VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

McEliece viešojo rakto kriptografinės sistemos saugumo tyrimas

Study of the security of McEliece public-key cryptosystem

Magistro baigiamasis darbas

Atliko: Tomas Vyčas *(parašas)*

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

Recenzentas: lekt. Irmantas Radavičius *(parašas)*

Vilnius – 2016

Santrauka

Darbo tema – viešojo rakto McEliece kriptografinės sistemos saugumo tyrimas. Darbe pateikta kriptografijos temų apžvalgą, klaidas taisančių kodų taikymą kriptografijoje ir McEliece kriptografinės sistemos apžvalgą. Taip pat pateikta galimų atakų prieš viešojo rakto kriptografinių sistemų klasifikavimą. Tyrimo metu išanalizuotos McEliece kriptografinės sistemos sudedamosios dalys, apžvelgta jų įtaka bendram sistemos saugumui. Darbe aprašytos ir realizuotos atakos prieš McEliece kriptografinę sistemą. Taip pat tyrime pateikta optimalūs t parametrai, atakų vidutiniai vykdymo laikai ir apskaičiuota kiek apytiksliai truktų kiekviena ataka su didesniais parametrais m ir t, kurie yra laikomi pakankamai saugiais. Pagal gautus rezultatus yra prognozuotas vykdymo laikas superkompiuteriui ir pateikti saugus kriptografinės sistemos parametrai.

Raktiniai žodžiai: McEliece, kriptografinė sistema, ataka, šifras, šifravimas, dešifravimas, viešasis raktas, privatus raktas, kriptoanalitikas, pranešimas, kriptografinės sistemos saugumas, kriptografija, apibendrinta informacijos dekodavimo ataka, žinomo dalinio teksto ataka, pranešimų persiuntimo ataka, susijusių pranešimų ataka.

Summary

The paper topic is security of the public key of the McEliece cryptographic system. The review of the cryptography topics, the error-correcting codes applied in the cryptography and the review of the McEliece cryptographic system are presented. Possible attacks against the classification of the cryptographic systems of the public key are also described. During the research, components of the McEliece cryptographic system are analysed as well as their influence to the overall security of the system are reviewed. In the paper, attacks against the McEliece cryptographic system are described and realised. In addition to this, optimal t parameters as well as average duration of different attacks are presented. In this paper there is also calculated how long approximately every attack would take with the bigger m and t parameters, which are considered to be sufficiently secure. According to the results execution time for supercomputers to provide a secure cryptographic system parameters are predicted.

Key words: McEliece, cryptographic system, attack, cypher, encryption, decryption, public key, private key, cryptanalysts, message, cryptographic system security, cryptography, generalized information set decoding attack, known partial plaintext attack, related message attack, message resend attack.

Turinys

Įvadas 6
1. Sąvokų žodynėlis 8
2. Kriptografijos temų apžvalga 9
2.1. Kriptografijos aspektai 9
2.2. Matematinės problemos 10
2.2.1. Didelių skaičių faktorizavimas 10
2.2.2. Diskretaus logaritmo problema 11
2.2.3. Elipsinės kreivės 11
2.2.4. Tiesinių kodų dekodavimas 11
2.3. Pagrindinės kriptografinės sistemos 12
2.4. Atakų tipai 13
3. Klaidas taisančių kodų taikymas kriptografijoje 14
3.1. Bendra informacija apie klaidas taisančius kodus 14
3.2. Kriptosistemos naudojančios tiesinius kodus 14
3.2.1. McEliece kriptografinės sistemos veikimas 15
3.2.1.1. Pavyzdinio pranešimo užšifravimas 16
4. McEliece kriptografinė sistemos sandara 17
4.1. Goppa kodai ir Goppa polinomai 17
4.2. Matricos S ir P 18
4.3. Klaidos vektorius 19
4.4. Kriptografinės sistemos silpnos vietos 19
4.5. Kriptografinės sistemos silpnų raktų išvengimas 20
5. Atakos prieš McEliece kriptografinę sistemą 20
5.1. Apibendrinta informacijos dekodavimo ataka 21
5.2. Žinomo dalinio teksto ataka 23
5.3. Pranešimų persiuntimo ataka 24
5.4. Susijusių pranešimų ataka 25
6. Atakų prieš McEliece kriptografinę sistemą tyrimas 26
6.1. Kriptografinės sistemos saugumas 27
6.2. Apibendrinta informacijos dekodavimo atakos realizacija 27
6.3. Apibendrintos informacijos dekodavimo atakos rezultatai 29
6.4. Žinomo dalinio teksto atakos realizacija 35
6.5. Žinomo dalinio teksto atakos rezultatai 35

6.6. Pranešimo persiuntimo atakos realizacija....38
6.7. Pranešimo persiuntimo atakos rezultatai....39
6.8. Susijusių pranešimų atakos realizacija....41
6.9. Susijusių pranešimų atakos rezultatai....41
Išvados ir rezultatai....44
Literatūros sąrašas....46
Priedas....49

Įvadas

Šiais laikais, sparčiai vystantis informacinėms technologijoms, galima siekti aukščiausių tikslų tiek programinės, tiek aparatūrinės įrangos srityse. Todėl, norint įgauti pranašumą verslo srityje ir išlaikyti pozicijas jame, būtina lanksčiai pritaikyti naujausias informacines technologijas, tenkinančias verslo poreikius. Taip pat naujausios informacinės technologijos padeda ir asmeniniame gyvenime: spartėjanti informacijos sklaida taupo mūsų laiką, suteikia komfortą ir naujas galimybes tobulėti.

Dėl vis didėjančio interneto vartotojų skaičiaus, atitinkamai didėja ir informaciniais kanalais siunčiamų svarbių dokumentų skaičius, atliekama vis daugiau elektroninės bankininkystės operacijų, publikuojama konfidenciali informacija, skirta tik tam tikram žmonių ratui. Vis daugiau viešųjų paslaugų yra perkeliama į elektroninę erdvę, todėl duomenų saugumo, informacijos siuntėjo ir gavėjo autorizavimo problemos tampa vis svarbesnės ir aktualesnės šiomis dienomis. Informacijos saugumas bus dar svarbesnis, kai atsiras kvantiniai kompiuteriai, nes manoma, kad jie galės pažeisti RSA, ElGamal ir elipsinių kreivių viešojo rakto kriptografines sistemas, kurios dabar yra plačiai naudojamos. Bet kaip [DMR11] publikacijoje rašoma, kad McEliece viešojo rakto kriptografinė sistema turėtų atsilaikyti prieš atakas.

Kritikai vienu iš didžiausių McEliece viešojo rakto kriptografinės sistemos minusų įvardija tai, kad užšifruotas pranešimas tampa gerokai didesniu negu originalus pranešimas [MOV96a]. Tačiau šiais laikais technologijoms sparčiai tobulėjant ir duomenų perdavimo greičiams augant šis trūkumas praranda reikšmę. Taigi, ateityje McEliece viešojo rakto kriptografinės sistemos panaudojimas praktikoje turėtų išaugti.

Šio baigiamojo darbo **tikslas** yra patikrinti McEliece viešojo rakto kriptografinės sistemos saugumą prieš galimas kriptografinės sistemos atakas.

Norint pasiekti darbo tikslą reikės atlikti šiuos **uždavinius**:

- išanalizuoti kriptografinius aspektus ir susipažinti su kriptografinėmis sistemomis (RSA, DES, A5, RC4);
- išnagrinėti McEliece kriptografinės sistemos sandarą ir jos įtaką saugumui;
- išnagrinėti atakas prieš McEliece kriptografinę sistemą;
- realizuoti McEliece kriptografinės sistemos šifravimą;
- realizuoti atakas prieš McEliece kriptografinę sistemą;
- surinkti ir įvertinti statistinius atakų vykdymo laikus su mažais McEliece kriptografinės sistemos parametrais;
- įvertinti, kokie būtų rezultatai su realiais Mceliece kriptografinės sistemos parametrais;

- įvertinti, kokie vykdymo laikai būtų naudojant superkompiuterį.

Kaip galima matyti iš užsibrėžtų darbo uždavinių, didelis dėmesys bus skiriamas praktiniams tyrimams. Rašant magistro darbą buvo remiamasi panašiais atliktais tyrimais, kurie yra aprašyti 1998 [CC98] ir 2000 [CS00] metų publikacijose. Publikacijose atlikti eksperimentai tyria saugumą, nagrinėjant dvejetainių operacijų skaičių, tačiau šiame darbe yra prognozuojamas vykdymo laikas su didesnėmis *n* parametro reikšmėmis. Taip pat šiame darbe be vykdymo laiko prognozavimo su tyrime naudotu personaliniu kompiuteriu buvo prognozuojamas laikas su šiuo metu galingiausiu Tianhe-2 superkompiuteriu.

1. Sąvokų žodynėlis

Sąvokų žodynėlyje pateikti sunkiau suprantamų terminų apibrėžimai, kurie buvo naudojami darbe.

- Kriptografija yra informacijų protokolų[1] rinkinys, užtikrinantis duomenų konfidencialumą (*angl. confidentiality*), vientisumą (*angl. integrity*) ir autentiškumą (*angl. authenticity*) [SLD+08].
- McEliece viešojo rakto kriptografinė sistema – kodavimo algoritmas, kuris buvo sukurtas 1978 metais ir pavadintas jo kūrėjo Robert J. McEliece garbei [McE78].
- Hammingo (*angl. Hamming weight)* svoris yra A žodžio nenulinių x abėcėlės bitų skaičius, kuris yra žymimas $Hw(x) = \sum_{i=1,\cdots,n} 1$. (pvz., Jei x = (0 1 0 1 0), tai $Hw(x) = 2$).
- $[n, k]$ yra žymimas tiesinis kodas C, kurio ilgis yra n, dimensija k.
- Tiesinio kodo C[n, k] virš F_q generuojančia matrica vadiname k × n matricą, kurios eilutės sudaro kodo C bazę [HP03].
- „Kodą $C \subset \mathbb{F}_k^n$, vadinsime tiesiniu, jei C yra tiesinė erdvė[2].<…> “ [HP03]
- Goppa kodai – viena iš tiesinių kodų šeimų, kurį pasižymi saugumu, kad būtų galima naudoti kriptografinėse sistemose. [HP03]

[1] http://www.oxforddictionaries.com/definition/english/protocol [žiūrėta 2016-05-23]
[2] http://www.math.cmu.edu/~wn0g/noll/2ch1a.pdf [žiūrėta 2016-05-21]

2. Kriptografijos temų apžvalga

2.1. Kriptografijos aspektai

Kriptografijos sąvoka yra plačiai žinoma daugelyje mokslo sričių. Universitetuose yra dėstomi kursai, susiję su kriptografija, tad teorinių traktatų, analizuojančių skirtingus aspektus, galima rasti daug ir įvairių. Juose aprašomi bendrieji kriptografijos principai, kriptografijos nagrinėjamos sritys. Pačią kriptografiją bendriausiu aspektu galima apibrėžti kaip: „<...> duomenų apsaugos uždavinių sprendimo matematiniais metodais mokslą“ [Sta07], arba „<...> kaip informacijos apsaugos priemonių kūrimo <...> sritį“ [Sta05]. Tiksliausias terminas, kuriuo vadovaujamasi šiame baigiamajame darbe, apibrėžia, kad – „kriptografija yra mokslas apie matematinius metodus skirtus informacijos saugumui užtikrinti“ [MOV96a].

Kiekviena kriptografinė sistema turi tenkinti tam tikrus saugumo uždavinius. McEliece kriptografinė sistema nėra išimtis, V. Stakėnas [Sta07] įvardija tris pagrindinius kriptografijos saugumo uždavinius:

- konfidencialumą (*angl. confidentiality*);
- duomenų vientisumą (*angl. integrity*);
- autentiškumą (*angl. authenticity*).

Tačiau, šiais technologijų laikais, be šių trijų pagrindinių kriptografijos uždavinių atsiranda ir ketvirtas – veiksmų neišsižadėjimas/neatsisakymas (*angl. non-repudiation*) [MOV96a].

Informacijos konfidencialumu siekiama užtikrinti, kad informacija būtų prieinama tik tiems asmenims, kuriems informacija buvo skirta. Šiam tikslui užtikrinti yra naudojama fizinė duomenų apsauga arba matematiniai algoritmai, kurių dėka informacija tampa sunkiai suprantama pašaliniams asmenims.

Duomenų vientisumu siekiama užtikrinti, kad pašaliniai asmenys negalėtų pakeisti (modifikuoti) siunčiamų duomenų. Jeigu tokia situacija įvyksta, tai užtikrinama, kad informacijos gavėjas būtų informuojamas apie informacijos pakeitimą.

Autentiškumo sąvoka yra plati, apimanti duomenų šaltinius ir jos naudotojų identifikaciją. Kai duomenų šaltinis nori užmegzti ryšį su duomenų gavėju, tai jie turi vienas kitą atpažinti.

Veiksmų neatsisakymu siekiama užtikrinti, kad atlikus tam tikrą veiksmą asmuo nebegali jo atšaukti. Palyginimui: asmuo pervedęs pinigus iš savo sąskaitos į kito asmens sąskaitą, tų pinigų

susigrąžinti nebegali. Jeigu šio saugumo kriterijaus nebūtų, tai labai paveiktų elektroninę prekybą ir visas kitas finansines operacijas.

Apžvelgus kriptografijos saugumo uždavinius reikia aptarti, kokio saugumo gali būti kriptografinės sistemos. Remiantis V. Stakėno knyga „Kodų ir šifrų. Informacijos kodavimo ir kriptografijos pagrindų“ išskiriami penki saugus kriptografinių sistemų apibrėžimai [Sta07]:

- besąlygišku saugumu (*angl. unconditional security*) vadinama, „<...> jei net ir turėdamas beribius skaičiavimo išteklius kriptoanalitikas negali be rakto iš šifro nustatyti, koks pranešimas buvo siųstas“.
- saugia sudėtingumo teorijos požiūriu kriptosistema vadinama (*angl. complexity-theoretic security*), „<...> jei jos negali įveikti asmuo, kurio skaičiavimo resursai leidžia jam taikyti tik polinominio laiko algoritmus <...>“.
- įrodomas saugumas (*angl. provable security*) yra „<...> jeigu galima įrodyti, kad sistemos įveikimas yra tolydus matematinio (dažniausiai skaičių teorijos) uždavinio, kuris laikomas sunkiu, sprendimui“.
- skaičiavimo požiūriu saugia kriptosistema vadinama (*angl. computational security*), „<...> jeigu pasiektas skaičiavimo resursų lygis yra pernelyg žemas, kad naudojant geriausias žinomas atakas, sistema būtų įveikta“.
- euristiškai saugi yra „<...> tokia sistema, kurios saugumą patvirtina tam tikri dažnai euristiniai argumentai“.

Kadangi kriptografijos saugumo lygį yra labai sunku įvertinti, todėl yra vertinamas operacijų skaičius, kuris yra reikalingas įveikti kriptografinę sistemą, taikant šiuo metu geriausius ir žinomiausius metodus [MOV96a]. Taigi logiška galvoti: kai atsiranda geresnis kriptografinės sistemos įveikimo metodas negu dabartiniai, tai saugumas sumažėja, o tai reiškia, kad reikia iš naujo įvertinti saugumą, ar kriptografinės sistemos naudojimui saugumas vis dar yra pakankamas.

2.2. Matematinės problemos

Kriptografinių sistemų veikimas remiasi tam tikromis matematinėmis problemomis. Šiame poskyryje apžvelgiama keletas matematinių problemų, kuriomis gali remtis kriptografinės sistemos.

2.2.1. Didelių skaičių faktorizavimas

Didelių skaičių pirminių daugiklių radimas vadinamas skaičiaus faktorizavimu. Skaidyti natūraliuosius skaičius pirminiais daugikliais yra sunkus skaičiavimo uždavinys [Sta07]. Iki šios dienos yra teigiama, kad neegzistuoja joks efektyvus algoritmas, kuris galėtų greitai

faktorizuoti didelius natūraliuosius skaičius. Todėl kriptografinės sistemos paremtos šia problema yra saugios tol, kol nebus rastas algoritmas galintis greitai spręsti didelių skaičių faktorizavimo problemą [Pom08].

