$. Iš analizės rasta, kad kai $m = 9$, tai $t = 20$, o vidutinis atakos laikas $2{,}04$ metų, o ant superkompiuterio $1{,}15\ min$. Kai $m = 10$, tai $t = 39$, o vidutinis atakos laikas $9{,}63 \times 10^{11}$ metų, o ant superkompiuterio daugiau negu 1 mln. metų. Galiausiai taip pat rasta, kad kai $m = 11$ - $t = 77$, vidutinis atakos laikas $2{,}15 \times 10^{35}$ metų, o ant superkompiuterio $2{,}30 \times 10^{29}$ metų.

Mažo svorio kodo žodžių radimo atakos tyrime gauta, kad su parašytąja implementacija, kai McEliece šifro parametras $m = 6$, tai saugiausias šifras yra su klaidos parametru $t = 4$ optimaliausi atakos parametrai yra yra $p = 1$, $\sigma = 7$, o ataka trunka vidutiniškai $0{,}9275\ ms$, su superkompiuteriu truktų $1{,}113 \times 10^{-6} ms$. Atitinkamai, kai $m = 7$, nustatytos vertės yra $t = 7$, $p = 1$, $\sigma = 7$ ataka vidutiniškai trunka $17.563\ ms$, superkompiuterio atveju - $1{,}594 \times 10^{-5} ms$. Tuo tarpu, kai $m = 8$, tai $t = 13$, $p = 1$, $\sigma = 9$, o vidutinis atakos laikas yra $2878{,}55\ ms$, superkompiuterio - $0{,}003275\ ms$. Iš analizės rasta, kad kai $m = 9$, tai $t = 25$, o vidutinis atakos laikas $35{,}797$ valandų, o ant superkompiuterio $138{,}136\ ms$. Kai $m = 10$, tai $t = 49$, o vidutinis

atakos laikas daugiau negu 7 $mln.$ metų, o ant superkompiuterio 7,792 metų. Taip pat rasta, kad kai $m = 11$, tai $t = 97$, vidutinis atakos laikas $2{,}30 \times 10^{25}$ metų, superkompiuterio $2{,}47 \times 10^{19}$ metų.

Pakartotinai siųstos žinutės atakos tyrime gauta, kad naudojant parašytąją implementaciją, kai McEliece šifro parametras $m = 6$, tai saugiausias šifras yra su klaidos para parametru $t = 2$, o atakos vidutinis laikas $0.1624\,ms$. Atitinkamai gauta, kad kai $m = 7$, tai $t = 4$ ir laikas $0.7615\,ms$. $m = 8$ atveju gauta, kad $t = 7$ o vidutinis atakos laikas $4.1994\,ms$. $m = 9$ atitinkamos vertės $t = 15$, o laikas $38.684\,ms$. Kai $m = 10$ gauta $t = 45$ ir vidutinis atakos laikas yra $352.391\,ms$. Galiausiai, kai $m = 11$ - $t = 56$ ir vidutinis atakos laikas $54816{,}18\,ms$.

Buvo gauta, kad apibendrintosios informacijos aibės dekodavimo atakos ir mažo kodo žodžių radimo atakos optimaliausi laikai saugiausiems t parametrams didėjant šifro ilgiui auga eksponentiškai. Nustatyta, kad McEliece kriptosistemos šifras prieš abi šias atakas, naudojant saugiausius t parametrus, kai $m = 10$ yra saugūs, prieš tyrime naudojamą asmeninį kompiuterį. Tačiau, galingiausio superkompiuterio atveju naudojant mažo kodo žodžių radimo ataką tokie parametrai nebėra saugūs ir reikėtų naudoti parametrus, kai $m = 11$ arba didesnius.

Galiausiai tyrimo metu buvo nustatyta, kad McEliece kriptosistema yra labai nesaugi prieš pakartotinai siųstos žinutės ataką, naudojant bet kokius McEliece kriptosistemos šifrus, nes tik $m = 22$ ir didesnėmis vertėmis sistema gali tenkinti išsikeltus saugumo reikalavimus. Buvo pasiūlyti būdai, kaip sustiprinti sistemą, kad nebūtų galima atlikti šios atakos.