2.2.2. Diskretaus logaritmo problema

Tegul *p* yra pirminis skaičius, o *a* ir *b* yra sveiki skaičiai, kurie *mod p* nėra lygūs nuliui. Tarkime, kad egzistuoja sveikas skaičius *k* (žr. į formulę, pateiktą žemiau):

$$a^k \equiv b \ (mod\ p).$$

Klasikinė diskretaus logaritmo problema sprendžia, kaip turint *a, b* ir *p* surasti sveikąjį skaičių *k* [Was08].

Lawrence‘o C. Washington‘o publikacijoje „Elliptic curves number theory and cryptography“ nagrinėja diskretaus logaritmo problemą ir ją naudojančias kriptografines sistemas. O V. Stakėno knygoje [Sta07] yra nagrinėjami metodai, kuriais sprendžiama diskretaus logaritmo problema.

2.2.3. Elipsinės kreivės

Elipsinės kreivės kriptografijoje buvo pradėtos naudoti 1980 m. Po 5 m. R. Schoof‘as paskelbė algoritmą, kuris skaičiavo elipsinės kreivės virš baigtinio lauko F_q taškus. Elipsinės kreivės apibrėžiamos kaip lygybės $y^2 = x^3 + ax + b$ sprendinių pora *(x, y),* kartu su papildomu tašku O, kuris vadinamas begalybės tašku.

Plačiai apie elipsines kreives rašė matematikas Lawrence‘as C. Washington‘as savo publikacijoje [Was08]. Joje autorius aptaria elipsines kreives. Išskiria, kad egzistuoja kriptografinis algoritmas, kuris remiasi elipsinėmis kreivėmis ir yra taikomas praktikoje.

2.2.4. Tiesinių kodų dekodavimas

Tiesiniai kodai yra žodžių aibės, kurios yra tiesinės žodžių erdvės poerdviai. V. Stakėnas tiesinį kodą apibrėžia: „kodą L, $L \subset F_q^n$, vadinsime tiesiniu, jei L yra tiesinis poerdvis F_q^n “ [Sta02] (tai yra n-matė tiesinė erdvė virš kūno F^n). „Jei L dimensija lygi *k,* o minimalus atstumas *d,* tai kodą L vadinsime *[n,k,d]*, arba tiesiog $[n, k]$, kodu “ [Sta02].

Tiesinės operacijos su žodžiais suteikia galimybę sudaryti patogias kodavimo ir dekodavimo procedūras. Tiesinių kodų dekodavimo algoritmas dažniausiai yra naudojamas duomenų perdavimui nesaugiais kanalais. Šis algoritmas plačiai aprašomas Viliaus Stakėno ([Sta02]) publikacijose.

Jeigu $L \subset F_q^n$ yra tiesinis kodas, tada erdvę F_q^n į aibes

$$L_x = c + L;$$

čia c$\in F_q^n$.

Tarkime, kad pranešimas koduojamas naudojant kodą L ir gaunamas užkoduotas pranešimas *c*. Tiesinio kodo dekodavimui panaudosime minimalaus atstumo taisyklę, kurią naudojant dekoduosime pranešimą *msg* ir jį tenkins:

$$\mathrm{h}(msg, c) = w(c - msg)$$

Pranešimas *d* yra kurioje nors klasėje L_i. Todėl norint dekoduoti reikia peržiūrėti klasę *L*, kurioje yra pranešimas c. Jeigu buvo pasirinktas didelis tiesinio kodo parametras *k*, tuomet reikės peržiūrėti didelį skaičių elementų [Sta02].

2.3. Pagrindinės kriptografinės sistemos

McEliece kriptografinė sistema yra asimetrinis dekodavimo algoritmas. Tačiau kriptografinės sistemos gali būti skirstomos ne vien į asimetrines, bet ir į simetrines sistemas. Apie šias kriptografines sistemas rašo V. Stakėnas ([Sta07]). Asimetrinę arba viešojo rakto (*angl. public-key*) kriptografinę sistemą vadinsime tokią sistemą, kuri šifravimui ir dešifravimui naudoja skirtingus raktus [Sta07]. Simetrinio rakto kriptografinėje sistemoje duomenys šifruojami ir dešifruojami tuo pačiu raktu. Taip pat pasitaiko atvejų, kai simetrinėje sistemoje „<...> dešifravimo raktas nesutampa su šifravimo raktu, tačiau gali būti iš jo nesunkiai surandamas“. [Sta07]

Simetrinės sistemos realizacija paprastai būna lengvesnė negu viešojo rakto. Tačiau simetrinės sistemos didžiausiu trūkumu yra įvardijamas rakto perdavimas duomenų (*informacijos*) gavėjui, nes reikia jį perduoti taip, kad pašaliniai asmenys nesužinotų. Simetrinės kriptografinės sistemos yra skirstomos pagal savo pobūdį į srautinius ir blokinius šifrus.

Šiuo metu viena iš populiariausių kriptografinės viešojo rakto sistemų yra RSA. Ši kriptografinė sistema buvo sukurta 1977 m., ji pavadinta kūrėjų pirmosiomis pavardžių raidėmis (Rivest, Shamir ir Adleman). Šios kriptografinės sistemos saugumas remiasi matematine faktorizavimo problema. Plačiau apie algoritmo veikimą galima rasti [Sta07] ir [KA08] publikacijose.

Taip pat gana plačiai buvo taikoma ir blokinė simetrinė kriptografinė sistema DES (*angl. Data Encryption Standard*) [Cop94], [Sta07], kuri buvo sukurta 1974 m. Nustojus naudoti šifravimo sistemą DES, ji „<...> buvo adaptuota 1977 m. Ir šiuo metu DES naudojama kaip palyginimo etalonas naujiems šifravimo algoritmams patikrinti“ [KA08]. Simetrinės

kriptografinės sistemos DES pagrindas yra Feistelio iteracijos, kuriose naudojamos sukonstruotos funkcijos. DES šifruoja 64 bitų ilgio blokus naudojant 56 bitų ilgio raktus. Šifravimo procesą sudaro 3 žingsniai. Plačiau apie DES aprašyta Dono Coppersmitho publikacijoje [Cop94].

1997 m. Jungtinių Amerikos Valstijų Nacionalinis standartų ir technologijų institutas paskelbė konkursą pakeisti DES šifrą. Po ilgų svarstymų buvo išrinktas Rijndeal šifras, kuriam buvo suteiktas AES *(angl. Advanced Encryption Standard)* pavadinimas. Po 5 m. trukusio vertinimo, AES kriptografinė sistema buvo paskelbta standartu. Jos detalų aprašymą galima rasti oficialioje Jungtinių Amerikos Valstijų Vyriausybės publikacijoje, kurioje pateikiamos AES standarto taikymo sritys, informacija, susijusi su realizacija, reikalingumo pagrindimas ir kt. [Sta07].

Keli plačiau paplitę simetrinio srauto šifrai yra RC4, A5 ir Bluetooth E0, tačiau plačiau panagrinėsime A5/1 ir RC4.

A5 algoritmas yra naudojamas srauto šifravimui komunikacijoje. Kai algoritmas atsirado jis buvo slaptas, tačiau po kurio laiko tapo žinomas dėl nutekintos informacijos. Šifro A5/1 rakto srauto sukūrimui yra naudojama sudėties moduliu 2 operacija su bitais, kuriuos generuoja trys tiesinių registrų sistemos. Tačiau nebūtinai kiekviename žingsnyje tiesinių registrų sistema pakeičia savo vertes (iš naujo suformuoja registrų reikšmes). Platesnį A5 algoritmo aprašymą galima rasti Hagen Fritsch publikacijoje [Fri07].

Kitas plačiai paplitęs programinės įrangos realizacijoje kriptografinis algoritmas yra RC4 (*angl. Rivest Cipher*). RC4 algoritmas buvo sukurtas 1987 m. ir iki 1994 m. buvo laikomas paslaptyje, tačiau gana greitai internete pasklido algoritmo realizacija. Nuo tada yra leidžiama realizuoti šį algoritmą. Jis dažniausiai taikomas TLS/SSL, WPA (*angl. Wi-Fi Protected Access*), WEP (*angl. Wired Equivalent Privacy*) [ABP+13] ir kituose protokoluose.

2.4. Atakų tipai

Priešingas kriptografijai procesas yra kriptoanalizė, t. y. pranešimo teksto ir slapto rakto radimas [KA08]. Kriptoanalizės tikslas yra „<...> naudojantis šifrais, atkurti juos atitinkančius tekstus, o galbūt – netgi nustatyti ir raktus.“ [Sta07].

Atakos skirstomos tipais pagal tai, kokią informaciją kriptoanalitikas turi. V. Stakėnas savo knygoje („Kodai ir šifrai. Informacijos kodavimo ir kriptografijos pagrindai“) išskiria penkis kriptosistemų atakų tipus [Sta07]:

1. pavienių šifrų ataka (*angl. Ciphertext-only attack*) yra, kai kriptoanalitikas turi tik užšifruotą pranešimą;

2. teksto – šifro porų ataka (*angl. Known-paintext attack*) yra, kai kriptoanalitikui be užšifruoto pranešimo pavyko gauti ir pradinius tekstus;
3. pasirinktų teksto šifro porų ataka (*angl. Chosen – plaintext attack*) yra, kai kriptoanalitikas turi galimybę dabartiniu laiku pateikti norimus pranešimus šifravimo sistemai ir gauti užšifruotus pranešimus;
4. adaptyvi pasirinktų teksto šifro porų ataka (*angl. Apdaptive – chosen – plaintext attack*) yra, kai kriptoanalitikas vieną kartą atlikęs pasirinktų teksto – šifro porų ataką ir atsižvelgęs į kriptoanalizės rezultatus, pasirenka naujus parnešimus ir gauna jų šifrus;
5. pasirinktų šifrų ataka (*angl. Chosen – ciphertext attack*) yra, kai kriptoanalitikas pasirenka šifrus ir gauna juos atitinkančius pranešimus.

Detalesnė aktualių McEliece kriptografinės sistemos atakų analizė yra aptariama 4.6 poskyryje.

3. Klaidas taisančių kodų taikymas kriptografijoje

3.1. Bendra informacija apie klaidas taisančius kodus

Siunčiant duomenis interneto informaciniais kanalais gali atsirasti įvairių duomenų iškraipymų. Norint sumažinti tokį iškraipymą, buvo pradėti naudoti metodai, kurie padeda aptikti duomenų pažeidimus (*modifikacijas*) ir juos ištaisyti. Prieš koduojant duomenis prie jų yra pridedama papildoma informacija, kuri gautuose duomenyse leidžia nustatyti ir ištaisyti tam tikrą skaičių klaidų, kurios įvyksta siunčiant duomenis.

Dekodavimo metu, naudojant papildomą informaciją, kuri buvo pridėta koduojant duomenis, yra aptinkamas ir ištaisomas tam tikras klaidų skaičius t. Tačiau kanale padarytų klaidų skaičius negali būti didesnis už klaidų skaičių t, kurį gali ištaisyti naudojamas klaidas taisantis kodas.

3.2. Kriptosistemos naudojančios tiesinius kodus

Yra daug kriptografinių sistemų, kurios naudoja tiesinius kodus, tačiau plačiau aptarsime tik McEliece kriptografinę sistemą. Nuo McEliece kriptografinės sistemos atsiradimo yra nemažai kriptografinės sistemos modifikacijų kaip Niederreiter kriptografinė sistema, kurios savo saugumu yra panašios ar lygiavertės į McEliece kriptografinę sistemą [LDW94][Can98].

3.2.1. McEliece kriptografinės sistemos veikimas

Ši kriptografinė sistema yra viena iš pirmųjų viešojo rakto kriptografinių sistemų, kuri naudoja klaidas taisančius kodus. Esminių McEliece kriptografinės sistemos saugumo spragų iki šiol niekas nesurado. Kaip ir kiekvienoje viešojo rakto kriptografinėje sistemoje, norint ją naudoti, reikia atlikti tris pagrindinius žingsnius:

- sugeneruoti privatų ir viešąjį raktą;
- užšifruoti duomenis;
- iššifruoti duomenis.

[Sti95] ir [LDW94] publikacijose, McEliece kriptografinės sistemos raktų generavimas, šifravimas ir dešifravimas aprašomas panašiai. Toliau darbe pateikiamos šios kriptografinės sistemos bendrosios nuostatos.

Privataus ir viešojo rakto generavimas [MOV96b]:

1. pasirenkami *m, t* parametrai ir apskaičiuojama

$$n = 2^m; \tag{3.2.1-1}$$

$$k = n - m * t. \tag{3.2.1-2}$$

2. sukonstruojama tiesinio *[n, k]* Goppa kodo C, galinčio ištaisyti t klaidų, $k \times n$ generuojančia matrica *G*;
3. sugeneruojama $k \times k$ neišsigimusi matrica *S*;
4. sugeneruojama $n \times n$ matrica *P*, kuri gaunama iš vienetinės matricos, panaudojus perstatas;
5. sugeneruojama matrica, kuri sudaro viešąjį raktą $K_v = < G', t >$,

$$G' = SGP; \tag{3.2.1-3}$$

6. privatus raktas yra visos trys matricos $K_p =< S, G, P >$

Šifravimas:

1. pranešimas *msg* (k ilgio dvejetainio formato pranešimas);
2. suskaičiuojama $msg\ G'$;
3. pridedamas klaidos vektorius *e* ir gaunamas užšifruotas pranešimas *c*. Klaidos vektorius *e*, parenkamas atsitiktinai ir jo svoris yra ne didesnis už *t*.

Dešifravimas:

1. apskaičiuojama $c' = cP^{-1}$;
2. žodyje c' ištaisomos klaidos, naudojant Goppa kodo C dekodavimo algoritmą ir gaunamas m';
3. apskaičiuojama $msg = m'S^{-1}$ ir gaunamas siųstas pranešimas *msg*.

3.2.1.1. Pavyzdinio pranešimo užšifravimas

Pranešimo šifravimui buvo naudojamas prieš tai aprašytas kriptografinės sistemos algoritmas. Tačiau demonstravimo dėlei yra naudojamas paprastas algoritmo pavyzdys, nes atlikti analogiškus veiksmus su realiais duomenimis būtų pernelyg sudėtinga.

Tarkime, kad turime pranešimą *msg*, kurio dvejetainis kodas yra (1 1 0 1). Pasinaudodami Goppa kodu šifruojame pradinį dvejetainį pranešimą, pagal išraišką $c = msg\ G' \oplus e$, kur:

c – užšifruotas pranešimas,

msg – originalus dvejetainis pranešimas,

G' – kodą generuojanti matrica,

e – klaidos vektorius.

Aptarsime G' matricos gavimo algoritmą.

Pagal algoritmo apibrėžimą $G' = SGP$, kur:

S – atsitiktinai pasirinkta neišsigimusi $k \times k$ matrica,

G – kodą generuojanti $k \times n$ matrica,

P – perstatos $n \times n$ matrica.

Dėl paprastumo šiame darbe neatlikinėsime pilnos generuojančios matricos išvedimo. Pradiniais matricos duomenimis laikysime literatūroje [Joc02] aprašytą S, G ir P matricas.

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix};$$

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

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Pasinaudojus anksčiau aprašyta formule (3.2.1-3) gauname G':

$$G' = SGP = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Kaip galime pastebėti, kad G ir G' matricos rangas liko tas pats. Tokiu atveju paslėpėme generuojančią matricą G ir galime atlikti šifravimo žingsnius.

Šifruodami dvejetainį pranešimą, atliekame sandaugą $msgG'$:

$$(1\ 1\ 0\ 1) * \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$
$$= (0\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1).$$

Galiausiai pagal algoritmą pridėję klaidos vektorių e gausime galutinį užšifruotą pranešimą. Klaidos vektorius yra parenkamas atsitiktiniu būdu. Pateiktame pavyzdyje klaidos vektoriuje gali būti nedaugiau negu dvi klaidos ($t = 2$).

Tarkime, kad klaidos vektorius yra $e = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$ ir jį pridėję prie atliktos $msgG'$ sandaugos gauname užšifruotą pranešimą:

c = (0 1 0 0 0 0 1 1 0 0 1 1).

4. McEliece kriptografinė sistemos sandara

Šiame skyriuje plačiau panagrinėsime McEliece kriptografinės sistemos sandarą. Taip pat be sandaros buvo aptartos kriptografinės sistemos silpnosios vietos ir ką reikėtų daryti, kad pavyktų išvengti šių silpnų kriptografinės sistemos vietų. Pagrindinis šio skyriaus poskyris yra McEliece kriptografinės sistemos atakų analizė.

4.1. Goppa kodai ir Goppa polinomai

Goppa kodų pavadinimas kilo nuo jo kūrėjo rusų matematiko pavardės (V. G. Goppa). Goppa kodai yra BCH[3] kodų generalizacija, bet BCH kodų saugumas per mažas, kad būtų galima jomis pagrįsti kriptografinę sistemą. Kriptografinė sistema paremta Goppa kodais buvo pristatyta R. J. McEliece 1978 m. [Joc02]. Goppa kodas $\Gamma\big(L, g(z)\big)$ yra apibrėžiamas Goppa polinomu g(z), kuris yra laipsnio *t* virš išplėstinės srities $GF(q^m)$, kur q yra pirminis ir $L \subseteq GF(q^m)$.

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

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

visiems $\alpha_i \in L$, kur $g(\alpha_i) \neq 0$.

[3] Bose-Chaudhuri-Hockuenghemo (BCH) kodai yra cikliniai kodai

Su vektoriais $c = (c_1, \dots, c_n)$ virš $GF(q)$ apibrėžiama funkcija:

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

kur $\frac{1}{z-\alpha_i}$ yra unikalus polinomas nuo $(z-\alpha_i) * \frac{1}{z-\alpha_i} \equiv 1 \left(mod\ g(z)\right)$.

Taigi galima suformuluoti apibrėžimą: Goppa kodas $\Gamma\left(L, g(z)\right)$ sudarytas iš visų vektorių *c*, kuriems galioja lygybė:

$$R_c(z) \equiv 0 \left(mod\ g(z)\right).$$

Thomas A. Bersonas savo publikacijoje [Ber97] pabrėžia, kad buvo mėginta pakeisti Goppa kodus kitais klaidas taisančiais kodais. Kaip pavyzdį autorius pateikė E. M. Gabidulino „Ideals over a non – commutative ring and their application in cryptology“ publikaciją, kurioje buvo mėginta Goppa kodus pakeisti kita, klaidas taisančių MRD (*angl. Maximum–rang–distance*) kodų klase. Tačiau Gibson'ui sukurta schema pasirodė nepakankamai saugi, kad būtų galima plačiai ją taikyti [Ber97].

4.2. Matricos S ir P

Ellenas Jochemszas savo [Joc02] publikacijoje aprašo S ir P matricas. S ir P matricos yra būtinos McEliece kriptografinės sistemos saugumui. Jeigu kriptoanalitikas atkurtų pagrindinę generuojančią matricą G, tai jam toliau nebūtų sunku surasti ir generuojantį polinomą g(z). Kai kriptoanalitikas suranda generuojančią matricą G ir polinomą g(z), tada, pasinaudojęs iššifravimo algoritmu, gali rasti pradinę informaciją.

Anksčiau buvo aptarta (žr.: 3.2.1. poskyrį), kad matrica S yra sudaryta iš $k \times k$ elementų. Pirmajai a_1 eilutei sudaryti yra $2^k - 1$ variantų. Antroje matricos eilutėje negali būti $0 * a_1$ ir $1 * a_1$, todėl variantų skaičius sumažėja ir yra $2^k - 2$. Todėl galimų $k \times k$ matricos elementų skaičius yra:

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

Jeigu matricos P dydis yra $n \times n$, tai gali būti n! skaičių kombinacijų.

McEliece pasiūlytoje situacijoje tikimybė atspėti matricą S yra:

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

Atspėti matricą P tikimybė yra šiek tiek didesnė:

$$\frac{1}{1024!} = 0{,}1845519398 * 10^{-2639}.$$

Iš pateikto S ir P matricų aprašymo, matyti, kad įveikimas McEliece kriptografinės sistemos atspėjant S ir P matricas yra praktiškai neįmanomas [Joc02].

4.3. Klaidos vektorius

McEliece kriptografinė sistema siūlo informaciją šifruoti, naudojant tiesinį kodą *C*, kuris turėtų greitą dekodavimo algoritmą. Informacija šifruojama, dauginant informaciją *msg* iš tiesinio kodo *C* generuojančios matricos *G*. Tačiau gautą, užšifruotą informaciją kriptoanalitikui nėra sunku iššifruoti, todėl prie užšifruotos informacijos yra pridedamas ir klaidos vektorius *e*. Informacijos gavėjas pasinaudojęs tiesinio kodo *C* dekodavimo algoritmu, suranda klaidos vektorių *e*, kurį atėmęs iš užšifruotos informacijos *c* gauna *msg*G*, iš kurio lengvai galima rasti *msg*.

Reikėtų atkreipti dėmesį, kad informacijos siuntėjas informaciją turėtų iškraipyti tik tiek, kiek tiesinio kodo *C* dekodavimo algoritmas gali ištaisyti klaidų. Šiai problemai spręsti yra naudojamas parametras *t*, kuris nurodo, kiek klaidų gali padaryti siuntėjas.

Taip pat reikėtų atkreipti dėmesį, kad dažnai publikacijose, kurios yra susijusios su klaidos vektoriaus naudojimu kriptografijoje, atkreipiamas dėmesys į tai, kad naudojant skirtingus klaidos vektorius šifruojant tą pačią informaciją, atsiranda rizika, jog sistema yra pažeidžiama [Ber97]. Šiam McEliece kriptografinės sistemos pažeidimui sumažinti reikėtų naudoti tą patį klaidos vektorių. Tokiu atveju kriptoanalitikas turėtų mažiau informacijos, spręsti apie siuntėjo privatų raktą.

4.4. Kriptografinės sistemos silpnos vietos

P ir S matricų atspėjimas galėtų būti įvardintas kaip silpnoji kriptografinės sistemos vieta. Tačiau kaip matyti iš ankščiau aprašytos temos (žr.: 4.2. poskyrį), pateikti skaičiavimai parodo, kad S ir P matricų atspėjimo tikimybė yra labai maža.

Viena iš pavojingiausių kriptografinės sistemos silpnų vietų yra ta, jei kriptoanalitikas sužinotų dalį originalaus pranešimo bitų sekos, tai kriptoanalitikui leistų sumažinti darbą prie originalaus pranešimo atskleidimo.

Kriptografinės sistemos silpna vieta galėtų būti klaidos vektorius. Thomas Bersonas savo publikacijoje [Ber97] aprašo, kaip dešifruoti pranešimą. Jeigu tas pats pranešimas buvo šifruotas panaudojant du skirtingus klaidos vektorius. Detalesnis atakų aprašymas yra pateiktas 5.3. ir 5.4. poskyriuose.

4.5. Kriptografinės sistemos silpnų raktų išvengimas

Pierre‘as Loidreau ir Nicolo Sendriero [LS01] publikacijoje nagrinėja, kaip surasti McEliece kriptografinės sistemos silpnus raktus. Įvardijamos dvi McEliece kriptografinės sistemos saugumo prielaidos:

1. kodo parametrai turi būti pakankamai dideli, kad būtų negalima jo iššifruoti pasinaudojus bendros paskirties tiesinių kodų dekodavimo algoritmu;
2. sunku būtų sukonstruoti greitą dekodavimo algoritmą, žinant viešąjį raktą [LS01].

5. Atakos prieš McEliece kriptografinę sistemą

Pasak Kazukuni Kobara ir Hideki Imai [KI01], McEliece kriptografinės sistemos atakos gali būti skirstomos į dvi kategorijas pagal jų pavojingumą:

- nekritinės (*angl. non–critical*) atakos;
- kritinės (*angl. critical*) atakos.

Nekritinių atakų išvengti yra paprasta, reikia didinti kriptografinės sistemos parametrus [KI01].

Nekritinės atakos gali būti šios:

- apibendrinta informacijos dekodavimo ataka (*angl. Generalized Information – Set – Decoding Attack*);
- mažo svorio kodo žodžių radimo ataka (*angl. Finding–Low–Weight–Codeword*).

Kritinių atakų išvengti yra šiek tiek sudėtingiau. Tačiau kaip išvengti kiekvienos kritinės atakos galima spręsti skirtingu atveju. Kaip pavyzdį, galima pateikti susijusių pranešimų ataką. Geriausias būdas jos išvengti – kiekvieną kartą siunčiant pranešimą, naudoti tą patį klaidos vektorių, jeigu buvo koduojamas tas pats pranešimas.

Kritinės atakos yra tokios:

- žinomo dalinio teksto (*angl. Known–Partial–Plaintext*) ataka;
- susijusių pranešimų (*angl. related–message*) ataka;
- pranešimo persiuntimo (*angl. message–resend*) ataka;
- reakcijos (*angl. Reaction*) ataka;
- Korzhik–Turkin ataka.

Norint plačiau panagrinėti atakas, jų aprašymus galima rasti įvairioje literatūroje. Pavyzdžiui, susijusių pranešimų ir pranešimo persiuntimo atakos plačiai yra nagrinėjamos amerikiečio kriptoanalitiko Thomo A. Bersono [Ber97] publikacijoje. Kazukuni Kobara, Hideki

Imai [KI02] ir D. Engelberto, E. Overbecko, A. Schmidto [EOS06] publikacijose pateikiami atakų aprašymai ir algoritmai. Atakų realizacija palengvina, kai kuriose publikacijose pateikti algoritmų pseudokodai[4].

Pagal pateikta sąrašą yra pasirinktos 4-rios atakos, kurios išskirtos pagal pavojingumą. Žinomos dalinio teksto atakos pavojingumas priklauso nuo kriptoanalitiko žinomo pradinio pranešimo dalies. Todėl buvo pasirinkta žinomo dalinio teksto ataka, kad būtų galima įvertinti, kaip priklauso pavojingumas nuo žinomo pradinio pranešimo dalies. Tačiau pasirinkta ataka negali dešifruoti pradinio pranešimo, todėl buvo pasirinkta apibendrinta informacijos dekodavimo ataka. Vertėtų paminėti, kad R. J. McEliece savo publikacijoje [McE78] yra pasiūlęs šią ataką.

Prieš tai buvusiame skyriuje buvo parašyta, kad labiausiai tikėtina McEliece kriptografinės sistemos silpnoji vieta yra klaidos vektorius. Norėdami, tai patikrinti iš kritinių atakų sąrašo yra pasirinkta susijusių pranešimų ir pranešimų persiuntimo atakos. Žemiau esančiuose skyriuose pateikiami pasirinktų atakų aprašymai.

5.1. Apibendrinta informacijos dekodavimo ataka

Apibendrintos informacijos dekodavimo ataka buvo pasiūlyta R. J. McElieco 1978 metais. Ataka yra nagrinėjama D. Engelbert, R. Overbeck, A. Schmidt ([EOS06]), Kazukuni Kobara, Hideki Imai ([KI02] ir [KI01]) publikacijose.

Ataka priklauso pavienių šifrų atakos tipui, kai kriptoanalitikas prieš atlikdamas ataką žino užšifruotą pranešimą c ir tiesinio klaidas taisančio kodo matricą G', kurios ilgis yra n, o dimensija k:

$$c = msgG' \oplus e\ ;$$

čia e yra t svorio klaidos vektorius.

Panagrinėsime atakos veikimą detaliau. Tarkime, kad $I \subset \{1,2,\dots,n\}$, kur $|I| = k$. Pasirenkame iš c, G', e nepriklausomų k stulpelių ir pažymėkime juos c_k, G_k ir e_k.

Tada:

$$c_k = msgG_k^{'} \oplus e_k \text{ [KI01].}$$

Galimi du atvejai:

[4] Neformalus, žmogui skirtas algoritmo užrašymo būdas, kuriame vartojami žymenys artimi natūraliai kalbai. (http://ims.mii.lt/EK%C5%BD/enciklo.html?word=pseudokodas)

1. Jeigu $e_k = 0$ ir $G_k^{'}$ nėra išsigimusi matrica, tada pradinį pranešimą galime gauti pasinaudoję formule:

$$msg = c_k {G_k^{'}}^{-1} \text{ [KI01].}$$

2. Jeigu vektoriaus e_k svoris ($Hw(e_k)$) yra labai mažas ir $G_k^{'}$ nėra išsigimusi matrica, tuomet pradinis pranešimas msg gali būti gautas spėjant vektorių e_k ir tikrinant sąlygą:

$$Hw\left((c_k \oplus e_k){G_k^{'}}^{-1} G' \oplus c\right) = t;$$

čia t yra klaidų skaičius.

Apibendrintos informacijos dekodavimo algoritmas atrodo taip [EOS06]:

1. Pasirenkame k nepriklausomų stulpelių iš G' ir suskaičiuojame

$$\hat{G}_k = {G'}_k^{-1} G'. \qquad (5.1\text{-}1)$$

I žymėkime pasirinktų stulpelių aibę, J – nepasirinktų stulpelių aibę.

2. ***while*** *pranešimas nerastas* ***do***

2.1. apskaičiuojame $w = c \oplus c_k \hat{G}_k$;

2.2. $if\ Hw(w) = t\ then\ msg = c_k {G'}_k^{-1}$;

2.3. $for\ i = 1\ to\ j\ do$

2.3.1. $for\ q = 1\ to\ \binom{n}{i}\ do$

2.3.1.1. pasirenkamas naujas w_k, kad $Hw(w_k) = i$;

2.3.1.2. $if\ Hw\left(w \oplus w_k \hat{G}_k\right) = t\ then\ msg = (c_k \oplus w_k){G'}_k^{-1}$;

2.4. Pakeičiame vieną I koordinatę iš J ir apskaičiuojame $\hat{G}_k = {G'}_k^{-1} G'$.

GISD atakai reikalingų dvejetainių operacijų skaičių pirmame algoritmo žingsnyje ${G'}_k^{-1} G'$ reikia:

$$\sum_{i=1}^{k} \frac{(k-1(n-i+1)}{4} = \frac{4(k-1)(2*n+1-k)}{8} \qquad (5.1\text{-}2)$$

bitų operacijų. 2.2. ir 2.3.1.1. algoritmo žingsniuose skaičiuojant svorį, nereikia suskaičiuoti visų n koordinačių. Vidutiniu atveju reikia suskaičiuoti 2t koordinates. Todėl 2.1. žingsnio dvejetainių operacijų skaičius yra $t * \frac{k}{2}$ ir 2.3.1.1. žingsnio $t * i$.