Modifikuotos McEliece kriptosistemos, skirtos išvengti pakartotinai siųstos žinutės atakos, tyrimo metu buvo atlikta modifikuota susijusių pranešimų ataka ir analizės metu nustatyta, kad šifras $m = 10$, $t = 45$ saugumo reikalavimus atitinka su modifikuotos sistemos parametru $a = 70$, o $m = 11$, $t = 56$ šifras su parametru $a = 76$.

LITERATŪRA

[BBC08] M. Baldi, M. Bodrato, F. Chiaraluce, "A New Analysis of the McEliece Cryptosystem Based on QC-LDPC Codes" *Security and Cryptography for Networks*, Lecture Notes in Computer Science, **5229**, Springer, Berlin, Heidelberg, 246-262 (2008).

[BBC13a] M. Baldi, M. Bianchi, F. Chiaraluce, "Optimization of the parity-check matrix density in QC-LDPC code-based McEliece cryptosystems", *Proceedings of the IEEE International Conference on Communications (ICC 2013) - Workshop on Information Security over Noisy and Lossy Communication Systems*, Budapest, Hungary (2013).

[BBC13b] M. Baldi, M. Bianchi, F. Chiaraluce, "Security and complexity of the McEliece Cryptosystem based on QC-LDPC codes" *IET Inf. Secur.* **7**, 212–220 (2013).

[BBC16] M. Bianchi, M. Baldi, F. Chiaraluce, "Enhanced Public Key Security for the McEliece Cryptosystem∗", *Journal of Cryptology* **29** (2016).

[BBC+14] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani, "Enhanced public key security for the McEliece cryptosystem*", Journal from cryptology* **29**, 1-27 (2014).

[BBM+13] M. Baldi, M. Bianchi, N. Maturo, F. Chiaraluce, "Improving the efficiency of the LDPC code-based McEliece cryptosystem through irregular codes", *Proceedings of the IEEE Symposium on Computers and Communications (ISCC 2013)*, Split, Croatia (2013).

[BEH18] M. Barakat, C. Eder, T. Hanke, "An Introduction to Cryptography" (2018).

[Ber97] T. A. Berson, "Failure of the McEliece Public-Key Cryptosystem under Message-Resend and Related-Message Attack", Burton S. Kaliski Jr. (Ed.), *Advances in Cryptology - CRYPTO '97, Lecture Notes in Computer Science* **1294**, 213-220, (1997).

[BMT78] E. Berlekamp, R. McEliece, H. van Tilborg, "On the inherent intractability of certain coding problems", IEEE Trans. Inform. Theory **24**, 384–386 (1978).

[Can96] A. Canteaut, "Attaques de Cryptosystèmes à Mots de Poids Faible et Construction de Fonctions t-Résilientes", PhD thesis, Université Paris VI (1996).

[CPB+17] S. Chaudhari, M. Pahade, S. Bhat, T. Sawant, C. Jadhav, "A Survey on Methods of Cryptography and Data Encryption" *Imperial Journal of Interdisciplinary Research* **3**, 11 (2017).

[CS98] A. Canteaut, N. Sendrier, "Cryptanalysis of the Original McEliece Cryptosystem" *Advances in Cryptology - ASIACRYPT '98*, International Conference on the Theory and

Applications of Cryptology and Information Security, Beijing, China, October 18-22, Springer **1514**, 187-199 (1998).

[DH76] W. Diffie, M. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory* **22**, 644-654 (1976).

[DK07] H. Delfs, H. Knebl, "Introduction to Cryptography Principles and Applications" Second Edition, Springer-Verlag, Berlin, Heidelberg (2007).

[EOS07] D. Engelbert, R. Overbeck, A. Schmidt, "A Summary of McEliece-Type Cryptosystems and their Security", *Journal of Mathematical Cryptology* **1**, 151-199 (2007).

[Goy12] S. Goyal, "A Survey on the Applications of Cryptography" *International Journal of Engineering and Technology* **2**, 3 (2012).

[Gop70] V. D. Goppa, "A New Class of Linear Correcting Codes", *Problems of Information Transmission* **6**, 207–212 (1970).

[Hei87] R. Heiman, "On the security of cryptosystems based on error correcting codes", Master's thesis, Feinberg Graduate School of the Weizman Institute of Science (1987).

[IH08] H. Imai, M. Hagiwara, "Error-correcting codes and cryptography", *Applicable Algebra in Engineering, Communication and Computing* **19**, 213-228 (2008).

[Jab01] A.A. Jabri, "A Statistical Decoding Algorithm for General Linear Block Codes" *Honary B. (eds) Cryptography and Coding. Cryptography and Coding 2001. Lecture Notes in Computer Science*, Springer **2260** (2001).

[Joc02] E. Jochemsz, "Goppa Codes & the McEliece Cryptosystem", Masters thesis, Vrije Universiteit Amsterdam (2002).

[KI03] K. Kobara, H. Imai, "On the one-wayness against chosen-plaintext attacks of the Loidreau's modified McEliece PKC", *IEEE Transactions on Information Theory*, **49**, 12, 3160 - 3168, (2003).

[Kir15] Z. Kirsch, "Quantum Computing: The Risk to Existing Encryption Methods" *Tufts University Computer Systems Security Computer Science* 116 (2015).

[Lou01] S. Loureiro, "Mobile Code Protection", PhD Thesis, ENST Paris / Institut Eurecom, Paris (2001).

[Mce77] R. J. McEliece, "The Theory of information and coding", *The Encyclopedia of Mathematics and Its Applications* **3** (1977).

[Mce78] R. J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory", *JPL DSN Progress Report* **42-44**, 114-116 (1978).

[Mol06] R. A. Mollin, "An introduction to cryptography", Second Edition (2006).

[Ple98] V. Pless, "Introduction to the Theory of Error-Correcting Codes", Third edition (1998).

[RZ14] M. Repka, P. Zajac, "Overview of the Mceliece Cryptosystem and its Security" *Tatra Mountains Mathematical Publications* **60** (2014).

[Ske20] G. Skersys, "Klaidas taisančių kodų teorija", Paskaitų konspektai, Vilniaus universitetas (2020).

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

[SSM+10] A. Shoufan, F. Strenzke, H.G. Molter, M. Stöttinger, "A Timing Attack against Patterson Algorithm in the McEliece PKC", *Lee D., Hong S. (eds) Information, Security and Cryptology – ICISC 2009. ICISC 2009. Lecture Notes in Computer Science*, Springer, **5984** (2010).

[Sta06] V. Stakėnas, "Kodai ir šifrai", Naujoji Vilnia (2006).

[Sti95] D. R. Stinson, "Cryptography: Theory and Practice", *CRC Press* (1995).

[Sun98] H. M. Sun, "Improving the Security of the McEliece Public-Key Cryptosystem" *Advances in Cryptology – ASIACRYPT '98*, International Conference on the Theory and Applications of Cryptology and Information Security, Berlin, Heidelberg, Springer **1514**, 200-213 (1998).

[UL10] V.G. Umana, G. Leander, "Practical key recovery attacks on two McEliece variants", *C. Cid, J.C. Faugère, (eds.) Proceedings of the 2nd International Conference on Symbolic Computation and Cryptography*, Egham, UK, 27-44 (2010).

[VDT02] E. Verheul, J. M. Doumen, H. C. A. van Tilborg, "Sloppy Alice Attacks! Adaptive Chosen Ciphertext Attacks on the McEliece cryptosystem", *Information, Coding and Mathematics*, Kluwer Academic Publishers, Boston, Massachusetts, 99-119, (2002).

[Woo10] L. Wood "The clock is ticking on encryption" *Computerworld*, Framingham, US, (2010).