Taigi 2.3. žingsnyje yra reikalingos dvejetainės operacijos [KI02]:

$$V_j = \sum_{i=1}^{j} t * i * \binom{k}{i}. \qquad (5.1\text{-}3)$$

2.5. žingsnyje atnaujinti $\hat{G}_k$ matricą, reikia:

$$\frac{(k-1)*(n-k)}{4} \qquad (5.1\text{-}4)$$

dvejetainių operacijų skaičiaus. Taip pat 2 žingsnis turės būti įvykdytas T_j kartų [KI02] [EOS06]:

$$T_j = \frac{\binom{n}{k}}{\sum_{i=0}^{j}\binom{t}{i}*\binom{n-t}{k-i}}, \qquad (5.1\text{-}5)$$

bendras atakos dvejetainių operacijų skaičius yra

$$W_j \approx \left\{\frac{(k-1)*(n-k)}{4} + \frac{t*k}{2} + V_j\right\} * T_j. \qquad (5.1\text{-}6)$$

Viską įsistatę į (5.1-6) gauname galutinę formulę:

$$W_j \approx \left\{\frac{(k-1)*(n-k)+2*t*k}{4} + \sum_{i=1}^{j} t*i*\binom{k}{i}\right\} * \frac{\binom{n}{k}}{\sum_{i=0}^{j}\binom{t}{i}*\binom{n-t}{k-i}}. \qquad (5.1\text{-}7)$$

5.2. Žinomo dalinio teksto ataka

Žinomo dalinio teksto ataka priklauso teksto – šifro atakos tipui, kai kriptoanalitikui be šifro, pavyko gauti ir dalį pradinio pranešimo. Ataka yra nagrinėjama Kazukuni Kobara, Hideki Imai [KI01]) ir D. Engelbert, R. Overbeck, A. Schmidt ([EOS06]) publikacijose. Šios atakos pavojingumas priklauso nuo to, kokią pradinę teksto dalį kriptoanalitikas žino. Kuo didesnę teksto dalį kriptoanalitikas žino, tuo ataka yra pavojingesnė, nes atakos vykdymo laikas sutrumpėja.

Tarkime, kad kriptoanalitikas žino msg_r – užšifruoto pranešimo dalį, ir msg_l – nežinoma užšifruoto pranešimo dalis.

Iš to gauname, kad:

$$msg = (msg_l || msg_r) \text{ [KI01].}$$

Jeigu žinome, kad G' matrica yra sudaryta iš k eilučių ir n stulpelių, tokiu atveju tariamai galime pažymėti žinomų eilučių skaičių k_r, o likusių eilučių skaičių k_l. Iš to gauname, jog:

$$k = k_l + k_r \text{ [KI01].}$$

Tada kriptoanalitikas gali bandyti sužinoti m_l pasinaudojęs formule:

$$msgG = msg_l G_l \oplus msg_r G_r.$$

Taigi, turime:

$$c \oplus msg_r G_r = msg_l G_l \oplus e;$$

$$c' = msg_l G_l \oplus e \text{ [EOS06].}$$

Čia c' yra apskaičiuojama iš c ir $msg_l G_l$ reikšmių, kur G_l reikšmė yra generuojančios matricos nežinomos k_l eilutės. Tai žinodami, galime įveikti McEliece kriptografinę sistemą, kurios

parametrai yra $[n, k_l]$. Ataka tik sumažina parametrus, bet neatskleidžia pradinio pranešimo. Tačiau sumažinus parametrus galime naudoti, kokią nors kitą ataką, kaip prieš tai aprašytą apibendrintos informacijos dekodavimo ataką.

5.3. Pranešimų persiuntimo ataka

Pranešimo persiuntimo ataka yra nagrinėjama D. Engelbert, R. Overbeck, A. Schmidt ([EOS06]) ir Thomas A. Berson ([Ber97]) publikacijose. Jeigu sėkmingai pavyksta įvykdyti susijusių pranešimų ataką, tai gautas rezultatas yra dešifruotas pranešimas, tačiau sėkmingai dešifravus pranešimą nepavyks surasti privataus rakto. Ataka remiasi paprasta idėja, jog tas pats pranešimas *m* yra užšifruotas su ta pačia generuojančia matrica *G**, tačiau panaudojus skirtingus klaidos vektorius e_a yra gaunami šifrai c_a, kur $a \in \{1,2, \dots\}$. Kaip galima pastebėti, kriptoanalitikas žino, kad buvo užkoduotas tas pats pranešimas *msg.* Todėl pranešimų persiuntimo ataką, galima priskirti prie teksto – šifro porų atakos tipo.

Panagrinėkime pavyzdį, kai pranešimas *m* yra užšifruojamas su dviem skirtingais klaidos vektoriais ir parametrais: k = 524, n = 1024, t = 50.

Tokiu atveju gauname du šifrus:

$$c_1 = mSGP + e_1$$
$$c_2 = mSGP + e_2$$
$$e_1 \neq e_2.$$

Iš turimų dviejų šifrų sudarome dvi sekas L_0 ir L_1. Seka L_0 yra sudaryta iš pozicijų numerių, kuriose $(c_1 + c_2)$ yra nuliai. Kita seka L_1 yra sudaryta iš pozicijų numerių, kuriose $(c_1 + c_2)$ yra vienetai. Thomas A. Berson savo publikacijoje sekas apibrėžia taip:

$$L_0 = \{l \in \{1,2, \dots ,1024\} : c_1(l) + c_2(l) = e_1(l) + e_2(l) = 0\}.$$
$$L_1 = \{l \in \{1,2, \dots ,1024\} : c_1(l) + c_2(l) = e_1(l) + e_2(l) = 1\}.$$

Iš šių dviejų sekų gauname, jog tikėtina, kad $l \in L_0$, tada, kai $c_1(l)$ ir $c_2(l)$ nebuvo pakeisti klaidos vektoriai, tačiau $l \in L_1$, kai vienas iš $c_1(l)$ arba $c_2(l)$ yra pakeistas klaidos vektorius. Tokiu atveju kiekvienas $l \in L_0$ reiškia, jog $e_1(l) = e_2(l) = 0$ arba $e_1(l) = e_2(l) = 1$. Jeigu sakysime, jog e_1 ir e_2 klaidos vektoriai buvo pasirinkti nepriklausomai, tada tikimybė, kad klaidos vektoriai $e_1(l) + e_2(l) = 1$ yra apskaičiuojama taip:

$$P(e_1(l) = e_2(l) = 1) = \binom{t}{n}^2 \text{ [Ber97][EOS06] ,}$$

kur t – klaidų skaičius, kuris gali būti klaidos vektoriuje, n – matricos stulpelių skaičius.

Jeigu naudotume prieš tai aprašytus standartinius parametrus (k = 524, n = 1024, t = 50), tuomet tikimybė, kad $e_1(l) + e_2(l) = 1$ būtų:

$$\binom{50}{1024}^2 \approx 0{,}0024.$$

Tuomet, pagal gautą tikimybę galime spręsti, kad dauguma $l \in L_0$. Taip pat, pagal nagrinėtus šaltinius galima apskaičiuoti kiek operacijų reikia, kad pavyktų įveikti pranešimą:

$$w = \alpha * \frac{\binom{n}{k}}{\binom{n-t}{k}} \text{ [Ber97],}$$

kur t – galimas klaidų skaičius klaidos vektoriuje,

α – operacijų skaičius, per kurį apverčiama k dimensijos matrica,

n – šifro ilgis,

k – pradinio pranešimo ilgis.

5.4. Susijusių pranešimų ataka

Pranešimo persiuntimo ataka yra panaši į susijusių pranešimų ataką ir taip pat galima priskirti prie teksto – šifro porų atakos tipo. Ši ataka yra nagrinėjama Thomas A. Berson ([Ber97]) ir Kazukuni Kobara, Hideki Imai ([KI01]) publikacijose.

Tarkime turime du šifrus c_1 ir c_2:

$$c_1 = m_1 SGP + e_1$$
$$c_2 = m_2 SGP + e_2$$
$$m_1 \neq m_2, e_1 \neq e_2. \text{ [Ber97].}$$

Kriptoanalitikas nežino m_1 ir m_2 pranešimų, tačiau jis žino jų sumą $m_1 + m_2$. Žinodami pranešimų sumą ir užšifruotus pranešimus, kriptoanalitikas gauna lygtį:

$$c_1 + c_2 = m_1 SGP + m_2 SGP e_1 + e_2.$$

Tada kriptoanalitikui reikia spręsti lygtį:

$$c_1 + c_2 + (m_1 + m_2)SGP = e_1 + e_2\, z.$$

Kaip galime pastebėti iš lygties, kad kriptoanalitikas gali sužinoti iškraipytų bitų pozicijas iš $c_1 + c_2 + (m_1 + m_2)SGP$. Spręsdamas lygtį kriptoanalitikas gali taikyti ataką, kuri yra aprašyta 5.3 poskyryje, vietoje $(c_1 + c_2)$ naudodamas $c_1 + c_2 + (m_1 + m_2)SGP$ [Ber97][KI01].

6. Atakų prieš McEliece kriptografinę sistemą tyrimas

Remiantis perskaityta ir išanalizuota literatūra, kurios apžvalga pateikta ankstesniuose skyriuose buvo pasirinktos atakos ir atlikta jų analizė. Išnagrinėjus kiekvienos atakos realizacijos aspektus, jos yra realizuotos. Taip pat reikėtų paminėti, kad pranešimo šifravimą pademonstruoti dokumente, kuomet kriptografinės sistemos parametrai yra pakankamai dideli yra praktiškai neįmanoma, dėl per didelės skaičiavimo apimties, todėl reikės realizuoti pačią McEliece kriptografinę sistemą. McEliece kriptografinės sistemos realizacija nėra esminis darbo pagrindas, todėl realizacijai yra panaudota modifikuota FlexiProvider[5] implementacija. FlexiProvider biblioteka yra modifikuota, kad būtų galima gauti viešąjį raktą. Taip pat FlexiProvider bibliotekos panaudojimą palengvino pateikta išsami bibliotekos dokumentacija.

Toliau aptariant atakų realizaciją, reikėtų paminėti, kad gautiems rezultatams nedaro įtakos, kokia programavimo kalba yra realizuotos atakos, nes realizuojant jas ta pačia programavimo kalba nei viena realizuota ataka neįgaus pranašumo dėl pasirinktos technologijos. Priešingai nutiktų, jei atakos būtų realizuotos skirtingomis programavimo kalbomis. Taigi dėl programavimo kalbos populiarumo ir taikytos kriptografinės sistemos FlexiProvider bibliotekos, realizacijai buvo pasirinkta Java programavimo kalba.

Vienas iš išsikeltų darbo uždavinių buvo realizuoti ir įvykdyti atakas. Tačiau su saugiais kriptografinės sistemos parametrais to padaryti neišeis, todėl dėl riboto laiko sąnaudų naudojami tokie parametrai, su kuriais kriptografinės sistemos nulaužimas trunka pakankamai priimtiną laiką (mažiau nei parą). Atakų įgyvendinimui naudojamas paprastas stacionarus kompiuteris, kurio parametrai:

- Procesorius: Intel(R) Core (TM) i5-4570 3.20 GHz;
- Adresuojama atminti: 8 GB;
- Operacinė sistema: 64 – bit Windows 7 Professional;
- Java versija: 1.7.0_67.

Statistiniai duomenys yra pateikiami grafikuose arba lentelėse (pavyzdys pateiktas žemiau):

1lentelė. Pavyzdinė duomenų lentelė.

Parametras m				
Parametras t				

[5] https://www.flexiprovider.de/download.html

Vykdymo laikas				

Surinktus duomenis reikės įvertinti ir pamėginti įvertinti, koks atakos vykdymo laikas būtų su realiais naudojamais parametrais (m, t), kuomet $m \geq 10$. Detalesni realizuotu atakų aprašymai yra pateikiami kituose skyriuose. Vertėtu atkreipti dėmesį, kad visos realizuotos atakos yra paleidžiamos iš pradinės programos, kurioje leidžiama pasirinkti norimą ataką. Pasirinktus ataką yra kviečiama atakos programa, kurioje yra užšifruojamas pranešimas ir iškviečiamas atakos algoritmas.

6.1. Kriptografinės sistemos saugumas

Tyrimo metu McEliece kriptografinės sistemos parametrus laikysime saugiais, jeigu šifro nepavyksta įveikti per 100 metų naudojant superkompiuterį. Teoriniams skaičiavimams remsimės šiuo metu greičiausiu pasaulio superkompiuterio Tianhe-2[6] charakteristika. Tianhe -2 superkompiuteris gali atlikti 33.86 petaflops[7], kai tyrimo metu naudotas kompiuteris gali atlikti 16.27[8] Gflops. Žinodami šiuos parametrus galime apskaičiuoti, kiek kartų superkompiuteris greičiau atliks veiksmus už tyrime naudota kompiuterį:

$$\frac{33.86 * 10^{15}}{16.27 * 10^{9}} = 2.0811 * 10^{6}.$$

Tačiau reikėtu nepamiršti, kad McEliece kriptografinės sistemos saugumas yra vertinamas tik nagrinėjamų atakų požiūriu ir su kitomis gauti rezultatai gali skirtis.

6.2. Apibendrinta informacijos dekodavimo atakos realizacija

Remiantis ankstesniame skyriuje (žr. 5.1. poskyrį) atlikta atakos analize yra realizuota apibendrintos informacijos dekodavimo ataka. Realizuotoje programoje yra sukurti du esminiai metodai, kurių vienas užšifruodavo dvejetainį pranešimą, o kitas bandydavo užšifruotą pranešimą įveikti.

Pirmo metodo realizaciją sudarė dvi esminės dalys. Pirmoje dalyje, kaip prieš tai minėta, yra panaudota FlexiProvider biblioteka, kurios pagalba yra sugeneruojamas viešasis raktas. Antroje metodo dalyje yra kviečiamas kitas metodas, kuris sugeneruodavo atsitiktinį dvejetainį

[6] http://www.extremetech.com/extreme/218078-chinas-tianhe-2-is-still-the-worlds-fastest-supercomputer-but-cray-is-on-a-resurgence
[7] Slankiojo kablelio operacijų per sekundę (http://whatis.techtarget.com/definition/FLOPS-floating-point-operations-per-second)
[8] https://setiathome.berkeley.edu/cpu_list.php

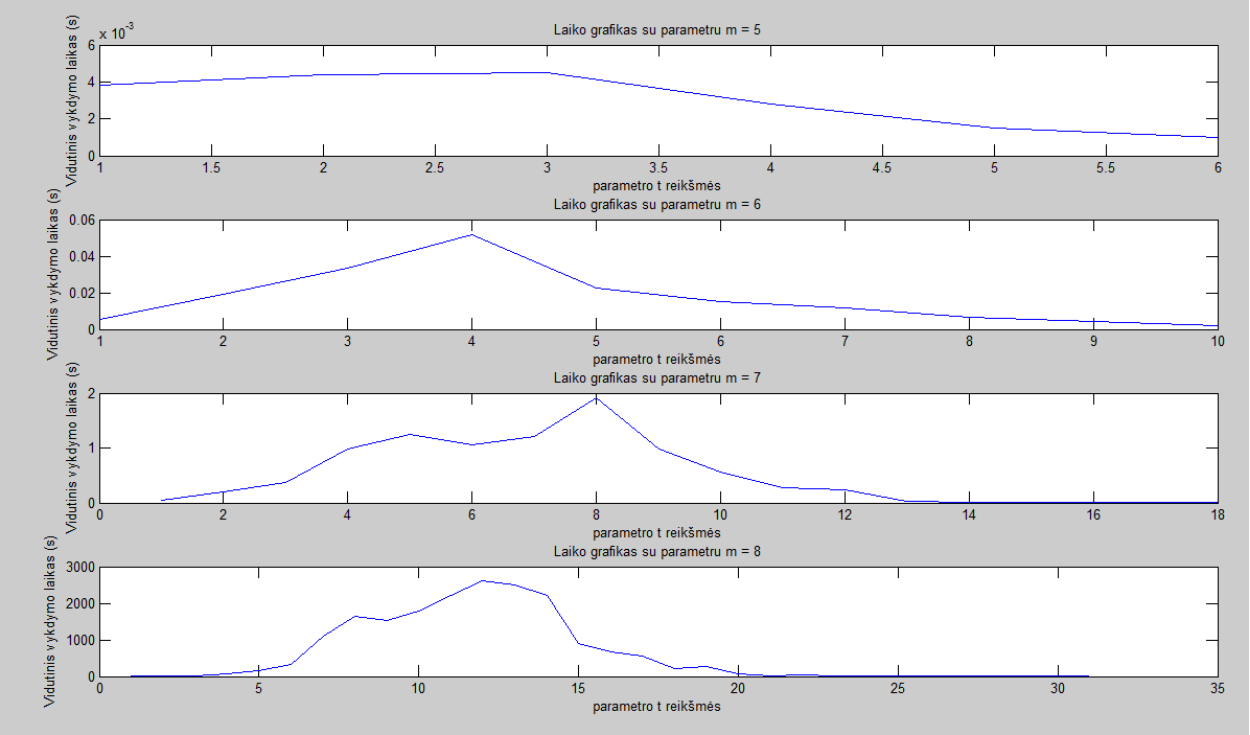
pranešimą. Gautas pranešimas, sugeneruotas viešasis raktas ir klaidos vektorius yra paduodamas į FlexiProvider biblioteką, kuriame jis užšifruojamas. Vartotojui nieko nebereikia perduoti į atakos metodą, nes užšifruotas pranešimas su viešu raktu ir parametrais n, k, t perduodamas į antrą metodą iš pirmo.

Antras metodas realizuotas pagal prieš tai pateiktą algoritmo aprašymą. Visų pirma iškviečiamas metodas, kuris pagal parametro n ilgį, sugeneruodavo sąrašą n pozicijų. Pagal sugeneruotas pozicijas pasirenkama k stulpelių ir įsirašoma į pasirinktų stulpelių sąrašą, o likusius stulpelius į nepasirinktų stulpelių sąrašą. Pagal pasirinktus stulpelius yra sugeneruojama matricą $\hat{G}_k$. Generuojant matricą yra tikrinama, ar generuojamos matricos eilutės nėra sudarytos iš nulių. Jeigu yra, tai pakeičiamas vienas stulpelis. Sugeneravus matricą yra apskaičiuojamas vektorius w, pagal algoritme pateiktą formulę. Jeigu apskaičiuoto w vektoriaus Hammingo svoris yra lygus klaidų skaičiui t, tai užšifruotas pranešimas įveiktas. Tačiau situacija pasikeičia, jei Hammingo svoris nėra lygus klaidų skaičiui. Tada, patenkama į kitą algoritmo dalį, kurioje yra du ciklai. Pirmas ciklas yra vykdomas iki kintamojo j, kuris yra pasirinktas 1. Iš to gauname, kad pirmasis ciklas yra vykdomas tik vieną kartą. Tačiau j galima pasirinkti ir didesnį. Antrasis ciklas yra vykdomas iki kombinatorinio derinio $\binom{n}{i}$ reikšmės. Kombinatorinio derinio skaičiavimui yra realizuotas metodas, kuris jį apskaičiuoja. Tačiau, kaip galima pastebėti, jei pirmojo ciklo kintamojo j reikšmė yra 1, tai antrasis ciklas yra vykdomas n kartų. Kai yra vykdomas ciklas jame yra pasirenkamas toks naujas vektorius w_k, kad Hammingo svoris būtų lygus i. Pagal parinktą vektorių w_k yra apskaičiuojama $w \oplus w_k \hat{G}_k$ išraiškos Hammingo svoris. Jeigu Hammingo svoris yra lygus klaidų skaičiui *t*, tuomet sąlyga tenkinama ir yra dešifruojamas pranešimas, o priešingu atveju yra parenkamas kitas vektorius. Tačiau galime ir nerasti tokio vektoriaus, tokiu atveju yra viskas kartojama iš naujo pasirenkant kitus k stulpelius.

Be apibendrintos informacijos, dekodavimo atakos realizacijos, yra realizuota ir kita programa. Realizuotoje programoje yra skaičiuojamas teorinis optimalus parametras t kiekvienam m parametrui. Programą sudaro 3 esminės dalys. Pirmoje ir antroje dalyje yra realizuotos atakos aprašyme pateiktos formulės (5.1-3) ir (5.1-5). Taip pat, kaip ir apibendrintos informacijos dekodavimo atakoje, yra realizuotas kombinatorinio derinio skaičiavimas. Tačiau skaičiaus faktorialo skaičiavimui galėjo nebeužtekti *long* tipo kintamojo, todėl yra panaudotas *BigDecimal* tipo kintamasis.

6.3. Apibendrintos informacijos dekodavimo atakos rezultatai

Šiame skyriuje aptariama apibendrintos informacijos dekodavimo atakos rezultatai, keičiant su parametrais, m ir t. Apibendrintos informacijos dekodavimo atakų su skirtingais parametrais buvo atlikta 10 skirtingų eksperimentų ir paimtas eksperimentų vykdymo laikų vidurkis. Vidutiniai vykdymo laikai yra pateikiami grafiškai, pagal prieduose pateiktą vykdymo laikų lentelę (lentelė 13).



1 pav. Praktiniai vidutiniai vykdymo laikai kiekvienam m parametrui.

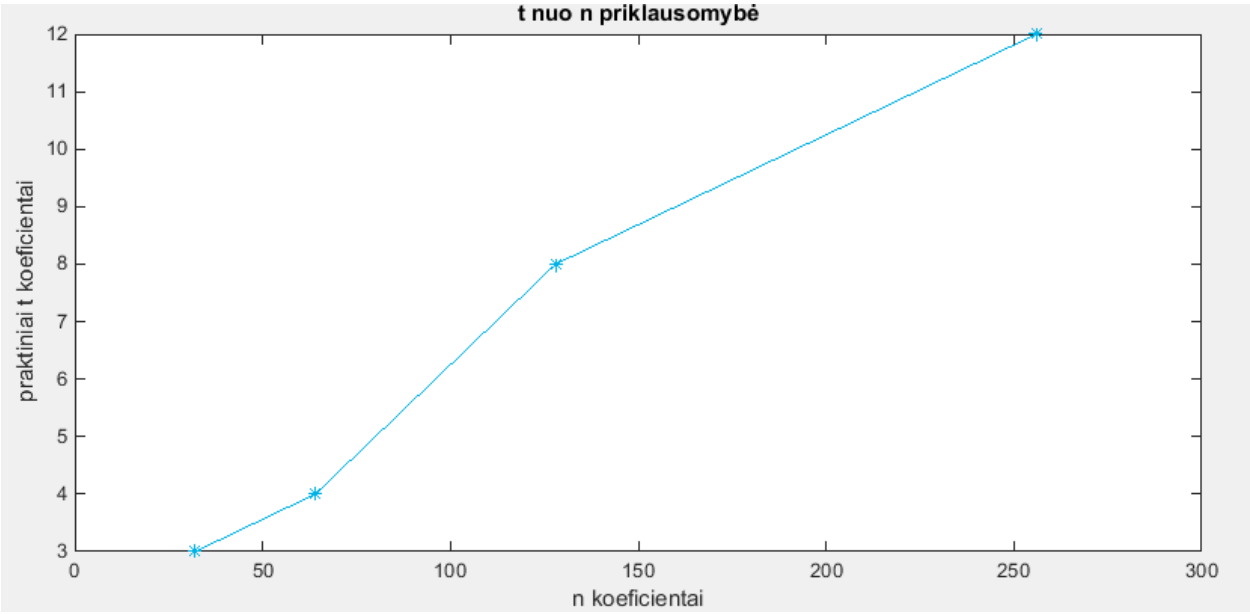
Pagal gautus vidutinius vykdymo laikus (žr. 13 lentelę), yra nustatyti optimalūs parametrai *t* kiekvienam *m* parametrui ir sudarytos optimalaus t parametro priklausomybės nuo parametro *m* (lentelė 2). Kurioje matosi, kokią *t* reikšmę parametrui *n* reikia rinktis, kad kriptografinė sistema būtų saugiausia.

2 lentelė. GISD atakos optimalios t reikšmės ir vid. vykdymo laiko priklausomybė nuo parametro m.

m	5	6	7	8
n	32	64	128	256
Vykdymo laikas (s)	0.0045	0.0518	1.9223	2607.913
t (optimalus)	3	4	8	12

Iš sudarytos lentelės (žr. 2 lentelę) yra sudarytas grafikas (žr. 2 pav.), pagal lentelę ir grafiką buvo išsikelta hipotezė, kad t priklausomybę nuo n būtų galima suvesti į tiesinį pavidalą. Pagal išsikeltą hipotezę buvo pasinaudota mažiausių kvadratų metodu, kad rastume tiesinę lygtį, kurios pavidalas yra:

$$y = A * x + b. \tag{6.3-1}$$



2 pav. Praktinių optimalių t parametrų priklausomybė nuo n.

Apibrėžta lygtis nusako tiesinės regresijos modelį, kur x ir y yra kintamieji, o a ir b parametrai, kurie apibūdina tam tikrą sistemą. Sistemos pagrindinis uždavinys yra gauti nežinomų a ir b parametrų reikšmes, naudojant kintamųjų matavimo reikšmes. Pagal lentelės (žr. 2 lentelę) duomenis buvo sudaryta lygčių sistema:

$$\begin{cases} 4 = 64 * a + b \\ 8 = 128 * a + b \\ 12 = 256 * a + b \\ 3 = 32 * a + b \end{cases} \tag{6.3-2}$$

Sudarytą lygčių sistemą perrašome matriciniu pavidalu:

$$\begin{bmatrix} 32 & 1 \\ 64 & 1 \\ 128 & 1 \\ 256 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 8 \\ 12 \end{bmatrix}. \tag{6.3-3}$$

Gautą matricinį pavidalą išskaidome į kintamuosius:

$$A = \begin{bmatrix} 32 & 1 \\ 64 & 1 \\ 128 & 1 \\ 256 & 1 \end{bmatrix}, b = \begin{bmatrix} 3 \\ 4 \\ 8 \\ 12 \end{bmatrix}, x^* = \begin{bmatrix} a \\ b \end{bmatrix}, \tag{6.3-4}$$

kur matrica *A* yra vadinama regresorių matrica, vektorius *b* – priklausomu kintamuoju, o parametro x^* elementai – regresijos koeficientais.

Užrašoma normalinė lygtis:

$$A^T A x^* = A^T b, \tag{6.3-5}$$

kur

$$A^T A = \begin{bmatrix} 32 & 64 & 128 & 256 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 32 & 1 \\ 64 & 1 \\ 128 & 1 \\ 256 & 1 \end{bmatrix} = \begin{bmatrix} 87040 & 480 \\ 480 & 4 \end{bmatrix}, \tag{6.3-6}$$

$$A^T b = \begin{bmatrix} 32 & 64 & 128 & 256 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 8 \\ 12 \end{bmatrix} = \begin{bmatrix} 4448 \\ 27 \end{bmatrix}. \tag{6.3-7}$$

Įsistatę į normalinę lygtį (6.3-5) koeficientus (6.3-6) ir (6.3-7) gauname:

$$\begin{bmatrix} 87040 & 480 \\ 480 & 4 \end{bmatrix} * x^* = \begin{bmatrix} 4448 \\ 27 \end{bmatrix}. \tag{6.3-8}$$

Jeigu į gautą matricinį sistemos pavidalą (6.3-8) įsistatome x^* (6.3-4) koeficientus:

$$\begin{bmatrix} 87040 & 480 \\ 480 & 4 \end{bmatrix} * \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 4448 \\ 27 \end{bmatrix}. \tag{6.3-9}$$

Gauname matricinį pavidalą (6.3-9), iš kurio galime užrašyti lygčių sistemą:

$$\begin{cases} 87040 * a + 480 * b = 4448 \\ 480 * a + 4 * b = 27 \end{cases}. \tag{6.3-10}$$

Išsprendę lygčių sistemą (6.3-10) gauname koeficiento a = 0.0413 ir b = 1.5 reikšmes. Jas įsistatę į (6.3-1) lygtį, gauname t priklausomybės nuo n tiesinę lygtį:

$$\mathrm{y} = 0.0408 * x + 1.6087. \tag{6.3-11}$$

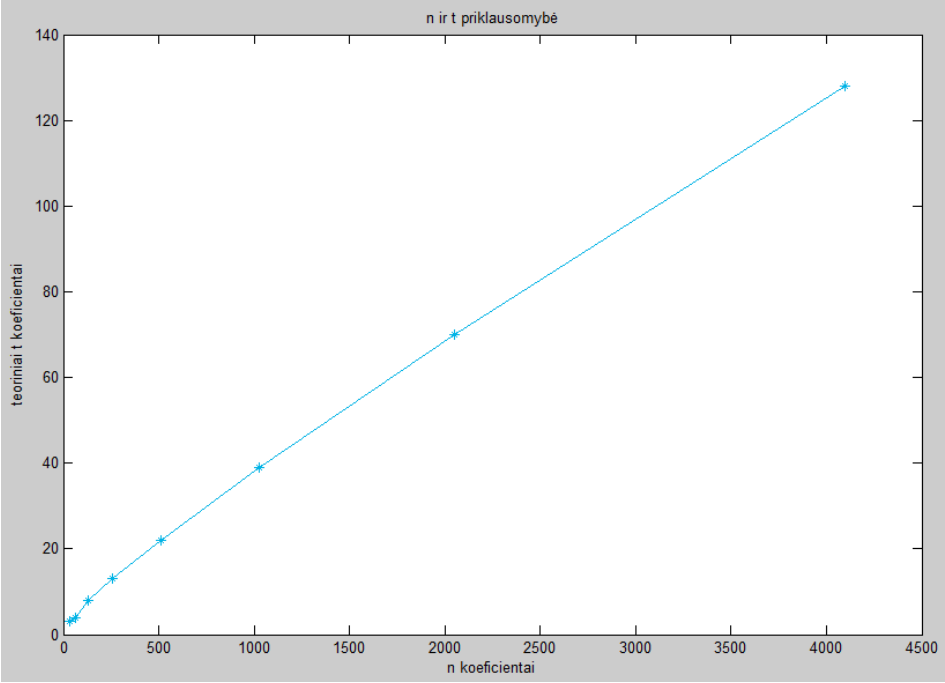
Pasinaudoję gautąja lygtimi (6.3-11), apskaičiuojame optimalius t koeficientus didesniems n parametrams. Pagal gautus optimalius koeficientus galima spręsti ar McEliece kriptografinė sistema yra saugi. Taip pat pagal formulę (5.1-7) buvo realizuota WorkFactor programa. Realizuotoje programoje, pagal gautą dvejetainį operacijų skaičių buvo atrenkamas optimaliausias teorinis koeficientas t, kuris yra nepriklausomas nuo tyrimui naudotų resursų.

Iš gautų praktinių (žr. 2 lentelę) ir prognozuotų t koeficientų buvo palyginimui sudaryta lentelė su apskaičiuotomis teorinėmis optimaliomis t reikšmėmis:

3 lentelė. Optimalių t reikšmių palyginimas

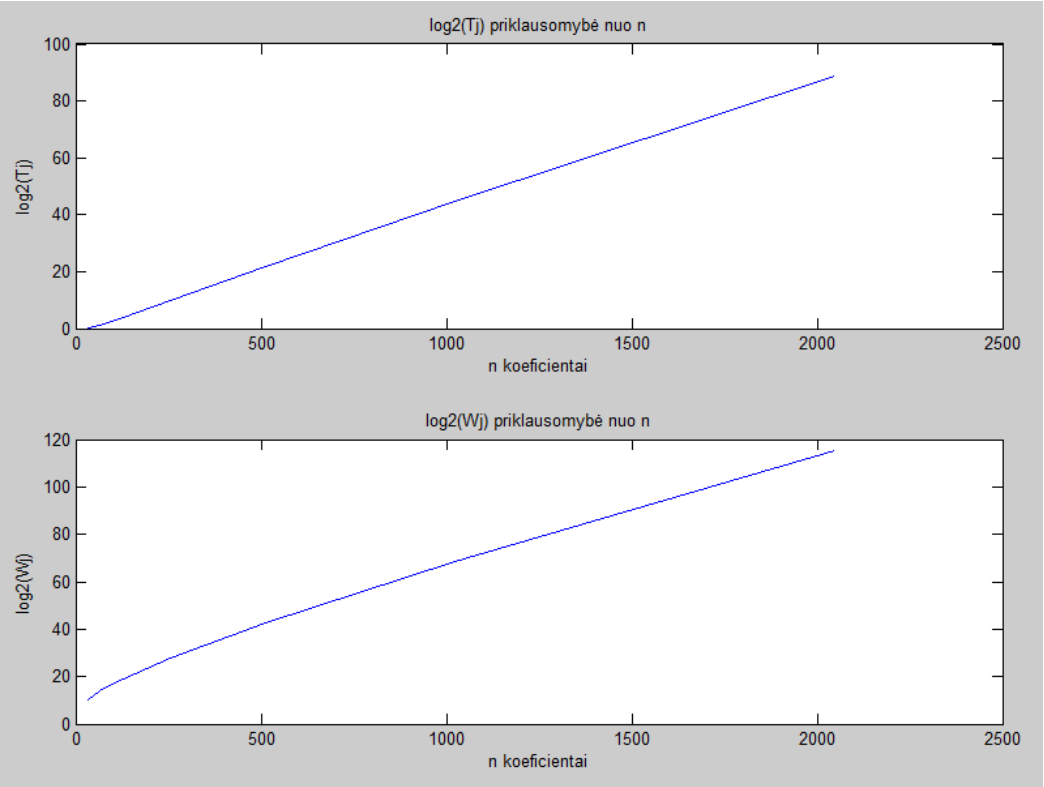
Parametras n	Optimalios t reikšmės		
	Praktiniai	Prognozuoti	Teoriniai
32	**3**	3	3
64	**4**	4	5
128	**8**	8	8
256	**12**	12	13
512	-	22	22
1024	-	43	39
2048	-	85	70
4096	-	168	128

Pagal gautus lentelės (žr. 3 lentelę) teorinius duomenis, buvo sudarytas t priklausomybės nuo n grafikas (žr. 3 pav.).



3 pav. Teorinė t koeficiento priklausomybė nuo parametro n.

Iš gautų rezultatų galima pamatyti, kad t priklausomybė nuo n yra panaši į laipsninę funkciją, todėl prieš tai iškelta hipotezė yra paneigta. Dėl didelės paklaidos prognozuotose t reikšmėse, tolesniems tyrimams buvo naudojamos gautos optimalios t teorinės reikšmės. Pagal teorines optimalias t reikšmes prognozuosime apibendrintos informacijos dekodavimo atakos algoritmo antro žingsnio įvykdymo T_j kartojimų ir dvejetainių operacijų skaičiaus priklausomybę nuo n grafikus (žr. *4* pav.).



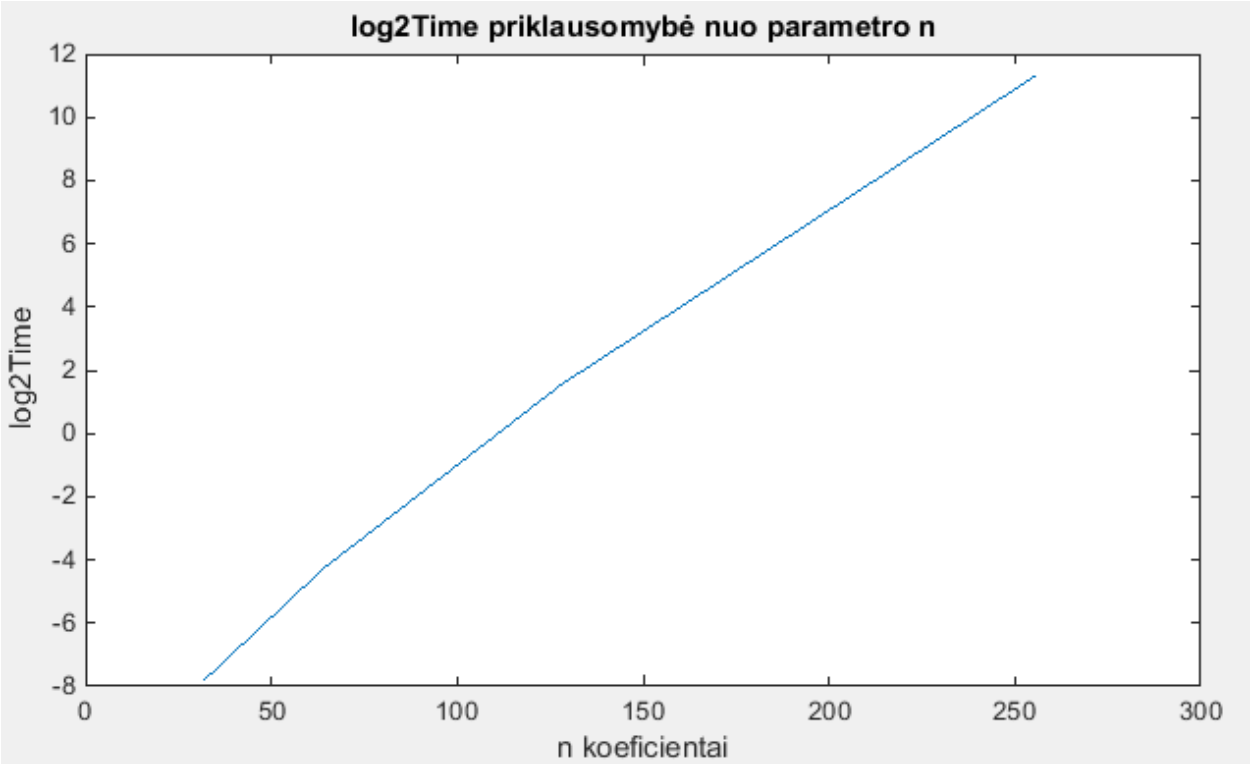
4 pav. log2(Tj) ir log2(Wj) priklausomybės nuo n parametro.

Pagal gautus grafikus galime pamatyti, kad $log_2 T_j$ n priklausomybė yra tiesinė. Pasinaudoję mažiausių kvadratų metodu gauname lygtį:

$$log_2 T_j = 0.0422 * n - 1.3717.$$

Tačiau laikas yra priklausomas nuo dvejetainių operacijų skaičiaus. Iš pateikto log2(Wj) priklausomybės nuo n grafiko (žr. 4 pav.) matome, kad priklausomybė yra šiek tiek laipsninė.

Atlikus ataką gavome vidutinius vykdymo laikus su kiekvienu m parametro t reikšmėmis. Iš gautų vidutinių vykdymo laikų lentelės (lentelė 13) buvo išrinkti didžiausi laikai, kurie yra pateikiami lentelėje (žr. 2 lentelę). Iš gautų duomenų gauname grafiką, kuriame yra pateikta $\log_2 Time$ priklausomybė nuo parametro n.



5 pav. Apibendrintos informacijos dekodavimo atakos laiko log_2 priklausomybė nuo parametro n.

Paprastumo dėlei tarkime, kad vidutinio vykdymo laiko logaritmas yra tiesiškai priklausomas nuo n. Tokiu atveju, pagal gautus rezultatus ir pasinaudojus mažiausių kvadratų metodu galima sudaryti tiesinę lygtį:

$$y_{Log_2^{Time}} = 0.0843 * n - 9.9055.$$

Gauname lygtį, kurią naudojant galima apskaičiuoti vidutinius vykdymo laikus didesniems n parametrams.

4 lentelė Apibendrintos informacijos dekodavimo atakos vykdymo laikų prognozavimas su superkompiuteriu

Parametras n	Vykdymo laikai		
	Praktiniai	Prognozuoti asmeniniam kompiuteriui	Prognozuoti superkompiuteriui
32	**0.0045 (s)**	0.0061 (s)	2.9547e-09 (s)
64	**0.0518 (s)**	0.0397 (s)	1.9087e-08 (s)
128	**1.9223 (s)**	1.6576 (s)	7.9648e-07 (s)
256	**2607.913 (s)**	2886.316 (s)	0.0013869 (s)
512	-	277.51 (metų)	4205.3393 (s)
1024	-	$2.55 * 10^{15}$ (metų)	$1.23 * 10^{9}$ (metų)
2048	-	$2.16 * 10^{41}$ (s)	$1.04 * 10^{35}$ (metų)

4096	-	$1.54 * 10^{93}$ (s)	$7.4 * 10^{86}$ (metų)

Įvertinę gautus rezultatus, galima teikti, kad šiuo metu pasirinkus kriptografinės sistemos parametrą $n \geq 1024$, kriptografinę sistemą galima laikyti saugia prieš apibendrintos informacijos dekodavimo ataką.

6.4. Žinomo dalinio teksto atakos realizacija

Šio tipo atakos realizacija praktiškai nesiskyrė nuo apibendrintos informacijos dekodavimo atakos. Žinomo dalinio teksto atakos realizaciją sudarė vienas esminis metodas iš trijų dalių. Pirmoje dalyje yra užšifruojamas pranešimas, kaip ir prieš tai minėtoje apibendrintos informacijos dekodavimo atakos realizacijoje. Antroje dalyje yra atliekamos gauto šifro ir pradinių duomenų transformacijos. Viena iš transformacijų yra ta, kad kriptoanalitikas žino dalį pranešimo, todėl reikia dešifruoti tik tą nežinomą dalį. Todėl po užšifravimo yra sumažinamas parametras k, iš jo atimant žinomų bitų skaičių ir gaunamas naujas k_l parametras. Taip pat pagal nežinomų bitų skaičių yra sukonstruojamas naujas viešas raktas, kurį sudaro k_l eilučių ir galiausiai trečioje dalyje yra apskaičiuojamas naujasis šifras *c'*.

Anksčiau buvo minėta (žr. 5.2. poskyrį), kad ši ataka negali dešifruoti pranešimo, o tik sumažina parametrus, kuriuos galima naudoti su kitomis atakomis. Tokiu atveju metodo trečioje dalyje yra panaudota apibendrintos informacijos dekodavimo ataka, kurioje yra perduodami transformuoti duomenis. Dėl sumažintų pradinių pranešimų, apibendrintos informacijos dekodavimo atakai reikėtų mažiau laiko įveikti šifrą.

6.5. Žinomo dalinio teksto atakos rezultatai

Atlikus ataką ir gavus rezultatus, reikėtų pažymėti, jog žinomo dalinio teksto atakos sėkmės tikimybė turėtų didėti, o vykdymo laikas mažėti, turint didesnį žinomą dalinį tekstą. Iš žemiau pateiktos lentelės galima matyti, kaip priklauso atakos vykdymo laikas nuo žinomo dalinio teksto dydžio su optimaliomis t reikšmėmis.

5 lentelė. Žinomo dalinio teksto atakos vidutiniai vykdymo laikai (s)

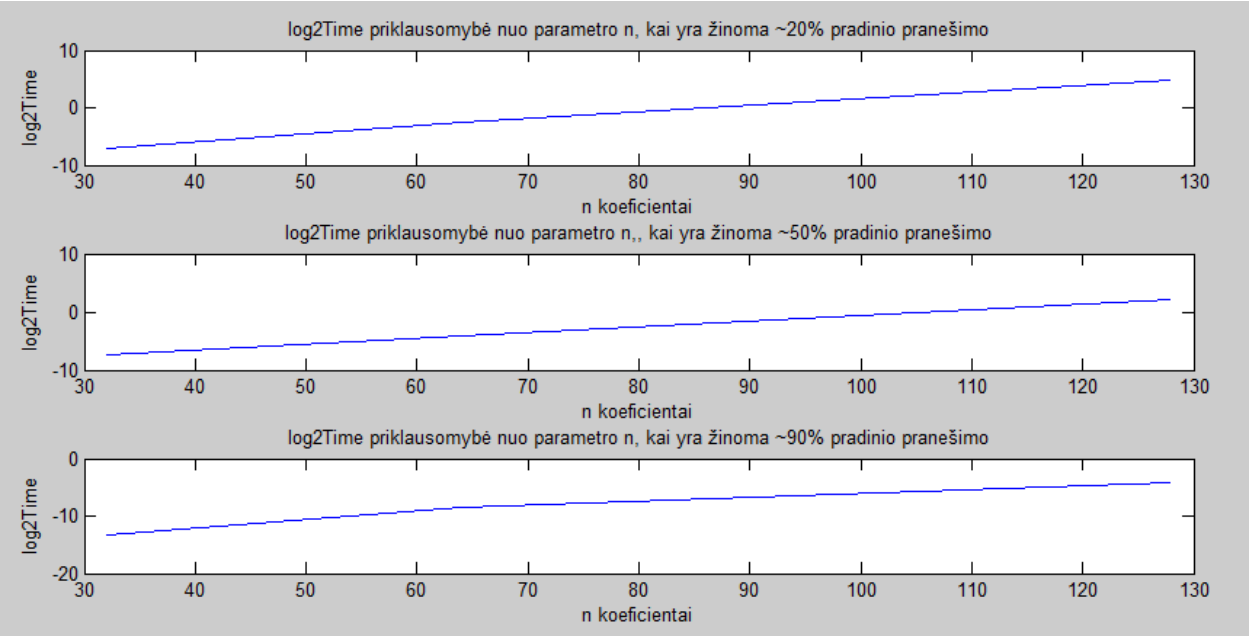
Žinoma teksto dalis	(m, t)			
	(5,3)	(6,5)	(7,8)	(8,13)
~20 %	0.007	0.1684	29.115	-
~50 %	0.0064	0.0561	4.2778	-[9]

[9] - reikšmės nepavyko surasti per priimtina laiką.

~90 %	0.0001	0.003	0.0519	10.3131

Iš gautų rezultatų (žr. 5 lentelę) galima teigti, kad kriptosistemos įveikimo laikas priklauso nuo žinomos pranešimo dalies. Tačiau reikėtų atkreipti dėmesį į tai, kad su kai kuriais parametrais kriptografinės sistemos įveikimas užtrunka ilgiau negu naudojant vien tik apibendrintos informacijos dekodavimo ataką. Tokį netikėtą rezultatą galėjo nulemti algoritme skaičiuojama $\hat{G}_k$ matrica. Šiame matricos skaičiavime būtų galima panaudoti Gauss Eliminavimo metodą, kuris minimas [KI02] algoritmo aprašyme. Panaudojus metodą nebebūtų bandoma apversti neapverčiamos matricos, kuri susidaro apibendrintos informacijos dekodavimo atakos algoritme pasirinkus *k* stulpelių.

Remiantis gautais rezultatais buvo sudaryti grafikai (žr. 6 pav.), iš kurių matoma, kad laiko logaritmo priklausomybė nuo n yra tiesinė:



6 pav. Žinomo dalinio teksto atakos grafikas, su skirtingomis žinomo pradinio teksto dalimis.

Pagal duomenis buvo apskaičiuoti tiesinės lygties koeficientai su skirtingais žinomo dalinio teksto kiekiais:

$$y_{20} = 0.1239 * n - 10.8753;$$

$$y_{50} = 0.0977 * n - 10.4141;$$

$$y_{90} = 0.0897 * n - 15.3441.$$

Remiantis gautomis tiesinėmis lygtimis, buvo prognozuoti vykdymo laikai, su didesnėmis n parametro reikšmėmis.

6 lentelė Žinomo dalinio teksto atakos vykdymo laikų prognozavimas žinant 20 % pradinio teksto

Parametras n	Žinoma teksto dalis %	Vykdymo laikai		
		Praktiniai	Prognozuoti asmeniniam kompiuteriui	Prognozuoti superkompiuteriui
32	**20**	0.007 (s)	0.008318 (s)	$3.9973 * 10^{-9}$ (s)
64	**20**	0.1684 (s)	0.1299 (s)	$6.2461 * 10^{-8}$ (s)
128	**20**	29.115 (s)	31.739 (s)	$1.5251 * 10^{-5}$(s)
256	**20**	-	$1.89 * 10^{6}$ (s)	0.90926 (s)
512	**20**	-	$2.13 * 10^{8}$ (metų)	102.48 (metų)
1024	**20**	-	$2.69 * 10^{27}$ (metų)	$1.29 * 10^{21}$ (metų)
2048	**20**	-	$4.3 * 10^{65}$ (metų)	$2.07 * 10^{59}$ (metų)
4096	**20**	-	$1.096 * 10^{142}$ (metų)	$5.27 * 10^{135}$ (metų)

Kriptoanalitikui žinant ~20 % pradinio pranešimo, kriptografinė sistema galima laikyti saugia (žr. 6 lentelę), kai sistemos parametras $n \geq 512$. Tačiau, kai kriptoanalitikas žino ~50 % pradinio pranešimo, tai naudoti kriptografinės sistemos parametrą $n = 512$ yra nebe patartina (žr. 7 lentelę). Todėl, jei numanoma, kad kriptoanalitikas gali žinoti ~50 % siunčiamo pranešimo, tai vertėtų padidinti tuo metu naudojamos kriptografinės sistemos parametrus $n \geq 1024$ ir t (žr. 3 lentelę).

7 lentelė Žinomo dalinio teksto atakos vykdymo laikų prognozavimas žinant 50 % pradinio teksto

Parametras n	Žinoma teksto dalis %	Vykdymo laikai		
		Praktiniai	Prognozuoti asmeniniam kompiuteriui	Prognozuoti superkompiuteriui
32	**50**	0.0064 (s)	0.006407 (s)	$3.0786 * 10^{-9}$ (s)
64	**50**	0.0561 (s)	0.056009 (s)	$2.6913 * 10^{-8}$ (s)
128	**50**	4.2778 (s)	4.2801 (s)	$2.0567 * 10^{-6}$ (s)
256	**50**	-	24995.53 (s)	0.012011 (s)
512	**50**	-	27031.3 (metų)	409619 (s)
1024	**50**	-	$3.14 * 10^{19}$ (metų)	$1.51 * 10^{13}$ (metų)
2048	**50**	-	$4.25 * 10^{49}$ (metų)	$2.04 * 10^{43}$ (metų)
4096	**50**	-	$7.78 * 10^{109}$ (metų)	$3.74 * 10^{103}$ (metų)

Taigi, kai kriptoanalitikas žino ~90 % pradinio pranešimo (žr. *8* lentelę), tai kriptografinę sistemą galima laikyti saugia, naudojant kriptografinės sistemos parametrą $n \geq 1024$.

8 lentelė Žinomo dalinio teksto atakos vykdymo laikų prognozavimas žinant 90 % pradinio teksto

Parametras n	Žinoma teksto dalis %	Vykdymo laikai		
		Praktiniai	Prognozuoti asmeniniam kompiuteriui	Prognozuoti superkompiuteriui
32	**90**	0.0001 (s)	0.000176 (s)	$8.4504 * 10^{-11}$ (s)
64	**90**	0.003 (s)	0.0012864 (s)	$6.1812 * 10^{-10}$ 10
128	**90**	0.0519 (s)	0.0688 (s)	$3.3072 * 10^{-8}$ (s)
256	**90**	10.3131 (s)	197.028 (s)	$9.4675 * 10^{-5}$ (s)
512	**90**	-	51.2 (metų)	775.8673 (s)
1024	**90**	-	$3.44 * 10^{15}$ (metų)	$1.65 * 10^{9}$ (metų)
2048	**90**	-	$1.55 * 10^{43}$ (metų)	$7.45 * 10^{36}$ (metų)
4096	**90**	-	$3.15 * 10^{98}$ (metų)	$1.51 * 10^{92}$ (metų)

Vertėtų atkreipti dėmesį, kad kai yra nežinoma, kokią pradinio pranešimo dalį žino kriptoanalitikas, tai rekomenduojama naudoti kriptografinės sistemos parametrą $n \geq 1024$.

6.6. Pranešimo persiuntimo atakos realizacija

Sukurtoje programoje buvo realizuotas vienas esminis metodas, kurio logika susidėjo iš trijų dalių. Pirmoje metodo dalyje buvo užšifruojamas tas pats pranešimas su dviem skirtingais klaidos vektoriais. Užšifravimui buvo naudojama ta pati FlexiProvider biblioteka. Iš pirmo metodo dalies buvo gaunami du šifrai ir vienas viešasis raktas.

Antroje metodo dalyje, pagal algoritmo aprašymą yra apskaičiuojamos iškraipytos, klaidos vektoriaus pozicijos iš šifrų c_1 ir $c_{2,}$ sumos, tačiau tai verčiau galima vadinti bandymu, negu pranešimo sužinojimu, nes aptiktas šifras gali būti ir nepradinis. Visų pirma randamos šifrų pozicijos, kuriuose yra vienetai, kurių skaičius turi būti ne didesnis negu padvigubintas galimų klaidų kiekviename šifre skaičius. Priklausomai nuo apskaičiuoto, vektoriuje esančių vienetų, skaičiaus yra generuojamos iškraipytų klaidų pozicijų vektorius.

Tarkime, kad iškraipytų vektorių skaičius yra lygus dvigubai leistinų klaidų skaičiui, esančių šifruose. Tokiu atveju viename ir kitame šifre buvo parinktos tarpusavyje nesutampančios pozicijos. Tada aišku, kad likusios pozicijos yra neiškraipyti bitai ir tokiu atveju belieka iš

iškraipytų pozicijų atsitiktiniu būdu, pasirinkti pozicijas, kuriose iš dviejų vektorių yra klaidinga informacija.

Situacija pasikeičia, kai dviejų šifrų sumos vienetų skaičius nėra lygus dvigubam galimų klaidų skaičiui. Tuomet tai reiškia, kad kažkurioje ar net keliose šifrų pozicijose yra klaidingi duomenys. Tokiu atveju, priklausomai nuo sutampančių pozicijų skaičiaus, surandamos vietos, kur galima tiksliai pasakyti, jog ten yra klaidingi bitai. Klaidingų vektoriaus pozicijų spėjimai daromi tokiu pačiu principu, kai klaidų pozicijos nesutampa. Surasti likusius iškraipytus bitus bandoma atsitiktinai spėjant, kurioje(-iose) pozicijoje(-se) yra klaidingi bitai abiejuose šifruose. Tokių spėjimų skaičių nėra sudėtinga apskaičiuoti, jeigu yra žinomas šifro ilgis. Tarkime, kad šifro ilgis yra *n*, o klaidingų bitų skaičių vektoriuje pažymėtume raide *t*. Tuomet teigiant, kad yra tik *p* pozicijų, kuriose sutampa klaidingi bitai, spėjimų skaičius yra apskaičiuojamas: C_{n-t+p}^{p}.

Parinkus klaidos vektorių, toliau belieka apskaičiuoti užšifruotą pranešimą. Pradinio pranešimo skaičiavimui buvo nuspręsta naudoti tiesinių lygčių sistemos sprendimo būdą. Buvo nuspręsta naudoti Gauss – Jordan tiesinių lygčių sprendimo metodą, kurį teko modifikuoti, kad dirbtų ne su skaičiais, o su bitais. Paprastumo dėlei gautą atsakymą iš tiesinių lygčių metodo pažymėkime msg', jį užšifruojame ir sudedame su šifru. Jei gautos sumos Hammingo svoris yra lygus iškraipytų pozicijų skaičiui, tai vykdymą nutraukiame, nes rastas msg' yra tikrasis pranešimas. Priešingu atveju rastas msg' nėra pradinis pranešimas ir viską vykdome iš naujo.

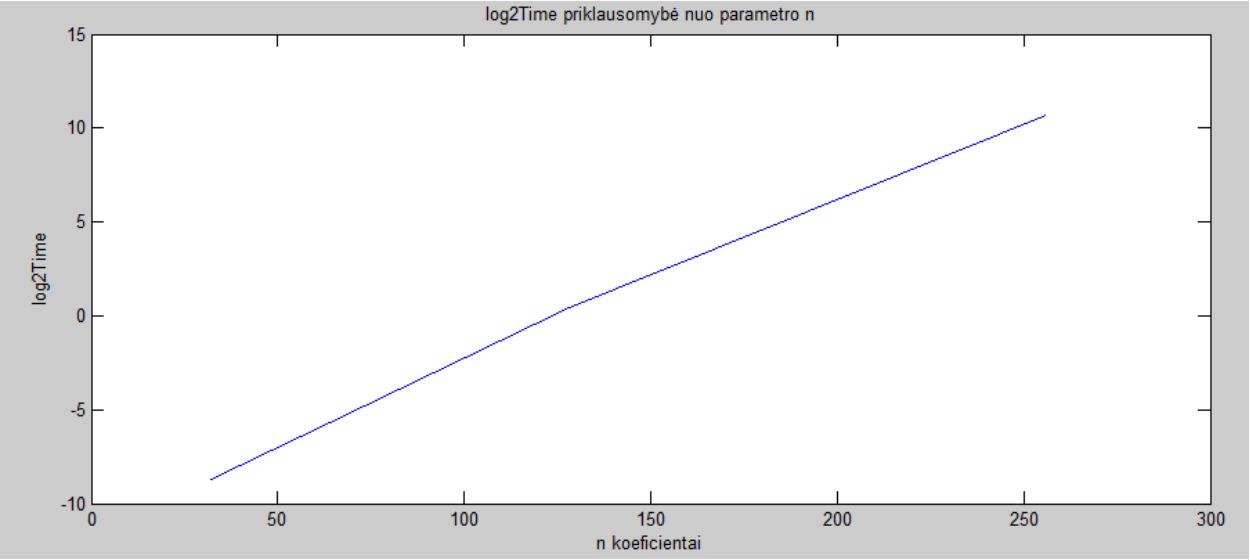
6.7. Pranešimo persiuntimo atakos rezultatai

Realizavus pranešimo persiuntimo ataką buvo atlikta 10 vienas nuo kito nepriklausomų eksperimentų. Su kiekvienu parametru m ir t yra sudaryta vidutinių vykdymų laikų lentelė.

9 lentelė. Pranešimo persiuntimo atakos vykdymo laikai

Parametrai	**m = 5, t = 3**	**m = 6, t = 4**	**m = 7, t = 8**	**m = 8, t = 13**
Vidutinis vykdymo laikas (s)	0.0024	0.0192	1.359	1664.5912

Pagal gautus vykdymo laikų rezultatus yra sudarytas grafikas:



7 pav. Pranešimo persiuntimo atakos laiko log_2 priklausomybė nuo parametro n.

Remiantis gautais rezultatais, pagal kuriuos buvo sudarytas grafikas (žr. 7 pav.) yra nuspręsta, kad laiko logaritmo priklausomybė nuo parametro n yra tiesinė. Todėl remiantis lentelės (žr. 9 lentelė) duomenimis yra sudaryta tiesinė lygtis:

$$y_{Log_2^{Time}} = 0.0841 * n - 10.0369.$$

,kurios pagalba galima prognozuoti didesniems McEliece kriptografinės sistemos n parametrams vykdymo laikus.

10 lentelė Pranešimų persiuntimo atakos vykdymo laikų prognozavimas su superkompiuteriu

Parametras n	Vykdymo laikai		
	Praktiniai	Prognozuoti asmeniniam kompiuteriui	Prognozuoti superkompiuteriui
32	0.0024 (s)	0.00292 (s)	$1.4029 * 10^{-9}$ (s)
64	0.0192 (s)	0.0198 (s)	$9.538 * 10^{-9}$ (s)
128	1.359 (s)	0.91753 (s)	$4.4089 * 10^{-7}$ (s)
256	1664.5912 (s)	1960.461 (s)	0.000942 (s)
512	-	283.81 (metų)	4300.7555 (s)
1024	-	$5.92 * 10^{15}$ (metų)	$2.84 * 10^{9}$ (metų)
2048	-	$2.57 * 10^{42}$ (metų)	$1.23 * 10^{36}$ (metų)
4096	-	$4.85 * 10^{95}$ (metų)	$2.33 * 10^{89}$ (metų)

Iš gautų rezultatų galima teigti, kad kriptografinė sistema yra saugi prieš pranešimų persiuntimo ataką. Vertėtų pastebėti, kad atakos pavojingumas yra didžiausias, kai klaidos

vektorių iškraipytų bitų pozicijos nesutampa, todėl rekomenduojama kriptografinę sistemą naudoti su parametrais – $n \geq 1024$. Taip pat reikėtų nepamiršti, kad atakos rezultatas yra dešifruotas pranešimas, o ne privatus raktas. Todėl net ir suradus pradinį pranešimą, kai pranešimas neteko savo vertės, kitais atvejais viską reikės vykdyti iš naujo

6.8. Susijusių pranešimų atakos realizacija

Susijusių pranešimų atakos realizacija yra labai panaši į pranešimų persiuntimo atakas. Skirtumas nuo prieš tai realizuotos atakos yra tas, kad kriptoanalitikas žino pranešimų bitų sumą ir yra siunčiami du skirtingi pranešimai. Todėl pranešimų persiuntimo atakoje naudotas algoritmas yra papildytas naujais funkcionalumais. Vienas iš funkcionalumų yra tai, kad suskaičiuojama pranešimų bitų suma ir ji užkoduojama. Taip pat, pranešimų persiuntimo atakoje klaidos vektorių iškraipytos pozicijos yra randamos sudėjus šifrus, šioje atakoje prie šifrų sumos yra pridedama užšifruoto pranešimo suma. Visi toliau naudojami veiksmai nieko nesiskyrė nuo pranešimų persiuntimo atakoje naudotų.

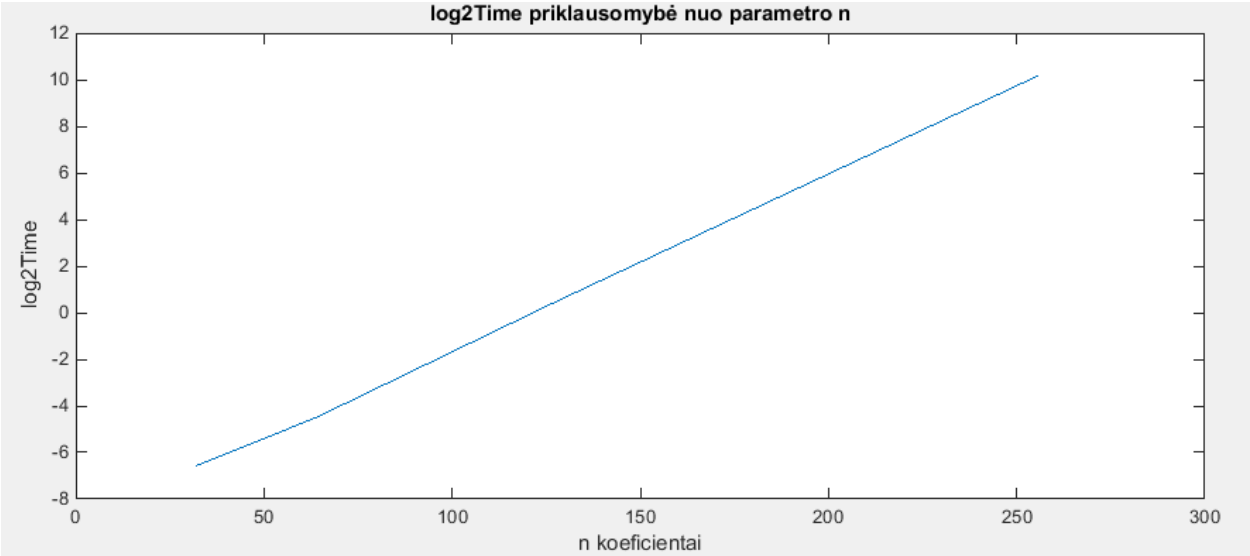
6.9. Susijusių pranešimų atakos rezultatai

Atlikus 10 vienas nuo kito nepriklausomų bandymų su susijusių pranešimų ataka, gauti rezultatai yra pateikiami lentelėje:

11 lentelė. Pranešimo persiuntimo atakos vykdymo laikai

Parametrai	**m = 5, t = 3**	**m = 6, t = 4**	**m = 7, t = 8**	**m = 8, t = 13**
Vidutinis vykdymo laikas (s)	0.0105	0.0445	1.4204	1170.3446

Pagal gautus rezultatus yra sudaryta vidutinių vykdymo laikų rezultatų grafikas:



8 pav. Pranešimo persiuntimo atakos laiko log_2 priklausomybė nuo parametro n.

Iš pateikto grafiko (žr. 8 pav.) ir rezultatų yra nuspręsta, kad laiko logaritmo priklausomybė nuo parametro n yra tiesinė. Todėl pagal lentelės (žr. 11 lentelė) duomenimis pasinaudojus mažiausių kvadratų metodu yra sudaryta tiesinė lygtis:

$$y_{Log_2^{Time}} = 0.0754 * n - 9.1407,$$

kuria pasinaudojus, galima įvertinti didesniems *n* parametrams vykdymo laikus.

12 lentelė Susijusių pranešimų atakos vykdymo laikų prognozavimas su superkompiuteriu

Parametras n	Vykdymo laikai		
	Praktiniai	Prognozuoti asmeniniam kompiuteriui	Prognozuoti superkompiuteriui
32	0.0105 (s)	0.00943 (s)	$2.4154 * 10^{-8}$ (s)
64	0.0445 (s)	0.0502 (s)	$2.4154 * 10^{-8}$ (s)
128	1.4204 (s)	1.4262 (s)	$6.8531 * 10^{-7}$(s)
256	1170.3446 (s)	1148.1415 (s)	0.000551 (s)
512	-	23.5948 (metų)	357.5442 (s)
1024	-	$9.91 * 10^{12}$ (metų)	$4.76 * 10^{6}$ (metų)
2048	-	$1.75 * 10^{36}$ (metų)	$8.4 * 10^{29}$ (metų)
4096	-	$5.44 * 10^{82}$ (metų)	$2.61 * 10^{76}$ (metų)

Pagal tiesinę lygtį, prognozavome vykdymo laikus (žr. 12 lentelę) naudotam stacionariam kompiuteriui su didesniais parametro n reikšmėmis. Taip pat buvo įvertinta, kiek laiko, šifro įveikimui užtruktu naudojant šiuo metu patį galingiausią superkompiuterį. Iš sudarytos lentelės

(žr. 12 lentelę) galima įvertinti, kad McEliece kriptografinė sistema yra saugi naudojant parametrą $n \geq 1024$.

Vertėtų atkreipti dėmesį, kad kaip ir prieš tai buvusi ataka, ji yra pavojingiausia, kai klaidos vektorių iškraipytų bitų pozicijos nesutampa. Tokiu atveju iškraipytą bitą reikia rinktis iš jau žinomų pozicijų. Norint padidinti atsparumą prieš ataką, galima prieš siunčiant pranešimą prie kiekvieno pranešimo pridėti atsitiktinius simbolius.

Išvados ir rezultatai

Darbe buvo išanalizuoti kriptografiniai aspektai ir susipažinta su populiariausiomis kriptografinėmis sistemomis. Apžvelgtos matematinės problemos, kuriomis remiasi kriptografinės sistemos. McEliece viešojo rakto kriptografinė sistema remiasi viena iš jų: tiesinių kodų dekodavimo problema.

Susipažinus su dažniausiai naudojamomis kriptografinėmis sistemomis buvo išnagrinėta viešojo rakto McEliece kriptografinė sistema. Išskirtos kriptografinės sistemos sudedamosios dalys:

- Gappa kodai;
- S ir P matricos;
- Klaidos vektorius.

Apžvelgta kiekvienos dalies įtaka kriptografinės sistemos saugumui. Nustatyta, kad kriptografinė sistema labiausiai yra pažeidžiama dėl klaidos vektoriaus. McEliece kriptografinės sistemos pranešimų šifravimui buvo panaudota FlexiProvider biblioteka, kurios pagalba buvo šifruojamas atsitiktinai sugeneruotas dvejetainis pranešimas. Prieš naudojant FlexiProvider biblioteką, ji modifikuota, kad būtų galima gauti viešąjį raktą.

Darbo metu buvo išskirtos kritinės ir nekritinės kriptografinės sistemos atakos. Didelis dėmesys buvo skirtas šioms atakoms: apibendrintos informacijos dekodavimo, žinomo dalinio teksto, pranešimų persiuntimo ir susijusių pranešimų atakoms. Kiekviena pasirinkta ataka buvo realizuota Java programavimo kalba ir atlikti nepriklausomi eksperimentai su įvairiais (m, t) dydžio McEliece kriptografinės sistemos parametrais.

Naudojant apibendrintos informacijos dekodavimo ataką, buvo išskirti optimalūs t parametrai kiekvienam m, su kuriais McEliece kriptografinė sistema yra saugiausia (tyrimams naudotas personalinis kompiuteris). Pagal gautus eksperimento rezultatus buvo išsikelta hipotezė, kad *t* priklausomybė nuo *n* yra tiesinė. Pasinaudojus mažiausių kvadratų metodu buvo sudaryta tiesinė lygtis, kurios pagalba yra prognozuojamas optimalus *t* parametras didesniems *n* parametrams. Taip pat, buvo realizuota papildoma programa, kuri apskaičiuodavo optimalius teorinius t parametrus ir binarinį operacijų skaičių. Pagal gautus rezultatus buvo nustatyta, kad hipotezė, jog *t* priklausomybė nuo n yra tiesinė, klaidinga. Todėl palyginus gautus rezultatus su prognozuotomis reikšmėmis nuspręsta tolimesniems tyrimams naudoti optimalias *t* reikšmes, kurios gautos iš teorinės formulės.

Atlikus apibendrintos informacijos dekodavimo atakos eksperimentus ir gavus vykdymo laukų rezultatus buvo sudaryta tiesinė lygtis, su kuria buvo galima prognozuoti, tyrimo metu

naudotam kompiuteriui, veikimo laikus su didesniais n parametrais. Pagal gautus prognozuotus vykdymo laikus buvo apskaičiuojamas vykdymo laikas, su šiuo metu galingiausiu Tianhe-2 superkompiuteriu. Iš gautų rezultatų paaiškėjo, kad McEliece kriptografinė sistema yra saugi naudojant $n \geq 1024$ parametrus.

Žinomo dalinio teksto atakos pavojingumas priklauso nuo to, kokią pradinio pranešimo dalį žino kriptoanalitikas. Todėl, su žinomo dalinio teksto ataka buvo atlikti tyrimai tariant, kad yra žinomas 20 %, 50 % ir 90 % pradinio pranešimo. Pagal gautus rezultatus, sudarytos tiesinės lygtis, kuriomis pasinaudojus buvo prognozuoti vykdymo laikai didesniems n parametrams. Pasinaudojus prognozuotais vykdymo laikais, nustatyti vykdymo laikai superkompiuteriu. Gauta, kad kriptoanalitikui žinant mažiau negu 20 % pradinio teksto, McEliece kriptografinę sistemą prieš šią ataką galima laikyti saugia su $n \geq 512$ parametru. Tačiau kriptoanalitikui žinant daugiau negu 20 % pradinio teksto, kriptografinė sistema su $n = 512$ parametru yra nesaugi. Todėl, jeigu žinoma, jog kriptoanalitikas gali turėti dalį siunčiamo pranešimo, rekomenduojama padidinti kriptografinės sistemos parametrus ir naudoti $n \geq 1024$.

Pranešimo persiuntimo ir susijusių pranešimų atakose yra realizuotas n lygčių su k nežinomųjų sprendimo metodas. Tiek vienoje, tiek kitoje atakoje yra bandoma atspėti klaidos vektorių, turint skirtingus pradinius duomenis. Atlikus eksperimentus su pranešimų persiuntimo ataka, buvo gauti vykdymo laikai. Pagal gautus vykdymo laikus sudarytos tiesinės lygtis, kuriomis pasinaudojus buvo prognozuoti vykdymo laikai su didesniais *n* parametrais. Pasinaudojus gautais vykdymo laikais buvo nustatyta, kiek laiko ataką užtruks vykdyti superkompiuteris. Gauta, kad McEliece kriptografinė sistemą išlieka saugi naudojant $n \geq 1024$. Vertėtų pastebėti, kad ataka būtų pavojingesnė, jei būtų bandoma kriptografinę sistemą įveikti turint daugiau negu du šifrus.

Susijusių pranešimų atakai prognozavus vykdymo laikus naudojant superkompiuterį, paaiškėjo, kad ataka yra pavojingesnė palyginus su prieš tai nagrinėtomis atakomis. Tačiau kriptografinę sistemą išlieka saugi naudojant parametrą $n \geq 1024$ ir optimalų klaidų skaičių (žr. 3 lentelę).

McEliece viešojo rakto kriptografinės sistemos kūrėjo Robert J. McEliece pasiūlyti parametrai $n \geq 1024, k \geq 524, t \geq 50$, kaip parodė eksperimentas yra saugūs prieš nagrinėtas atakas ir šiuo metu nėra įveikiami. Todėl kriptografinę sistemą galima laikyti saugia prieš nagrinėtas atakas, naudojant parametrą $n \geq 1024$ ir optimalias parametro t reikšmes. Bendru atveju tiesinę lygti padauginę iš tyrime naudoto kompiuterio flops ir padalinus iš kito kompiuterio flops, gauname atakų universalias lygtis. Taip pat pagal gautus atakų vykdymo laikų rezultatus buvo nustatyta, kad atakų vykdymo laiko logaritmo priklausomybė nuo parametro n yra tiesinė.

Literatūros sąrašas

[ABP+13] Nadhem J. AlFardan, Daniel J. Bernstein, Kenneth G. Paterson, Bertram Poettering, Jacob C. N. Schuldt, On the security of RC4 in TLS and WPA, USENIX Securaty Symposium, p. 1 – 31, 2013.

[Ber97] Thomas 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, Proceedings of 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1997, Lecture Notes in Computer Science 1294, p. 213 – 220, 1997.

[Can98] Anne Canteaut, Public – key cryptosystems based on error – correcting codes. [žiūrėta 2016-04-15]. Prieiga per internetą: <https://www.rocq.inria.fr/secret/Anne.Canteaut/English/mceliece.html>

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

[Cop94] Don Coppersmith, The Data Encryption Standard (DES) and its strenght against attacks, IBM J. Res. Develop, volume 38, 1994 kovo 3. [žiūrėta 2015-04-02] Prieiga per internetą: < http://simson.net/ref/1994/coppersmith94.pdf >

[CS00] Anne Canteaut and Nicolas Sendrier, Cryptanalysis of the Original McEliece Cryptosystem, Springer–Verlag Berlin Heidelberg, p. 187 – 199, 2000.

[DMR11] Hang Dinh, Cristopher Moore, Alexander Russell, McEliece and Niederreiter Cryptosystems that Resist Quantum Fourier Sampling Attacks, 2011. [žiūrėta 2014-11-10] Prieiga per internetą: <http://www.santafe.edu/media/workingpapers/11-06-021.pdf>

[EOS06] D. Engelbert, R. Overbeck and A. Schmidt: A summary of McEliece – type cryptosystems and their security. Cryptology ePrint Archive, 2006, Report 2006/162. [žiūrėta 2015-05-28] Prieiga per internetą: <http://eprint.iacr.org/2006/162>

[Fri07] Hagen Fritsch, Cryptography in GSM, Computer Security Al Akhawayn University, Morocco, 2007. [žiūrėta 2015-04-10] Prieiga per internetą: <https://itooktheredpill.irgendwo.org/stuff/studium/gsm.pdf >

[HP03] W. Cary Huffman, Vera Pless, Fundamentals of Error – Correcting Codes, Cambridge University Press, 2003.

[Joc02] Ellen Jochemsz, Goppa Codes & the McEliece Cryptosystem. Masters thesis, Vrije Universiteit Amsterdam, 2002.

[KA08] Stasys Kašėta, Tomas Adomkus, Telefonijos informacijos ir VoIP sauga, Kauno technologijos universitetas, 2008.

[KI01] Kazukuni Kobara and Hideki Imai. Semantically Secure McEliece Public – Key Cryptosystems – Conversions for McEliece PKC. K. Kim (Ed.), Public Key Cryptography, Proceedings of 4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001, Cheju Island, Korea, February 13 – 5, 2001, Lecture Notes in Computer Science 1992, p. 19 – 35, 2001.

[KI02] Kazukuni Kobara, Hideki Imai. New Chosen – Plaintext Attacks on the One – Wayness of the Modified McEliece PKC Proposed at Asiacrypt 2000. D. Naccache, P. Paillier (Eds.): Proceedings of 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 2002, Lecture Notes in Computer Science 2274, 237 – 251, 2002.

[LDW94] Yuan Xing Li, Robert H. Deng, Xin Mei Wang, On the Equivalence of McEliece's and Niederreiter's Public – Key Cryptosystems. IEEE – IT 40(1), p. 271 – 273, Sausis 1994.

[LS01] Pierre Loidreau and Nicolas Sendrier, Weak keys in the McEliece Public – Key Cryptosystem. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 47, No. 3, p. 1207 – 1211, Kovo 2001.

[Nie86] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. Probl. Control and Inform. Theory 15(2), p. 159 – 165, 1986

[McE78] R. J. McEliece. A Public – Key Cryptosystem Based On Algebraic Coding Theory. JPL DSN Progress Report 42 – 44, p. 114 – 116, April 15, 1978. Prieiga per internetą < http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF> [žiūrėta 2016-05-22]

[MOV96a] Alfred J. Menezes, Paul C. van Oorschot ir Scott A. Vanstone. Handbook of applied cryptography. Press, October 1996, p. 1 – 49, Chapter 1. Prieiga per internetą < http://cacr.uwaterloo.ca/hac/about/chap1.pdf > [žiūrėta 2014-10-05]

[MOV96b] Alfred J. Menezes, Paul C. van Oorschot ir Scott A. Vanstone. Handbook of

applied cryptography. Press, October 1996, p. 283 – 319, Chapter 8. Prieiga per internetą < http://cacr.uwaterloo.ca/hac/about/chap8.pdf > [žiūrėta 2016-05-23]

[Pom08] Carl Pomerance. Smooth numbers and the quadratic sieve. Algorithm Number Theory MSRI Publications Volume 44, 2008.

[Sta02] Vilius Stakėnas. Kodavimo teorija. Paskaitų kursas. 2002.

[Sta05] Vilius Stakėnas. Šifrų istorijos. TEV, 2005.

[Sta07] Vilius Stakėnas. Kodai ir šifrai. Informacijos kodavimo ir kriptografijos pagrindai. TEV, 2007.

[Sti95] Douglas R. Stinson. Cryptography: Theory and Practice. CRC Press, 1995.

[SLD+08] E. Sakalauskas, N. Listopadskis, G. S. Dosinas, K. Lukšys, A. Katvickis, Kriptografinės sistemos – mokomoji knyga, Kauno technologijos universitetas, 2008.

[Was08] Lawrence C. Washington, Elliptic Curves: Number Theory and Cryptography, Second Edition, Chapman and Hall/CRC, 2008.

Priedas

13 lentelė. Praktinių tyrimų gauti vykdymo laikai (s)

t	m			
	5	6	7	8
1	0.0038	0.0056	0.0392	0.4542
2	0.0044	0.0193	0.196	3.2977
3	0.0045	0.0335	0.375	13.1575
4	0.0028	0.0518	0.9884	64.5729
5	0.0015	0.0226	1.2562	168.6791
6	0.001	0.015	1.0656	318.1471
7	-	0.012	1.2095	1093.3793
8	-	0.0067	1.9223	1633.2216
9	-	0.0043	0.9825	1534.6241
10	-	0.0019	0.5636	1779.0608
11	-	-	0.2835	2216.6053
12	-	-	0.2363	2607.913
13	-	-	0.0355	2501.6137
14	-	-	0.0167	2215.7746
15	-	-	0.0143	903.2564
16	-	-	0.0072	674.8474
17	-	-	0.007	549.9312
18	-	-	0.001	227.7834
19	-	-	-	270.5702
20	-	-	-	61.7863
21	-	-	-	27.5418
22	-	-	-	37.8893
23	-	-	-	9.9487
24	-	-	-	3.0173
25	-	-	-	4.6988
26	-	-	-	0.5566
27	-	-	-	0.155
28	-	-	-	0.0481
29	-	-	-	0.025
30	-	-	-	0.0105
31	-	-	-	0.0056